Web Services documentation of Geoconcept Web

GEOCONCEPT SAS

Geoconcept Web

Version: 2022-1

Date: 3/24/22

This documentation concerns version 2021.0.x of Geoconcept Web.

Introduction

Geoconcept Web is supplied with a series of tools designed so integrators can:

- add new Widgets in the Designer;
- use the map in autonomous interfaces with the Javascript API;
- exploit numerous web services (SOAP/REST) provided with the solution;
- as well as integration examples.

Widgets Development Kit

This section explains how to create widgets so they can be integrated in Geoconcept Web. These new widgets can then take advantage of the features provided by the mechanism set up in the solution.

Introduction

Already present

The first and foremost requirement is that the project is ISO-functional and ISO-compatible with geoweb-eaasy v1. To this end, automated migration procedures have been developed to ensure the installation of geoweb-easy v1 with a database of existing geoweb-easy data. These migration procedures will be explained in the section on the server.

Technological choices

Unlike geoweb-easy v1, geoweb-easy v2 will integrate a full JavaScript engine on the client side. If we take the five parts of the project:

1. The widgets part will be based on the last version of YUI: 3.5.1
2. The Java part is based on Java 5 and will use the Sax libraries for managing widgets
3. The database part uses the current version of Hibernate used as Standard in geoweb, that is

```
Hibernate 3.2
```

1. The Dispatcher part of the URL is based on Spring MVC 25
2. The REST Web Services part is based on Apache CXF, and notably uses the Web plugins manager

```
Services de Geoweb.
```

```
The Jackson 1.8.5 library handles client-server mapping and the project will integrate cityportal-jar for
 managing the POI.
```

Widgets

Introduction

The widget is a graphics interface component (any kind of HTML/JavaScript type element) and is also a tool enabling retrieval of information via the server, or an internal/external datasource.

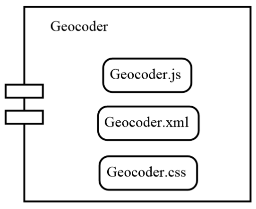
The advantage of using widgets is that custom client components can be designed without having to alter the Easy Geoweb code. A widget must therefore be interpreted by the client in a generic and autonomous way.

This chapter presents details of the composition, dependencies, and configuration of widgets.

Architecture

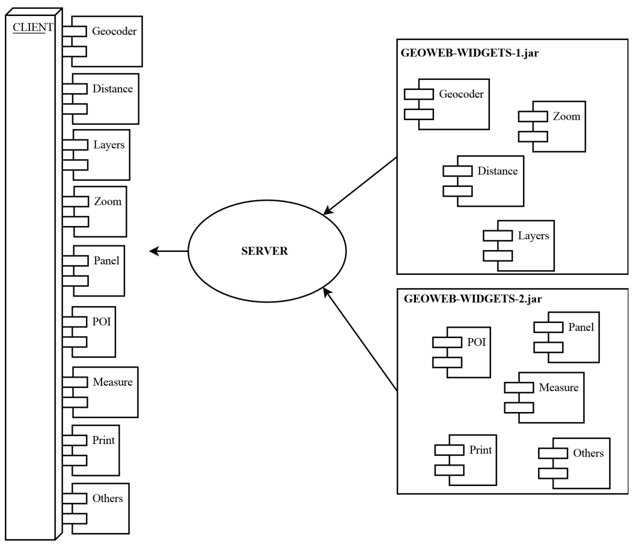
Each widget is independent of any others. It is made up of an XML description file, a JavaScript source file, and a CSS style file:

Architecture of a widget



The client is composed of a widgets manager at JavaScript level, and is extended and enhanced with a series of widgets.

Global architecture of widgets



The principle is to describe the widget via an XML configuration file that will point towards the corresponding source file. When the widget is loaded by the widgets (server) engine, this will parse the XMl file in order to extract from it the necessary information. It will put these information items in JSON format, and return it to the client.

XML file

General description

The XML file is a widget description file. It is parsed by the widget engine in order to extract information that will be of use to the widget engine, this information then being returned to the client in JSON format. This file must respect the same rules for all widgets:

- `id` : the widget ID must be unique
- `icon` : the related filepath (in relation to the jar) of the widget icon

- `js` : the corresponding source file
- `module` : the YUI module for the widget, which must be unique, and identical to that of the source file
- `display` : the widgets can inherit other widgets or YUI classes that we don't necessarily need to display, this attribute allows us to specify if the widget is to be displayed in the builder/portal or not
- `groups` : groups define which users will have access to the widget in the case of a private portal. Groups can be defined directly in the design tool.
- `lang` : the resources file allows internationalisation of the widget. The filepath refers to a properties file. In the example below, the file searched for is by default resources/widgets.properties. A widget can support several languages, for example resources/widgets_fr.properties or resources/widgets_en.properties. The language will be displayed as a function of the browser language.
- `category` : the category allows you to define the positioning of the widget in the designer library. If this field is not filled, the category will be defined as a function of the directory in which the widget will be stored. The list of categories is available in the EasyGeowebCategories.xml file. From this file, it is possible to add new categories (see the list of categories).
- `position` : The field position allows you to define the positioning of the widget in the widget library. If this value is not filled, the widget will be placed at the end.

```
<widget id="geocoder"
        category="navigation"
        position="1"
    icon="images/geocoder.png"
    js="view/GeocoderWidget.js"
    module="geoconcept-widget-geocoder"
    display="true"
        groups="Standard"
        lang="resources/widgets">
</widget>
```

It is possible to add properties to configure the widget in the designer in the XML file. Depending on what has been entered, the properties are generated automatically in the widget configuration window in the composer.

The property label displayed is structured as follows: "widget." + widget id + "." + property id. The value to add in the properties file is, for example, "widget.geocoder.mode=Mode d'affichage".

The *params* field allows you to define a default value for the property:

```
<widget id="geocoder"
                category="navigation"
                position="1"
                icon="resources/images/Geocode_32.png"
                js="navigation/GeocoderWidget.js"
                module="geoconcept-widget-geocoder"
                display="true"
                groups="Standard"
                lang="resources/widgets">
        <properties>
                <property id="mode" type="radiobutton-2">
                        <params value="0"/>
                </property>
        </properties>
```

```
</widget>
```

List of properties

Here is a list of properties available for the widgets:

- **`checkbox`** : handles true or false values.

```
<property id="checkboxProperty" type="checkbox">
        <params value="true"/>
</property>
```

- **`text`** : handles text values.

```
<property id="textProperty" type="text">
        <params value="default text"/>
</property>
```

- **`slider`** : handles numeric values. Maximum values are defined by min and max. The selected value corresponds to value.

```
<property id="sliderProperty" type="slider">
        <params min="1" max="12" value="6"/>
</property>
```

- **`colorpicker`** : handles hexadecimal values.

```
<property id="colorpickerProperty" type="colorpicker">
        <params value="#ffffff"/>
</property>
```

- **`combobox`** : handles text values. The list of available values is based on the `getComboboxOptions` and `setComboboxOptions` methods (see the widgets API).

```
<property id="comboboxProperty" type="combobox">
        <params value="comboboxValueName"/>
</property>
```

- **`datagrid`** : handles text values. The default values are separated by ; characters. The list of available values is based on the `getDgOptions` and `setDgOptions` methods (see the widgets API).

```
<property id="datagridProperty" type="datagrid">
        <params value="datagridValueName"/>
</property>
```

- **`radiobutton-2`** : handles numeric values. Possible values are 0 or 1. To display a label on each radio-button, you need to add 2 keys in the properties file, ending them with ".left" and ".right"

```
<property id="radiobuttonProperty" type="radiobutton-2">
        <params value="0"/>
</property>
```

- **`checkbox-2`** : handles numeric values. Functions in an identical manner to the "radiobutton-2" property, except that it is possible to select both the property values.

```
<property id="checkbox2Property" type="checkbox-2">
        <params value="0"/>
</property>
```

Each of these properties can then be used in the widget using the `getPropertyValue` method (see the widgets API).

List of categories

A series of categories are defined in a project configuration file *`EasyGeowebCategories.xml`*. Each widget is dependent on one of these categories. To do this the widget must be placed in a directory bearing the name of a category. Example: our geocode widget is placed in the file *`navigation/Distance.js`*,and it then belongs to the navigation category. The distance widget is placed in *`measure/Distance.js`* therefore it belongs to the *measure* category.

It is possible to add new categories in *`EasyGeowebCategories.xml`*. To do this, simply add a tag *entry*. *key* allows the user to define the positioning in the accordion display.

```
<properties>
        <entry key="0">layout</entry>
        <entry key="1">general</entry>
        <entry key="2">navigation</entry>
        <entry key="3">view</entry>
        <entry key="4">measure</entry>
        <entry key="5">annotation</entry>
        <entry key="6">data</entry>
        <entry key="7">modification</entry>
</properties>
```

The name of the category allows you to handle internationalisation. For example, for a new category named *`graphic`*, you need to add the following key in *`easy.properties`*

```
accordion.category.graphic = graphique
```

JavaScript file

WidgetBase

The JavaScript file is the widget source file. This will be loaded by the combo to be injected in the client. Here is an outline source file:

```
YUI.add("module-name", function(Y) {

        function WidgetClassName(config) {
        WidgetClassName.superclass.constructor.apply(this, arguments);
        }

        WidgetClassName.NAME = "WidgetClassName";

        Y.extend(WidgetClassName, Y.GCUI.WidgetBase,  {
            initializer: function(config) {
            },

            destructor: function() {
            },

            getWidget: function() {
                // return html description widget
            },

            configureAction: function() {
                WidgetClassName.superclass.configureAction.apply(this);

                // some actions
            }

        });
        Y.namespace("GCUI");
        Y.GCUI.WidgetClassName = WidgetClassName;

}, "1.0.0", {requires: ["project-widgetbase"] });
```

The `getWidget()` function will be called in the first place by the WidgetManager (client) to display the widget. This means that this method returns the representation of the widget. It may contain all that the developer wants to display. The only constraint is to remain in an HTML tag `<div></div>`. In effect, the widget will only be displayed if it is contained in this tag.

`WidgetBase` displays TreeView, listbox and combobox (and so on) items.

The `configureAction()` function will be called by the WidgetManager to configure the widget once it has been inserted in the DOM. `WidgetBase` suggests a series of basic functions forming the API of widgets. So, all widgets have access to the map, to the detail of the project, but also to a panel that one can use to display diverse information items.

All widgets inheriting from WidgetBase possess default properties: Display of the label, a label designating a label, and display of a label separator.

This the section called “Widgets by example” part along with the tutorial part present real-life examples of the utilisation of `WidgetBase` methods.

WidgetButton

`WidgetButton` functions in the same way as `WidgetBase`; it will suffice to inherit this class and to add “requires **`project-widgetbutton`** ”.

There are 3 types of button to define in `initializer()`:

- **`BUTTON`** : basic button, this action executes at the moment of the mouse click
- **`TOOL`** : after the mouse click, the button is activated and remains activated. To de-activate it, click on another button of the TOOL type.
- **`TOGGLE`** : a single mouse click will activate the button, and a second click will de-activate it.

Here is the outline code for a `WidgetButton`

```
YUI.add("geoconcept-widget-button", function(Y) {

        function SchemaButton(config) {
                SchemaButton.superclass.constructor.apply(this, arguments);
        }

        SchemaButton.NAME = "schemaButton";

        Y.extend(SchemaButton, Y.GCUI.WidgetButton,  {
                initializer: function(config) {
                        SchemaButton.superclass.initializer.apply(this);
                        this.setType(Y.GCUI.WidgetButton.TYPE_TOOL);
                },

                destructor: function() {
                        SchemaButton.superclass.destructor.apply(this);
                },

                configureAction: function() {
                        SchemaButton.superclass.configureAction.apply(this);
                },

                // Button type action
                trigger:function() {
                        SchemaButton.superclass.trigger.apply();
                        // some actions
            },

                // Toggle and tool type action
                activate:function() {
                        SchemaButton.superclass.activate.apply();
                        // some actions
            },

            // Toggle and tool type action
            deactivate:function() {
                SchemaButton.superclass.deactivate.apply();
            }

        });
        Y.namespace("GCUI");
        Y.GCUI.SchemaButton = SchemaButton;

}, "1.0.0", {requires: ["project-widgetbutton"] });
```

All widgets inheriting from WidgetButton possess default properties: Label display, a label designating a label, and default widget in the case of a toggle or tool type button.

API

Method

```
getId() : {String} Retourne l'identifiant unique du widget.
```

```
getJson() : {JSON} Retourne la description JSON du widget.
```

```
getJsonProject() : {JSON} Retourne la description JSON du projet.
```

```
getJsonLayers() : {JSON} Retourne la description JSON de la liste des couches définie pour le gestionnaire
 de couches
```

```
getWidgetName() : {String} Retourne le nom internationalisé du widget.
```

```
getWidgetDescription() : {String} Retourne la description internationalisé du widget.
```

```
getEGWApi() : {Y.GCUI.EGWRestAPI} Retourne l'objet EGWRestAPI afin d'avoir accès aux méthodes du service
 Rest de Easy-geoweb.
```

```
getParameters() : {JSON} Retourne les paramètres par défaut de l'application.
```

```
getMap() : {GCUI.Map} Retourne l'objet carte.
```

```
getNode() : {Node object} Retourne le widget sous format javascript.
```

```
getContextUrl() : {String} Retourne le contexte de l'url par exemple "geoweb".
```

```
getWidgetPath() : {String} Retourne l'url pour accéder aux fichiers présent dans le JAR par exemple "geoweb/
widget?".
```

```
getHost() : {String} Retourne le host de l'url par exemple "http://localhost:8080/".
```

```
getImageSrc() : {String} Retourne l'url pour accéder à l'image définie dans icon du fichier XML.
```

```
isPortal() : {Boolean} Retourne true si l'application est démarrée dans le portail, false si c'est dans
 l'outil de conception.
```

```
getWrapper() : {String} Retourne le wrapper, utilisé pour désactiver le widget dans l'outil de conception.
```

```
getPropertyValue(id) : {String} Retourne la valeur de la propriété définie dans le fichier XML en fonction
 de son identifiant (id).
```

```
updatePropertyValue(id,value) : Met à jour la valeur (value) de la propriété définie dans le fichier XML en
 fonction de son identifiant (id).
```

```
getPropertiesPanel() : {Node object} Retourne l'objet panneau de propriété sous l'accordéon dans le but
 d'ajouter de nouvelles propriétés.
```

```
emptyPropertiesPanel() : Vide le panneau de propriété.
```

```
addDynamicProperty(order, propertyId, label, type, defaultValue, defaultMin, defaultMax) : Permet de créer
 une propriété sans
le renseigné dans le fichier XML du widget. order comme position d'insertion parmi les autres propriétés.
 propertyId comme
identifiant. label d'affichage. type, à choisir parmi les types disponibles (par exemple "text").
 defaultValue, defaultMin
et defaultMax pour définir les valeurs par défaut, minimale (optionnel) et maximale (optionnel).
```

```
movePropertyToOrder(property, order) : Déplacer une propriété. property comme identifiant de la propriété et
 order en numéro de
placement parmi les autres propriétés.
```

```
getComboboxOptions(id) : List{String} Retourne la liste des valeurs disponibles pour la propriété "combobox"
```

```
en fonction de l'id de la propriété.
```

```
setComboboxOptions(id, data) : Enregistre en mémoire la liste des valeurs pour la propriété "combobox" avec
 en paramètre l'id de la
propriété et data en liste de string.
```

```
getDgOptions(name) : List{String} Retourne la liste des valeurs disponibles pour la propriété "datagrid"
en fonction de l'id de la propriété.
```

```
setDgOptions(name, data) : Enregistre en mémoire la liste des valeurs pour la propriété "datagrid" avec en
 paramètre l'id de la
propriété et data en liste de string.
```

```
_(key) : {String} Converti une clé en chaine de caractère internationalisable.
```

Empty method to enrich for a utilisation

```
reinitView() : Méthode permettant de réinitialiser l'action du widget.
```

```
getWidget() : {String} Méthode permettant de définir la représentation du widget.
```

```
checkboxPropertyAction(propertyId, value) : Action appelée après l'enregistrement d'une propriété de type
 checkbox. PropertyId
comme identifiant de la propriété enregistré. Value comme valeur enregistré.
```

```
textPropertyAction(propertyId, value) : Action appelée après l'enregistrement d'une propriété de type text.
 PropertyId
comme identifiant de la propriété enregistré. Value comme valeur enregistré.
```

```
hiddenPropertyAction(propertyId, value) : Action appelée après l'enregistrement d'une propriété utilisant
 des input de type
hidden (datagrid, combobox...). PropertyId comme identifiant de la propriété enregistré. Value comme valeur
 enregistré.
```

```
radioPropertyAction(propertyId, value) : Action appelée après l'enregistrement d'une propriété de type
 radiobutton. PropertyId
comme identifiant de la propriété enregistré. Value comme valeur enregistré.
```

The group of methods of WidgetBase are re-useable in WidgetButton.

Constant

```
WidgetButton.NAME_PANEL : Nom du panneau
```

```
WidgetButton.TYPE_TOGGLE : Bouton de type Toggle (activable et désactivable par l'utilisateur).
```

```
WidgetButton.TYPE_BUTTON : Bouton de type Bouton (action executé au moment du clic).
```

```
WidgetButton.TYPE_TOOL : Bouton de type Tool (un seul bouton de type Tool peut être est activé, si
 l'utilisateur active un autre bouton de type Tool, tous les autres sont désactivés).
```

Method

```
getType()       : {String} Retourne le type du bouton (Tool, Toggle ou Button).
```

```
isActive() : {Boolean} Retourne true si le bouton est activé, false si non.
```

```
getPanel() : {Panel node} Retourne l'objet panel lié au widget. Chaque widget possède son propre panel par
 défaut.
```

```
getWidget() : {Widget node} Retourne l'objet représentant le widget sous forme de bouton avec une icone
 défini dans le fichier XML.
```

```
reinitView() : Réinitialise le widget en désactivant le bouton.
```

```
createPanel(title) : Créé le panel par défaut avec le titre (title) en paramètre.
```

```
showPanel() : Affiche le panel.
```

```
hidePanel() : Masque le panel.
```

```
hidePopup() : Masque les popups de type _GCUI.Control.Popup_.
```

Empty method to enrich for a utilisation

```
activate() : Méthode appelée suite à l'activation d'un bouton de type Tool ou Toggle.
```

```
deactivate() : Méthode appelée suite à l'désactivation d'un bouton de type Tool ou Toggle.
```

```
trigger() : Méthode appelée suite au clic sur un bouton de type Button.
```

```
onClosePanel() : Méthode appelée suite à la fermeture du panel.
```

Widgets by example

Here is the detail of certain widgets to study their implementation.

Hand

This widget allows the user to move around the map. It is of the button type and will therefore inherit from the WidgetButton class.

```
YUI.add("geoconcept-widget-hand", function(Y) {

        /* Hand class constructor */
        function Hand(config) {
                Hand.superclass.constructor.apply(this, arguments);
        }

        /*
         * Required NAME static field, to identify the Widget class and
         * used as an event prefix, to generate class names etc. (set to the
         * class name in camel case).
         */
        Hand.NAME = "hand";

        Y.extend(Hand, Y.GCUI.WidgetButton,  {
                initializer: function(config) {
                        Hand.superclass.initializer.apply(this);
                        this.setType(Y.GCUI.WidgetButton.TYPE_TOOL);
                },

                destructor: function() {
                        Hand.superclass.destructor.apply(this);
                },

                configureAction: function() {
                        Hand.superclass.configureAction.apply(this);
                },

                activate:function() {
                        Hand.superclass.activate.apply();
                        var map = this.getMap();

                        DynMapSetMouseMode(map,DynMapMoveSelectionMode());
            },

            deactivate:function() {
                Hand.superclass.deactivate.apply();
            }

        });
        Y.namespace("GCUI");
        Y.GCUI.Hand = Hand;

}, "1.0.0", {requires: ["project-widgetbutton"] });
```

This widget takes the standard example described above. The `activate()` method enables the widget to call the API Javascript [http://api.geoconcept.net/htc/index.html] method to enable the pan mode on the map.

Here is its XML description file:

```
<widget id="hand"
                icon="resources/images/hand.png"
                js="navigation/Hand.js"
                module="geoconcept-widget-hand"
                display="true"
                lang="resources/widgets">
</widget>
```

The name of the module corresponds to that defined in the source file. In addition, the user may notice that the widget is found in the position category of the accordion.

MeasureArea

This widget enables measurement of surface areas.

```
YUI.add("geoconcept-widget-measure-area", function(Y) {

        [...]

        Y.extend(MeasureArea, Y.GCUI.MeasureBase,  {
                [...]

            /**
             * Manage widget action
             */
                activate:function() {
                        MeasureArea.superclass.activate.apply(this,[this.getCallbacks()]);
                        this.createPanelMeasure();
                        this.showPanel();
                },

                deactivate:function() {
                        MeasureArea.superclass.deactivate.apply(this);
                },

                /**
                 * Manage default panel
                 */
                // Create panel for area information
                createPanelMeasure:function() {
                var panelNode = this.createPanel(this._("widget.measureArea.popup.title"));
                        var id = this.getPanelId();

                var areaInfo = Y.Node.create("<div id='" + id + "' class='areaMeasure'></div>");
                areaInfo.append("<div class='areaSegmentTitle'>" +
 this._("widget.measureArea.segment.measure") + "</div>");
                areaInfo.append("<div id='" + MeasureArea.ID_PANEL_SEGMENT_MEASURE + id + "'
 class='areaSegmentValue'>0.000 " + this._("widget.measureArea.segment.kilometer") + "<sup>2</sup></div>");

                        panelNode.appendChild(areaInfo);
            }
        });
        Y.namespace("GCUI");
        Y.GCUI.MeasureArea = MeasureArea;

}, "1.0.0", {requires: ["geoconcept-widget-measure-base"] });
```

This widget extends Y.GCUI.MeasureBase using the Control OpenLayers of measurement *OpenLayers.Control.Measure* : it uses the panel to display its result.

Layers

The layers widget is not a button but displays a TreeView on a given div.

```
YUI.add("geoconcept-widget-layers", function(Y) {

        [...]

        Y.extend(Layers, Y.GCUI.LayersBase, {

                [...]

                configureAction : function() {
                        Layers.superclass.configureAction.apply(this);

                        if (!this.isExistingLayers) {
                                var yuiNode = Y.one("#"+this.getId());
                                if (!this.isPortal()) {
                                        yuiNode.setContent(this.getWrapper());
                }
```

```
                                var map = this.getMap();
                                var wLayers = this;
                                var options = {
                                        div : document.getElementById(this.getId())
                                };
                                var ls = new GCUI.Control.LayerSwitcher(options);
                                map.addControl(ls);

                                var json = this.initJson();
                                ls.initFromJson(json);

                                yuiNode.addClass("addedWidget");

                                if (this.isPortal()) {
                        var layers = map.layers;
                                        for (var i = 0; i < layers.length; i++) {
                                                var layer = layers[i];
                                                layer.events.on({
                                    "visibilitychanged" : wLayers.layerChangeVisibility,
                                    scope : wLayers
                                });
                                        }
                                        this.showLegends(layers);
                }
                        }
                },

                [...]

        });
        Y.namespace("GCUI");
        Y.GCUI.Layers = Layers;

}, "1.0.0", {requires : ["geoconcept-widget-layers-base"] });
```

This widget extends the Y.GCUI.LayerBase that itself extends Y.GCUI.WidgetBase. The TreeView is generated by GCUI.Control.LayerSwitcher. It displays the list of layers to display in a JSON.

Contact

This widget is special in that it uses properties, these properties are defined in its XML description file:

```
<widget id="contact"
                icon="resources/images/contact.png"
                js="general/Contact.js"
                module="geoconcept-widget-contact"
                display="true"
                lang="resources/widgets">
        <properties>
                <property id="mailContact" type="text"/>
        </properties>
</widget>
```

From the client it will therefore be possible to define the mail address in a text field.

The source file takes into account these properties that are injected into it in JSON by the client. In effect, the widgets engine will read the widget’s XML file, inject the JSON corresponding to the client that will create a widget instance and the client will handle instantiatiation of the widget with the JSON descriptiion. This JSON description is *set* to the instance. Each time one of the property values is changed, the JSON is updated. When the builder is saved, the JSON file for each instance is retrieved to then be saved by the builder. (cf the client part)

Below is the source code for the widget:

```
YUI.add("geoconcept-widget-contact", function(Y) {

        [...]

        Contact.MAIL_CONTACT_PROPERTY = "mailContact"

        [...]

        Y.extend(Contact, Y.GCUI.WidgetButton,  {

                [...]

                configureAction: function() {
                        Contact.superclass.configureAction.apply(this);
                        var defaultMail = this.getPropertyValue(Contact.MAIL_CONTACT_PROPERTY);
                        if (defaultMail == undefined || defaultMail == "") {
                                this.updatePropertyValue(Contact.MAIL_CONTACT_PROPERTY,
 this.getParameters().contact);
                        }
                },

                trigger: function() {
                        window.location = "mailto:" + this.getPropertyValue(Contact.MAIL_CONTACT_PROPERTY);
                }

        });
        Y.namespace("GCUI");
        Y.GCUI.Contact = Contact;

}, "1.0.0", {requires: ["project-widgetbutton"] });
```

The important part here is the handling of the properties. By parsing the XML file, the widgets engine will call the `addProperties()` when the instance is selected in the design tool if the widget possesses properties. `updatePropertyValue()` is used here to update the value of the property in the JSON. It will then be saved in the project description and then stored in the database.

Tutorial

Introduction

First and foremost, we cannot recommend strongly enough that users read the documentation on building widgets at the following address: YUI [http://yuilibrary.com/] and OpenLayers [http://openlayers.org/]. Many examples are also available at YUI exemples [http://yuilibrary.com/yui/docs/guides/] and OpenLayers exemples [http://openlayers.org/dev/examples/]. In addition, the community around these 2 libraries is very active, and the forums are also well-stocked with many answers to questions that will be of interest.

This tutorial describes several examples of how to create your own widget. Before you start, make sure that Eclipse is installed on your workstation as well as the Maven module. This will allow you to generate a JAR file that contains your widgets.

Easy Geoweb accepts several JAR files with widgets, but each widget must possess a unique identifier.

A widget is a module that will be dynamically loaded in the portal, and which breaks down into 3 files:

- `fichier XML` to describe the resources needed (javascript, poperties, image) and to define the identification of the module.
- `fichier JS` to configure the actions and the interactions of the widget with the portal.
- `CSSfile` to configure the style of the widget.

The set of sources needed to complete this tutorial is supplied with the documentation.

Maven Project

Creation of the project

In Eclipse, create a new Maven project.

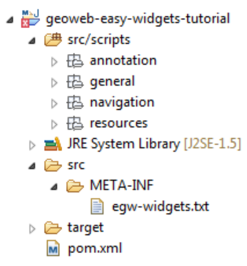
Choose Create a simple project (skype archetype selection). For this tutorial, the description of the artifact is:

- `Group Id` : com.geoconcept.geoweb.product
- `Artifact Id` : geoweb-easy-widgets-tutorial
- `Version` : 0.0.1-SNAPSHOT
- `Packaging` : jar

Once the new project has been created, delete the main java class and the *test* and *main* directories in *src* . In the *src* directory, create a *scripts* directory with Source folder (right-click on the project, then select New → Source Folder).

Next, create a *META-INF* directory in *src*. This will contain the *egw-widgets.txt* file. Inside this file, it will be necessary to inscribe a character string, for example "easy-geoweb widgets module". This file identifies and filters the JARs so only those that possess widgets are conserved.

The final structure is as follows:



Create 4 new directories under *src/scripts*:

- general: this will contain the files in the new widgets. This directory will be visible in the designer accordion. There are currently 8 categories/directories available for the designer (Annotation/ /*annotation*, Information on objects/*data*, General/*general*, Page layout/*layout*, Measures/*measure*, Data Modifications/*modification*, Navigation/*navigation*, View/*view*). In addition to the General directory, you should add the *annotation* and *navigation* directories in order to work the examples.
- resources: this will contain the files of resources (images, localisation, …).

In Resources, create:

- images: this will contain all the images used
- exampleLang.properties: this file contains all the default localisation strings. To add new languages to the widgets, you need to duplicate this file while adding to the name of the desired localisation file. For example, for a French localisation, the file will have to be named exampleLang_fr.properties.

Important note: a widget JAR must ONLY contain files linked to the widgets.

Configuration of the POM

In the file called pom.xml, add the following elements in order to construct the project in JAR format:

```
<build>
        <plugins>
                <plugin>
                        <groupId>org.apache.maven.plugins</groupId>
                        <artifactId>maven-jar-plugin</artifactId>
                        <configuration>
                                <classesDirectory>src</classesDirectory>
                                <outputDirectory>D:/widgets</outputDirectory>
                                <excludes>
                                        <exclude>**/pom.xml</exclude>
                                </excludes>
                                <archive>
                                        <addMavenDescriptor>false</addMavenDescriptor>
                                </archive>
                        </configuration>
                </plugin>
        </plugins>
</build>
```

classesDirectory is the access filepath to the widget files.

outputDirectory is the filepath in which the JAR file generated will be placed.

Display a legend widget

Objective

This widget allows you to display the map legend. This will consist of a button type widget. The legend corresponds to an image displayed in a panel.

The widget is called ExampleLegend.

Initialisation

Create the 3 widget files in the *general* directory:

- ExampleLegend.xml
- ExampleLegend.js
- ExampleLegend.css

ExampleLegend.xml

This example is the simplest. Here is the configuration file:

```
<widget id="exampleLegend"
        category="standard"
        position="12"
        icon="resources/images/exampleLegend.png"
        js="general/ExampleLegend.js"
        module="geoconcept-widget-example-legend"
        display="true"
        lang="resources/exampleLang">
</widget>
```

- **`id`** : unique identifier
- **`icon`** : filepath to access the image resource for the display as a button.
- **`js`** : filepath to access the configuration file for the widget action.
- **`module`** : name of the module that will be called in the widget manager to load the widget instance.
- **`display`** : a Boolean to determine whether the widget will be displayed in the designer. Use false to create a sub-widget that will provide functionalities that are common to other widgets.
- **`lang`** : access filepath to the localisation file.

For further information, see the description of XML files in the chapter on widgets.

ExampleLegend.js

The js file allows you to build a new module. The widget created is a single button, and the action will be executed each time the user will apply pressure on the button. The method called following a mouse-click is trigger:function(). This method is defined in WidgetBase but here it suits our purpose to overload it in order to display the image of the legend. Here is the final method:

```
trigger:function() {
        var panelNode = this.createPanel(this._("widget.exampleLegend.popup.title"));

        var linkNode = Y.Node.create(this.getImageHtml());
        panelNode.appendChild(linkNode);

        this.showPanel();
},
```

The _(key), createPanel(name) and showPanel() methods belong to the basic API of a widget. Each widget has a window (a pane) that is linked with a unique identifier. It is possible to create other windows, but they must be defined directly from the widget. The localisation key must be added to the properties file of the jar.

The HTML part of the interior of the window is defined by the getImageHtml() method. A node is then created and then added to the window:

```
getImageHtml:function() {
        var imageUrl = "/" + this.getContextUrl() + "/combo?resources/images/sample_legende.png";
        var html = "<img src='" + imageUrl + "' class='imageLegend'>";
        return html;
}
```

In this method, the HTML tag is created. The getContextUrl() method belongs to the basic widget API. The image is stored under the widget resources. A Class is associated in order to define the position of the image in the window.

The full description of this widget is currently as follows:

```
YUI.add("geoconcept-widget-example-legend", function(Y) {

        /* ExampleLegend class constructor */
        function ExampleLegend(config) {
                ExampleLegend.superclass.constructor.apply(this, arguments);
        }

        /*
         * Required NAME static field, to identify the Widget class and
         * used as an event prefix, to generate class names etc. (set to the
         * class name in camel case).
         */
        ExampleLegend.NAME = "exampleLegend";

        Y.extend(ExampleLegend, Y.GCUI.WidgetButton,  {
                initializer: function(config) {
                        ExampleLegend.superclass.initializer.apply(this);
                },

                destructor: function() {
                        ExampleLegend.superclass.destructor.apply(this);
                },

                configureAction: function() {
                        ExampleLegend.superclass.configureAction.apply(this);
                },

                /*
                 * Action after click on button
                 */
                trigger:function() {
                        var panelNode = this.createPanel(this._("widget.exampleLegend.popup.title"));

                        var linkNode = Y.Node.create(this.getImageHtml());
                        panelNode.appendChild(linkNode);

                        this.showPanel();
                },

                /*
                 * Get HTML description for image legend
                 * return string html
```

```
                   */
                  getImageHtml:function() {
                          var imageUrl = "/" + this.getContextUrl() + "/combo?resources/images/
sample_legende.png";
                          var html = "<img src='" + imageUrl + "' class='imageLegend'>";
                          return html;
                  }
          });
          Y.namespace("GCUI");
          Y.GCUI.ExampleLegend = ExampleLegend;

}, "1.0.0", {requires: ["project-widgetbutton"] });
```

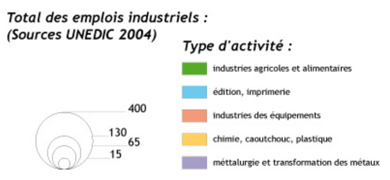
ExampleLegend.css

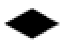
In this example, a class has been added to define the margins around the legend to improve its positioning in the window.

```
.imageLegend {
        margin: 5px;
}
```

Resources

Here are the images needed for the creation of this widget.


Total des emplois industriels :
(Sources UNEDIC 2004)
400
130
65
15
Type d'activité :
🟩 industries agricoles et alimentaires
🟦 édition, imprimerie
🟥 industries des équipements
🟨 chimie, caoutchouc, plastique
🟪 méttalurgie et transformation des métaux



Place these images in the *src/scripts/resources/images* directory.

By default, a button possesses 2 localisation keys, *label* and *description*. *label* is used in the designer accordion, and *description* is displayed when the mouse cursor glides over the button.

Take care to see that the keys ALWAYS start by widget.(id for the widget).(end of the key). For this widget, the key for the label will be widget.exampleLegend.label

Here are the base keys to put in the properties file:

```
widget.exampleLegend.label=Exemple de l\u00E9gende
widget.exampleLegend.description=Afficher la l\u00E9gende principale
```

Here is the key to display a title in the window:

```
widget.exampleLegend.popup.title=L\u00E9gende
```

The Draw a point widget

Objective

This widget allows you to draw a point on the map. It is a push button type widget, by clicking on it, the user can activates the feature. A property in the Composer will determine the image stored in the database to be displayed after a click on the map.

The widget is named ExampleDrawPOI.

Initialisation

Create the 3 files of the widget in the *annotation* directory:

- ExampleDrawPOI.xml
- ExampleDrawPOI.js
- ExampleDrawPOI.css

ExampleDrawPOI.xml

This example allows the user to add a widget property. Here is the configuration of the description file:

```
<widget id="exampleDrawPOI"
                category="standard"
                position="13"
                icon="resources/images/exampleDrawPOI.png"
                js="annotation/ExampleDrawPOI.js"
                module="geoconcept-widget-example-draw-poi"
                display="true"
                lang="resources/exampleLang">
        <properties>
                <property id="defaultImage" type="combobox">
                        <params value="poi_blue_small"/>
                </property>
        </properties>
</widget>
```

- **`properties`** : serves to add the widget properties.
- **`property`** : detail of a property composed of a unique identifier and a Class.
- **`params`** : default parameter for the corresponding parameter. This tag is not mandatory in the case where there is no default value.

For further information, see the description of XML files in the chapter on widgets.

ExampleDrawPOI.js

The widget created is a TOOL type button, and when the user clicks on the button, it remains activated. A single TOOL type button can be activated at any one time, and all the other buttons of this type are then disabled. The two methods used are activate() and de-activate. On activation, the widget calls a function to enable the listener via a click on the map. When it comes to disabling the same, the listener will be deleted. Here are 2 basic methods for the widget:

```
activate:function() {
        ExampleDrawPOI.superclass.activate.apply(this);
        this.createPoiListener();
},

deactivate:function() {
        ExampleDrawPOI.superclass.deactivate.apply(this);
        this.removeCreatePoiListener();
},
```

The creation of a listener following a mouse-click on the map is based on the address: API Javascript [http://api.geoconcept.net/htc/index.html]

```
createPoiListener:function() {
        var wExampleDrawPOI = this;

        function clickListener(dynMap){}

        clickListener.prototype.onSelect = function(x,y,xpx,ypx){
                wExampleDrawPOI.createObject(x,y);
        };

        var listener = new clickListener();
        DynMapAddMouseSelectionEventListener(this.getMap(), "clickOnMap", listener)
},
```

The method draws on the DynMapAddMouseSelectionEventListener function, to which it is possible to pass a listener (*listener*) as parameter. A mouse-click activates the createObject (x,y) function. The x and y coordinates will enable the user to position the object on the map.

The addition of the point is made in the API Javascript object layer. The image displayed to reperesent the point is defined from a url.

```
var imageUrl = this.getHost() + this.getContextUrl() + "/Image/showImage.do?name=" +

 this.getPropertyValue(ExampleDrawPOI.IMAGE_PROPERTY_NAME);
```

The getHost(), getContextUrl() and getPropertyValue(name) methods belong to the basic widget API. ExampleDrawPOI.IMAGE_PROPERTY_NAME is a constant defined in the module in the following way:

```
ExampleDrawPOI.IMAGE_PROPERTY_NAME = "defaultImage";
```

getPropertyValue(name) is the method allowing you to search for the value of a property. Each widget can benefit from properties that you need to define in its description file. All widgets possess, by default, a property relating to the choice of groups having access to this widget in the case of a private portal. Widgets of the button type possess, in addition, a property defining whether the widget is active by default. The other properties are created automatically as a function of the description file but if the type of property does not exist, you can then proceed to use the `addProperties(groups)` method.

To retrieve the images from the database, you need to call the getImages method of the REST API. The result is returned in the onSuccess() or onError() method, in the case of success or failure of a search action.

```
this.getEGWApi().getImages({
        onSuccess:function(response) {
        },
        onError:function(response) {
        }
});
```

Following retrieval of images, you need to associate the result to the drop-down list for the property using the `setComboboxOptions(propertyId, data)` method.

```
this.setComboboxOptions(ExampleDrawPOI.IMAGE_PROPERTY_NAME, images);
```

The full description of this widget is currently as follows:

```
YUI.add("geoconcept-widget-example-draw-poi", function(Y) {

        /* ExampleDrawPOI class constructor */
        function ExampleDrawPOI(config) {
                ExampleDrawPOI.superclass.constructor.apply(this, arguments);
        }

        /*
         * Required NAME static field, to identify the Widget class and
         * used as an event prefix, to generate class names etc. (set to the
         * class name in camel case).
         */
        ExampleDrawPOI.NAME = "exampleDrawPOI";
        ExampleDrawPOI.IMAGE_PROPERTY_NAME = "defaultImage";

        Y.extend(ExampleDrawPOI, Y.GCUI.WidgetButton,  {
                initializer: function(config) {
                        ExampleDrawPOI.superclass.initializer.apply(this);
                        this.setType(Y.GCUI.WidgetButton.TYPE_TOOL);
                        this.getImages();
                },

                destructor: function() {
                        ExampleDrawPOI.superclass.destructor.apply(this);
                },

                configureAction: function() {
                        ExampleDrawPOI.superclass.configureAction.apply(this);
                },

                activate:function() {
                        ExampleDrawPOI.superclass.activate.apply(this);
                        this.createPoiListener();
            },

            deactivate:function() {
                ExampleDrawPOI.superclass.deactivate.apply(this);
                this.removeCreatePoiListener();
            },

            createPoiListener:function() {
                        var wExampleDrawPOI = this;

                        function clickListener(dynMap){}

                        clickListener.prototype.onSelect = function(x,y,xpx,ypx){
                                wExampleDrawPOI.createObject(x,y);
                        };

                        var listener = new clickListener();
                        DynMapAddMouseSelectionEventListener(this.getMap(), "clickOnMap", listener)
                },

                // Remove create poi listener
                removeCreatePoiListener:function() {
                        DynMapRemoveListener(this.getMap(),"click","clickOnMap");
                },

            // add points from list of citysite
                createObject:function(x,y) {
                        var map = this.getMap();
                        var imageUrl = this.getHost() + this.getContextUrl() + "/Image/showImage.do?name=" +

 this.getPropertyValue(ExampleDrawPOI.IMAGE_PROPERTY_NAME);
                        var poi = {
                        mapx : x,
                        mapy : y,
                        imgsrc : imageUrl
                    };
                    map.objectLayer.addObject(poi);
                        map.objectLayer.refresh();
                },

                /**
                 * Method: Get images on geoweb server. Call egw api rest method
                 */
                getImages : function() {
                        var wExampleDrawPOI = this;

                        this.getEGWApi().getImages({
                                onSuccess : function(response) {
                                        // For images properties, we add the images of geoweb
                                        // database
                                        var result = response.result;
                                        var images = [""];

                                        for (i = 0; i < result.length; i++) {
                                                var image = result[i];
                                                images.push(image.name);
                                        }


 wExampleDrawPOI.setComboboxOptions(ExampleDrawPOI.IMAGE_PROPERTY_NAME, images);
                                },
                                onError : function(response) {
                                        alert("Poi getImages error " + response.status + ", "
                                                        + response.message);
                                }
                        });
                }
        });
        Y.namespace("GCUI");
        Y.GCUI.ExampleDrawPOI = ExampleDrawPOI;

}, "1.0.0", {requires: ["project-widgetbutton"] });
```

ExampleDrawPOI.css

In this example, we will not use any default style.

Resources

Here is the image needed for the creation of this widget.

Here are the base keys to put in the properties file:

```
widget.exampleDrawPOI.label=Exemple de points
widget.exampleDrawPOI.description=Cr\u00E9er un point sur la carte
```

To the title name of the property of the choice of image, it will be necessary to add the following key:

```
widget.exampleDrawPOI.defaultImage=Images :
```

For more information about resources, refer to the widget called ExampleLegend.

The Display X and Y coordinates widget

Objective

This widget allows display of X and Y coordinates following a click on the map.

The widget is called ExampleXY.

Initialisation

Create the 3 widget files in the *navigation* directory:

- ExampleXY.xml
- ExampleXY.js
- ExampleXY.css

ExampleXY.xml

Here is the configuration of the description file:

```
<widget id="exampleXY"
        category="standard"
        position="14"
        icon="resources/images/exampleXY.png"
        js="navigation/ExampleXY.js"
        module="geoconcept-widget-example-xy"
        display="true"
        lang="resources/exampleLang">
</widget>
```

For further information, see the description of XML files in the chapter on widgets.

ExampleXY.js

The widget created is of the basic type. It has no structure, and it is up to the developer to create its interface. The man-machine-interface for the widget has to be defined in the getWidget() method. For the display in the design tool, the widget must be de-activated. To do this, 2 methods are available in the basic widget API: isPortal() and getWrapper(). The first of these defines whether the application is displayed in the portal or the design tool. The second enables retrieval of the template that can be superposed over it to disable the widget.

```
getWidget: function() {
        var htmlContent = "<div class='xyBG'><div class='xyField'><label>"+this._("widget.exampleXY.x") +
        "</label></br><input type='text' id='x' size='15' disabled='disabled'/></div>" +
    "<div class='xyField'><label>" + this._("widget.exampleXY.y") +
    "</label></br><input type='text' id='y' size='15' disabled='disabled'/></div>" +
    "<div class='xyButton egw-btn'><button id='searchXY'>" + this._("widget.exampleXY.button") + "</
button></div>";

        if (!this.isPortal()) {
                htmlContent = htmlContent + this.getWrapper();
        }
        htmlContent += "</div>";
        return ("<div id='"+this.getId()+"'>"+htmlContent+"</div>");
},
```

As for the example, this widget deploys a listener on the mouse-click event. The activation will then occur on a mouse-click on the *searchXY button. The event positioned in the _configureAction (1)* function will activate the listener. The _configureAction() function is called following construction of the widget in the man-machine-interface.

```
configureAction: function() {
        ExampleXY.superclass.configureAction.apply(this);
        if (this.isPortal()) {
                Y.one("#searchXY").on("click", this.clickOnMapListener, this);
        }
},
```

To disable the listener, the overloaded function is *reinitView()*. This function can be called by the Y.fire("reinitView") event (used notably by the Erase widget supplied by default).

```
reinitView: function() {
        this.removeClickOnMapListener();
},
```

Following the activation of this widget, each click on the map will allow the display of x and y coordinates of the mouse.

```
setXY: function(x,y) {
        var map = this.getMap();
        Y.one("#x").set("value", x*map.precision);
        Y.one("#y").set("value", y*map.precision);
}
```

The full description of this widget is currently as follows:

```
YUI.add("geoconcept-widget-example-xy", function(Y) {

        /* ExampleXY class constructor */
        function ExampleXY(config) {
                ExampleXY.superclass.constructor.apply(this, arguments);
        }

        /*
         * Required NAME static field, to identify the Widget class and
         * used as an event prefix, to generate class names etc. (set to the
         * class name in camel case).
         */
        ExampleXY.NAME = "exampleXY";

        Y.extend(ExampleXY, Y.GCUI.WidgetBase,  {
                initializer: function(config) {
                        ExampleXY.superclass.initializer.apply(this);
                },

                destructor: function() {
                        ExampleXY.superclass.destructor.apply(this);
                },

                configureAction: function() {
                        ExampleXY.superclass.configureAction.apply(this);
                        if (this.isPortal()) {
                                Y.one("#searchXY").on("click", this.clickOnMapListener, this);
                        }
                },

                reinitView: function() {
                        this.removeClickOnMapListener();
                },

                getWidget: function() {
                        var htmlContent = "<div class='xyBG'><div
 class='xyField'><label>"+this._("widget.exampleXY.x") +
                        "</label></br><input type='text' id='x' size='15' disabled='disabled'/></div>" +
                    "<div class='xyField'><label>" + this._("widget.exampleXY.y") +
                    "</label></br><input type='text' id='y' size='15' disabled='disabled'/></div>" +
                    "<div class='xyButton egw-btn'><button id='searchXY'>" +
 this._("widget.exampleXY.button") + "</button></div>";

                        if (!this.isPortal()) {
                                htmlContent = htmlContent + this.getWrapper();
                        }
                        htmlContent += "</div>";
                        return ("<div id='"+this.getId()+"'>"+htmlContent+"</div>");
                },

                clickOnMapListener: function() {
                        var wExampleXY = this;

                        function clickListener(){}

                        clickListener.prototype.onSelect = function(x,y,xpx,ypx){
                                wExampleXY.setXY(x,y);
                        };

                        var listener = new clickListener();
                        DynMapAddMouseSelectionEventListener(this.getMap(), "getXY", listener)
                },

                // Remove create poi listener
                removeClickOnMapListener: function() {
                        DynMapRemoveListener(this.getMap(),"click","getXY");
                },

                setXY: function(x,y) {
                        var map = this.getMap();
                        Y.one("#x").set("value", x*map.precision);
                        Y.one("#y").set("value", y*map.precision);
                }
        });
        Y.namespace("GCUI");
        Y.GCUI.ExampleXY = ExampleXY;

}, "1.0.0", {requires: ["project-widgetbase"] });
```

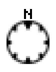
ExampleXY.css

In this example, as the widget is not a button, the css file must be enriched to personalise the tool. Below is one of the possible displays for this widget.

```
.xyField {
        padding: 0 10px;
        float: left;
}

.xyButton {
        float: left;
        padding: 13px 0px 0px;
}

.xyBG {
        width: 100%;
        height: 40px;
}

.xyField label{
        font-weight: bold;
}
```

Resources

Here is the image needed for the creation of this widget.



Here are the base keys to put in the properties file:

```
widget.exampleXY.label=Exemple de coordonn\u00E9es
widget.exampleXY.description=R\u00E9cup\u00E9rer les coordonn\u00E9es \u00E0 partir d\'un clic
```

For the other widget texts, it will be necessary to add the following keys:

```
widget.exampleXY.button=Activer
widget.exampleXY.x=X
widget.exampleXY.y=Y
```

For more information about resources, refer to the widget called ExampleLegend.

Javascript API

The Javascript API provides a comprehensive library for interacting with the map in a web interface.

Use the following documentation [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-js-intro.html] to access a description of Javascript functions available for you to build your web application and enrich it with the functionalities you require.

Web Services

The various types of Web services provided by Geoconcept Web are explained in the following sections of this guide, taking as starting point the different types of architecture that are possible. For each type of web service, this guide describes which functionalities are offered, and how to configure and use them, with the help of examples of queries and responses, or again FAQ.

WMS and WMTS

In addition to the components that are configurable and utilisable via Geoconcept Web, as described in the previous pargraphs, Geoconcept Web enables publishing of Geoconcept maps, while respecting the standards of the Open Geospatial Consortium (OGC). In this way, two types of Web Services enable publishing a Geoconcept map:

- Web Map Service (WMS): this is an OGC web service standard that enables maps to be produced dynamically from georeferenced data; the server is called each time there is a query from a client workstation. The Geoconcept map can then be shared on a server and is visible via client workstations reading WMS format, as is the Geoconcept GIS.
- Web Map Tiles Service (WMTS): this is a standard describing the way to publish cartographic data in the form of predefined tiles. The WMTS is a complement to the WMS. As compared to the WMS, the main advantage of the WMTS is that it offers better performances in the publishing of cartographic data, as the tiles are stored in a cache at the time of the first generation. On the other hand, the WMTS is not recommended when the data are updated frequently, since it then means the cache has to be invalidated for the data to be brought up to date.

Minimum requirements

The `Administration` module for the web application offers a functionality for publishing view tabs in Geoconcept maps in WMS or WMTS format, while respecting OGC standards.

It is therefore necessary to have deployed the web application, following the steps described in the Geoconcept Web installation guide.

Creating layers in WMS or WMTS format

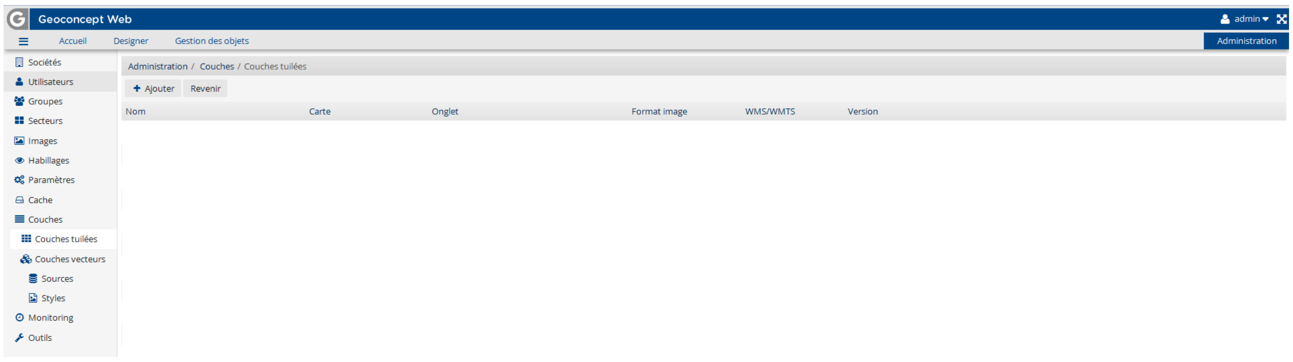
This functionality features an interface for creating layers to be published in WMS or WMTS format.

Don’t forget to activate WMS or WMTS services (refer to the section in the Geoconcept Web Installation Guide / Settings / Activating Geoconcept Web WMS and WMTS.

The procedure for publishing a view tab in WMS or WMTS format is as follows:

- In the `Administration` tab and then the `Layers` / `Tiled layers` option, select `Add` :

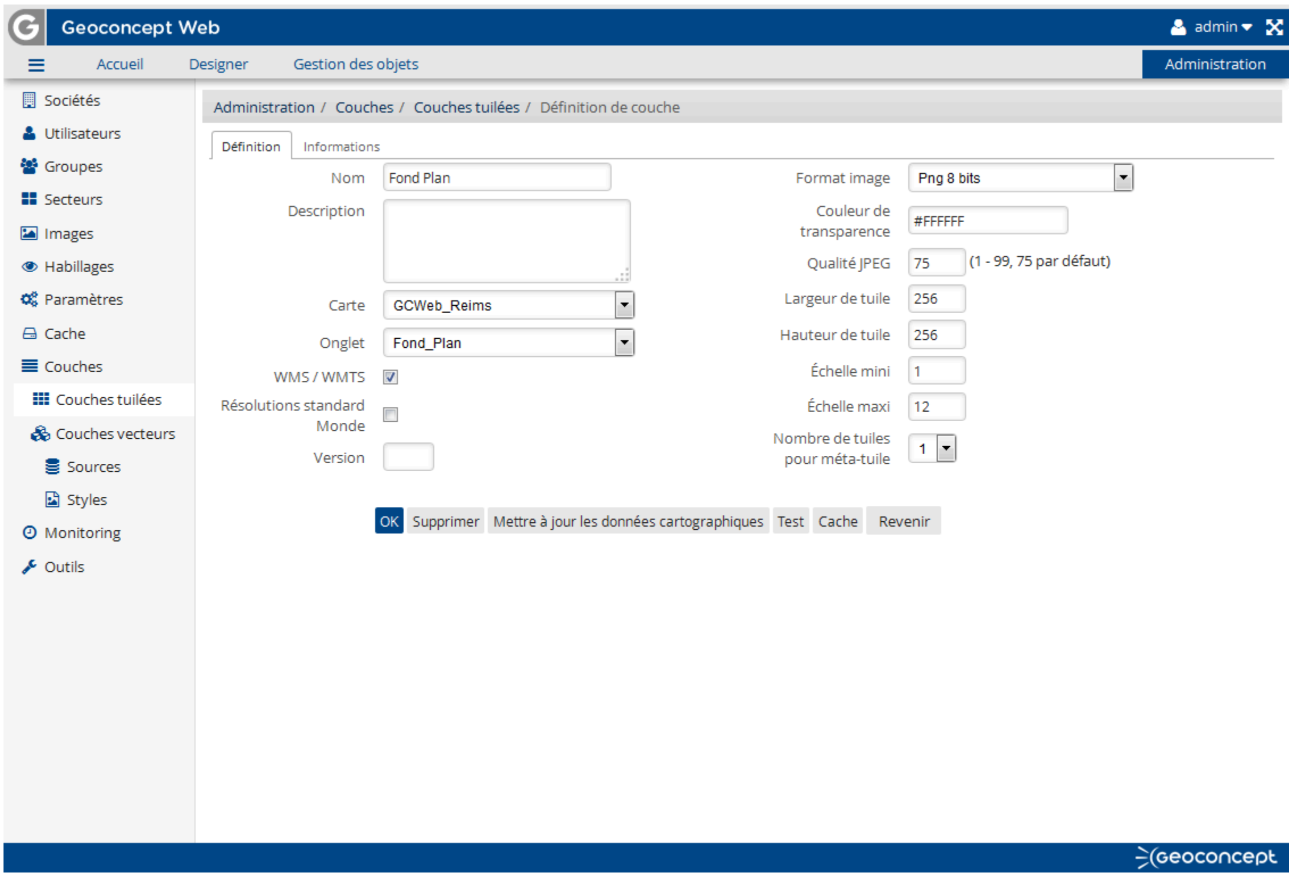
Adding a layer



Nom	Carte	Onglet	Format image	WMS/WMTS	Version

- In the text entry interface, fill in values for the following fields:
  - Name : type in the name that will be visible to users when they consult the WMS or WMTS service from a third party application (for example, the Geoconcept GIS),
  - the '*WMS/WMTS*' check-box should be checked,
  - Map: choose the map to publish from those shown in the drop-down menu,
  - Tab : select the map view tab to publish,
  - Select the image format required:
    - PNG format is recommeded as an image format because it allows a better resolution for images generated,
    - The format called PNG with transparency allows transformation of «the colour white» on a Geoconcept map so it is transparent in the image generated. This format is useful for displaying layers that are superimposed in relation to one another.

Type in the items related to the Geoconcept view tab



Nom	Fond Plan	Format image	Png 8 bits
Description		Couleur de transparence	#FFFFFF
		Qualité JPEG	75 (1 - 99, 75 par défaut)
Carte	GCWeb_Reims	Largeur de tuile	256
Onglet	Fond_Plan	Hauteur de tuile	256
WMS / WMTS	✓	Échelle mini	1
Résolutions standard Monde	☐	Échelle maxi	12
Version		Nombre de tuiles pour méta-tuile	1

To be on the safe side, do not put any special characters or any spaces in the layer name, the map name or the view tab name.

- Once these parameters have been assigned values, click on OK .

Results

A layer configured in the portal is accessible in WMS or WMTS format with the same configuration. The format requested depends on the client : if they want WMS, they specify this in the query with the term WMS. The procedure is the same for WMTS.

Each published layer is available in three projection systems:

- the original Geoconcept map projection
- in WGS84 latitude / longitude (epsg:4326)
- in spherical Mercator (epsg:3857)

WMS - Web Map Service

GetCapabilities:

http://<server>/geoconcept-web/wms?request=GetCapabilities

http://<server>/geoconcept-web/wms?request=GetCapabilities

The view tab for your Geoconcept map is published in WMS format. It is accessible via a third party display tool that is capable of reading WMS format via the following query :

http://<server>/<webapp>/wms

Example of a WMS image request query: http://gcweb.geoconcept.com/gws/wms?request=GetMap&VERSION=1.3.0&CRS=epsg:27572&BBOX=551533.765952,2313284.041120,566370.515430,2321072
%2fpng&WIDTH=800&HEIGHT=600

WMTS - Web Map Tile Service

KVP and REST protocols are supported for WMTS by the application. WMTS 1.0.0 OGC is also supported (http://www.opengeospatial.org/standards/wmts).

GetCapabilities:

- KVP encoding : http://<server>/<webapp>/wmts?request=GetCapabilities&version=1.0.0&service=WMTS

Example: http://gcweb.geoconcept.com/gws/wmts?request=GetCapabilities&version=1.0.0&service=WMTS

- REST encoding : http://<server>/<webapp>/wmts/1.0.0/WmtsCapabilities.xml

Example: http://gcweb.geoconcept.com/gws/wmts/1.0.0/WmtsCapabilities.xml

http://<server>/<webapp>/gws/wmts?request=GetCapabilities

GetTile:

- KVP encoding : http://<server>/<webapp>/wmts?SERVICE=WMTS&REQUEST=GetTile&VERSION=1.0.0&LAYER=<layerIdentifier>&STYLE=<StyleIdentifier>&TILEMA
  png

Example: http://gcweb.geoconcept.com/gws/wmts?SERVICE=WMTS&REQUEST=GetTile&VERSION=1.0.0&LAYER=France&STYLE=Server2&TILEMATRIXSET=epsg%3A27572&TILEMATRIX=0&TILEROW=0&TILECOL=0&FORMAT=image%2Fpng

- REST encoding : la ResourceURL d’un layer est décrite dans le getCapabilities : http://<server>/<webapp>/wmts/<layerIdentifier>/<styleIdentifier>/<TileMatrixSet>/<TileMatrix>/<TileRow>/<TileCol>.png

Example: http://gcweb.geoconcept.com/gws/wmts/France/Server2/epsg:27572/0/0/0.png

- The view tab for your Geoconcept map is published in WMTS format. It is accessible via a third party display tool that is capable of reading WMTS format via the following query:

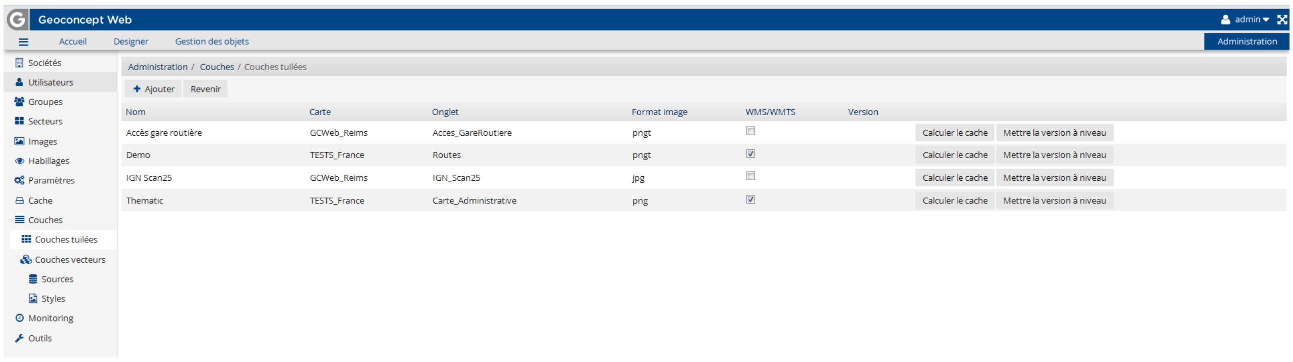
http://<server>/<webapp>/wmts

Example of a WMS result

The following example was configured for displaying in the Standalone Geoconcept GIS solution (client for reading WMS layers) layers configured in the WMS server described earlier.

The Demo and Thematic view tabs have been configured as WMS or WMTS layers.

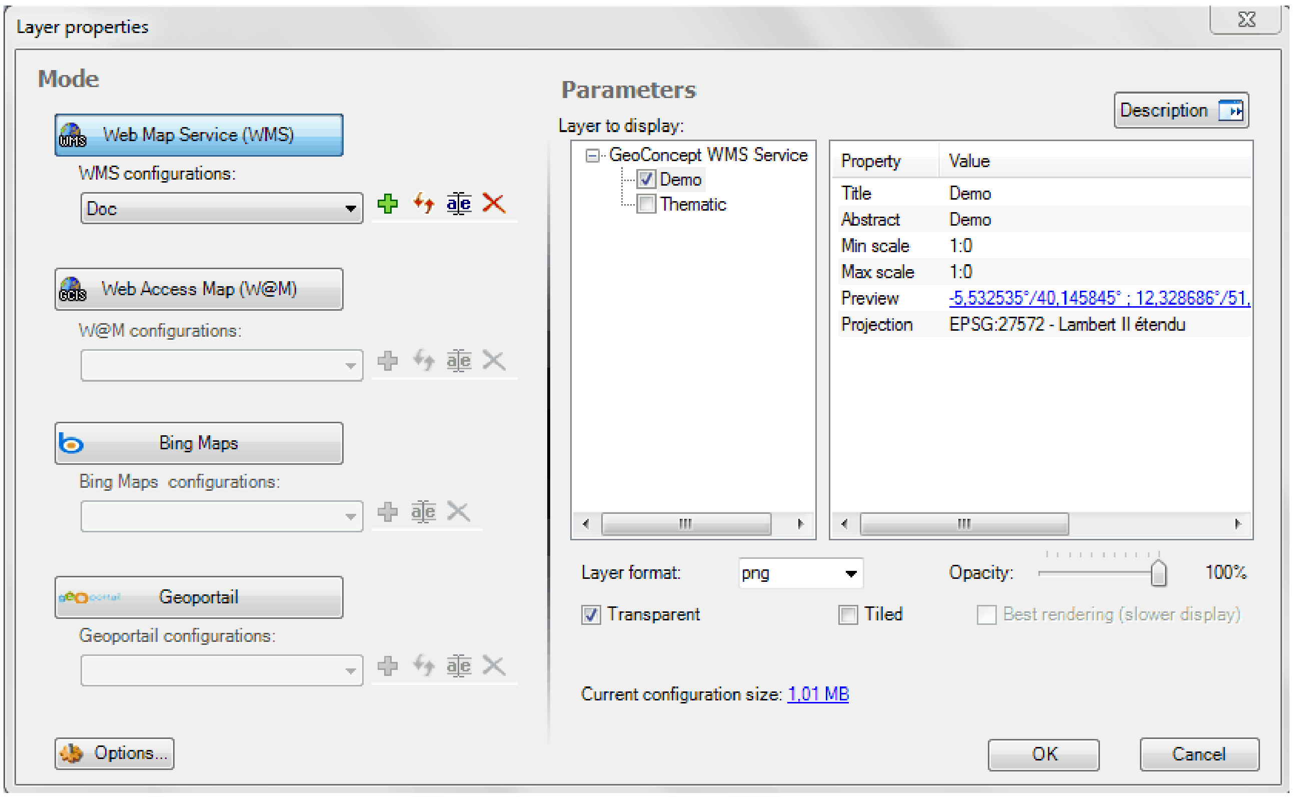
Configuration of WMS / WMTS layers



Nom	Carte	Onglet	Format image	WMS/WMTS	Version		
Accès gare routière	GCWeb_Reims	Acces_GareRoutiere	pngt	☐		Calculer le cache	Mettre la version à niveau
Demo	TESTS_France	Routes	pngt	✓		Calculer le cache	Mettre la version à niveau
IGN Scan25	GCWeb_Reims	IGN_Scan25	jpg	☐		Calculer le cache	Mettre la version à niveau
Thematic	TESTS_France	Carte_Administrative	png	✓		Calculer le cache	Mettre la version à niveau

When the WMS server is called from Geoconcept, the dialogue box will allow the display of those layers that have been configured :

Request for WMS layers via Geoconcept



WMS configurations: Doc

Layer to display:
- GeoConcept WMS Service
  - [✓] Demo
  - [ ] Thematic

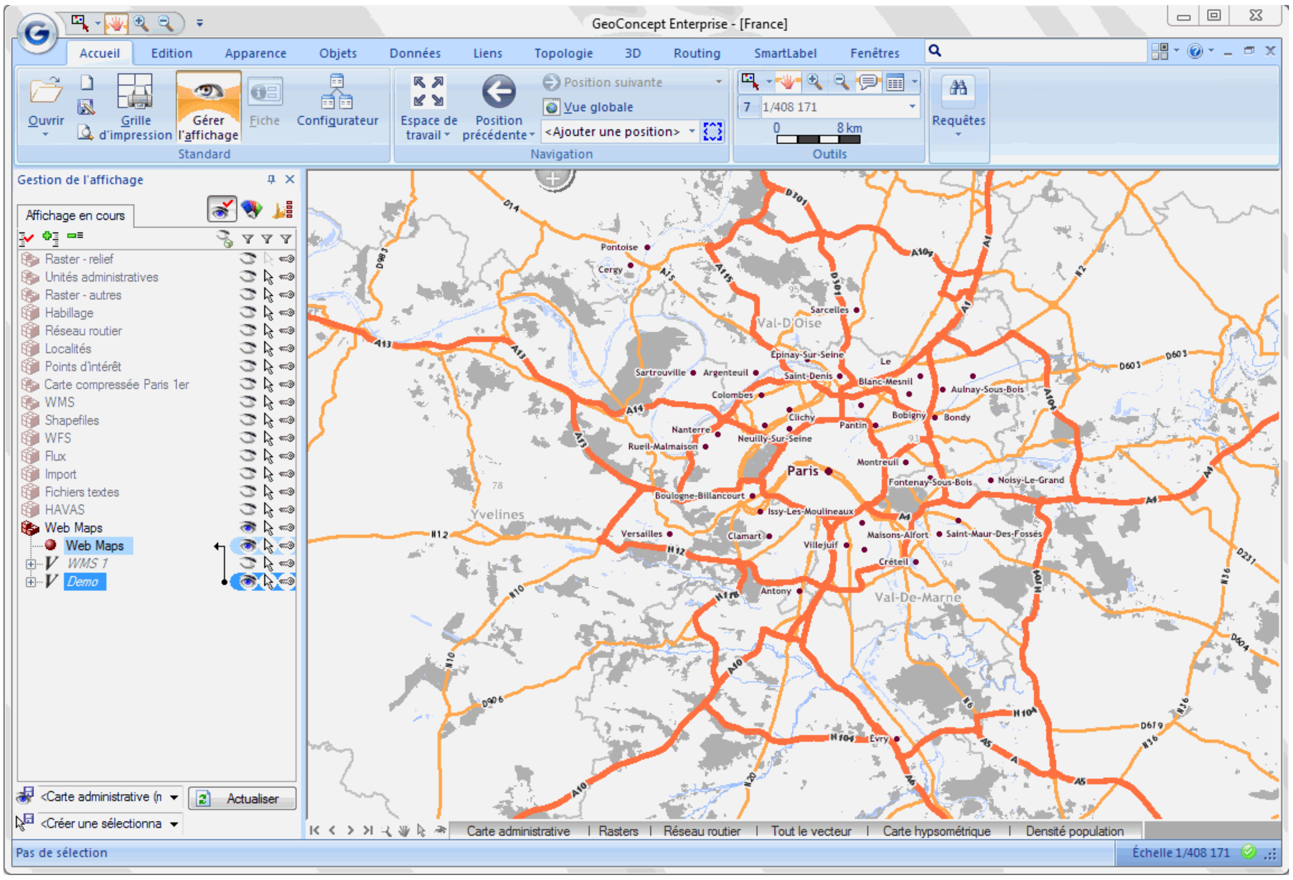
Property	Value
Title	Demo
Abstract	Demo
Min scale	1:0
Max scale	1:0
Preview	-5,532535°/40,145845° ; 12,328686°/51,
Projection	EPSG:27572 - Lambert II étendu

Layer format: png
Opacity: 100%
[✓] Transparent
[ ] Tiled
[ ] Best rendering (slower display)

Current configuration size: 1,01 MB

The Demo layer is displayed via Geoconcept Webmaps.

Displaying WMS layers in Geoconcept



List of layers in a project

(fr) Cette page n'est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

This web service returns a list of all the layers in a particular project/portal.

Availability

This web service is available at all times with Geoconcept Web.

V1

Settings / properties

Input

parameter	description	optional	default
name	Project name	no	

Output

parameter	type	min/max	description
type	string	1/1	Types of layers - raster: tiled layers - vector: vector layers - groupLayer: hybrid layers - group: groups of layers
		name	string
1/1	Layer name	label	string
1/1	Label for the layer displayed in the layer manager	depth	string
1/1	0 = base level, 1 = layers in a group	internalName	string
1/1	non-used	rights	string
1/1	non-used	disabled	string
1/1	non-used	metadataUrl	string
1/1	Metadata URL	legendUrl	string
1/1	Legend URL	format	string
1/1	image format (PNG, JPG, PNG24)	transparent	string
1/1	PNG image transparency (true/ false)	visibility	string
1/1	1 = visible , O = invisible in the layer manager	opacity	string
1/1	Layer opacity (0 - 100)	background	string
1/1	Main layer (true/false)	singleTiledLayer	string
1/1	dynamic layer (true/ false)	expanded	string
1/1	Developed group (true/ false)	layerId	string
1/1	Layer identifier	legend	string
1/1	Display legend (true/ false)	isDefaultLayer	string
1/1	Main layer (true/false)	isWebmap	string
1/1	Webmap layer (true/ false)	tabname	string
1/1	Map tab	map	string

REST (GET)

Query

JSON query

```
http://<server>/<webapp>/api/easy/project/layers.json?name=Loire-Atlantique
```

Response

The response is always encoded in UTF-8.

JSON format

```
{
  "result": {
    "layers": [
      {
        "type": "vector",
        "name": "samplePoints",
        "label": "Letterbox",
        "depth": 0,
        "internalName": null,
        "rights": "",
        "disabled": false,
        "metadataUrl": null,
        "legendUrl": null,
        "visibility": 0,
        "opacity": 100,
        "layerId": "122",
        "legend": true
      },
      {
        "type": "layer",
        "name": "ADMINISTRATIVE",
        "label": "Administrative",
        "depth": 0,
        "internalName": null,
        "rights": "",
        "disabled": false,
        "metadataUrl": null,
        "legendUrl": null,
        "format": "pngt",
        "transparent": true,
        "visibility": 0,
        "opacity": 100,
        "background": false,
        "singleTiledLayer": false,
        "legend": true,
        "isDefaultLayer": false,
        "isWebmap": false,
        "tabname": "ADMINISTRATIVE",
        "map": "Loire-Atlantique"
      },
      {
        "type": "layer",
        "name": "BASEMAP",
        "label": "BASEMAP",
        "depth": 0,
        "internalName": null,
        "rights": "",
        "disabled": false,
        "metadataUrl": null,
        "legendUrl": null,
        "format": "pngt",
        "transparent": true,
        "visibility": 1,
        "opacity": 100,
        "background": true,
        "singleTiledLayer": false,
        "legend": false,
        "isDefaultLayer": true,
        "isWebmap": false,
        "tabname": "STANDARD",
        "map": "Loire-Atlantique"
      },
      {
        "type": "layer",
        "name": "COMPLETE",
        "label": "COMPLETE",
        "depth": 0,
        "internalName": null,
        "rights": "",
        "disabled": false,
        "metadataUrl": null,
        "legendUrl": null,
        "format": "png",
        "transparent": false,
        "visibility": 0,
        "opacity": 100,
        "background": false,
        "singleTiledLayer": false,
        "legend": true,
        "isDefaultLayer": false,
        "isWebmap": false,
        "tabname": "COMPLETE",
        "map": "Loire-Atlantique"
      }
    ]
  },
  "message": "Layers in project Loire-Atlantique",
  "status": "OK"
}
```

List of layers

Basic principles

This web service returns the list of all the layers declared in Geoconcept Web.

Availability

This web service is available with Geoconcept Web by setting the *services.activate.getAllLayers* parameter to true (cf. parameters section).

V1

Parameters / properties

Input

No parameter as input

Output

parameter	type	min/max	description
id	long	1/1	Identifier for the layer.
name	string	1/1	Name of the layer.
kind	string	1/1	Types of layers - RASTER: for tiled layers - VECTOR: for vector layers - HYBRID: for hybrid layers

REST (GET)

Query

JSON query

```
http://<server>/<webapp>/api/lbs/layers/getAll
```

Response

The response is always in UTF-8 format.

JSON format

```
{
  "layers": [
    {
      "id": "123",
      "name": "ADMINISTRATIVE",
      "kind": "RASTER"
    },
    {
      "id": "124",
      "name": "BASEMAP",
      "kind": "RASTER"
    },
    {
      "id": "128",
      "name": "ADMINISTRATIVE MAP",
      "kind": "RASTER"
    },
    {
      "id": "130",
      "name": "EMPTY",
      "kind": "RASTER"
    },
    {
      "id": "133",
      "name": "COMPLETE",
      "kind": "RASTER"
    },
    {
      "id": "122",
      "name": "samplePoints",
      "kind": "VECTOR"
    }
  ]
}
```

Object geometries

Basic principles

This web service returns the geometry of objects in geojson format, filtered by layer, and optionally by bounding box. Available only for vector type layers (cf. the web service for the list of layers).

Availability

This web service is available with Geoconcept Web by setting the *services.activate.getFeatures* parameter to true (cf. advanced settings section).

V1

Settings / properties

Input

parameter	description	optional	default
layerId	Vector layer identifier	no	
bbox	The filter bounding box must be in the same projection as the objects.	yes	

Output

parameter	type	min/max	description
message	string	1/1	"Get geojson features"
status	string	1/1	"OK" or "ERROR"
result	array	0/ unlimited	Result

Result (result)

parameter	type	min/max	description
id	string	1/1	null
geojson	string	1/1	geojson
pagination	string	1/1	null
sort	string	1/1	null
distinctValues	string	1/1	null
fields	array	0/ unlimited	Description of fields used in the geojson

Description of fields used in the geojson (fields)

parameter	type	min/max	description
name	string	1/1	Fieldname for the field in the database.
alias	string	1/1	Alias for the field.

parameter	type	min/max	description
type	string	1/1	Field typing.

REST (POST)

Query

Query

```
http://<server>/<webapp>/api/lbs/layers/getFeatures
```

Data (JSON)

```
{
   "layerId":"136",
   "bbox":"-262000,5982000,-258000,5988000"
}
```

Response

The response is always coded in UTF-8.

JSON format

```
{
   "message":"Get geojson features",
   "status":"OK",
   "result":{
      "id":null,
      "geojson":"{[geojson]}",
      "pagination":null,
      "sort":null,
      "distinctValues":null,
      "fields":[
         {
            "name":"co_insee_com",
            "alias":"city_code",
            "type":"Integer"
         },
         {
            "name":"lb_com",
            "alias":"city",
            "type":"String"
         },
         {
            "name":"lb_voie_ext",
            "alias":"street",
            "type":"String"
         },
         {
            "name":"va_no_voie",
            "alias":"number",
            "type":"Integer"
         }
      ]
   }
}
```

Geojson

```
{
   "type":"FeatureCollection",
   "features":[
      {
         "type":"Feature",
         "geometry":{
            "type":"Point",
            "coordinates":[
               -258650.09,
               5982645.79
            ]
         },
         "properties":{
            "co_insee_com":"44132",
            "lb_com":"PORNICHET",
            "lb_voie_ext":"AVENUE DE BONNE SOURCE",
            "va_no_voie":"150"
         },
         "id":"gw_sample_point.A5U8T4"
      },
      [...]
      }
   ]
}
```

Possible responses

The case of a layer being absent

```
{"message":null,"status":"ERROR","result":
{"id":null,"geojson":null,"pagination":null,"sort":null,"distinctValues":null,"fields":null}}
```

the case of a faulty Bounding box

```
{"message":"Error to get feature source : gw_sample_point","status":"ERROR","result":
{"id":null,"geojson":null,"pagination":null,"sort":null,"distinctValues":null,"fields":null}}
```

Coordinates transformation service

(fr) Cette page n'est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

This web service transforms pairs of coordinates from one projection to another. For example, from Lambert 2 Extended to WGS84.

Availability

This web service is available at all times with Geoconcept Web.

V1

Parameters / properties

Input

parameter	description	optional	default
inSrs	input projection (EPSG code such as epsg:4326 or wgs84)	no	
outSrs	output projection (EPSG code such as epsg:4326 or wgs84)	no	
point (or points in JSON / JSON-P)	Coordinates for points in SOAP, you have to fill the coordinates in each of the dedicated <x></x> and <y></y> tags in REST the points are separated by “,”. The points are characterised by an XY coordinate pair separated by “;”.	no	

Output

Transformed points (srsTransformResult)

parameter	type	min/max	description
point (or points in JSON / JSON-P)	geographicPoint (or array in JSON / JSON-P)	0/ unlimited	transformed points

Points(geographicPoint)

parameter	type	min/max	description
x	double	1/1	first coordinate or longitude
y	double	1/1	second coordonnée or latitude

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/srsTransformService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:srsTransform>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <inSrs>epsg:27572</inSrs>
            <!--Optional:-->
            <outSrs>epsg:4326</outSrs>
            <!--Zero or more repetitions:-->
            <point>
               <x>246087</x>
               <y>2262265</y>
            </point>
             <point>
               <x>255081</x>
               <y>2262679</y>
            </point>
            <point>
               <x>298976</x>
               <y>2254643</y>
            </point>
         </request>
      </sch:srsTransform>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:srsTransformResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <srsTransformResult>
            <status>OK</status>
            <point>
               <x>-2.343140132718342</x>
               <y>47.26525078744117</y>
            </point>
            <point>
               <x>-2.2248034057039128</x>
               <y>47.27372258177934</y>
            </point>
            <point>
               <x>-1.6400114587002539</x>
               <y>47.22297983102492</y>
            </point>
         </srsTransformResult>
      </ns2:srsTransformResponse>
   </soap:Body>
</soap:Envelope>
```

REST (POST)

Query

Query

```
http://<server>/<webapp>/api/lbs/srsTransform.xml
```

Data (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<srsTransformRequest>
  <inSrs>epsg:27572</inSrs>
  <outSrs>epsg:4326</outSrs>
  <point>
    <x>246087</x>
    <y>2262265</y>
  </point>
```

```
    <point>
      <x>255081</x>
      <y>2262679</y>
     </point>
     <point>
      <x>298976</x>
      <y>2254643</y>
      </point>
</srsTransformRequest>
```

Response

The response is always in UTF-8 format.

XML format

```
<srsTransformResult>
   <status>OK</status>
   <point>
      <x>-2.343140132718342</x>
      <y>47.26525078744117</y>
   </point>
   <point>
      <x>-2.2248034057039128</x>
      <y>47.27372258177934</y>
   </point>
   <point>
      <x>-1.6400114587002539</x>
      <y>47.22297983102492</y>
   </point>
</srsTransformResult>
```

REST (GET)

Query

JSON query

```
http://<server>/<webapp>/api/lbs/srsTransform.json?
&inSrs=epsg:27572&outSrs=epsg:4326&points=246087,2262265;255081,2262679;298976,2254643
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/srsTransform.json?
&inSrs=epsg:27572&outSrs=epsg:4326&points=246087,2262265;255081,2262679;298976,2254643&callback=MyCallBack
```

XML query

```
http://<server>/<webapp>/api/lbs/srsTransform.xml?
&inSrs=epsg:27572&outSrs=epsg:4326&points=246087,2262265;255081,2262679;298976,2254643
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message": null,
   "status": "OK",
   "points":    [
            {
         "x": -2.343140132718342,
         "y": 47.26525078744117
      },
            {
         "x": -2.2248034057039128,
         "y": 47.27372258177934
      },
            {
         "x": -1.6400114587002539,
         "y": 47.22297983102492
      }
   ]
}
```

JSON-P format

```
myCallback(
    {
       "message": null,
       "status": "OK",
       "points":    [
                {
             "x": -2.343140132718342,
             "y": 47.26525078744117
          },
                {
             "x": -2.2248034057039128,
             "y": 47.27372258177934
          },
                {
             "x": -1.6400114587002539,
             "y": 47.22297983102492
          }
       ]
    }
);
```

XML format

```
<srsTransformResult>
   <status>OK</status>
   <point>
      <x>-2.343140132718342</x>
      <y>47.26525078744117</y>
   </point>
   <point>
      <x>-2.2248034057039128</x>
      <y>47.27372258177934</y>
   </point>
   <point>
      <x>-1.6400114587002539</x>
      <y>47.22297983102492</y>
   </point>
</srsTransformResult>
```

Possible responses

Case of an itinerary found (routeTransformResult/status is OK)

```
<srsTransformResult>
    <status>OK</status>
    <point>
       <x>-2.343140132718342</x>
       <y>47.26525078744117</y>
    </point>
    <point>
       <x>-2.2248034057039128</x>
       <y>47.27372258177934</y>
    </point>
    <point>
       <x>-1.6400114587002539</x>
       <y>47.22297983102492</y>
    </point>
</srsTransformResult>
```

Case of an empty start projection

```
<soap:Fault>
    <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
    <faultstring>inSrs can't be null or empty</faultstring>
</soap:Fault>
```

Case of an empty destination projection

```
<soap:Fault>
    <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
    <faultstring>outSrs can't be null or empty</faultstring>
</soap:Fault>
```

Case of an incorrect projection

```
<soap:Fault>
    <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
    <faultstring>inSrs or outSrs do not exist</faultstring>
</soap:Fault>
```

Case of a coordinate for an empty point

```
<soap:Fault>
    <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
    <faultstring>Unmarshalling Error:</faultstring>
</soap:Fault>
```

Case of a faulty typing entry on the coordinates for a series of points

```
<soap:Fault>
    <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
    <faultstring>Unmarshalling Error: 488degree
488degree
488degree
488degre</faultstring>
</soap:Fault>
```

Autocompletion

Basic principles

This automatic completion service, or autocomplete, saves internauts time when entering a location or address. The function suggests likely locations as the letters of an address are typed in, or even as they are deleted, in the localisation input field.

example: the internaut enters "avenue des cham», and the service returns «Avenue des Champs-Élysées, Paris», as well as other addresses starting with the same string.

This service takes as input a character string, as an option, an identifier for a territory, and an identifier of the type of entity being sought. From an optimised index, it is capable of instantly providing an extract of all the location names (addresses) starting with this string, in this territory, and of this type.

The location names are ranked or graded in reference files according to a weighting that allows the most probable solutions to be presented first, taking into account the size of the town.

You will need to copy the reference files and configure the service thanks to the Administration interface of Geoconcept Web, see chapter Installing Geoconcept Web / Settings/ UGC parameter settings / Autocomplete web service.

This web service is not part of the geocoding operation. The geocoding web service can be used with the data received from the autocomplete service, taking the address found by the autocomplete module as an entry parameter of the geocoder.

To ensure optimised performances, we advise building the geocoding from the result of the autocomplete operation with separate fields, and not basing the geocoding on fulltext fields (both of these are provided in the autocomplete result).

We recommend a 64-bit UGC Server architecture since autocompletion requires significant resources.

For performance reasons, (since the service has to be called each time the internaut types a character at the keyboard) it will be published in the form of a JSON service. As an option, it can also be called in the form of a JSONP service.

Availability

This web service is available at all times with Geoconcept Web with the reference files.

Version change

Earlier versions of the web service are conserved in Geoconcept Web to ensure compatibility with previous developments. It is recommended that you use the most recent version.

Changes in relation to v2

- "zipcode" parameter is renamed "postCode"

V2

Parameters / properties

Input

parameter	description	optional	default
text	Location address to enter	yes	
type	Allows selection of the source of the autocomplete files indicating the alias for the latter: if a single autocomplete source exists, the value should remain empty if several sources exist for the autocomplete function, you will need to use the aliases of the latter, defined beforehand cf. Geoconcept Web / Administration / Parameters `geocoder.autocomplete.alias.<alias_name>.datasource` the value of this key must indicate the name of the directory containing the series of files needed for autocompletion.	yes	
terr	Filter on the territory where one wants to search for the localising elements in question. example, if you want the query to apply to a town, it will be necessary to enter a postcode : 92220 if you want the search to be applied to several towns or departments, you need to separate the territories with a ";" : 75;78;92 if the value remains empty, the search will be applied to all of the territory covered by the autocomplete files and the results will be sorted in increasing streets alphabetical order.	yes	
maximumResponses	maximum number of results for addresses in the response. if the value is left empty, the parameter defined in the Administration will apply	yes	cf. Geoconcept Web / Administration / Parameters `geocoder.autocomplete.ma`

As output

property	type	min/max	description
city	string	0/1	town found
classification	integer	1/1	code indicating in which category the town returned is found by the webservice. example, the categorisation of HERE tables ranges from 1 (the biggest towns) to 8 (village or hamlet)
country	string	0/1	country on two letters (ISO code 3166-1 or ccTLD) for example "fr" when the result is found in France
fulltext	string	0/1	street found with a specific nomenclature including the post code and the name of the town
street	string	1/1	street found

property	type	min/max	description
postCode	string	0/1	post code found

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/autoCompleteService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:autocompleteV2>
         <!--Optional:-->
         <text>rue noire</text>
         <!--Optional:-->
         <type></type>
         <!--Optional:-->
         <terr>44</terr>
         <!--Optional:-->
         <maximumResponses>3</maximumResponses>
      </sch:autocompleteV2>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:autocompleteV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <AutoCompleteResults>
            <status>OK</status>
            <result>
               <city>NANTES</city>
               <classification>2</classification>
               <country>FR</country>
               <fulltext>RUE NOIRE, 44000 NANTES</fulltext>
               <postCode>44000</postCode>
               <street>RUE NOIRE</street>
            </result>
            <result>
               <city>BLAIN</city>
               <classification>5</classification>
               <country>FR</country>
               <fulltext>RUE NOIRE, 44130 BLAIN</fulltext>
               <postCode>44130</postCode>
               <street>RUE NOIRE</street>
            </result>
            <result>
               <city>NOTRE-DAME-DES-LANDES</city>
               <classification>7</classification>
               <country>FR</country>
               <fulltext>RUE NOIRE, 44130 NOTRE-DAME-DES-LANDES</fulltext>
               <postCode>44130</postCode>
               <street>RUE NOIRE</street>
            </result>
         </AutoCompleteResults>
      </ns2:autocompleteV2Response>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/autocomplete.json?text=rue%20noire&terr=44&maximumResponses=3
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/autocomplete.json?text=rue
%20noire&terr=44&maximumResponses=3&callback=myFonction
```

XML query

```
http://<server>/<webapp>/api/lbs/autocomplete.xml?text=rue%20noire&terr=44&maximumResponses=3
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message":null,
   "status":"OK",
   "results":[
      {
         "fulltext":"RUE NOIRE, 44000 NANTES",
         "country":"FR",
         "classification":2,
         "street":"RUE NOIRE",
         "city":"NANTES",
         "zipcode":"44000"
      },
      {
         "fulltext":"RUE NOIRE, 44130 BLAIN",
         "country":"FR",
         "classification":5,
         "street":"RUE NOIRE",
         "city":"BLAIN",
         "zipcode":"44130"
      },
      {
         "fulltext":"RUE NOIRE, 44130 NOTRE-DAME-DES-LANDES",
         "country":"FR",
         "classification":7,
         "street":"RUE NOIRE",
         "city":"NOTRE-DAME-DES-LANDES",
         "zipcode":"44130"
      }
   ]
}
```

JSON-P format

```
myFonction(
    {
   "message":null,
   "status":"OK",
   "results":[
      {
         "fulltext":"RUE NOIRE, 44000 NANTES",
         "country":"FR",
         "classification":2,
         "street":"RUE NOIRE",
         "city":"NANTES",
         "zipcode":"44000"
      },
      {
         "fulltext":"RUE NOIRE, 44130 BLAIN",
         "country":"FR",
         "classification":5,
         "street":"RUE NOIRE",
         "city":"BLAIN",
         "zipcode":"44130"
      },
      {
         "fulltext":"RUE NOIRE, 44130 NOTRE-DAME-DES-LANDES",
         "country":"FR",
         "classification":7,
         "street":"RUE NOIRE",
         "city":"NOTRE-DAME-DES-LANDES",
         "zipcode":"44130"
      }
   ]
}
);
```

XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<autoCompleteResults>
   <status>OK</status>
   <result>
      <city>NANTES</city>
      <classification>2</classification>
      <country>FR</country>
      <fulltext>RUE NOIRE, 44000 NANTES</fulltext>
      <street>RUE NOIRE</street>
      <zipcode>44000</zipcode>
   </result>
   <result>
      <city>BLAIN</city>
      <classification>5</classification>
      <country>FR</country>
      <fulltext>RUE NOIRE, 44130 BLAIN</fulltext>
      <street>RUE NOIRE</street>
      <zipcode>44130</zipcode>
   </result>
   <result>
      <city>NOTRE-DAME-DES-LANDES</city>
      <classification>7</classification>
      <country>FR</country>
      <fulltext>RUE NOIRE, 44130 NOTRE-DAME-DES-LANDES</fulltext>
      <street>RUE NOIRE</street>
```

```
        <zipcode>44130</zipcode>
    </result>
</autoCompleteResults>
```

Possible responses

The case of addresses that are found (autocompleteResponse/AutoCompleteResults/status est OK)

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:autocompleteV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <AutoCompleteResults>
            <status>OK</status>
            <result>
               <city>NANTES</city>
               <classification>2</classification>
               <country>FR</country>
               <fulltext>RUE NOIRE, 44000 NANTES</fulltext>
               <postCode>44000</postCode>
               <street>RUE NOIRE</street>
            </result>
            <result>
               <city>BLAIN</city>
               <classification>5</classification>
               <country>FR</country>
               <fulltext>RUE NOIRE, 44130 BLAIN</fulltext>
               <postCode>44130</postCode>
               <street>RUE NOIRE</street>
            </result>
            <result>
               <city>NOTRE-DAME-DES-LANDES</city>
               <classification>7</classification>
               <country>FR</country>
               <fulltext>RUE NOIRE, 44130 NOTRE-DAME-DES-LANDES</fulltext>
               <postCode>44130</postCode>
               <street>RUE NOIRE</street>
            </result>
         </AutoCompleteResults>
      </ns2:autocompleteV2Response>
   </soap:Body>
</soap:Envelope>
```

The case of an address that does not exist (autocompleteResponse/AutoCompleteResults/status is ERROR)

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:autocompleteV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <AutoCompleteResults>
            <status>ERROR</status>
         </AutoCompleteResults>
      </ns2:autocompleteV2Response>
   </soap:Body>
</soap:Envelope>
```

The case of a query having an XML error or not respecting the WSDL ⇒ error with faultstring that contains the description

```
<soap:Fault>
```

```
        <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
        <faultstring>Message part {http://geoconcept.com/gc/schemas}geocodeABCD was not recognized.  (Does
 it exist in service WSDL?)</faultstring>
</soap:Fault>
```

The case of a query with a non-existant territory (autocompleteResponse/AutoCompleteResults/status est ERROR)

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:autocompleteResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <AutoCompleteResults>
            <status>ERROR</status>
         </AutoCompleteResults>
      </ns2:autocompleteResponse>
   </soap:Body>
</soap:Envelope>
```

The case of a maximum number of responses query with an input error. For example, putting a number with a comma in it, or any other character. The value entered in the Designer Administration is the one that will be used in priority cf. Geoconcept Web / Administration / Parameters `geocoder.maxCandidates`

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:autocompleteResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <AutoCompleteResults>
            <status>OK</status>
            <result>
               <city>PARIS</city>
               <classification>1</classification>
               <country>FR</country>
               <fulltext>AVENUE DU GÉNÉRAL BALFOURIER, 75016 PARIS</fulltext>
               <kind/>
               <street>AVENUE DU GÉNÉRAL BALFOURIER</street>
               <x>0.0</x>
               <y>0.0</y>
               <postCode>75016</postCode>
            </result>
         </AutoCompleteResults>
      </ns2:autocompleteResponse>
   </soap:Body>
</soap:Envelope>
```

V1

Parameters / properties

Input

parameter	description	optional	default
text	Location address to enter	yes	
type	Allows selection of the source of the autocomplete files indicating the alias for the latter: if a single autocomplete source exists, the value should remain empty if several sources exist for the autocomplete function, you will need to use the aliases of the latter, defined	yes	

parameter	description	optional	default
	beforehand cf. Geoconcept Web / Administration / Parameters **`geocoder.autocomplete.alias.<alias_name>.datasource`** the value of this key must indicate the name of the directory containing the series of files needed for autocompletion.		
terr	Filter on the territory where one wants to search for the localising elements in question. example, if you want the query to apply to a town, it will be necessary to enter a postcode : 92220 if you want the search to be applied to several towns or departments, you need to separate the territories with a “;” : 75;78;92 if the value remains empty, the search will be applied to all of the territory covered by the autocomplete files and the results will be sorted in increasing streets alphabetical order.	yes	
maximumResponses	maximum number of results for addresses in the response. if the value is left empty, the parameter defined in the Administration will apply	yes	cf. Geoconcept Web / Administration / Parameters **`geocoder.maxCandidates`**

As output

property	type	min/max	description
city	string	0/1	town found
classification	integer	1/1	code indicating in which category the town returned is found by the webservice. example, the categorisation of HERE tables ranges from 1 (the biggest towns) to 8 (village or hamlet)
country	string	0/1	country on two letters (ISO code 3166-1 or ccTLD) for example "fr" when the result is found in France
fulltext	string	0/1	street found with a specific nomenclature including the post code and the name of the town
street	string	1/1	street found
zipcode	string	0/1	post code found

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/autoCompleteService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:autocomplete>
         <!--Optional:-->
         <text>Avenue Général</text>
         <!--Optional:-->
```

```
          <type></type>
          <!--Optional:-->
          <terr>75</terr>
          <!--Optional:-->
          <maximumResponses>3</maximumResponses>
          <!--Optional:-->
          <repeat>?</repeat>
       </sch:autocomplete>
    </soapenv:Body>
 </soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:autocompleteResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <AutoCompleteResults>
            <status>OK</status>
            <result>
               <city>PARIS</city>
               <classification>1</classification>
               <country>FR</country>
               <fulltext>AVENUE DU GÉNÉRAL BALFOURIER, 75016 PARIS</fulltext>
               <kind/>
               <street>AVENUE DU GÉNÉRAL BALFOURIER</street>
               <x>0.0</x>
               <y>0.0</y>
               <zipcode>75016</zipcode>
            </result>
            <result>
               <city>PARIS</city>
               <classification>1</classification>
               <country>FR</country>
               <fulltext>AVENUE DU GÉNÉRAL CLAVERY, 75016 PARIS</fulltext>
               <kind/>
               <street>AVENUE DU GÉNÉRAL CLAVERY</street>
               <x>0.0</x>
               <y>0.0</y>
               <zipcode>75016</zipcode>
            </result>
            <result>
               <city>PARIS</city>
               <classification>1</classification>
               <country>FR</country>
               <fulltext>AVENUE DU GÉNÉRAL DÉTRIE, 75007 PARIS</fulltext>
               <kind/>
               <street>AVENUE DU GÉNÉRAL DÉTRIE</street>
               <x>0.0</x>
               <y>0.0</y>
               <zipcode>75007</zipcode>
            </result>
         </AutoCompleteResults>
      </ns2:autocompleteResponse>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/autocomplete.json?text=rue%20general&terr=75&maximumResponses=3
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/autocomplete.json?text=rue
%20general&terr=75&maximumResponses=3&callback=myFonction
```

XML query

```
http://<server>/<webapp>/api/lbs/autocomplete.xml?text=rue%20general&terr=75&maximumResponses=3
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message": null,
   "status": "OK",
   "results":    [
            {
         "fulltext": "RUE DU GÉNÉRAL ANSELIN, 75016 PARIS",
         "country": "FR",
         "classification": 1,
         "street": "RUE DU GÉNÉRAL ANSELIN",
         "city": "PARIS",
         "zipcode": "75016",
         "x": 0,
         "y": 0,
         "kind": ""
      },
            {
         "fulltext": "RUE DU GÉNÉRAL ANSELIN, 75116 PARIS",
         "country": "FR",
         "classification": 1,
         "street": "RUE DU GÉNÉRAL ANSELIN",
         "city": "PARIS",
         "zipcode": "75116",
         "x": 0,
         "y": 0,
         "kind": ""
      },
            {
         "fulltext": "RUE DU GÉNÉRAL APPERT, 75116 PARIS",
         "country": "FR",
         "classification": 1,
         "street": "RUE DU GÉNÉRAL APPERT",
         "city": "PARIS",
         "zipcode": "75116",
         "x": 0,
         "y": 0,
         "kind": ""
      }
   ]
}
```

JSON-P format

```
myFonction({
        "message":null,"status":"OK",
"results":[
                {
                        "fulltext":"RUE DU GÉNÉRAL ANSELIN, 75016
 PARIS","country":"FR","classification":1,"street":"RUE DU GÉNÉRAL
 ANSELIN","city":"PARIS","zipcode":"75016","x":0.0,"y":0.0,"kind":""
                },
                {
                        "fulltext":"RUE DU GÉNÉRAL ANSELIN, 75116
 PARIS","country":"FR","classification":1,"street":"RUE DU GÉNÉRAL
 ANSELIN","city":"PARIS","zipcode":"75116","x":0.0,"y":0.0,"kind":""
                },
                {
                        "fulltext":"RUE DU GÉNÉRAL APPERT, 75116
 PARIS","country":"FR","classification":1,"street":"RUE DU GÉNÉRAL
 APPERT","city":"PARIS","zipcode":"75116","x":0.0,"y":0.0,"kind":""}
]});
```

XML format

```
<autoCompleteResults>
   <status>OK</status>
   <result>
      <city>PARIS</city>
      <classification>1</classification>
      <country>FR</country>
      <fulltext>RUE DU GÉNÉRAL ANSELIN, 75016 PARIS</fulltext>
      <kind/>
      <street>RUE DU GÉNÉRAL ANSELIN</street>
      <x>0.0</x>
      <y>0.0</y>
      <zipcode>75016</zipcode>
   </result>
   <result>
      <city>PARIS</city>
      <classification>1</classification>
      <country>FR</country>
      <fulltext>RUE DU GÉNÉRAL ANSELIN, 75116 PARIS</fulltext>
      <kind/>
      <street>RUE DU GÉNÉRAL ANSELIN</street>
      <x>0.0</x>
      <y>0.0</y>
      <zipcode>75116</zipcode>
   </result>
   <result>
      <city>PARIS</city>
      <classification>1</classification>
      <country>FR</country>
      <fulltext>RUE DU GÉNÉRAL APPERT, 75116 PARIS</fulltext>
      <kind/>
      <street>RUE DU GÉNÉRAL APPERT</street>
      <x>0.0</x>
      <y>0.0</y>
      <zipcode>75116</zipcode>
   </result>
</autoCompleteResults>
```

Possible responses

The case of addresses that are found (autocompleteResponse/AutoCompleteResults/status est OK)

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:autocompleteResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <AutoCompleteResults>
            <status>OK</status>
            <result>
               <city>PARIS</city>
               <classification>1</classification>
               <country>FR</country>
               <fulltext>AVENUE DU GÉNÉRAL BALFOURIER, 75016 PARIS</fulltext>
               <kind/>
               <street>AVENUE DU GÉNÉRAL BALFOURIER</street>
               <x>0.0</x>
               <y>0.0</y>
               <zipcode>75016</zipcode>
            </result>
            <result>
               <city>PARIS</city>
               <classification>1</classification>
               <country>FR</country>
               <fulltext>AVENUE DU GÉNÉRAL CLAVERY, 75016 PARIS</fulltext>
               <kind/>
               <street>AVENUE DU GÉNÉRAL CLAVERY</street>
               <x>0.0</x>
               <y>0.0</y>
               <zipcode>75016</zipcode>
            </result>
            <result>
               <city>PARIS</city>
               <classification>1</classification>
               <country>FR</country>
               <fulltext>AVENUE DU GÉNÉRAL DÉTRIE, 75007 PARIS</fulltext>
               <kind/>
               <street>AVENUE DU GÉNÉRAL DÉTRIE</street>
               <x>0.0</x>
               <y>0.0</y>
               <zipcode>75007</zipcode>
            </result>
         </AutoCompleteResults>
      </ns2:autocompleteResponse>
   </soap:Body>
</soap:Envelope>
```

The case of an address that does not exist (autocompleteResponse/AutoCompleteResults/status is ERROR)

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:autocompleteResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <AutoCompleteResults>
            <status>ERROR</status>
         </AutoCompleteResults>
      </ns2:autocompleteResponse>
   </soap:Body>
</soap:Envelope>
```

The case of a query having an XML error or not respecting the WSDL ⇒ error with faultstring that contains the description

```
<soap:Fault>
```

```
        <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
        <faultstring>Message part {http://geoconcept.com/gc/schemas}geocodeABCD was not recognized.  (Does
 it exist in service WSDL?)</faultstring>
</soap:Fault>
```

The case of a query with a non-existant territory (autocompleteResponse/AutoCompleteResults/status est ERROR)

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:autocompleteResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <AutoCompleteResults>
            <status>ERROR</status>
         </AutoCompleteResults>
      </ns2:autocompleteResponse>
   </soap:Body>
</soap:Envelope>
```

The case of a maximum number of responses query with an input error. For example, putting a number with a comma in it, or any other character. The value entered in the Designer Administration is the one that will be used in priority cf. Geoconcept Web / Administration / Parameters `geocoder.maxCandidates`

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:autocompleteResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <AutoCompleteResults>
            <status>OK</status>
            <result>
               <city>PARIS</city>
               <classification>1</classification>
               <country>FR</country>
               <fulltext>AVENUE DU GÉNÉRAL BALFOURIER, 75016 PARIS</fulltext>
               <kind/>
               <street>AVENUE DU GÉNÉRAL BALFOURIER</street>
               <x>0.0</x>
               <y>0.0</y>
               <zipcode>75016</zipcode>
            </result>
         </AutoCompleteResults>
      </ns2:autocompleteResponse>
   </soap:Body>
</soap:Envelope>
```

Geocoding

(fr) Cette page n’est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

The query includes an input address, the service returns one or several possible responses (if there is any ambiguity), including the recognised address, the position, the geocoding score, and the geocoding type.

This geocoding service consults the reference table configured via the Administration interface of Geoconcept Web.

Availability

This web service is available at all times with Geoconcept Web and a reference table.

Version change

Earlier versions of the web service are conserved in Geoconcept Web to ensure compatibility with previous developments. Use of the most recent version is recommended.

Changes in relation to v2

- Deletion of the "projection" parameter, replaced by "srs"
- Deletion of the "geocodeScore" parameter, replaced by "score"
- Deletion of "projection", "srs" and "maxResponses" parameters in the "initialAddress" block
- The "postalCode" parameter has been renamed "postCode"

V2

Parameters / properties

Input

parameter	description	optional	default
countryCode	countries on two or three letters (ISO code 3166-1 or ccTLD) for example, "fr" or"fra", this parameter can be used to distinguish between several repositories for a single territory (IGN, HERE, …)	yes	cf. Geoconcept Web / Administration / Parameters `geocoder.datasource`
postCode	post code	yes *	
addressLine	address including number, repetition index, type of street and street name.	yes *	
city	city	yes *	
region	State, County, …	yes	
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	Without any projection, the result is in the native projection of the geocoding index, which is

parameter	description	optional	default
			usually wgs84
maxResponses	maximum number of address results in the response	yes	cf. Geoconcept Web / Administration / Parameters **`geocoder.maxCandidate`**
streetMinScore	the value of this parameters varies, as for the score, between 0 and 100. The streetMinScore has the effect of filtering possible candidates in the following way: If the candidate score is equal to or higher than the score, the candidate is selected as it is, If the candidate score is lower than the score and no candidate match has already been selected, the geocoding is suggested at town level, If the candidate score is lower than the score, and another candidate has already been selected, then the candidate will not be presented unless it fulfills the condition of the parameter **`geocoder.maxCandidates`** The value of streetMinScore must always be higher than **`filterMinScore`** , if this is not the case, the geocoding forces the value of **`filterMinScore`** for streetMinScore. The value defined for streetMinScore replaces that defined in the **`geocoder.geocoder.streetMinScore`** paremeter. The **`geocoder.maxCandidates`** and **`geocoder.minscore`** parameters are described in Geoconcept Web / Administration / Parameters.	yes	

(*) At least one of the three parameters postCode, addressLine and city must be assigned a value.

Output

parameter	type	min/max	description
geocodedAddress (or geocodedAddresses in JSON / JSON-P)	geocodedAddress (or array in JSON / JSON-P)	0/ unlimited	Geocoded addresses
initialAddress	geocodeInitialAddress	0/1	Initial address

Geocoded addresses (geocodedAddress)

parameter	type	min/max	description
addressLine	string	0/1	street found and, if there is one, the number
city	string	0/1	town found
region	string	0/1	State, County, ..; found, varies as a function of country, can also be empty
countryCode	string	1/1	cf. description of the input parameter
postCode	string	0/1	post code found
secondaryZone	string	0/1	zone that depends on the geocoding index, usually in France this will be an IRIS code
score	double	1/1	geocoding score from 0 to 100, with 100 for a perfect correspondance

parameter	type	min/max	description
geocodeType	int	1/1	type of geocoding: - town = 1 - street = 2 - improved street = 3 - street number = 4 - non-geocoded = 0
x	double	1/1	X coordinates
y	double	1/1	Y coordinates
place (or places in JSON / JSON-P)	string (or array in JSON / JSON-P)	0/ unlimited	list of attributes. Value of attributes for the address found, in relation to *placeTypes* (for example ["751010206","930005Y001XCHE"]). Depends on the repository used.
placeType (or placeTypes en JSON / JSON-P)	string (or array in JSON / JSON-P)	0/ unlimited	list of types of attributes (for example ["IRIS","FANTOIR"]). Depends on the reference table used.
streetNumber	string	0/1	street number
streetWayType	string	0/1	type of street (avenue, street, etc)
streetWayName	string	0/1	street name
streetWay	string	0/1	full name of the street

Initial address (initialAddress)

parameter	type	min/max	description
addressLine	string	0/1	street found and, if there is one, the number
city	string	0/1	town found
region	string	0/1	State, County, ..; found, varies as a function of country, can also be empty
countryCode	string	1/1	cf. description of the input parameter
postCode	string	0/1	post code found

SOAP

WSDL

http://`<server>`/`<webapp>`/api/ws/geocodeService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:geocodeV2>
         <!--Optional:-->
         <countryCode>FR</countryCode>
         <!--Optional:-->
         <postCode>44000</postCode>
         <!--Optional:-->
         <addressLine>45 Rue Noire</addressLine>
         <!--Optional:-->
         <city>nantes</city>
         <!--Optional:-->
         <region></region>
```

```
            <!--Optional:-->
            <srs>epsg:4326</srs>
            <maxResponses>2</maxResponses>
            <streetMinScore></streetMinScore>
         </sch:geocodeV2>
      </soapenv:Body>
   </soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:geocodeV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <GeocodeResult>
            <status>OK</status>
            <geocodedAddress>
               <addressLine>45 RUE NOIRE</addressLine>
               <city>Nantes</city>
               <countryCode>FR</countryCode>
               <postCode>44109</postCode>
               <srs>epsg:4326</srs>
               <secondaryZone/>
               <score>94.95</score>
               <geocodeType>4</geocodeType>
               <x>-1.563901</x>
               <y>47.224987</y>
               <places>
                  <place/>
               </places>
               <placeTypes>
                  <placeType>Attribut</placeType>
               </placeTypes>
               <streetNumber>45</streetNumber>
               <streetWayType>RUE</streetWayType>
               <streetWayName>NOIRE</streetWayName>
               <streetWay>RUE NOIRE</streetWay>
            </geocodedAddress>
            <initialAddress>
               <addressLine>45 Rue Noire</addressLine>
               <city>nantes</city>
               <region/>
               <countryCode>FR</countryCode>
               <postCode>44000</postCode>
            </initialAddress>
         </GeocodeResult>
      </ns2:geocodeV2Response>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/geocode/v2.json?addressLine=45%20Rue
%20Noire&postCode=44000&city=nantes&countryCode=fr&srs=epsg:4326
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/geocode/v2.json?addressLine=45%20Rue
%20Noire&postCode=44000&city=nantes&countryCode=fr&srs=epsg:4326&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/geocode/v2.xml?addressLine=45%20Rue
%20Noire&postCode=44000&city=nantes&countryCode=fr&srs=epsg:4326
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message":null,
   "status":"OK",
   "geocodedAddresses":[
      {
         "addressLine":"45 RUE NOIRE",
         "city":"Nantes",
         "countryCode":"FR",
         "postCode":"44109",
         "srs":"epsg:4326",
         "secondaryZone":"",
         "score":94.95,
         "geocodeType":4,
         "x":-1.563901,
         "y":47.224987,
         "places":[
            ""
         ],
         "placeTypes":[
            "Attribut"
         ],
         "streetNumber":"45",
         "streetWayType":"RUE",
         "streetWayName":"NOIRE",
         "streetWay":"RUE NOIRE"
      }
   ],
   "initialAddress":{
      "addressLine":"45 Rue Noire",
      "city":"nantes",
      "countryCode":"fr",
      "postCode":"44000"
   }
}
```

JSON-P format

```
MyCallBack({
   "message":null,
   "status":"OK",
   "geocodedAddresses":[
      {
         "addressLine":"45 RUE NOIRE",
         "city":"Nantes",
         "countryCode":"FR",
         "postCode":"44109",
         "srs":"epsg:4326",
         "secondaryZone":"",
         "score":94.95,
         "geocodeType":4,
         "x":-1.563901,
         "y":47.224987,
         "places":[
            ""
         ],
         "placeTypes":[
            "Attribut"
         ],
         "streetNumber":"45",
         "streetWayType":"RUE",
         "streetWayName":"NOIRE",
         "streetWay":"RUE NOIRE"
      }
   ],
   "initialAddress":{
      "addressLine":"45 Rue Noire",
      "city":"nantes",
      "countryCode":"fr",
      "postCode":"44000"
   }
});
```

XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<geocodeResultV2>
   <status>OK</status>
   <geocodedAddress>
      <addressLine>45 RUE NOIRE</addressLine>
      <city>Nantes</city>
      <countryCode>FR</countryCode>
      <postCode>44109</postCode>
      <srs>epsg:4326</srs>
      <secondaryZone />
      <score>94.95</score>
      <geocodeType>4</geocodeType>
      <x>-1.563901</x>
      <y>47.224987</y>
      <places>
         <place />
      </places>
      <placeTypes>
         <placeType>Attribut</placeType>
      </placeTypes>
      <streetNumber>45</streetNumber>
      <streetWayType>RUE</streetWayType>
      <streetWayName>NOIRE</streetWayName>
      <streetWay>RUE NOIRE</streetWay>
   </geocodedAddress>
   <initialAddress>
      <addressLine>45 Rue Noire</addressLine>
      <city>nantes</city>
      <countryCode>fr</countryCode>
      <postCode>44000</postCode>
   </initialAddress>
</geocodeResultV2>
```

API JavaScript

Include the Javascript library.

```
var geocoder = new GCUI.Control.GeoCode();
geocoder.geocode({url : 'http://<server>/<webapp>/api/lbs/geocode/v2.json', city : 'Nantes', postCode :
 '44000', addressLine : '45 Rue Noire', countryCode : 'fr', srs : 'wgs84',
callback : function(result) {...} }) ;
```

The `result` variable is in JSON format as described above. The `callback` function passed as parameter is called at the end of the geocoding operation.

Possible responses

Case of an address found (geocodeResponse/GeocodeResult/status is OK)

```
<ns2:geocodeV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
    <?xml version="1.0" encoding="UTF-8"?>
    <geocodeResultV2>
       <status>OK</status>
       <geocodedAddress>
          <addressLine>45 RUE NOIRE</addressLine>
          <city>Nantes</city>
          <countryCode>FR</countryCode>
          <postCode>44109</postCode>
          <srs>epsg:4326</srs>
          <secondaryZone />
          <score>94.95</score>
          <geocodeType>4</geocodeType>
          <x>-1.563901</x>
          <y>47.224987</y>
          <places>
             <place />
          </places>
          <placeTypes>
             <placeType>Attribut</placeType>
          </placeTypes>
          <streetNumber>45</streetNumber>
          <streetWayType>RUE</streetWayType>
          <streetWayName>NOIRE</streetWayName>
          <streetWay>RUE NOIRE</streetWay>
       </geocodedAddress>
       <initialAddress>
          <addressLine>45 Rue Noire</addressLine>
          <city>nantes</city>
          <countryCode>fr</countryCode>
          <postCode>44000</postCode>
       </initialAddress>
    </geocodeResultV2>
</ns2:geocodeV2Response>
```

Case of an address that is not found (geocodeResponse/GeocodeResult/status is OK and no geocodedAddress)

```
<ns2:geocodeV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
        <GeocodeResult>
                <status>OK</status>
                <initialAddress>
```

```
                        <city>#hdkvnjsdvn</city>
                </initialAddress>
        </GeocodeResult>
</ns2:geocodeV2Response>
```

Case of a query containing an XML error or not respecting the WSDL ⇒ error with faultstring that contains the description

```
<soap:Fault>
        <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
        <faultstring>Message part {http://geoconcept.com/gc/schemas}geocodeABCD was not recognized.  (Does
 it exist in service WSDL?)</faultstring>
</soap:Fault>
```

Case of a query with a non-existant reprojection system ⇒ error with faultstring that contains the description

```
<soap:Fault>
        <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
        <faultstring>Geocode failed Failed to process geocoding task Unsupported coordinate system
 'epsg:432666666'</faultstring>
</soap:Fault>
```

V1

Parameters / properties

Input

parameter	description	optional	default
countryCode	countries on two letters (ISO code 3166-1 or ccTLD) for example, "fr", this parameter can be used to distinguish between several repositories for a single territory (IGN, HERE, …)	yes	cf. Geoconcept Web / Administration / Parameters **geocoder.datasource**
postalCode	post code	yes *	
addressLine	address including number, repetition index, type of street and street name.	yes *	
city	city	yes *	
region	State, County, …	yes	
projection	depreciated, replaced by srs	yes	
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	Without any projection, the result is in the native projection of the geocoding index,

parameter	description	optional	default
			which is usually wgs84
maxResponses	maximum number of address results in the response	yes	cf. Geoconcept Web / Administration / Parameters **`geocoder.maxCandidates`**
streetMinScore	the value of this parameters varies, as for the score, between 0 and 100. The streetMinScore has the effect of filtering possible candidates in the following way: If the candidate score is equal to or higher than the score, the candidate is selected as it is, If the candidate score is lower than the score and no candidate match has already been selected, the geocoding is suggested at town level, If the candidate score is lower than the score, and another candidate has already been selected, then the candidate will not be presented unless it fulfills the condition of the parameter **`geocoder.maxCandidates`** The value of streetMinScore must always be higher than **`filterMinScore`** , if this is not the case, the geocoding forces the value of **`filterMinScore`** for streetMinScore. The value defined for streetMinScore replaces that defined in the **`geocoder.geocoder.streetMinScore`** paremeter. The **`geocoder.maxCandidates`** and **`geocoder.minscore`** parameters are described in Geoconcept Web / Administration / Parameters.	yes	

(*) At least one of the three parameters postalCode, addressLine and city must have a value assigned.

Output

parameter	type	min/max	description
geocodedAddress (or geocodedAddresses in JSON / JSON-P)	geocodedAddress (or array in JSON / JSON-P)	0/ unlimited	Geocoded addresses
initialAddress	geocodeInitialAddress	0/1	Initial address

Geocoded addresses (geocodedAddress)

parameter	type	min/max	description
secondaryZone	string	0/1	zone that depends on the geocoding index, usually in France this will be an IRIS code
score	double	1/1	geocoding score from 0 to 100, with 100 for a perfect correspondance
geocodeScore	double	1/1	Deprecated, replaced by score: geocoding score from 0 to 20, with 20 for a perfect correspondance
geocodeType	int	1/1	type of geocoding: - town = 1 - street = 2 - improved street = 3 - street number = 4 - non-geocoded = 0
x	double	1/1	X coordinates
y	double	1/1	Y coordinates

parameter	type	min/max	description
place (or places in JSON / JSON-P)	string (or array in JSON / JSON-P)	0/ unlimited	list of attributes. Value of attributes for the address found, in relation to *placeTypes* (for example ["751010206","930005Y001XCHE"]). Depends on the repository used.
placeType (or placeTypes en JSON / JSON-P)	string (or array in JSON / JSON-P)	0/ unlimited	list of types of attributes (for example ["IRIS","FANTOIR"]). Depends on the reference table used.
streetNumber	string	0/1	street number
streetWayType	string	0/1	type of street (avenue, street, etc)
streetWayName	string	0/1	street name
streetWay	string	0/1	full name of the street

Initial address (initialAddress)

parameter	type	min/max	description
addressLine	string	0/1	street found and, if there is one, the number
city	string	0/1	town found
region	string	0/1	State, County, ..; found, varies as a function of country, can also be empty
countryCode	string	1/1	cf. description of the input parameter
postalCode	string	0/1	post code found
projection	string	1/1	Deprecated, replaced by srs
srs	string	1/1	cf. description of the input parameter
maxResponses	string	0/1	cf. description of the input parameter

SOAP

WSDL

http://`<server>`/`<webapp>`/api/ws/geocodeService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:geocode>
         <!--Optional:-->
         <countryCode></countryCode>
         <!--Optional:-->
         <postalCode>75013</postalCode>
         <!--Optional:-->
         <addressLine>25, rue de toldiac</addressLine>
         <!--Optional:-->
         <city>paris</city>
         <!--Optional:-->
         <region></region>
         <!--Optional:-->
         <projection></projection>
         <!--Optional:-->
         <srs></srs>
         <maxResponses>2</maxResponses>
         <streetMinScore></streetMinScore>
      </sch:geocode>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:geocodeResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <GeocodeResult>
            <status>OK</status>
            <geocodedAddress>
               <addressLine>25 RUE DE TOLBIAC</addressLine>
               <city>PARIS</city>
               <countryCode>FR</countryCode>
               <postalCode>75013</postalCode>
               <projection>epsg:27572</projection>
               <srs>epsg:27572</srs>
               <secondaryZone>751135014</secondaryZone>
               <geocodeScore>19.8101</geocodeScore>
               <score>99.05</score>
               <geocodeType>4</geocodeType>
               <x>602722.1475334</x>
               <y>2425598.4510822</y>
               <places>
                  <place>751135014</place>
               </places>
               <placeTypes>
                  <placeType>IRIS</placeType>
               </placeTypes>
               <streetNumber>25</streetNumber>
               <streetWayType>RUE</streetWayType>
               <streetWayName>DE TOLBIAC</streetWayName>
               <streetWay>RUE DE TOLBIAC</streetWay>
            </geocodedAddress>
            <geocodedAddress>
               <addressLine>PONT DE TOLBIAC</addressLine>
               <city>PARIS</city>
               <countryCode>FR</countryCode>
               <postalCode>75013</postalCode>
               <projection>epsg:27572</projection>
               <srs>epsg:27572</srs>
               <secondaryZone>751135099</secondaryZone>
               <geocodeScore>19.0505</geocodeScore>
               <score>95.25</score>
               <geocodeType>2</geocodeType>
               <x>603221.0049634</x>
               <y>2426021.5068995</y>
               <places>
                  <place>751135099</place>
               </places>
               <placeTypes>
                  <placeType>IRIS</placeType>
               </placeTypes>
               <streetNumber/>
               <streetWayType>PONT</streetWayType>
               <streetWayName>DE TOLBIAC</streetWayName>
               <streetWay>PONT DE TOLBIAC</streetWay>
            </geocodedAddress>
```

```
               <initialAddress>
                  <addressLine>25, rue de toldiac</addressLine>
                  <city>paris</city>
                  <region/>
                  <countryCode/>
                  <postalCode>75013</postalCode>
                  <projection/>
                  <srs/>
               </initialAddress>
            </GeocodeResult>
         </ns2:geocodeResponse>
      </soap:Body>
   </soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/geocode.json?countryCode=fr&addressLine=25%20rue%20de
%20toldiac&postalCode=75013&city=Paris&srs=epsg:4326&maxResponses=2
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/geocode.json?countryCode=fr&addressLine=25%20rue%20de
%20toldiac&postalCode=75013&city=Paris&srs=epsg:4326&maxResponses=2&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/geocode.xml?countryCode=fr&addressLine=25%20rue%20de
%20toldiac&postalCode=75013&city=Paris&maxResponses=2
```

Response

The response is always in UTF-8 format.

JSON format

```
{
        "message":null,"status":"OK",
        "geocodedAddresses":    [
                {
                        "addressLine":"25 RUE DE
 TOLBIAC","city":"PARIS","countryCode":"FR","postalCode":"75013",

 "projection":"epsg:4326","srs":"epsg:4326","maxResponses":null,"secondaryZone":"751135014",
                        "geocodeScore":19.8101,"score":99.05,"geocodeType":4,"x":2.373562,"y":48.828757,
                        "places":["751135014"],"placeTypes":["IRIS"],
                        "streetNumber":"25","streetWayType":"RUE","streetWayName":"DE
 TOLBIAC","streetWay":"RUE DE TOLBIAC"
                },
                {
                        "addressLine":"PONT DE
 TOLBIAC","city":"PARIS","countryCode":"FR","postalCode":"75013",

 "projection":"epsg:4326","srs":"epsg:4326","maxResponses":null,"secondaryZone":"751135099",
                        "geocodeScore":19.0505,"score":95.25,"geocodeType":2,"x":2.380356,"y":48.832557,
                        "places":["751135099"],"placeTypes":["IRIS"],
                        "streetNumber":"","streetWayType":"PONT","streetWayName":"DE
 TOLBIAC","streetWay":"PONT DE TOLBIAC"
                }
        ],
        "initialAddress":               {
                "addressLine":"25 rue de toldiac","city":"Paris","countryCode":"fr","postalCode":"75013",
                "projection":"epsg:4326","srs":"epsg:4326"
        }
}
```

JSON-P format

```
myCallback({
        "message":null,"status":"OK",
        "geocodedAddresses":    [
                {
                        "addressLine":"25 RUE DE
 TOLBIAC","city":"PARIS","countryCode":"FR","postalCode":"75013",

 "projection":"epsg:4326","srs":"epsg:4326","maxResponses":null,"secondaryZone":"751135014",
                        "geocodeScore":19.8101,"score":99.05,"geocodeType":4,"x":2.373562,"y":48.828757,
                        "places":["751135014"],"placeTypes":["IRIS"],
                        "streetNumber":"25","streetWayType":"RUE","streetWayName":"DE
 TOLBIAC","streetWay":"RUE DE TOLBIAC"
                },
                {
                        "addressLine":"PONT DE
 TOLBIAC","city":"PARIS","countryCode":"FR","postalCode":"75013",

 "projection":"epsg:4326","srs":"epsg:4326","maxResponses":null,"secondaryZone":"751135099",
                        "geocodeScore":19.0505,"score":95.25,"geocodeType":2,"x":2.380356,"y":48.832557,
                        "places":["751135099"],"placeTypes":["IRIS"],
                        "streetNumber":"","streetWayType":"PONT","streetWayName":"DE
 TOLBIAC","streetWay":"PONT DE TOLBIAC"
                }
        ]
        "initialAddress":       {
                "addressLine":"25 rue de toldiac","city":"Paris","countryCode":"fr","postalCode":"75013",
                "projection":"epsg:4326","srs":"epsg:4326"
        }
});
```

XML format

```
<geocodeResult>
        <status>OK</status>
        <geocodedAddress>
                <addressLine>25 RUE DE TOLBIAC</addressLine>
                <city>PARIS</city>
                <countryCode>FR</countryCode>
                <postalCode>75013</postalCode>
                <projection>epsg:27572</projection>
                <srs>epsg:27572</srs>
                <secondaryZone>751135014</secondaryZone>
                <geocodeScore>19.8101</geocodeScore>
                <score>99.05</score>
                <geocodeType>4</geocodeType>
                <x>602722.1475334</x>
                <y>2425598.4510822</y>
                <places>
```

```
                        <place>751135014</place>
                </places>
                <placeTypes>
                        <placeType>IRIS</placeType>
                </placeTypes>
                <streetNumber>25</streetNumber>
                <streetWayType>RUE</streetWayType>
                <streetWayName>DE TOLBIAC</streetWayName>
                <streetWay>RUE DE TOLBIAC</streetWay>
        </geocodedAddress>
        <geocodedAddress>
                <addressLine>PONT DE TOLBIAC</addressLine>
                <city>PARIS</city>
                <countryCode>FR</countryCode>
                <postalCode>75013</postalCode>
                <projection>epsg:27572</projection>
                <srs>epsg:27572</srs>
                <secondaryZone>751135099</secondaryZone>
                <geocodeScore>19.0505</geocodeScore>
                <score>95.25</score>
                <geocodeType>2</geocodeType>
                <x>603221.0049634</x>
                <y>2426021.5068995</y>
                <places>
                        <place>751135099</place>
                </places>
                <placeTypes>
                        <placeType>IRIS</placeType>
                </placeTypes>
                <streetNumber/>
                <streetWayType>PONT</streetWayType>
                <streetWayName>DE TOLBIAC</streetWayName>
                <streetWay>PONT DE TOLBIAC</streetWay>
        </geocodedAddress>
        <initialAddress>
                <addressLine>25 rue de toldiac</addressLine>
                <city>Paris</city>
                <countryCode>fr</countryCode>
                <postalCode>75013</postalCode>
        </initialAddress>
</geocodeResult>
```

API JavaScript

Include the Javascript library.

```
var geocoder = new GCUI.Control.GeoCode();
geocoder.geocode({url : 'http://<server>/<webapp>/api/lbs/geocode.json', city : 'Paris', postalCode :
 '75013', addressLine : '25 rue de Toldiac', countryCode : 'fr', srs : 'wgs84',
callback : function(result) {...} }) ;
```

The `result` variable is in JSON format as described above. The `callback` function passed as parameter is called at the end of the geocoding operation.

Possible responses

Case of an address found (geocodeResponse/GeocodeResult/status is OK)

```
<ns2:geocodeResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
        <GeocodeResult>
```

```
                <status>OK</status>
                <geocodedAddress>
                        <addressLine/>
                        <city>BAGNEUX</city>
                        <countryCode>FR</countryCode>
                        <postalCode>92220</postalCode>
                        <projection>epsg:4326</projection>
                        <srs>epsg:4326</srs>
                        <secondaryZone>920070110</secondaryZone>
                        <geocodeScore>20.0</geocodeScore>
                        <score>100.0</score>
                        <geocodeType>1</geocodeType>
                        <x>2.30811</x>
                        <y>48.79692</y>
                        <place>920070110</place>
                        <placeType>IRIS</placeType>
                        <streetNumber/>
                        <streetWayType/>
                        <streetWayName/>
                        <streetWay/>
                </geocodedAddress>
                <initialAddress>
                        <city>Bagneux</city>
                        <postalCode>92</postalCode>
                        <projection>epsg:4326</projection>
                </initialAddress>
        </GeocodeResult>
</ns2:geocodeResponse>
```

Case of an address that is not found (geocodeResponse/GeocodeResult/status is OK and no geocodedAddress)

```
<ns2:geocodeResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
        <GeocodeResult>
                <status>OK</status>
                <initialAddress>
                        <city>#hdkvnjsdvn</city>
                </initialAddress>
        </GeocodeResult>
</ns2:geocodeResponse>
```

Case of a query containing an XML error or not respecting the WSDL ⇒ error with faultstring that contains the description

```
<soap:Fault>
        <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
        <faultstring>Message part {http://geoconcept.com/gc/schemas}geocodeABCD was not recognized.  (Does
 it exist in service WSDL?)</faultstring>
</soap:Fault>
```

Case of a query with a non-existant reprojection system ⇒ error with faultstring that contains the description

```
<soap:Fault>
        <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
        <faultstring>Geocode failed Failed to process geocoding task Unsupported coordinate system
 'epsg:432666666'</faultstring>
</soap:Fault>
```

FAQ

1. How is the score obtained in the *score* to be interpreted?

   A strict correspondance (equivalent to a score of 100) is equivalent to a *no tolerance* solution in effect, for a file that must not contain any errors at all. Choosing this score, you only accept suggestions of perfect matches between the reference table and the address sought. The higher the tolerance (in other words, the lower this score is set) the more the user accepts letters that are close matches between the addresses in the file to geocode and the reference table. The more likely it will be, also, that the software will assign false coordinates to certain addresses in the file due to a faulty interpretation. It is recommended that you only keep any candidates scoring 90 or higher. This score is a good compromise to the extent that one or two approximations between the addresses to geocode and the reference table are tolerated. It is strongly recommended that candidates with a score of less that 75 are NOT retained: this corresponds to at least 4 spelling or syntax mistakes in the address. Different criteria will have a bearing on the calculation of this score:

   - the number of equivalent letters in the two strings;
   - the number of words;
   - the length of each word making up the string.

   In the case of correspondances between addresses, only the names of streets have a bearing on the search for a sppelling match. The type of street has an additional incidence on the score, but only in the case where the type of street sought and the one suggested are identical.

2. How many characters are returned in the tag *postCode* ?

   Depends on the characteristics of postal codes for each country. Moreover, the returned string will be truncated to the first N characters if the address is not precise enough.

3. What rules exist with regard to X-Y coordinates (format, coordinates range)?

   These are names in decimal format, with . as a separator of the decimal part. In the case where the geographic coordinates are in degrees, the return must be included within the range -180 and 180 (X=longitude) and -90 and 90 (Y=latitude), with 6 figures after the comma. In the case of projected coordinates, the result is in metres, with two figures after the comma.

4. Can this service be used to find and deliver a close-matching address?

   If the town does not exist, it will return an empty list.

5. How are *Cedex* entities handled?

   The grammar, described in the UGC.PDF documentation, applies a filter so as not to take Cedex and the codes that follow it into account during the search for addresses.

6. Can you have all the components of an address as a single input item, in just one field?

   Yes, the *addressline* component can be used to perform a fulltext geocoding via a unique field containing the concatenation of all fields: street number, type of street, street name, post code and name of the town. The addresses input via the *addressLine* component are analysed in two phases:

- The mechanism for searching on word pattern will find, in the first instance, the town or towns present in the string. A search on the pattern then very quickly finds a label corresponding to a dictionary, in this case a dictionary of towns. This label can be badly spelt, with predictable or repeated errors in the user's entering text at the keyboard when using Internet: duplicated latters, letters omitted, replacement of one letter with another given a similar sound to the resulting word (C instead of K, for example) part of a town name where the town name consists of several sub-entities. Following these lines of enquiry, a series of candidate towns is found in the shortest of timescales.
- Next, the pairs of towns found + the initial string without the "town" part that has just been recognised are given to the geocoder to perform the usual geocoding operation.

Note: when the name of a town is also present in the name of a street, for example, "12 Rue d'Aix Antibes" there are potentially two candidates: Rue d'Aix à Antibes, and Rue d'Antibes à Aix. The best candidate is determined by its relative position in the string.

7. Can you use several repositories at the same time?

   Yes, there are two methods: The first consists of deploying a configuration via the *'geocoding world'* Cf. Administration / Tools / World Geocoding. The second, rather more technical, is deployed over a series of steps:

- Stop the Tomcat service,
- Put the referencial geocoding (8 files with .ugc.xxi extensions) in the appropriate directory. For example: Here, two tables are placed in the following directory: C:\[Home]\Geoconcept\UGC \JEE\ugc\reftables, in the framework of a default configuration.
- Edit the *'Service.xml'* file associated to the JEE UGC module. The latter can be found here: C: \[Home]\Geoconcept\UGC\JEE\ugc\conf.
- In the text editor of your choice, copy/paste the text block framed by the <default-datasource> tags as many times as you have reference tables (in our example, there are 2 different sources, to copy/paste twice as a result),
- For each of the blocks copied/pasted, rename the <default-datasource> tag as <datasource>,
- The <file /> tag should indicate the name of one of the two reference tables. The <name /> tag allows you to give an alias to this source, for example here: "HERE".

Edition of the first block associated to the first reference table

```
<datasource>
  <file>france_dom_M20.ugc.mdi</file>
  <name>HERE</name>
  <title />
  <abstract />
  <online-resource />
  <zone-meaning />
  <uniqueId-meaning />
  <secondaryZone-meaning />
  <roadId-meaning />
  <country />
  <version />
  <min-processors>0</min-processors>
  <max-processors>1</max-processors>
```

```
      <cache-enabled>true</cache-enabled>
      <cache-max-size>65536</cache-max-size>
      <city-scoring-method>levenshtein</city-scoring-method>
      <street-scoring-method>levenshtein</street-scoring-method>
      <min-streets>0</min-streets>
      <default-hierachy-search-depth>-1</default-hierachy-search-depth>
      <city-name-prefix-index-size>0</city-name-prefix-index-size>
      <strategy>
        <on-no-exact-city-match score-all-candidates="true" />
        <on-no-exact-street-match score-all-candidates="true" />
        <city-search-scoring max-error-threshold="85" />
        <street-search-scoring max-error-threshold="85" />
        <search hirerachy-depth="-1" />
        <!-- <force-search-two-keys /> -->
        <!-- <filter-zone filters="&gt;##" /> -->
      </strategy>
      <country />
      <coordinateSystem />
      <bounding-box />
      <max-candidates>10</max-candidates>
      <find-type>street number</find-type>
      <tolerate-find-type>7</tolerate-find-type>
      <max-meter-error>50</max-meter-error>
      <discrepancy>0.0</discrepancy>
      <discrepancy-along-street>0.0</discrepancy-along-street>
      <favor-city-match-element>favor city name</favor-city-match-element>
      <zone-match-digits>auto</zone-match-digits>
      <score-threshold>0.0</score-threshold>
      <score-threshold-to-tolerate-street-geocoding-type>0.0</score-threshold-to-tolerate-street-geocoding-
  type>
      <score-threshold-to-accept-city>0.88</score-threshold-to-accept-city>
    </datasource>
```

- Leave the other parameters by default,
- Repeat the steps for copying the <datasource> block, and setting up the <file/> and <name/> tags to configure the second data source.

Create the second block associated to the other reference table that will be exploited

```
    <datasource>
      <file>BD_ADRESSE_france_M20.ugc.mdi</file>
      <name>IGN</name>
      <title />
      <abstract />
      <online-resource />
      <zone-meaning />
      <uniqueId-meaning />
      <secondaryZone-meaning />
      <roadId-meaning />
      <country />
      <version />
      <min-processors>0</min-processors>
      <max-processors>1</max-processors>
      <cache-enabled>true</cache-enabled>
      <cache-max-size>65536</cache-max-size>
      <city-scoring-method>levenshtein</city-scoring-method>
      <street-scoring-method>levenshtein</street-scoring-method>
      <min-streets>0</min-streets>
      <default-hierachy-search-depth>-1</default-hierachy-search-depth>
      <city-name-prefix-index-size>0</city-name-prefix-index-size>
      <strategy>
```

```
        <on-no-exact-city-match score-all-candidates="true" />
        <on-no-exact-street-match score-all-candidates="true" />
        <city-search-scoring max-error-threshold="85" />
        <street-search-scoring max-error-threshold="85" />
        <search hirerachy-depth="-1" />
        <!-- <force-search-two-keys /> -->
        <!-- <filter-zone filters="&gt;##" /> -->
      </strategy>
      <country />
      <coordinateSystem />
      <bounding-box />
      <max-candidates>10</max-candidates>
      <find-type>street number</find-type>
      <tolerate-find-type>7</tolerate-find-type>
      <max-meter-error>50</max-meter-error>
      <discrepancy>0.0</discrepancy>
      <discrepancy-along-street>0.0</discrepancy-along-street>
      <favor-city-match-element>favor city name</favor-city-match-element>
      <zone-match-digits>auto</zone-match-digits>
      <score-threshold>0.0</score-threshold>
      <score-threshold-to-tolerate-street-geocoding-type>0.0</score-threshold-to-tolerate-street-geocoding-
  type>
      <score-threshold-to-accept-city>0.88</score-threshold-to-accept-city>
    </datasource>
```

- Back-up the "XML Service" file and rerun the Tomcat service,
- Connect to the Geoconcept Web portal and go into the Administration and then Parameters ,
- Add a new parameter key via the Add button, indicating the following information: Name = geocoder.countryDatasource and value = %country%, then validate by clicking on OK .
- Now, when you call the Geocoding WebService, the user will have to call one of the two resources in their query, indicating in the <countrycode> parameter, the alias that has been defined beforehand, via the <name> tag. If no alias has been passed, the source defined in the geocoder.datasource parameter will be used by default.

Batch geocoding

(fr) Cette page n'est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

The query includes an address or several addresses as input, the service returns for each item, one or several possible responses (if there is any ambiguity), including the address recognised, the position, the geocoding score and the type of geocoding.

This geocoding service interrogates the configured reference table thanks to the Geoconcept Web Administration interface.

Availability

This web service is available at all times with gGoconcept Web and a reference table.

Version change

Earlier versions of the web service are conserved in Geoconcept Web to ensure compatibility with previous developments. We recommend using the most recent version.

Changes in relation to v2

- Deletion of the "projection" parameter, replaced by "srs".
- Deletion of the "geocodeScore" parameter, replaced by "score"
- Deletion of the "projection", "srs" and "maxResponses" parameters in the "initialAddress" block.
- The "postalCode" parameter has been renamed "postCode".

V2

Parameters / properties

Input

parameter	description	optional	default
address (geocodeInitialAddress)	The addresses to geocode	no	
streetMinScore	the value of this parameter will vary, like the score, between 0 and 100 (Cf. full description in the geocoding web service)	yes	
srs	projection (EPSG code such as epsg:4326 or wgs 84)	yes	Without projection, the result is in native projection of the geocoding index, usually wgs84.
maxResponses	maximum number of address results in the response	yes	

Addresses (geocodeInitialAddress)

parameter	description	optional	default
addressLine	address including the number, repetition index, type of street and street name.	yes *	
city	town	yes *	
region	State, County, ..;	yes	
countryCode	countries on 2 or 3 letters (3166-1 ISO code) for example "fr" or "fra",	yes	
postCode	post code	yes *	

(*) At least one of the three parameters postCode, addressLine and city must be filled.

Output

parameter	type	min/max	description
geocodedAddress (or geocodedAddresses in JSON / JSON-P)	geocodedAddress (or array in JSON / JSON-P)	0/ unlimited	Geocoded addresses
initialAddress	geocodeInitialAddress	0/1	Initial address

Geocoded addresses (geocodedAddress)

parameter	type	min/max	description
addressLine	string	0/1	street found and, where appropriate, the number
city	string	0/1	town found
region	string	0/1	State, County, found, varies as a function of country, and could also be empty
countryCode	string	1/1	cf. description of the input parameter
postCode	string	0/1	post code found
secondaryZone	string	0/1	zone that depends on the geocoding index, usually in France this will be the IRIS code
score	double	1/1	geocoding score, ranging from 0 to 100, with 100 for a perfect match.
geocodeType	int	1/1	type of geocoding: - town = 1 - street = 2 - improved street = 3 - street number = 4 - non-geocoded = 0
x	double	1/1	X coordinates
y	double	1/1	Y coordinates
place (or places in JSON / JSON-P)	string (or array in JSON / JSON-P)	0/ unlimited	list of attributes. Value of attributes for the address found, in relation with *placeTypes* (for example["751010206","930005Y001XCHE"]). Varies as a function of the repository used.
placeType (or placeTypes in JSON / JSON-P)	string (or array in JSON / JSON-P)	0/ unlimited	list of attribute types (for example ["IRIS","FANTOIR"]). Varies as a functiion of the repository used.
streetNumber	string	0/1	street number
streetWayType	string	0/1	type of street (avenue, street, etc)
streetWayName	string	0/1	name of the street
streetWay	string	0/1	full name of the street

Initial address (initialAddress)

parameter	type	min/max	description
addressLine	string	0/1	street found and, where appropriate, the number
city	string	0/1	town found
region	string	0/1	State, County, found, varies as a function of country, and could also be empty
countryCode	string	1/1	cf. description of the input parameter

parameter	type	min/max	description
postCode	string	0/1	post code found

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/geocodeService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:batchGeocodeV2>
         <!--Optional:-->
         <addresses>
            <!--Zero or more repetitions:-->
            <address>
               <!--Optional:-->
               <addressLine>23, rue de la gare</addressLine>
               <!--Optional:-->
               <city>Saint-Herblain</city>
               <!--Optional:-->
               <region></region>
               <!--Optional:-->
               <countryCode>FR</countryCode>
               <!--Optional:-->
               <postCode>44800</postCode>
            </address>
            <address>
               <!--Optional:-->
               <addressLine>32 Route de Pornic</addressLine>
               <!--Optional:-->
               <city>Bouguenais</city>
               <!--Optional:-->
               <region></region>
               <!--Optional:-->
               <countryCode>FR</countryCode>
               <!--Optional:-->
               <postCode>44340</postCode>
            </address>
            <address>
               <!--Optional:-->
               <addressLine>5, Avenue Victor Hugo</addressLine>
               <!--Optional:-->
               <city>Sainte-Luce-sur-Loire</city>
               <!--Optional:-->
               <region></region>
               <!--Optional:-->
               <countryCode>FR</countryCode>
               <!--Optional:-->
               <postCode>44980</postCode>
            </address>
            <streetMinScore></streetMinScore>
            <!--Optional:-->
            <srs></srs>
            <maxResponses>2</maxResponses>
         </addresses>
      </sch:batchGeocodeV2>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:batchGeocodeV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <BatchGeocodeResult>
            <status>OK</status>
            <result>
               <status>OK</status>
               <geocodedAddress>
                  <addressLine>23 RUE DE LA GARE</addressLine>
                  <city>SAINT-HERBLAIN</city>
                  <countryCode>FR</countryCode>
                  <postCode>44800</postCode>
                  <srs>epsg:4326</srs>
                  <secondaryZone>441620701</secondaryZone>
                  <score>100.0</score>
                  <geocodeType>4</geocodeType>
                  <x>-1.656851</x>
                  <y>47.21028</y>
                  <places>
                     <place>441620701</place>
                     <place>441620701_010</place>
                     <place>4401621080</place>
                     <place>44162</place>
                  </places>
                  <placeTypes>
                     <placeType>IRIS</placeType>
                     <placeType>ILOT</placeType>
                     <placeType>FANTOIR</placeType>
                     <placeType>INSEE</placeType>
                  </placeTypes>
                  <streetNumber>23</streetNumber>
                  <streetWayType>RUE</streetWayType>
                  <streetWayName>DE LA GARE</streetWayName>
                  <streetWay>RUE DE LA GARE</streetWay>
               </geocodedAddress>
               <initialAddress>
                  <addressLine>23, rue de la gare</addressLine>
                  <city>Saint-Herblain</city>
                  <region/>
                  <countryCode>FR</countryCode>
                  <postCode>44800</postCode>
               </initialAddress>
            </result>
            <result>
               <status>OK</status>
               <geocodedAddress>
                  <addressLine>ROUTE DE PORNIC</addressLine>
                  <city>BOUGUENAIS</city>
                  <countryCode>FR</countryCode>
                  <postCode>44340</postCode>
                  <srs>epsg:4326</srs>
                  <secondaryZone>440200106</secondaryZone>
                  <score>100.0</score>
                  <geocodeType>2</geocodeType>
                  <x>-1.581521</x>
                  <y>47.189556</y>
                  <places>
                     <place>440200106</place>
                     <place>440200106_038</place>
                     <place>4400202800</place>
                     <place>44020</place>
                  </places>
                  <placeTypes>
                     <placeType>IRIS</placeType>
                     <placeType>ILOT</placeType>
                     <placeType>FANTOIR</placeType>
                     <placeType>INSEE</placeType>
                  </placeTypes>
                  <streetNumber/>
                  <streetWayType>ROUTE</streetWayType>
                  <streetWayName>DE PORNIC</streetWayName>
                  <streetWay>ROUTE DE PORNIC</streetWay>
               </geocodedAddress>
               <initialAddress>
                  <addressLine>32 Route de Pornic</addressLine>
                  <city>Bouguenais</city>
                  <region/>
                  <countryCode>FR</countryCode>
                  <postCode>44340</postCode>
               </initialAddress>
            </result>
            <result>
               <status>OK</status>
               <geocodedAddress>
                  <addressLine/>
                  <city>SAINTE-LUCE-SUR-LOIRE</city>
                  <countryCode>FR</countryCode>
                  <postCode>44980</postCode>
                  <srs>epsg:4326</srs>
                  <secondaryZone>44172</secondaryZone>
                  <score>100.0</score>
                  <geocodeType>1</geocodeType>
                  <x>-1.48669</x>
                  <y>47.24961</y>
                  <places>
                     <place>44172</place>
                  </places>
                  <placeTypes>
                     <placeType>IRIS</placeType>
                  </placeTypes>
                  <streetNumber/>
                  <streetWayType/>
                  <streetWayName/>
                  <streetWay/>
               </geocodedAddress>
               <initialAddress>
                  <addressLine>5, Avenue Victor Hugo</addressLine>
                  <city>Sainte-Luce-sur-Loire</city>
                  <region/>
                  <countryCode>FR</countryCode>
                  <postCode>44980</postCode>
               </initialAddress>
            </result>
         </BatchGeocodeResult>
      </ns2:batchGeocodeV2Response>
   </soap:Body>
</soap:Envelope>
```

REST (POST)

Query

Query

```
http://<server>/<webapp>/api/lbs/geocode/batch/v2.xml
```

Data (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<batchGeocodeRequestV2>
  <address>
    <addressLine>23, rue de la gare</addressLine>
    <city>Saint-Herblain</city>
    <region></region>
    <countryCode>FR</countryCode>
    <postCode>44800</postCode>
  </address>
  <address>
    <addressLine>32 Route de Pornic</addressLine>
    <city>Bouguenais</city>
    <region></region>
    <countryCode>FR</countryCode>
    <postCode>44340</postCode>
  </address>
    <address>
    <addressLine>5, Avenue Victor Hugo</addressLine>
    <city>Sainte-Luce-sur-Loire</city>
    <region></region>
    <countryCode>FR</countryCode>
    <postCode>44980</postCode>
  </address>
  <streetMinScore></streetMinScore>
  <srs>epsg:4326</srs>
  <maxResponses>2</maxResponses>
</batchGeocodeRequestV2>
```

Response

The response is always in UTF-8 format.

XML format

```
<batchGeocodeResultV2>
   <status>OK</status>
   <result>
      <status>OK</status>
      <geocodedAddress>
         <addressLine>23 RUE DE LA GARE</addressLine>
         <city>SAINT-HERBLAIN</city>
         <countryCode>FR</countryCode>
         <postCode>44800</postCode>
         <srs>epsg:4326</srs>
         <secondaryZone>441620701</secondaryZone>
         <score>100.0</score>
         <geocodeType>4</geocodeType>
         <x>-1.656851</x>
         <y>47.21028</y>
         <places>
```

```
            <place>441620701</place>
            <place>441620701_010</place>
            <place>4401621080</place>
            <place>44162</place>
         </places>
         <placeTypes>
            <placeType>IRIS</placeType>
            <placeType>ILOT</placeType>
            <placeType>FANTOIR</placeType>
            <placeType>INSEE</placeType>
         </placeTypes>
         <streetNumber>23</streetNumber>
         <streetWayType>RUE</streetWayType>
         <streetWayName>DE LA GARE</streetWayName>
         <streetWay>RUE DE LA GARE</streetWay>
      </geocodedAddress>
      <initialAddress>
         <addressLine>23, rue de la gare</addressLine>
         <city>Saint-Herblain</city>
         <region/>
         <countryCode>FR</countryCode>
         <postCode>44800</postCode>
      </initialAddress>
   </result>
   <result>
      <status>OK</status>
      <geocodedAddress>
         <addressLine>ROUTE DE PORNIC</addressLine>
         <city>BOUGUENAIS</city>
         <countryCode>FR</countryCode>
         <postCode>44340</postCode>
         <srs>epsg:4326</srs>
         <secondaryZone>440200106</secondaryZone>
         <score>100.0</score>
         <geocodeType>2</geocodeType>
         <x>-1.581521</x>
         <y>47.189556</y>
         <places>
            <place>440200106</place>
            <place>440200106_038</place>
            <place>4400202800</place>
            <place>44020</place>
         </places>
         <placeTypes>
            <placeType>IRIS</placeType>
            <placeType>ILOT</placeType>
            <placeType>FANTOIR</placeType>
            <placeType>INSEE</placeType>
         </placeTypes>
         <streetNumber/>
         <streetWayType>ROUTE</streetWayType>
         <streetWayName>DE PORNIC</streetWayName>
         <streetWay>ROUTE DE PORNIC</streetWay>
      </geocodedAddress>
      <initialAddress>
         <addressLine>32 Route de Pornic</addressLine>
         <city>Bouguenais</city>
         <region/>
         <countryCode>FR</countryCode>
         <postCode>44340</postCode>
      </initialAddress>
   </result>
   <result>
```

```
      <status>OK</status>
      <geocodedAddress>
         <addressLine/>
         <city>SAINTE-LUCE-SUR-LOIRE</city>
         <countryCode>FR</countryCode>
         <postCode>44980</postCode>
         <srs>epsg:4326</srs>
         <secondaryZone>44172</secondaryZone>
         <score>100.0</score>
         <geocodeType>1</geocodeType>
         <x>-1.48669</x>
         <y>47.24961</y>
         <places>
            <place>44172</place>
         </places>
         <placeTypes>
            <placeType>IRIS</placeType>
         </placeTypes>
         <streetNumber/>
         <streetWayType/>
         <streetWayName/>
         <streetWay/>
      </geocodedAddress>
      <initialAddress>
         <addressLine>5, Avenue Victor Hugo</addressLine>
         <city>Sainte-Luce-sur-Loire</city>
         <region/>
         <countryCode>FR</countryCode>
         <postCode>44980</postCode>
      </initialAddress>
   </result>
</batchGeocodeResultV2>
```

Query

JSON query

```
http://<server>/<webapp>/api/lbs/geocode/batch/v2.json
```

JSON

```
{
  "addresses" :
  [
  {
    "addressLine" : "23, rue de la gare",
    "city" : "Saint-Herblain",
    "region" : "",
    "countryCode" : "FR",
    "postCode" : "44800"
  },
  {
    "addressLine" : "32 Route de Pornic",
    "city" : "Bouguenais",
    "region" : "",
    "countryCode" : "FR",
    "postCode" : "44340"
  },
  {
    "addressLine" : "5, Avenue Victor Hugo",
    "city" : "Sainte-Luce-sur-Loire",
    "region" : "",
    "countryCode" : "FR",
    "postCode" : "44980"
  }
 ],
  "streetMinScore" : 80 ,
  "srs" : "epsg:4326",
  "maxResponses" : 2
}
```

Response

The response is always in UTF-8 format.

JSON format

```
  {
     "message": null,
     "status": "OK",
     "results":    [
              {
           "message": null,
           "status": "OK",
           "geocodedAddresses": [         {
              "addressLine": "23 RUE DE LA GARE",
              "city": "SAINT-HERBLAIN",
              "countryCode": "FR",
              "postCode": "44800",
              "srs": "epsg:4326",
              "secondaryZone": "",
              "score": 100,
              "geocodeType": 4,
              "x": -1.65801,
              "y": 47.20903,
              "places": [""],
              "placeTypes": ["IRIS"],
              "streetNumber": "23",
              "streetWayType": "RUE",
              "streetWayName": "DE LA GARE",
              "streetWay": "RUE DE LA GARE"
           }],
           "initialAddress":          {
              "addressLine": "23, rue de la gare",
              "city": "Saint-Herblain",
              "region": "",
              "countryCode": "FR",
              "postCode": "44800"
           }
        },
              {
           "message": null,
           "status": "OK",
           "geocodedAddresses": [         {
              "addressLine": "32 ROUTE DE PORNIC",
              "city": "BOUGUENAIS",
              "countryCode": "FR",
              "postCode": "44340",
              "srs": "epsg:4326",
              "secondaryZone": "",
              "score": 100,
              "geocodeType": 3,
              "x": -1.58862,
              "y": 47.18768,
              "places": [""],
              "placeTypes": ["IRIS"],
              "streetNumber": "32",
              "streetWayType": "ROUTE",
              "streetWayName": "DE PORNIC",
              "streetWay": "ROUTE DE PORNIC"
           }],
           "initialAddress":          {
              "addressLine": "32 Route de Pornic",
              "city": "Bouguenais",
              "region": "",
              "countryCode": "FR",
              "postCode": "44340"
           }
        },
              {
           "message": null,
           "status": "OK",
           "geocodedAddresses": [         {
              "addressLine": "",
              "city": "SAINTE-LUCE-SUR-LOIRE",
              "countryCode": "FR",
              "postCode": "44980",
              "srs": "epsg:4326",
              "secondaryZone": "",
              "score": 100,
              "geocodeType": 1,
              "x": -1.48669,
              "y": 47.24961,
              "places": [""],
              "placeTypes": ["IRIS"],
              "streetNumber": "",
              "streetWayType": "",
              "streetWayName": "",
              "streetWay": ""
           }],
           "initialAddress":          {
              "addressLine": "5, Avenue Victor Hugo",
              "city": "Sainte-Luce-sur-Loire",
              "region": "",
              "countryCode": "FR",
              "postCode": "44980"
           }
        }
     ]
  }
```

Possible responses

Case of an address found (batchGeocodeResponse/batchGeocodeResult/status is OK)

```
<ns2:batchGeocodeV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
 <BatchGeocodeResult>
    <status>OK</status>
    <result>
       <status>OK</status>
       <geocodedAddress>
          <addressLine>23 RUE DE LA GARE</addressLine>
          <city>SAINT-HERBLAIN</city>
          <countryCode>FR</countryCode>
          <postCode>44800</postCode>
          <srs>epsg:4326</srs>
          <secondaryZone>441620701</secondaryZone>
          <score>100.0</score>
          <geocodeType>4</geocodeType>
          <x>-1.656851</x>
          <y>47.21028</y>
          <places>
             <place>441620701</place>
             <place>441620701_010</place>
             <place>4401621080</place>
             <place>44162</place>
          </places>
          <placeTypes>
             <placeType>IRIS</placeType>
             <placeType>ILOT</placeType>
             <placeType>FANTOIR</placeType>
             <placeType>INSEE</placeType>
          </placeTypes>
          <streetNumber>23</streetNumber>
          <streetWayType>RUE</streetWayType>
          <streetWayName>DE LA GARE</streetWayName>
          <streetWay>RUE DE LA GARE</streetWay>
       </geocodedAddress>
       <initialAddress>
          <addressLine>23, rue de la gare</addressLine>
          <city>Saint-Herblain</city>
          <region/>
          <countryCode>FR</countryCode>
          <postCode>44800</postCode>
       </initialAddress>
    </result>
    <result>
        [...]
    </result>
 </BatchGeocodeResult>
</ns2:batchGeocodeV2Response>
```

Case of an address that is not found (batchGeocodeResponse/BatchGeocodeResult/status is OK and no geocodedAddress)

```
<BatchGeocodeResult>
<status>OK</status>
<result>
   <status>OK</status>
   <initialAddress>
      <addressLine/>
      <city>#hdkvnjsdvn</city>
      <region/>
      <countryCode/>
      <postCode/>
   </initialAddress>
</result>
</BatchGeocodeResult>
```

Case of a query with an XML error or that does not respect the WSDL ⇒ error with faultstring that contains the description

```
<soap:Fault>
 <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
 <faultstring>Message part {http://geoconcept.com/gc/schemas}batchGeocodeV2fff was not recognized.  (Does it
 exist in service WSDL?)</faultstring>
```

```
</soap:Fault>
```

Case of a query with a non-existent reprojection system ⇒ error with faultstring that contains the description

```
<soap:Fault>
        <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
        <faultstring>Geocode failed Failed to process geocoding task Unsupported coordinate system
 'epsg:432666666'</faultstring>
</soap:Fault>
```

V1

Parameters / properties

Input

parameter	description	optional	default
address (geocodeInitialAddress)	The addresses to geocode	no	
streetMinScore	the value of this parameter will vary, like the score, between 0 and 100 (Cf. full description in the geocoding web service)	yes	
srs	projection (EPSG code such as epsg:4326 or wgs 84)	yes	Without projection, the result is in native projection of the geocoding index, usually wgs84.
maxResponses	maximum number of address results in the response	yes	

Addresses (geocodeInitialAddress)

parameter	description	optional	default
addressLine	address including the number, repetition index, type of street and street name.	yes *	
city	town	yes *	
region	State, County, ..;	yes	
countryCode	country in two letters (ISO code 3166-1 or ccTLD) for eample "fr",	yes	
postalCode	post code	yes *	
projection	deprecated	yes	
srs	deprecated	yes	
maxResponses	deprecated	yes	

(*) At least one of the three parameters postalCode, addressLine and city must be assigned values.

Output

parameter	type	min/max	description
geocodedAddress (or geocodedAddresses in JSON / JSON-P)	geocodedAddress (or array in JSON / JSON-P)	0/ unlimited	Geocoded addresses
initialAddress	geocodeInitialAddress	0/1	Initial address

Geocoded addresses (geocodedAddress)

parameter	type	min/max	description
secondaryZone	string	0/1	zone that depends on the geocoding index, usually in France this will be the IRIS code
score	double	1/1	geocoding score, ranging from 0 to 100, with 100 for a perfect match.
geocodeType	int	1/1	type of geocoding: - town = 1 - street = 2 - improved street = 3 - street number = 4 - non-geocoded = 0
x	double	1/1	X coordinates
y	double	1/1	Y coordinates
place (or places in JSON / JSON-P)	string (or array in JSON / JSON-P)	0/ unlimited	list of attributes. The value of attributes for the address found, in relation to *placeTypes* (for example, ["751010206","930005Y001XCHE"]). Depends on the repository used.
placeType (or placeTypes in JSON / JSON-P)	string (or array in JSON / JSON-P)	0/ unlimited	list of types of attributes (for example ["IRIS","FANTOIR"]). Depends on the reference table used.
streetNumber	string	0/1	street number
streetWayType	string	0/1	type of street (avenue, street, etc)
streetWayName	string	0/1	name of the street
streetWay	string	0/1	full name of the street

Initial address (initialAddress)

parameter	type	min/max	description
addressLine	string	0/1	street found and, where appropriate, the number
city	string	0/1	town found
region	string	0/1	State, County, found, varies as a function of country, and could also be empty
countryCode	string	1/1	cf. description of the input parameter
postalCode	string	0/1	post code found

SOAP

WSDL

http://`<server>`/`<webapp>`/api/ws/geocodeService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:batchGeocode>
         <!--Optional:-->
         <addresses>
            <!--Zero or more repetitions:-->
            <address>
               <!--Optional:-->
               <addressLine>23, rue de la gare</addressLine>
               <!--Optional:-->
               <city>Saint-Herblain</city>
               <!--Optional:-->
               <region></region>
               <!--Optional:-->
               <countryCode>FR</countryCode>
               <!--Optional:-->
               <postalCode>44800</postalCode>
               <!--Optional:-->
               <projection></projection>
               <!--Optional:-->
               <srs>epsg:4326</srs>
               <!--Optional:-->
               <maxResponses></maxResponses>
            </address>
            <address>
               <!--Optional:-->
               <addressLine>32 Route de Pornic</addressLine>
               <!--Optional:-->
               <city>Bouguenais</city>
               <!--Optional:-->
               <region></region>
               <!--Optional:-->
               <countryCode>FR</countryCode>
               <!--Optional:-->
               <postalCode>44340</postalCode>
               <!--Optional:-->
               <projection></projection>
               <!--Optional:-->
               <srs>epsg:4326</srs>
               <!--Optional:-->
               <maxResponses></maxResponses>
            </address>
            <address>
               <!--Optional:-->
               <addressLine>5, Avenue Victor Hugo</addressLine>
               <!--Optional:-->
               <city>Sainte-Luce-sur-Loire</city>
               <!--Optional:-->
               <region></region>
               <!--Optional:-->
               <countryCode>FR</countryCode>
               <!--Optional:-->
               <postalCode>44980</postalCode>
               <!--Optional:-->
               <projection></projection>
               <!--Optional:-->
               <srs>epsg:4326</srs>
               <!--Optional:-->
               <maxResponses></maxResponses>
            </address>
            <streetMinScore></streetMinScore>
            <!--Optional:-->
            <srs>epsg:4326</srs>
            <maxResponses>2</maxResponses>
         </addresses>
      </sch:batchGeocode>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:batchGeocodeResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <BatchGeocodeResult>
            <status>OK</status>
            <result>
               <status>OK</status>
               <geocodedAddress>
                  <addressLine>23 RUE DE LA GARE</addressLine>
                  <city>SAINT-HERBLAIN</city>
                  <countryCode>FR</countryCode>
                  <postalCode>44800</postalCode>
                  <projection>epsg:4326</projection>
                  <srs>epsg:4326</srs>
                  <secondaryZone>441620701</secondaryZone>
                  <geocodeScore>20.0</geocodeScore>
                  <score>100.0</score>
                  <geocodeType>4</geocodeType>
                  <x>-1.6568714</x>
                  <y>47.2102641</y>
                  <places>
                     <place>441620701</place>
                  </places>
                  <placeTypes>
                     <placeType>IRIS</placeType>
                  </placeTypes>
                  <streetNumber>23</streetNumber>
                  <streetWayType>RUE</streetWayType>
                  <streetWayName>DE LA GARE</streetWayName>
                  <streetWay>RUE DE LA GARE</streetWay>
               </geocodedAddress>
               <initialAddress>
                  <addressLine>23, rue de la gare</addressLine>
                  <city>Saint-Herblain</city>
                  <region/>
                  <countryCode>FR</countryCode>
                  <postalCode>44800</postalCode>
                  <projection/>
                  <srs>epsg:4326</srs>
                  <maxResponses/>
               </initialAddress>
            </result>
            <result>
               <status>OK</status>
               <geocodedAddress>
                  <addressLine>ROUTE DE PORNIC</addressLine>
                  <city>BOUGUENAIS</city>
                  <countryCode>FR</countryCode>
                  <postalCode>44340</postalCode>
                  <projection>epsg:4326</projection>
                  <srs>epsg:4326</srs>
                  <secondaryZone>440200106</secondaryZone>
                  <geocodeScore>20.0</geocodeScore>
                  <score>100.0</score>
                  <geocodeType>2</geocodeType>
                  <x>-1.5799805</x>
                  <y>47.1901969</y>
                  <places>
                     <place>440200106</place>
                  </places>
                  <placeTypes>
                     <placeType>IRIS</placeType>
                  </placeTypes>
                  <streetNumber/>
                  <streetWayType>ROUTE</streetWayType>
                  <streetWayName>DE PORNIC</streetWayName>
                  <streetWay>ROUTE DE PORNIC</streetWay>
               </geocodedAddress>
               <initialAddress>
                  <addressLine>32 Route de Pornic</addressLine>
                  <city>Bouguenais</city>
                  <region/>
                  <countryCode>FR</countryCode>
                  <postalCode>44340</postalCode>
                  <projection/>
                  <srs>epsg:4326</srs>
                  <maxResponses/>
               </initialAddress>
            </result>
            <result>
               <status>OK</status>
               <geocodedAddress>
                  <addressLine/>
                  <city>SAINTE-LUCE-SUR-LOIRE</city>
                  <countryCode>FR</countryCode>
                  <postalCode>44980</postalCode>
                  <projection>epsg:4326</projection>
                  <srs>epsg:4326</srs>
                  <secondaryZone>441720105</secondaryZone>
                  <geocodeScore>20.0</geocodeScore>
                  <score>100.0</score>
                  <geocodeType>1</geocodeType>
                  <x>-1.4787216</x>
                  <y>47.2565577</y>
                  <places>
                     <place>441720105</place>
                  </places>
                  <placeTypes>
                     <placeType>IRIS</placeType>
                  </placeTypes>
                  <streetNumber/>
                  <streetWayType/>
                  <streetWayName/>
                  <streetWay/>
               </geocodedAddress>
               <initialAddress>
                  <addressLine>5, Avenue Victor Hugo</addressLine>
                  <city>Sainte-Luce-sur-Loire</city>
                  <region/>
                  <countryCode>FR</countryCode>
                  <postalCode>44980</postalCode>
                  <projection/>
                  <srs>epsg:4326</srs>
                  <maxResponses/>
```

```
                  </initialAddress>
               </result>
            </BatchGeocodeResult>
         </ns2:batchGeocodeResponse>
      </soap:Body>
   </soap:Envelope>
```

REST (POST)

Query

Query

```
http://<server>/<webapp>/api/lbs/geocode/batch/v1.xml
```

Data (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<batchGeocodeRequest>
  <address>
    <addressLine>23, rue de la gare</addressLine>
    <city>Saint-Herblain</city>
    <region></region>
    <countryCode>FR</countryCode>
    <postalCode>44800</postalCode>
    <projection></projection>
  </address>
  <address>
    <addressLine>32 Route de Pornic</addressLine>
    <city>Bouguenais</city>
    <region></region>
    <countryCode>FR</countryCode>
    <postalCode>44340</postalCode>
    <projection></projection>
  </address>
    <address>
    <addressLine>5, Avenue Victor Hugo</addressLine>
    <city>Sainte-Luce-sur-Loire</city>
    <region></region>
    <countryCode>FR</countryCode>
    <postalCode>44980</postalCode>
    <projection></projection>
  </address>
  <streetMinScore></streetMinScore>
  <srs>epsg:4326</srs>
  <maxResponses>2</maxResponses>
</batchGeocodeRequest>
```

Response

The response is always in UTF-8 format.

XML format

```
<batchGeocodeResult>
   <status>OK</status>
   <result>
      <status>OK</status>
      <geocodedAddress>
         <addressLine>23 RUE DE LA GARE</addressLine>
```

```
            <city>SAINT-HERBLAIN</city>
            <countryCode>FR</countryCode>
            <postalCode>44800</postalCode>
            <projection>epsg:4326</projection>
            <srs>epsg:4326</srs>
            <secondaryZone>441620701</secondaryZone>
            <geocodeScore>20.0</geocodeScore>
            <score>100.0</score>
            <geocodeType>4</geocodeType>
            <x>-1.6568714</x>
            <y>47.2102641</y>
            <places>
               <place>441620701</place>
            </places>
            <placeTypes>
               <placeType>IRIS</placeType>
            </placeTypes>
            <streetNumber>23</streetNumber>
            <streetWayType>RUE</streetWayType>
            <streetWayName>DE LA GARE</streetWayName>
            <streetWay>RUE DE LA GARE</streetWay>
         </geocodedAddress>
         <initialAddress>
            <addressLine>23, rue de la gare</addressLine>
            <city>Saint-Herblain</city>
            <region/>
            <countryCode>FR</countryCode>
            <postalCode>44800</postalCode>
            <projection/>
         </initialAddress>
      </result>
      <result>
         <status>OK</status>
         <geocodedAddress>
            <addressLine>ROUTE DE PORNIC</addressLine>
            <city>BOUGUENAIS</city>
            <countryCode>FR</countryCode>
            <postalCode>44340</postalCode>
            <projection>epsg:4326</projection>
            <srs>epsg:4326</srs>
            <secondaryZone>440200106</secondaryZone>
            <geocodeScore>20.0</geocodeScore>
            <score>100.0</score>
            <geocodeType>2</geocodeType>
            <x>-1.5799805</x>
            <y>47.1901969</y>
            <places>
               <place>440200106</place>
            </places>
            <placeTypes>
               <placeType>IRIS</placeType>
            </placeTypes>
            <streetNumber/>
            <streetWayType>ROUTE</streetWayType>
            <streetWayName>DE PORNIC</streetWayName>
            <streetWay>ROUTE DE PORNIC</streetWay>
         </geocodedAddress>
         <initialAddress>
            <addressLine>32 Route de Pornic</addressLine>
            <city>Bouguenais</city>
            <region/>
            <countryCode>FR</countryCode>
            <postalCode>44340</postalCode>
            <projection/>
         </initialAddress>
      </result>
      <result>
         <status>OK</status>
         <geocodedAddress>
            <addressLine/>
            <city>SAINTE-LUCE-SUR-LOIRE</city>
            <countryCode>FR</countryCode>
            <postalCode>44980</postalCode>
            <projection>epsg:4326</projection>
            <srs>epsg:4326</srs>
            <secondaryZone>441720105</secondaryZone>
            <geocodeScore>20.0</geocodeScore>
            <score>100.0</score>
            <geocodeType>1</geocodeType>
            <x>-1.4787216</x>
            <y>47.2565577</y>
            <places>
               <place>441720105</place>
            </places>
            <placeTypes>
               <placeType>IRIS</placeType>
            </placeTypes>
            <streetNumber/>
            <streetWayType/>
            <streetWayName/>
            <streetWay/>
         </geocodedAddress>
         <initialAddress>
            <addressLine>5, Avenue Victor Hugo</addressLine>
            <city>Sainte-Luce-sur-Loire</city>
            <region/>
            <countryCode>FR</countryCode>
            <postalCode>44980</postalCode>
            <projection/>
         </initialAddress>
      </result>
   </batchGeocodeResult>
```

Possible responses

Case of an address found (batchGeocodeResponse/batchGeocodeResult/status is OK)

```
   <ns2:batchGeocodeResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
            <BatchGeocodeResult>
               <status>OK</status>
               <result>
                  <status>OK</status>
                  <geocodedAddress>
                     <addressLine>23 RUE DE LA GARE</addressLine>
                     <city>SAINT-HERBLAIN</city>
                     <countryCode>FR</countryCode>
                     <postalCode>44800</postalCode>
                     <projection>epsg:4326</projection>
                     <srs>epsg:4326</srs>
                     <secondaryZone>441620701</secondaryZone>
                     <geocodeScore>20.0</geocodeScore>
                     <score>100.0</score>
                     <geocodeType>4</geocodeType>
                     <x>-1.6568714</x>
                     <y>47.2102641</y>
```

```
                    <places>
                       <place>441620701</place>
                    </places>
                    <placeTypes>
                       <placeType>IRIS</placeType>
                    </placeTypes>
                    <streetNumber>23</streetNumber>
                    <streetWayType>RUE</streetWayType>
                    <streetWayName>DE LA GARE</streetWayName>
                    <streetWay>RUE DE LA GARE</streetWay>
                 </geocodedAddress>
                 <initialAddress>
                    <addressLine>23, rue de la gare</addressLine>
                    <city>Saint-Herblain</city>
                    <region/>
                    <countryCode>FR</countryCode>
                    <postalCode>44800</postalCode>
                    <projection/>
                    <srs>epsg:4326</srs>
                    <maxResponses/>
                 </initialAddress>
              </result>
              <result>
              ...
              </result>
           </BatchGeocodeResult>
</ns2:batchGeocodeResponse>
```

Case of an address that is not found (batchGeocodeResponse/BatchGeocodeResult/status is OK and no geocodedAddress)

```
<ns2:batchGeocodeResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
        <BatchGeocodeResult>
                <status>OK</status>
                <initialAddress>
                        <city>#hdkvnjsdvn</city>
                </initialAddress>
        </BatchGeocodeResult>
</ns2:batchGeocodeResponse>
```

Case of a query with an XML error or that does not respect the WSDL ⇒ error with faultstring that contains the description

```
<soap:Fault>
        <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
        <faultstring>Message part {http://geoconcept.com/gc/schemas}geocodeABCD was not recognized.  (Does
 it exist in service WSDL?)</faultstring>
</soap:Fault>
```

Case of a query with a non-existent reprojection system ⇒ error with faultstring that contains the description

```
<soap:Fault>
        <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
        <faultstring>Geocode failed Failed to process geocoding task Unsupported coordinate system
 'epsg:432666666'</faultstring>
</soap:Fault>
```

FAQ

See Geocoding Web service.

1. How can one geocode via the cURL command?

   You will need to call an address file, here in json format, with the following command line:

```
curl -X POST "https://<server>/<webapp>/api/lbs/geocode/batch/v2.json" -H "Content-Type: application/json"
 --data-binary @"adresses.json"
```

```
Fichier : adresses.json
```

```
{
   "addresses":[
      {
         "addressLine":"200 Quai Charles de Gaulle",
         "city":"Lyon",
         "region":"",
         "countryCode":"FR",
         "postCode":"69006"
      },
      {
         "addressLine":"Bruno Kreisky Platz 1",
         "city":"Wien",
         "region":"",
         "countryCode":"AT",
         "postCode":"1220"
      },
      {
         "addressLine":"Route des Morillons 15",
         "city":"Genève",
         "region":"",
         "countryCode":"CH",
         "postCode":"1202"
      }
   ],
   "streetMinScore":80,
   "srs":"epsg:4326",
   "maxResponses":2
}
```

Address abbreviation

(fr) Cette page n’est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

The query takes as input an address (broken down into an address line, post code, town) and the size of the required standardised abreviation (32 or 38).

The service returns as output an abbreviation of the address line part following the rules expressed in the corresponding AFNOR standards (32 character standard and 38 character standard).

Concerning the restitution function of the 38-character address line, only the AFNOR NF Z10-011 standard dating from January 2013 (in 38 characters) will be applicable.

Concerning the restitution function of the 32-character address line;

- In the first place the AFNOR NF Z10-011 standard from January 2013 (in 38 characters) will be aplicable:
  - with regard to the rules for reduction, but up to the target of 32 characters,
  - with regard to the abbreviation of words, including those that have not been defined in Appendix B of the standard NF Z10-011 dating from August 1989 (in 32 characters).
- The state of play currently is that, following the application of AFNOR standard NF Z10-011 in January 2013, the NF Z10-011 standard dating from August 1989 (in 32 characters) will be applicable in relation to abbreviations of words, but only if the maximum length requested is 32 and if the resulting length remains higher than 32 characters. In this case, it will be limited to application of abbreviations from Appendix B of the NF Z10-011 standard dating from August 1989 (in 32 characters) for which the AFNOR standard NF Z10-011 of January 2013 in its various appendices does not suggest any abbreviation.

This serfvice interrogates files of abbreviations, copying the appendices of the AFNOR standards cited.

Availability

This web service is available at all times with Geoconcept Web.

V1

Parameters / properties

Input

parameter	description	optional	default
maxLength	maxLength defines, in the form of an integer, the reference size to attain for the abbreviation of the address line. Values accepted are exclusivement 32 or 38.	yes	38
addressLine	The address line receiving, EITHER the street in the form number, repetition index, type of street and street name, OR the place name (in France this is the LIEU-DIT). The accuracy of the address abbreviation as output from the WS can only be ensured on condition that an address line parameter resulting from a validated address using an Address repository is supplied.	no	
city	Town corresponding to the address line entered. No treatment will be applied to this parameter.	yes	
postCode	Post code corresponding to the address line supplied. No treatment will be applied to this parameter.	yes	

Output

parameter	type	min/max	description
id	long	1/1	Automated increment of the response identifier.

parameter	type	min/max	description
status	string	0/1	Indicates the success or failure of the abbreviation process «OK» or «ERROR».
statusDetails	string	0/1	Empty, in the case of a succesful abbreviation process. With the corresponding error message, in the event of failure, "Invalid maxLength parameter. It needs to be either 32 or 38". "The address contraction can’t reach the expected maxLength". "The supplied addressline was above 100 characters".
maxLength	int	1/1	Takes the value supplied in input «32» or «38».
address	array (normalizedAddress)	0/1	Standardised address.
normedAddressLineLength	int	1/1	Length of the address line obtained, following the operations of abbreviation applied by the service.

Standardised address (normalizedAddress)

parameter	type	min/max	description
normedAddressLine	string	0/1	Final result of the contraction of the address line applied by the service.
city	string	0/1	Returns the town supplied as input, without modification, (otherwise empty).
postCode	string	0/1	Returns the post code supplied as input, without modification, (otherwise empty).

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/addressNormalizerService?wsdl

Query

The example query takes into account the following address line: ``123 Bis, Boulevard du MARÉCHAL Jean de-Lattre-de-Tassigny Inférieur".

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
  <soapenv:Header/>
  <soapenv:Body>
     <sch:normalize>
        <maxLength>38</maxLength>
        <addressLine>123 Bis, Boulevard du MARÉCHAL Jean de-Lattre-de-Tassigny Inférieur</addressLine>
        <!--Optional:-->
        <city></city>
        <!--Optional:-->
        <postCode></postCode>
     </sch:normalize>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

```
    <soap:Body>
       <ns2:normalizeResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
          <NormalizerResponse>
             <id>2</id>
             <status>OK</status>
             <statusDetails/>
             <maxLength>38</maxLength>
             <address>
                <normedAddressLine>123 B BD MAL J LATTRE TASSIGNY INF</normedAddressLine>
                <city/>
                <postCode/>
             </address>
             <normedAddressLineLength>34</normedAddressLineLength>
          </NormalizerResponse>
       </ns2:normalizeResponse>
    </soap:Body>
  </soap:Envelope>
```

REST

Query

The example query takes into account the following address line: ``123 Bis, Boulevard du MARÉCHAL Jean de-Lattre-de-Tassigny Inférieur".

JSON query

```
http://<server>/<webapp>/api/lbs/normalizer.json?addressLine=123%20Bis,%20Boulevard%20du%20MAR%C3%89CHAL
%20Jean%20de-Lattre-de-Tassigny%20Inf%C3%A9rieur
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/normalizer.json?addressLine=123%20Bis,%20Boulevard%20du%20MAR%C3%89CHAL
%20Jean%20de-Lattre-de-Tassigny%20Inf%C3%A9rieur&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/normalizer.xml?addressLine=123%20Bis,%20Boulevard%20du%20MAR%C3%89CHAL
%20Jean%20de-Lattre-de-Tassigny%20Inf%C3%A9rieur
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "id": 3,
   "status": "OK",
   "statusDetails": "",
   "maxLength": 38,
   "address":    {
      "normedAddressLine": "123 B BD MAL J LATTRE TASSIGNY INF",
      "city": "",
      "postCode": ""
   },
   "normedAddressLineLength": 34
}
```

JSON-P format

```
myCallback({
           "id": 3,
           "status": "OK",
           "statusDetails": "",
           "maxLength": 38,
           "address":    {
                  "normedAddressLine": "123 B BD MAL J LATTRE TASSIGNY INF",
                  "city": "",
                  "postCode": ""
           },
           "normedAddressLineLength": 34
});
```

XML format

```
<normalizerResponse>
   <id>4</id>
   <status>OK</status>
   <statusDetails/>
   <maxLength>38</maxLength>
   <address>
      <normedAddressLine>123 B BD MAL J LATTRE TASSIGNY INF</normedAddressLine>
      <city/>
      <postCode/>
   </address>
   <normedAddressLineLength>34</normedAddressLineLength>
</normalizerResponse>
```

Possible responses

case of a standardised address (NormalizerResponse/NormalizerResponse/status is OK)

```
<ns2:normalizeResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
 <NormalizerResponse>
        <id>2</id>
        <status>OK</status>
        <statusDetails/>
        <maxLength>38</maxLength>
        <address>
           <normedAddressLine>123 B BD MAL J LATTRE TASSIGNY INF</normedAddressLine>
           <city>Paris</city>
           <postCode/>
        </address>
        <normedAddressLineLength>34</normedAddressLineLength>
 </NormalizerResponse>
</ns2:normalizeResponse>
```

Case of setting the reduction size to a value other than «32» or «38» (NormalizerResponse/NormalizerResponse/status is ERROR)

```
<ns2:normalizeResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
 <NormalizerResponse>
        <id>5</id>
        <status>ERROR</status>
        <statusDetails>Invalid maxLength parameter. It needs to be either 32 or 38</statusDetails>
        <maxLength>0</maxLength>
        <address>
           <normedAddressLine/>
           <city/>
           <postCode/>
        </address>
        <normedAddressLineLength>0</normedAddressLineLength>
 </NormalizerResponse>
</ns2:normalizeResponse>
```

Case of an address of more than 100 characters (NormalizerResponse/NormalizerResponse/status is ERROR)

```
<ns2:normalizeResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
 <NormalizerResponse>
        <id>6</id>
        <status>ERROR</status>
        <statusDetails>The supplied addressline was above 100 characters</statusDetails>
        <maxLength>0</maxLength>
        <address>
           <normedAddressLine/>
           <city/>
           <postCode/>
        </address>
        <normedAddressLineLength>0</normedAddressLineLength>
 </NormalizerResponse>
</ns2:normalizeResponse>
```

Case of failure in contraction of the address line in terms of the expected size (NormalizerResponse/NormalizerResponse/status is ERROR)

```
<ns2:normalizeResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
 <NormalizerResponse>
        <id>7</id>
        <status>ERROR</status>
        <statusDetails>The address contraction can't reach the expected maxLength</statusDetails>
        <maxLength>0</maxLength>
        <address>
           <normedAddressLine>123 B BD A F ZQ S E MAL J L TASSIGNY INF</normedAddressLine>
           <city>Paris</city>
           <postCode/>
        </address>
        <normedAddressLineLength>40</normedAddressLineLength>
 </NormalizerResponse>
</ns2:normalizeResponse>
```

FAQ

1. Is it possible to modify, add or delete the strings used for abbreviations?

   Yes, these information items can be edited. In the “<TOMCAT_HOME>”\webapps\geoconcept-web\WEB-INF\lib\ folder, open the geoweb-ws-(version).jar file, follow the com\geoconcept \lbs\webservices filepath, and inside you will find the Substitutions.xml file. This file contains the vocabulary used for standardisation (primary way types, secondary way types, way type extensions, articles, repetitions indicator, first names and other abbreviations. Also, at the end you will find a list defining those words used only for the AFNOR 32-character standard.

Reverse geocoding

(fr) Cette page n’est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

The query includes a pair of coordinates as input, the service returns one or several possible responses (depending on the maximum number of responses desired), including the address, the post code, the town, the country, and the distance between the result and the start point (coordinates indicated as input).

This reverse geocoding service consults the road network installed previously in the dedicated directory associated to it.

Availability

This web service is available at all times with Geoconcept Web and a graph.

Version change

Earlier versions of the web service have been conserved in Geoconcept Web to ensure compatibility with previous developments. We recommend using the most recent version.

Changes in relation to v3

- The "coordinates" parameter has been added to the response

Changes in relation to v2

- The "postalCode" parameter has been renamed "postCode"
- The "distanceToLocation" parameter has been renamed "distanceMetersToLocation"
- The "srs" parameter has been added to the response

V2

Parameters / properties

Input

parameter	description	optional	default
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
locations	Coordinates for the start point for reverse geocoding in SOAP, it will be necessary to supply coordinates in each of the dedicated tags <x></x> and <y></y> + in REST, the points are separated by the “;” character. The points are characterised by a pair of X, Y coordinates separated by a ‘,’.	no	
graphName	indicate the name of the network or graph to use so the WebService can operate.	yes	
maxDistance	maximum search distance to find neighbouring streets at the reverse geocoding start point	no	1000

parameter	description	optional	default
	if the parameter is passed, it is advisable to assign a value that is significantly higher than 1 (in metres).		
maxCandidates	maximum number of result addresses in the response if the parameter is passed, the minimum value to indicate is 1.	yes	1
maxDistanceBetweenCandidates	Maximum distance (in metres) between each candidate. if the parameter is passed, it is advisable to assign a value that is significantly higher than 1 (in metres).	no	100

Output

Reminder for the coordinates of the start point of the search(reverseGeocodingV2Response)

property	type	min/max	description
location	string	0/1	X,Y coordinates for the start point for the reverse geocoding
srs	string	0/1	projection (EPSG code such as epsg:4326 or wgs84)
address/addresses		0/ unlimited	list of (candidate) addresses returned

Candidates returned (Address)

property	type	min/max	description
addressLine	string	0/1	street found
postCode	string	0/1	post code found
city	string	0/1	town associated with the candidate returned
country	string	0/1	country associated to the candidate returned
distanceMetersToLocation	double	1/1	Distance (in metres) between the candidate returned and the start point
coordinates	string	1/1	Address coordinates

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/reverseGeocodingService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:reverseGeocodingV3>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <srs>epsg:4326</srs>
            <!--Zero or more repetitions:-->
            <locations>
               <x>-1.565769</x>
               <y>47.226236</y>
            </locations>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <maxDistance>1000</maxDistance>
            <!--Optional:-->
            <maxCandidates>2</maxCandidates>
            <!--Optional:-->
            <maxDistanceBetweenCandidates>100</maxDistanceBetweenCandidates>
         </request>
      </sch:reverseGeocodingV2>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:reverseGeocodingV3Response xmlns:ns2="http://geoconcept.com/gc/schemas">
        <reverseGeocodingResultsV3>
           <status>OK</status>
           <reverseGeocodingResult>
              <location>-1.565769;47.226236</location>
              <srs>epsg:4326</srs>
              <addresses>
                 <address>
                    <addressLine>26 RUE DU MAINE</addressLine>
                    <postCode>44000</postCode>
                    <city>NANTES</city>
                    <country>France</country>
                    <distanceMetersToLocation>1.2300580212448586</distanceMetersToLocation>
                    <coordinates>-1.565757,47.226229</coordinates>
                 </address>
                 <address>
                    <addressLine>2 RUE JOYAU</addressLine>
                    <postCode>44000</postCode>
                    <city>NANTES</city>
                    <country>France</country>
                    <distanceMetersToLocation>27.310568595900534</distanceMetersToLocation>
                    <coordinates>-1.566068,47.226099</coordinates>
                 </address>
              </addresses>
           </reverseGeocodingResult>
        </reverseGeocodingResultsV3>
      </ns2:reverseGeocodingV2Response>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/reverseGeocoding/v3.json?
locations=-1.565769,47.226236&maxDistance=1000&maxCandidates=2&maxDistanceBetweenCandidates=100
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/reverseGeocoding/v3.json?
locations=-1.565769,47.226236&maxDistance=1000&maxCandidates=2&maxDistanceBetweenCandidates=100&fonction=myFonction
```

XML query

```
http://<server>/<webapp>/api/lbs/reverseGeocoding/v3.xml?
locations=-1.565769,47.226236&maxDistance=1000&maxCandidates=2&maxDistanceBetweenCandidates=100
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message": null,
   "status": "OK",
   "reverseGeocodingResults": [
      {
         "location": "-1.565769;47.226236",
         "srs": null,
         "addresses": [
            {
               "addressLine": "26 RUE DU MAINE",
               "postCode": "44000",
               "city": "NANTES",
               "country": "France",
               "distanceMetersToLocation": 1.2300581,
               "coordinates": "-1.565757,47.226229"
            },
            {
               "addressLine": "2 RUE JOYAU",
               "postCode": "44000",
               "city": "NANTES",
               "country": "France",
               "distanceMetersToLocation": 27.310568,
               "coordinates": "-1.566068,47.226099"
            }
         ]
      }
   ]
}
```

JSON-P format

```
myCallback(
    {
       "message": null,
       "status": "OK",
       "reverseGeocodingResults": [
          {
             "location": "-1.565769;47.226236",
             "srs": null,
             "addresses": [
                {
                   "addressLine": "26 RUE DU MAINE",
                   "postCode": "44000",
                   "city": "NANTES",
                   "country": "France",
                   "distanceMetersToLocation": 1.2300581,
                   "coordinates": "-1.565757,47.226229"
                },
                {
                   "addressLine": "2 RUE JOYAU",
                   "postCode": "44000",
                   "city": "NANTES",
                   "country": "France",
                   "distanceMetersToLocation": 27.310568,
                   "coordinates": "-1.566068,47.226099"
                }
             ]
          }
       ]
    }
)
```

XML format

```
<reverseGeocodingResultsV3>
   <status>OK</status>
   <reverseGeocodingResult>
      <location>-1.565769;47.226236</location>
      <addresses>
         <address>
            <addressLine>26 RUE DU MAINE</addressLine>
            <postCode>44000</postCode>
            <city>NANTES</city>
            <country>France</country>
            <distanceMetersToLocation>1.2300580212448586</distanceMetersToLocation>
            <coordinates>-1.565757,47.226229</coordinates>
         </address>
         <address>
            <addressLine>2 RUE JOYAU</addressLine>
            <postCode>44000</postCode>
            <city>NANTES</city>
            <country>France</country>
            <distanceMetersToLocation>27.310568595900534</distanceMetersToLocation>
            <coordinates>-1.566068,47.226099</coordinates>
         </address>
      </addresses>
   </reverseGeocodingResult>
</reverseGeocodingResultsV3>
```

Possible responses

Example of a correct reverse geocoding (reverseGeocodingResult/status is OK)

```
<reverseGeocodingResultsV3>
   <status>OK</status>
   <reverseGeocodingResult>
      <location>-1.565769;47.226236</location>
      <addresses>
         <address>
            <addressLine>26 RUE DU MAINE</addressLine>
            <postCode>44000</postCode>
            <city>NANTES</city>
            <country>France</country>
            <distanceMetersToLocation>1.2300580212448586</distanceMetersToLocation>
            <coordinates>-1.565757,47.226229</coordinates>
         </address>
         <address>
            <addressLine>2 RUE JOYAU</addressLine>
            <postCode>44000</postCode>
            <city>NANTES</city>
            <country>France</country>
            <distanceMetersToLocation>27.310568595900534</distanceMetersToLocation>
            <coordinates>-1.566068,47.226099</coordinates>
         </address>
      </addresses>
   </reverseGeocodingResult>
</reverseGeocodingResultsV3>
```

Case of a reverse geocoding operation where the maxDistance does not respect a value of > 1 (a minimum value of 100 is recommended). (reverseGeocodingResult/status is OK)

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceResult>
   <message>ServiceException: Error in reverse geocoding computation Error in smartrouting Failed to execute
 ReverseGeocode com.geoconcept.smartrouting.SmartRoutingNativeException: failed to connect { -1.565769,
 47.226236, 0.000000 } failed to connect { -1.565769, 47.226236, 0.000000 }</message>
   <status>ERROR</status>
</serviceResult>
```

In the case of a query where one of the parameters (<locations>,…) is not set. (reverseGeocodingResult/status is ERROR)

```
<reverseGeocodingResultsV3>
    <message>locations must be not null</message>
    <status>ERROR</status>
</reverseGeocodingResultsV3>
```

Case of a query that does not snap to the graph (reverseGeocodingResult/status is ERROR)

```
<serviceResult>
    <message>ServiceException: Error in reverse geocoding computation Error in smartrouting Failed
 to execute ReverseGeocode com.geoconcept.smartrouting.SmartRoutingNativeException: failed to connect
 { -1.565769, 147.226236, 0.000000 } failed to connect { -1.565769, 147.226236, 0.000000 }
    </message>
    <status>ERROR</status>
</serviceResult>
```

V2

Parameters / properties

Input

parameter	description	optional	default
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
locations	Coordinates for the start point for reverse geocoding in SOAP, it will be necessary to supply coordinates in each of the dedicated tags <x></x> and <y></y> + in REST, the points are separated by the “;” character. The points are characterised by a pair of X, Y coordinates separated by a ‘,’.	no	
graphName	indicate the name of the network or graph to use so the WebService can operate.	yes	
maxDistance	maximum search distance to find neighbouring streets at the reverse geocoding start point	no	1000

parameter	description	optional	default
	if the parameter is passed, it is advisable to assign a value that is significantly higher than 1 (in metres).		
maxCandidates	maximum number of result addresses in the response if the parameter is passed, the minimum value to indicate is 1.	yes	1
maxDistanceBetweenCandidates	Maximum distance (in metres) between each candidate. if the parameter is passed, it is advisable to assign a value that is significantly higher than 1 (in metres).	no	100

Output

Reminder for the coordinates of the start point of the search(reverseGeocodingV2Response)

property	type	min/max	description
location	string	0/1	X,Y coordinates for the start point for the reverse geocoding
srs	string	0/1	projection (EPSG code such as epsg:4326 or wgs84)
address/addresses		0/ unlimited	list of (candidate) addresses returned

Candidates returned (Address)

property	type	min/max	description
addressLine	string	0/1	street found
postCode	string	0/1	post code found
city	string	0/1	town associated with the candidate returned
country	string	0/1	country associated to the candidate returned
distanceMetersToLocation	double	1/1	Distance (in metres) between the candidate returned and the start point

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/reverseGeocodingService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:reverseGeocodingV2>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <srs>epsg:4326</srs>
            <!--Zero or more repetitions:-->
            <locations>
               <x>-1.565769</x>
               <y>47.226236</y>
            </locations>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <maxDistance>1000</maxDistance>
            <!--Optional:-->
            <maxCandidates>2</maxCandidates>
            <!--Optional:-->
            <maxDistanceBetweenCandidates>100</maxDistanceBetweenCandidates>
         </request>
      </sch:reverseGeocodingV2>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:reverseGeocodingV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <ReverseGeocodingResults>
            <status>OK</status>
            <reverseGeocodingResult>
               <location>-1,565769;47,226236</location>
               <srs>epsg:4326</srs>
               <addresses>
                  <address>
                     <addressLine>24 RUE DU MAINE</addressLine>
                     <postCode>44000</postCode>
                     <city>NANTES</city>
                     <country>France</country>
                     <distanceMetersToLocation>1.2509365114145279</distanceMetersToLocation>
                  </address>
                  <address>
                     <addressLine>2 RUE JOYAU</addressLine>
                     <postCode>44000</postCode>
                     <city>NANTES</city>
                     <country>France</country>
                     <distanceMetersToLocation>27.312077690953934</distanceMetersToLocation>
                  </address>
               </addresses>
            </reverseGeocodingResult>
         </ReverseGeocodingResults>
      </ns2:reverseGeocodingV2Response>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/reverseGeocoding/v2.json?
locations=-1.565769,47.226236&maxDistance=1000&maxCandidates=2&maxDistanceBetweenCandidates=100
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/reverseGeocoding/v2.json?
locations=-1.565769,47.226236&maxDistance=1000&maxCandidates=2&maxDistanceBetweenCandidates=100&fonction=myFonction
```

XML query

```
http://<server>/<webapp>/api/lbs/reverseGeocoding/v2.xml?
locations=-1.565769,47.226236&maxDistance=1000&maxCandidates=2&maxDistanceBetweenCandidates=100
```

Response

The response is always in UTF-8 format.

JSON format

```
{
  "message": null,
  "status": "OK",
  "reverseGeocodingResults": [
    {
      "location": "-1.565769;47.226236",
      "addresses": [
        {
          "addressLine": "26 RUE DU MAINE",
          "postalCode": "44000",
          "city": "NANTES",
          "country": "France",
          "distanceToLocation": 1.2370980024960025
        },
        {
          "addressLine": "2 RUE JOYAU",
          "postalCode": "44000",
          "city": "NANTES",
          "country": "France",
          "distanceToLocation": 27.302664051548852
        }
      ]
    }
  ]
}
```

JSON-P format

```
myCallback(
    {
      "message": null,
      "status": "OK",
      "reverseGeocodingResults": [
        {
          "location": "-1.565769;47.226236",
          "addresses": [
            {
              "addressLine": "26 RUE DU MAINE",
              "postalCode": "44000",
              "city": "NANTES",
              "country": "France",
              "distanceToLocation": 1.2370980024960025
            },
            {
              "addressLine": "2 RUE JOYAU",
              "postalCode": "44000",
              "city": "NANTES",
              "country": "France",
              "distanceToLocation": 27.302664051548852
            }
          ]
        }
        ]
      }
   )
```

XML format

```
<reverseGeocodingResultsV2>
   <status>OK</status>
   <reverseGeocodingResult>
      <location>-1.565769;47.226236</location>
      <srs>epsg:4326</srs>
      <addresses>
         <address>
            <addressLine>26 RUE DU MAINE</addressLine>
            <postCode>44000</postCode>
            <city>NANTES</city>
            <country>France</country>
            <distanceMetersToLocation>1.2370980024960025</distanceMetersToLocation>
         </address>
         <address>
            <addressLine>2 RUE JOYAU</addressLine>
            <postCode>44000</postCode>
            <city>NANTES</city>
            <country>France</country>
            <distanceMetersToLocation>27.302664051548852</distanceMetersToLocation>
         </address>
      </addresses>
   </reverseGeocodingResult>
</reverseGeocodingResultsV2>
```

Possible responses

Example of a correct reverse geocoding (reverseGeocodingResult/status is OK)

```
<reverseGeocodingResultsV2>
   <status>OK</status>
   <reverseGeocodingResult>
      <location>-1.565769;47.226236</location>
      <srs>epsg:4326</srs>
      <addresses>
         <address>
            <addressLine>26 RUE DU MAINE</addressLine>
            <postCode>44000</postCode>
            <city>NANTES</city>
            <country>France</country>
            <distanceMetersToLocation>1.2370980024960025</distanceMetersToLocation>
         </address>
         <address>
            <addressLine>2 RUE JOYAU</addressLine>
            <postCode>44000</postCode>
            <city>NANTES</city>
            <country>France</country>
            <distanceMetersToLocation>27.302664051548852</distanceMetersToLocation>
         </address>
      </addresses>
   </reverseGeocodingResult>
</reverseGeocodingResultsV2>
```

Case of a reverse geocoding operation where the maxDistance does not respect a value of > 1 (a minimum value of 100 is recommended). (reverseGeocodingResult/status is OK)

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceResult>
   <message>ServiceException: Error in reverse geocoding computation Error in smartrouting Failed to execute
 ReverseGeocode com.geoconcept.smartrouting.SmartRoutingNativeException: failed to connect { -1.565769,
 47.226236, 0.000000 } failed to connect { -1.565769, 47.226236, 0.000000 }</message>
   <status>ERROR</status>
</serviceResult>
```

In the case of a query where one of the parameters (<locations>,…) is not set. (reverseGeocodingResult/status is ERROR)

```
<reverseGeocodingResultsV2>
    <message>locations must be not null</message>
    <status>ERROR</status>
</reverseGeocodingResultsV2>
```

Case of a query that does not snap to the graph from version 2.0.9 of SmartRouting Server (reverseGeocodingResult/status is ERROR)

```
<serviceResult>
    <message>ServiceException: Error in reverse geocoding computation Error in smartrouting Failed
 to execute ReverseGeocode com.geoconcept.smartrouting.SmartRoutingNativeException: failed to connect
 { -1.565769, 147.226236, 0.000000 } failed to connect { -1.565769, 147.226236, 0.000000 }
    </message>
    <status>ERROR</status>
</serviceResult>
```

V1

Parameters / properties

Input

parameter	description	optional	default
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
locations	Coordinates for the start point for reverse geocoding in SOAP, it will be necessary to supply coordinates in each of the dedicated tags <x></x> and <y></y> + in REST, the points are separated by the “;” character. The points are characterised by a pair of X, Y coordinates separated by a ','.	no	
graphName	indicate the name of the network or graph to use so the WebService can operate.	yes	
maxDistance	maximum search distance to find neighbouring streets at the reverse geocoding start point if the parameter is passed, it is advisable to assign a value that is significantly higher than 1 (in metres).	no	
maxCandidates	maximum number of result addresses in the response if the parameter is passed, the minimum value to indicate is 1.	yes	
maxDistanceBetweenCandidates	Maximum distance (in metres) between each candidate. if the parameter is passed, it is advisable to assign a value that is significantly higher than 1 (in metres).	no	

Output

Reminder for the coordinates of the start point of the search (location)

property	type	min/max	description
location	string	0/1	X,Y coordinates for the start point for the reverse geocoding
address/addresses		0/ unlimited	list of (candidate) addresses returned

Candidates returned (Address)

property	type	min/max	description
addressLine	string	0/1	street found
postalCode	string	0/1	post code found
city	string	0/1	town associated with the candidate returned
country	string	0/1	country associated to the candidate returned
distanceToLocation	integer	1/1	Distance (in metres) between the candidate returned and the start point

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/reverseGeocodingService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:reverseGeocoding>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <srs></srs>
            <!--Zero or more repetitions:-->
            <locations>
               <x>2.324382</x>
               <y>48.803305</y>
            </locations>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <maxDistance>1000</maxDistance>
            <!--Optional:-->
            <maxCandidates>10</maxCandidates>
            <!--Optional:-->
            <maxDistanceBetweenCandidates>100</maxDistanceBetweenCandidates>
         </request>
      </sch:reverseGeocoding>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
```

```
        <ns2:reverseGeocodingResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
           <ReverseGeocodingResults>
              <status>OK</status>
              <reverseGeocodingResult>
                 <location>2.324382;48.803305</location>
                 <addresses>
                    <address>
                       <addressLine>AVENUE ARISTIDE BRIAND</addressLine>
                       <postalCode>92220</postalCode>
                       <city>BAGNEUX</city>
                       <country>FRANCE</country>
                       <distanceToLocation>18.661267300047538</distanceToLocation>
                    </address>
                    <address>
                       <addressLine>RUE DU MIDI</addressLine>
                       <postalCode>94110</postalCode>
                       <city>ARCUEIL</city>
                       <country>FRANCE</country>
                       <distanceToLocation>29.691380904228758</distanceToLocation>
                    </address>
                 </addresses>
              </reverseGeocodingResult>
           </ReverseGeocodingResults>
        </ns2:reverseGeocodingResponse>
     </soap:Body>
  </soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/reverseGeocoding.json?
locations=2.324382,48.803305&maxDistance=1000&maxCandidates=2&maxDistanceBetweenCandidates=100
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/reverseGeocoding.json?
locations=2.324382,48.803305&maxDistance=1000&maxCandidates=2&maxDistanceBetweenCandidates=100&fonction=myFonction
```

XML query

```
http://<server>/<webapp>/api/lbs/reverseGeocoding.xml?
locations=2.324382,48.803305&maxDistance=1000&maxCandidates=2&maxDistanceBetweenCandidates=100
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message": null,
   "status": "OK",
   "reverseGeocodingResults": [   {
      "location": "2.324382;48.803305",
      "addresses":       [
                  {
```

```
            "addressLine": "AVENUE ARISTIDE BRIAND",
            "postalCode": "92220",
            "city": "BAGNEUX",
            "country": "FRANCE",
            "distanceToLocation": 18.661267300047538
         },
                 {
            "addressLine": "RUE DU MIDI",
            "postalCode": "94110",
            "city": "ARCUEIL",
            "country": "FRANCE",
            "distanceToLocation": 29.691380904228758
         }
      ]
   }]
}
```

JSON-P format

```
myCallback({
        "message":null,"status":"OK",
        "reverseGeocodingResults":[
                {
                        "location":"2.324382;48.803305",
                                "addresses":[
                                        {
                                                "addressLine":"AVENUE ARISTIDE BRIAND",
                                                "postalCode":"92220",
                                                "city":"BAGNEUX",
                                                "country":"FRANCE",
                                                "distanceToLocation":18.661267300047538
                                        },
                                        {
                                                "addressLine":"RUE DU MIDI",
                                                "postalCode":"94110",
                                                "city":"ARCUEIL",
                                                "country":"FRANCE",
                                                "distanceToLocation":29.691380904228758
                                        }
]}]}
```

XML format

```
<reverseGeocodingResults>
   <status>OK</status>
   <reverseGeocodingResult>
      <location>2.324382;48.803305</location>
      <addresses>
         <address>
            <addressLine>AVENUE ARISTIDE BRIAND</addressLine>
            <postalCode>92220</postalCode>
            <city>BAGNEUX</city>
            <country>FRANCE</country>
            <distanceToLocation>18.661267300047538</distanceToLocation>
         </address>
         <address>
            <addressLine>RUE DU MIDI</addressLine>
            <postalCode>94110</postalCode>
            <city>ARCUEIL</city>
            <country>FRANCE</country>
            <distanceToLocation>29.691380904228758</distanceToLocation>
         </address>
      </addresses>
   </reverseGeocodingResult>
</reverseGeocodingResults>
```

Possible responses

Example of a correct reverse geocoding (reverseGeocodingResult/status is OK)

```
<reverseGeocodingResults>
   <status>OK</status>
   <reverseGeocodingResult>
      <location>2.324382;48.803305</location>
      <addresses>
         <address>
            <addressLine>AVENUE ARISTIDE BRIAND</addressLine>
            <postalCode>92220</postalCode>
            <city>BAGNEUX</city>
            <country>FRANCE</country>
            <distanceToLocation>18.661267300047538</distanceToLocation>
         </address>
         <address>
            <addressLine>RUE DU MIDI</addressLine>
            <postalCode>94110</postalCode>
            <city>ARCUEIL</city>
            <country>FRANCE</country>
            <distanceToLocation>29.691380904228758</distanceToLocation>
         </address>
      </addresses>
   </reverseGeocodingResult>
</reverseGeocodingResults>
```

Case of a reverse geocoding operation where the maxDistanceBetweenCandidates does not respect a value of > 1 (a minimum value of 100 is recommended). (reverseGeocodingResult/status is OK)

```
<reverseGeocodingResults>
   <status>OK</status>
   <reverseGeocodingResult>
      <location>2.324382;48.803305</location>
      <addresses>
         <address>
            <addressLine>AVENUE ARISTIDE BRIAND</addressLine>
            <postalCode>92220</postalCode>
            <city>BAGNEUX</city>
            <country>FRANCE</country>
            <distanceToLocation>18.661267300047538</distanceToLocation>
         </address>
      </addresses>
   </reverseGeocodingResult>
</reverseGeocodingResults>
```

Case of a reverse geocoding operation where the maxDistance does not respect a value of > 1 (a minimum value of 100 is recommended). (reverseGeocodingResult/status is OK)

```
<reverseGeocodingResults>
   <status>OK</status>
   <reverseGeocodingResult>
      <location>2.324382;48.803305</location>
      <addresses/>
   </reverseGeocodingResult>
```

```
</reverseGeocodingResults>
```

Case of a query where the <location> parameter is not indicated. (reverseGeocodingResult/status is ERROR)

```
<reverseGeocodingResults>
   <message>locations must be not null</message>
   <status>ERROR</status>
</reverseGeocodingResults>
```

Case of a query that does not snap to the graph from version 2.0.9 of SmartRouting Server (serviceResult/status is ERROR)

```
<serviceResult>
    <message>ServiceException: Error in reverse geocoding computation
    Error in smartrouting
    Failed to execute ReverseGeocode
com.geoconcept.smartrouting.SmartRoutingNativeException: failed to connect { 2.165931, 49.218690, 0.000000 }
failed to connect { 2.165931, 49.218690, 0.000000 }
    </message>
    <status>ERROR</status>
</serviceResult>
```

FAQ

1. How is the maximum distance interpreted?

   The <maxDistance> tag is mandatory and must have a value higher than 1. Take care, nonetheless, if the maximum distance is not high enough, as the risk would be not to have any candidate returned at all.

2. What is the role of the maxCandidates parameter?

   This parameter allows you to define the number of responses you want to obtain, in the event that there are a large number of potential candidates. If no value is indicated, it is the iti.maxTargets value assigned in the Geoconcept Web Administration that is taken.

3. What is the significance of the maxDistanceBetweenCandidates parameter?

   If this parameter is filled in incorrectly, the webservice cannot return a positive response. We strongly advise assigning a value of > 1. If no value is mentioned, it is the iti.candidatesDeltaDistance value assigned in the Geoconcept Web Administration that will predominate. Care is needed, as if the distance between the candidates is not large enough, there is a significant risk of not having any candidates returned at all.

4. Can I use aliases instead of .siti file names to call a datasource?

   Yes. A specific configuration has to be performed to enable exploitation of the alias enabling utilisation of several sources for graphs. The steps to respect are the following:

   - Stop the Tomcat service,
   - Store the graphs (.siti and as an option the .siti.MG and .siti.GA files) in the appropriate directory. For example: here two graphs are stored in the following directory: [Home]\Geoconcept \SmartRouting\server\SRJEE\smartrouting\graphs, in the framework of a default parameter.
   - Edit the '*Service.xml*' file associated to the JEE SmartRouting module. This can be found here [Home]\Geoconcept\SmartRouting\server\SRJEE\smartrouting\conf.

- In the text editor of your choice, copy/paste as many times as you have graphs, the text block framed by the <datasource> tags shown below (in our example, there are 2 different sources, to be copied/pasted twice as a result),
- Copy this <datasource> block under the <default-datasource> tag of the initial block,
- The <file /> tag should indicate the name of one of the two graphs. The tag allows you to give an alias to this source, for example here: «HEREQ414».

Editing the first block associated to the first graph

```
        <datasource>
        <file>navteq_maps_for_geoconcept_Q414_france_v1.siti</file>
        <name>HEREQ414</name>

        <load-graph-settings>
            <full-load>false</full-load>
            <primary-language>en</primary-language>
        </load-graph-settings>

        <calculate-route-options>
            <vehicle-type/>
            <!-- a vehicle type name present in the vehicle types catalog -->
            <vehicle-type-id/>
            <!-- a vehicle type identifier present in the vehicle types catalog -->
            <route-search-criteria>time</route-search-criteria>
            <!-- distance, time, or an integer matching cost table index (graph dependant) -->
            <max-graph-snap-distance>500</max-graph-snap-distance>
            <!-- in meters -->
            <graph-snap-speed>1.11111111</graph-snap-speed>
            <!-- in m/s -->
            <route-descr-items/>
            <!-- composition (separated by comma) -->
            <reference-level>4</reference-level>
            <reject-flags/>
            <!-- composition (separated by comma) -->
            <speed-profile/>
            <input-coordinate-system/>
            <output-coordinate-system/>
            <use-meta-graph>true</use-meta-graph>
            <use-graph-accelerator>true</use-graph-accelerator>
            <fields/>
            <!-- composition (separated by comma) -->
            <tour-optimization>auto</tour-optimization>
            <language/>
            <!-- default output language -->
        </calculate-route-options>

        <calculate-route-settings>
            <calculate-ecotax>true</calculate-ecotax>
            <!-- calculate ecotax when available  -->
        </calculate-route-settings>

        <!-- if a section named 'search-around-options' (same definition as calculate-route-options) is
 present it defines particular default options for search around -->
        <!-- if a section named 'calculate-route-matrix-options' (same definition as calculate-route-
options) is present it defines particular default options for calculate route matrix -->
        <!-- if a section named 'calculate-tour-options' (same definition as calculate-route-options) is
 present it defines particular default options for calculate tour -->

        <!-- if a section named 'calculate-route-matrix-settings' (same definition as calculate-route-
matrix-settings) is present it defines particular settings for calculate route matrix -->
```

```
        <!-- if a section named 'search-around-settings' (same definition as calculate-route-settings) is
present it defines particular settings for search around -->

        <reverse-geocode-options>

            <max-distance>1000</max-distance>
            <!-- in meters -->
            <max-distance-between-candidates>2</max-distance-between-candidates>
            <!-- in meters -->
            <max-candidates>10</max-candidates>

            <input-coordinate-system/>
            <output-coordinate-system/>

            <include-street-number>true</include-street-number>
            <accept-emtpy-segments>true</accept-emtpy-segments>

            <language/>
            <places/>

        </reverse-geocode-options>

        <!-- reverse geocoding configuration usually depends on each particular graph -->
        <reverse-geocode-settings>
            <address-line-field>${built-in-field:name}</address-line-field>
            <city-name-field>City name</city-name-field>

            <postal-code-field>Postcode</postal-code-field>

            <country-field>Country</country-field>

            <right-suffix> right</right-suffix>
            <left-suffix> left</left-suffix>

            <number-begin-left-field>Begin left number</number-begin-left-field>
            <number-end-left-field>End left number</number-end-left-field>
            <number-begin-right-field>Begin right number</number-begin-right-field>
            <number-end-right-field>End right number</number-end-right-field>

        </reverse-geocode-settings>

        <calculate-tour-settings>
            <tour-optimization-auto-threshold>10</tour-optimization-auto-threshold>
        </calculate-tour-settings>

        <calculate-concentric-reachable-areas-options>
            <vehicle-type/>
            <!-- a vehicle type name present in the vehicle types catalog -->
            <vehicle-type-id/>
            <!-- a vehicle type identifier present in the vehicle types catalog -->
            <route-search-criteria>time</route-search-criteria>
            <!-- distance, time, or an integer matching cost table index (graph dependant) -->
            <max-graph-snap-distance>500</max-graph-snap-distance>
            <!-- in meters -->
            <graph-snap-speed>1.11111111</graph-snap-speed>
            <!-- in m/s -->
            <route-descr-items/>
            <!-- composition (separated by comma) -->
            <reference-level>4</reference-level>
            <reject-flags/>
            <!-- composition (separated by comma) -->
            <speed-profile/>
            <input-coordinate-system/>
            <output-coordinate-system/>
            <use-meta-graph>true</use-meta-graph>
            <use-graph-accelerator>true</use-graph-accelerator>

            <default-reachable-area-specification>
                <row-col>2000</row-col>
                <tile-resolution>50</tile-resolution>
                <max-holes>0</max-holes>
                <min-hole-size>0</min-hole-size>
                <extended-smoothing>false</extended-smoothing>
                <accept-non-connectable-links>true</accept-non-connectable-links>
            </default-reachable-area-specification>

        </calculate-concentric-reachable-areas-options>

        <calculate-concentric-reachable-areas-settings>
        </calculate-concentric-reachable-areas-settings>
    </datasource>
```

- leave the other default parameters, except where you decide to create a configuration that is specific to your installation.
- Repeat the steps of copying the <datasource> block and the editing of the <file/> and <name/> tags to configure the second data source.

Editing the second block associated to the other reference table that will be exploited

```
        <datasource>
        <file>BD_ADRESSE_France_v4.siti</file>
        <name>IGN</name>

        <load-graph-settings>
            <full-load>false</full-load>
            <primary-language>en</primary-language>
        </load-graph-settings>

        <calculate-route-options>
            <vehicle-type/>
            <!-- a vehicle type name present in the vehicle types catalog -->
            <vehicle-type-id/>
            <!-- a vehicle type identifier present in the vehicle types catalog -->
            <route-search-criteria>time</route-search-criteria>
            <!-- distance, time, or an integer matching cost table index (graph dependant) -->
            <max-graph-snap-distance>500</max-graph-snap-distance>
            <!-- in meters -->
            <graph-snap-speed>1.11111111</graph-snap-speed>
            <!-- in m/s -->
            <route-descr-items/>
            <!-- composition (separated by comma) -->
            <reference-level>4</reference-level>
            <reject-flags/>
            <!-- composition (separated by comma) -->
            <speed-profile/>
            <input-coordinate-system/>
            <output-coordinate-system/>
            <use-meta-graph>true</use-meta-graph>
            <use-graph-accelerator>true</use-graph-accelerator>
            <fields/>
            <!-- composition (separated by comma) -->
            <tour-optimization>auto</tour-optimization>
            <language/>
            <!-- default output language -->
        </calculate-route-options>

        <calculate-route-settings>
            <calculate-ecotax>true</calculate-ecotax>
            <!-- calculate ecotax when available  -->
        </calculate-route-settings>

        <!-- if a section named 'search-around-options' (same definition as calculate-route-options) is
 present it defines particular default options for search around -->
        <!-- if a section named 'calculate-route-matrix-options' (same definition as calculate-route-
options) is present it defines particular default options for calculate route matrix -->
        <!-- if a section named 'calculate-tour-options' (same definition as calculate-route-options) is
 present it defines particular default options for calculate tour -->

        <!-- if a section named 'calculate-route-matrix-settings' (same definition as calculate-route-
matrix-settings) is present it defines particular settings for calculate route matrix -->
        <!-- if a section named 'search-around-settings' (same definition as calculate-route-settings) is
 present it defines particular settings for search around -->

        <reverse-geocode-options>

            <max-distance>1000</max-distance>
            <!-- in meters -->
            <max-distance-between-candidates>2</max-distance-between-candidates>
            <!-- in meters -->
            <max-candidates>10</max-candidates>

            <input-coordinate-system/>
            <output-coordinate-system/>

            <include-street-number>true</include-street-number>
            <accept-emtpy-segments>true</accept-emtpy-segments>

            <language/>
            <places/>

        </reverse-geocode-options>

        <!-- reverse geocoding configuration usually depends on each particular graph -->
        <reverse-geocode-settings>
            <address-line-field>${built-in-field:name}</address-line-field>
            <city-name-field>City name</city-name-field>

            <postal-code-field>Postcode</postal-code-field>

            <country-field>Country</country-field>

            <right-suffix> right</right-suffix>
            <left-suffix> left</left-suffix>

            <number-begin-left-field>Begin left number</number-begin-left-field>
            <number-end-left-field>End left number</number-end-left-field>
            <number-begin-right-field>Begin right number</number-begin-right-field>
            <number-end-right-field>End right number</number-end-right-field>

        </reverse-geocode-settings>

        <calculate-tour-settings>
            <tour-optimization-auto-threshold>10</tour-optimization-auto-threshold>
        </calculate-tour-settings>

        <calculate-concentric-reachable-areas-options>
            <vehicle-type/>
            <!-- a vehicle type name present in the vehicle types catalog -->
            <vehicle-type-id/>
            <!-- a vehicle type identifier present in the vehicle types catalog -->
            <route-search-criteria>time</route-search-criteria>
            <!-- distance, time, or an integer matching cost table index (graph dependant) -->
            <max-graph-snap-distance>500</max-graph-snap-distance>
            <!-- in meters -->
            <graph-snap-speed>1.11111111</graph-snap-speed>
            <!-- in m/s -->
            <route-descr-items/>
            <!-- composition (separated by comma) -->
            <reference-level>4</reference-level>
            <reject-flags/>
            <!-- composition (separated by comma) -->
            <speed-profile/>
            <input-coordinate-system/>
            <output-coordinate-system/>
            <use-meta-graph>true</use-meta-graph>
            <use-graph-accelerator>true</use-graph-accelerator>

            <default-reachable-area-specification>
                <row-col>2000</row-col>
                <tile-resolution>50</tile-resolution>
                <max-holes>0</max-holes>
                <min-hole-size>0</min-hole-size>
                <extended-smoothing>false</extended-smoothing>
                <accept-non-connectable-links>true</accept-non-connectable-links>
            </default-reachable-area-specification>

        </calculate-concentric-reachable-areas-options>

        <calculate-concentric-reachable-areas-settings>
        </calculate-concentric-reachable-areas-settings>
    </datasource>
```

- Save the «Service.xml» file and rerun the Tomcat service,
- From now on, whenever you call the WebService, the user should call one of the two resources in a query, indicating in the <GraphName> parameter the alias that has been defined previously in the <name> tag. If no alias is passed, the source defined in the `iti.graphname` parameter will be used by default.

Find an object

(fr) Cette page n’est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

This web service uses an object’s geographic coordinates to retrieve information contained in the fields of the object in a Geoconcept map. The call to the web service requires as parameters the layer name, the structure of the map and of the field(s) to interrogate.

Availability

This web servce is available at all times with Geoconcept Web and a Geoconcept map.

Version change

Earlier versions of the web service are conserved in Geoconcept Web to ensure compatibility with earlier software versioning and development. We recommend using the most recent version.

Changes in V2

- The name of the web service has changed from "findObjectsUnder" to "findObjectV2"
- The Web service is no longer available in REST GET
- Replacement of "mapName" by "layerName"
- Addition of several "findObjectsTarget" search parameters
- Addition of "targetID", "maxDistance" and "maxCandidates" parameters
- Addition of "distance" and "wktGeometry" output parameters
- Deletion of the "topology" parameter

V2

Parameters / properties

Input

parameter	description	optional	default
srs	projection (EPSG code such as epsg:4326 or wgs84)	no	
geometries	Coordinates to find (series of Id,X,Y triplets separated by the ";" character)	no	
layerName	Name of the layer to use	no	
targets/target (findObjectsTarget)	Properties of searched objects	no	

Targets (findObjectsTarget)

parameter	description	optional	default
targetID	Identifier	yes	
className	Name of the Class	no	
subclassName	Name of the Subclass	no	
fields	Name of the fields separated by the ";" character	no	
maxDistance	Maximum search radius (in metres)	yes	
maxCandidates	Maximum number of candidates to return	yes	

Output

parameter	type	min/max	description
id	string	0/1	Objects identifier
targets/target (findTargetResult)	array (objectInfo)	0/ unlimited	Objects

Target Infos (findTargetResult)

parameter	type	min/max	description
targetID	string	0/1	Identifier
objects/object (objectInfoV2)	string	0/ unlimited	Object descriptions

Object Info (objectInfoV2)

parameter	type	min/max	description
distance	integer	1/1	Distance in metres betwen the origin point and the found object -1 if the calculated distance is not available
wktGeometry	string	0/1	Object geometry

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/findObjectsService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:findObjectV2>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <srs>epsg:4326</srs>
            <geometries>1,-1.80280,47.15524</geometries>
            <layerName>Fond de plan</layerName>
            <!--Optional:-->
            <targets>
               <!--1 or more repetitions:-->
               <target>
                  <!--Optional:-->
                  <targetId>1</targetId>
                  <className>Unité administrative</className>
                  <!--Optional:-->
                  <subClassName>Commune</subClassName>
                  <fields>Name;Code gouvernement</fields>
                  <!--Optional:-->
                  <maxDistance>1000</maxDistance>
                  <!--Optional:-->
                  <maxCandidates>5</maxCandidates>
               </target>
            </targets>
         </request>
      </sch:findObjectV2>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
```

```
         <ns2:findObjectV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
            <FindObjectResult>
               <status>OK</status>
               <results>
                  <id>1</id>
                  <targets>
                     <target>
                        <targetId>1</targetId>
                        <objects>
                           <object>
                              <fields>
                                 <field>Port-Saint-Père</field>
                                 <field>44133</field>
                              </fields>
                              <distance>0</distance>
                           </object>
                           <object>
                              <fields>
                                 <field>Rouans</field>
                                 <field>44145</field>
                              </fields>
                              <distance>447</distance>
                           </object>
                        </objects>
                     </target>
                  </targets>
               </results>
            </FindObjectResult>
         </ns2:findObjectV2Response>
      </soap:Body>
   </soap:Envelope>
```

REST (POST)

Query

Query

```
http://<server>/<webapp>/api/lbs/find/findObject/v2.xml
```

Data (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<findObjectRequestV2>
  <srs>epsg:4326</srs>
  <geometries>1,-1.80280,47.15524</geometries>
  <layerName>Fond de plan</layerName>
  <targets>
    <target>
      <targetId>1</targetId>
      <className>Unité administrative</className>
      <subClassName>Commune</subClassName>
      <fields>Name;Code gouvernement</fields>
      <maxDistance>0</maxDistance>
      <maxCandidates>1</maxCandidates>
    </target>
  </targets>
</findObjectRequestV2>
```

Response

The response is always in UTF-8 format.

XML format

```
<findObjectResultsV2>
   <status>OK</status>
   <results>
      <id>1</id>
      <targets>
         <target>
            <targetId>1</targetId>
            <objects>
               <object>
                  <fields>
                     <field>Port-Saint-Père</field>
                     <field>44133</field>
                  </fields>
                  <distance>0</distance>
               </object>
            </objects>
         </target>
      </targets>
   </results>
</findObjectResultsV2>
```

Query

JSON query

```
http://<server>/<webapp>/api/lbs/find/findObject/v2.json
```

JSON

```
{
  "srs" : "epsg:4326",
  "geometries" : "1,-1.80280,47.15524",
  "layerName" : "Fond de plan",
  "targets" : [ {
  "className" : "Unité administrative",
  "subClassName" : "Commune",
  "fields" : "Name;Code gouvernement"
  }]
}
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message": null,
   "status": "OK",
   "results": [   {
      "id": "1",
      "targets": [{"objects": [      {
         "distance": 0,
         "fields":         [
            "Port-Saint-Père",
            "44133"
         ]
      }]}]
   }]
}
```

Possible responses

Case of a response that is found (findObjectsResults/status est OK)

```
<findObjectResultsV2>
   <status>OK</status>
   <results>
      <id>1</id>
      <targets>
         <target>
            <targetId>1</targetId>
            <objects>
               <object>
                  <fields>
                     <field>Port-Saint-Père</field>
                     <field>44133</field>
                  </fields>
                  <distance>0</distance>
               </object>
            </objects>
         </target>
      </targets>
   </results>
</findObjectResultsV2>
```

Absence of an object on the position searched (findObjectsResultsV2/status is OK)

```
<findObjectsResultsV2>
   <status>OK</status>
   <results>
      <id>1</id>
      <objects/>
   </results>
</findObjectsResultsV2>
```

Faulty projection / srs (serviceResult/status is ERROR)

```
<serviceResult>
    <message>
        CoordinateTransformEngineException: Coordinate transformation failed
    </message>
    <status>ERROR</status>
</serviceResult>
```

Case, absence of an argument for ***geometries*** (serviceResult/status est ERROR)

```
<serviceResult>
   <message>IllegalArgumentException: Geometry has too few fields : String[][{1,-1.80280}]</message>
   <status>ERROR</status>
</serviceResult>
```

Case of a layer that is not found (serviceResult/status is ERROR)

```
<serviceResult>
   <message>WebServiceException: Layer name 'Fond de lan' does not exist</message>
   <status>ERROR</status>
</serviceResult>
```

Case of a Class that is not found (serviceResult/status est ERROR)

```
<serviceResult>
   <message>WebServiceException: Error in finding objects
failed to execute text request
failed to execute gcis request (text response)
failed to execute gcis request
exception occured while servicing request
Failed to service request
com.geoconcept.gc.GcException: native returned exception (code=1)
native returned exception
[FindObject-88304] unknown type 'Unité administrtive'</message>
   <status>ERROR</status>
</serviceResult>
```

Case of a Subclass that is not found (serviceResult/status est ERROR)

```
<serviceResult>
   <message>WebServiceException: Error in finding objects
failed to execute text request
failed to execute gcis request (text response)
failed to execute gcis request
exception occured while servicing request
Failed to service request
com.geoconcept.gc.GcException: native returned exception (code=1)
native returned exception
[FindObject-88305] unknown subtype 'Commun'</message>
   <status>ERROR</status>
</serviceResult>
```

Case of a field not found (serviceResult/status est ERROR)

```
<serviceResult>
   <message>WebServiceException: Error in finding objects
failed to execute text request
failed to execute gcis request (text response)
failed to execute gcis request
exception occured while servicing request
Failed to service request
com.geoconcept.gc.GcException: native returned exception (code=1)
native returned exception
[FindObject-88300] unknown field 'Code governement'</message>
   <status>ERROR</status>
</serviceResult>
```

V1

Parameters / properties

Input

parameter	description	optional	default
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
geometries	Coordinates to find (series of Id,X,Y triplets separated by the ";" character)	no	
topology	Topological relationship between the `geometries` and the objects to find	yes	
mapName	Name of the map	no	
className	Name of the Class	no	
subclassName	Name of the Subclass	no	
fields	Name of the fields separated by the ";" character	no	

Output

Objects (findObjectsResult)

parameter	type	min/max	description
id	string	0/1	Objects identifier
objects	array (objectInfo)	0/ unlimited	Objects

Object Info (objectInfo)

parameter	type	min/max	description
fields	string	0/ unlimited	Field attributes

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/findObjectsService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:findObjectsUnder>
         <!--Optional:-->
         <srs>epsg:4326</srs>
         <!--Optional:-->
         <geometries>1,-1.80280,47.15524</geometries>
         <!--Optional:-->
         <topology></topology>
         <!--Optional:-->
         <mapName>Loire.gcm</mapName>
         <!--Optional:-->
         <className>Unité administrative</className>
         <!--Optional:-->
         <subclassName>Commune</subclassName>
         <!--Optional:-->
         <fields>Nom;Population</fields>
      </sch:findObjectsUnder>
   </soapenv:Body>
```

```
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:findObjectsUnderResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <FindObjectsResult>
            <status>OK</status>
            <results>
               <id>1</id>
               <objects>
                  <object>
                     <fields>
                        <field>Port-Saint-Père</field>
                        <field>2724</field>
                     </fields>
                  </object>
               </objects>
            </results>
         </FindObjectsResult>
      </ns2:findObjectsUnderResponse>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/find/under.json?
srs=epsg:4326&geometries=1,-1.80280,47.15524&mapName=Loire.gcm&className=Unit
%C3%A9%20administrative&subclassName=Commune&fields=Nom;Population
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/find/under.xml?
srs=epsg:4326&geometries=1,-1.80280,47.15524&mapName=Loire.gcm&className=Unit
%C3%A9%20administrative&subclassName=Commune&fields=Nom;Population&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/find/under.xml?
srs=epsg:4326&geometries=1,-1.80280,47.15524&mapName=Loire.gcm&className=Unit
%C3%A9%20administrative&subclassName=Commune&fields=Nom;Population
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message":null,
   "status":"OK",
   "results":[
      {
         "id":"1",
         "objects":[
            {
               "fields":[
                  "Port-Saint-Père",
                  "2724"
               ]
            }
         ]
      }
   ]
}
```

JSON-P format

```
myCallback({
           "message":null,
           "status":"OK",
           "results":[
                  {
                         "id":"1",
                         "objects":[
                                {
                                   "fields":[
                                          "Port-Saint-Père",
                                          "2724"
                                   ]
                                }
                         ]
                  }
           ]
        }
);
```

XML format

```
<findObjectsResults>
        <status>OK</status>
        <results>
                <id>1</id>
                <objects>
                        <object>
                                <fields>
                                        <field>Port-Saint-Père</field>
                                        <field>2724</field>
                                </fields>
                        </object>
                </objects>
        </results>
</findObjectsResults>
```

Possible responses

Case of a response that is found (findObjectsResults/status est OK)

```
<findObjectsResults>
        <status>OK</status>
        <results>
```

```
                   <id>1</id>
                   <objects>
                           <object>
                                   <fields>
                                           <field>Port-Saint-Père</field>
                                           <field>2724</field>
                                   </fields>
                           </object>
                   </objects>
           </results>
   </findObjectsResults>
```

Absence of an object on the searched position (findObjectsResults/status is OK)

```
<findObjectsResults>
   <status>OK</status>
   <results>
      <id>1</id>
      <objects/>
   </results>
</findObjectsResults>
```

Faulty projection / srs (serviceResult/status is ERROR)

```
<serviceResult>
    <message>
        CoordinateTransformEngineException: Coordinate transformation failed
    </message>
    <status>ERROR</status>
</serviceResult>
```

Case, absence of an argument for ***geometries*** (serviceResult/status est ERROR)

```
<serviceResult>
   <message>IllegalArgumentException: Candidate has too few fields : [Ljava.lang.String;@5890d398</message>
   <status>ERROR</status>
</serviceResult>
```

Case of a map that is not found (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: getLayerInfo : Error reading map info on Loir.gcm
No metadata for this map has been found</message>
   <status>ERROR</status>
</serviceResult>
```

Case of a Class that is not found (serviceResult/status est ERROR)

```
<serviceResult>
   <message>WebServiceException: Error in finding objects
Error on ObjClass</message>
   <status>ERROR</status>
</serviceResult>
```

Case of a Subclass that is not found (serviceResult/status est ERROR)

```
<serviceResult>
```

```
   <message>WebServiceException: Error in finding objects
Error on ObjSubclass</message>
   <status>ERROR</status>
</serviceResult>
```

Case of a field not found (serviceResult/status est ERROR)

```
<serviceResult>
   <message>WebServiceException: Error in finding objects
Error on ObjField</message>
   <status>ERROR</status>
</serviceResult>
```

FAQ

1. Is it possible to use other topological relationships between objects?

   No, for the moment, only the *Intersect* relationship is available.

2. How should we process maps with non-latin encoding?

   In ADministration / Parameters, change the value of "geographics.server.gcis.gcisServerCharset" by "UTF-8".

3. Is it possible to pass several *geometries* when calling a Web Service?

   Yes, it will suffice to specify more triplets separated by the ";" character

   ```
   <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
   geoconcept.com/gc/schemas">
      <soapenv:Header/>
      <soapenv:Body>
         <sch:findObjectsUnder>
            <!--Optional:-->
            <srs>epsg:4326</srs>
            <!--Optional:-->
            <geometries>1,6.01501,49.350;2,6.01701,49.390</geometries>
            <!--Optional:-->
            <topology></topology>
            <!--Optional:-->
            <mapName>geowebSmp.gcm</mapName>
            <!--Optional:-->
            <className>Pyr</className>
            <!--Optional:-->
            <subclassName>P2L2</subclassName>
            <!--Optional:-->
            <fields>Nom;X;Y</fields>
         </sch:findObjectsUnder>
      </soapenv:Body>
   </soapenv:Envelope>
   ```

   Return:

+

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
        <soap:Body>
                <ns2:findObjectsUnderResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
```

```
                            <FindObjectsResult>
                                    <status>OK</status>
                                    <results>
                                            <id>1</id>
                                            <objects>
                                                    <object>
                                                            <fields>
                                                                    <field>FONTOY</field>
                                                                    <field>868858</field>
                                                                    <field>2489508</field>
                                                            </fields>
                                                    </object>
                                            </objects>
                                    </results>
                                    <results>
                                            <id>2</id>
                                            <objects>
                                                    <object>
                                                            <fields>
                                                                    <field>HAVANGE</field>
                                                                    <field>864342.4</field>
                                                                    <field>2494239.52</field>
                                                            </fields>
                                                    </object>
                                            </objects>
                                    </results>
                            </FindObjectsResult>
                    </ns2:findObjectsUnderResponse>
            </soap:Body>
    </soap:Envelope>
```

Route calculation

(fr) Cette page n’est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

This web service calculates an itinerary between two points and returns a full route sheet. Adding intermediate steps is an option for the user, in the context of the configured graph or network, the name of which has been specified in the interface.

Availability

This web service is available at all times with Geoconcept Web and a graph or network.

Version change

Earlier versions of the web service are conserved in Geoconcept Web to ensure compatibility with earlier software developments. We reommend using the most recent version.

Changes in relation to v6

- Addition of the "formatItems" and "fields" parameters.

Changes in relation to v5

- Addition of the "maxCost", "timeOut" and "computeOptions" parameters.
- Addition to the "nodes" snapMethod of snap to the nearest nodes.
- "compressedgeometry" and "compressedsimplifiedgeometry" formats have been added, with utilization of the *encoded polyline* compression algorithm (Encoded Polyline Algorithm Format [https://developers.google.com/maps/documentation/utilities/polylinealgorithm]) enabling storage of coordinates in a single ASCII character string, thereby significantly reducing overall data volumes.
- Addition of the "tollcost" format, requires an access license to the HERE platform.

Changes in relation to v4

- Addition of the notion of node, faster, to snap to graph nodes rather than to geographic coordinates. Addition of the following elements: "originNode", "destinationNode", "waypointNodes", "node" in the format parameter and "nodes" in the snapMethod parameter.
- Addition in the response of the following items: "originNode", "destinationNode", "waypointNodes" and "carbonFootprint".
- Addition of the "avoidArea" and "configName" parameters.

Changes in relation to v3

- Addition of the "snapMethod" parameter
- Deletion of the "projection" parameter, replaced by "srs"
- Deletion of the "RejectFlags" parameter, replaced by "exclusions"
- Change in the parameter type for the "distanceMeters" and "durationSeconds" parameters from String/ Double to Double
- The "geometryWkt" and "simplifiedWkt" parameters have been renamed "wktGeometry" and "wktSimplifiedGeometry" respectively
- The "navInstruvtion" parameter has been renamed "navigationInstruction"
- In RouteResultV3, addition of the "srs" parameter
- The "elapsedTimeSeconds" parameter has been renamed "durationSeconds" and its type has been changed from String to Double
- The "coordinates" parameter has been renamed "location"

Changes in relation to V2

- Addition of the "timeLine" parameter
- Deletion of the "map" and "applyMapPrecisionOut" parameters
- The "distance" and "duration" parameters have been renamed "distanceMeters" and "durationSeconds" respectivly

V6

Parameters / properties

Input

parameter	description	optional	default
origin	Start point coordinates. The longitude and latitude coordinates are separated by , characters.	no	
originNode	Start node. Care: a physical node does not have the same ID in another graph.	yes *	
destination	Finish point coordinates. The longitude and latitude coordinates are separated by the , character	no	
destinationNode	Arrival node. Care: a physical node does not have the same ID in another graph.	yes *	
waypoints	Coordinates for route stops. Each pair of coordinates for a stop is framed by the <waypoint> tag The longitude and latitude coordinates are separated by the , character	yes	
waypointNodes	Step nodes. Each pair of coordinates for a step is framed by the <waypointNode> tag Care: a physical node does not have the same ID in another graph.	yes *	
method	the shortest (distance) and fastest (time) route	yes	time
format	- standard: result = route summary / waypoints / geometry in wkt format / simplified geometry in wkt format / list of concatenated segments (without geometries) - extended: result = route summary / waypoints / simplified geometry in wkt format / list of non-concatenated segments (with geometries) - summary: result = route summary - geometry: result = route summary / waypoints / geometry in wkt format - simplifiedgeometry: result = route summary / waypoints / simplified geometry in wkt format - geometries: result = route summary / waypoints / geometry in wkt format / simplified geometry in wkt format - brief: result = route summary / list of segments with duration and distance - standardext: result = route summary / waypoints / list of concatenated segments with duration and distance and geometry - compressedgeometry: result = route summary / waypoints / compressed geometry with polyline encoding - compressedsimplifiedgeometry: result = route summary / waypoints / simplified compressed geometry with polyline encoding - node: result = route summary + Id of snapped nodes - completegeometry : result = route summary / waypoints / eometry in wkt format / list of concatenated segments with geometry in wkt format + - compresscompletegeometry : result = route summary / waypoints / compressed geometry with polyline encoding / list of concatenated segments and compressed geometry with polyline encoding - tollcost : résultat = route summary / cost of tolls. The cost is calculated according to the type of vehicle defined *configName* as well as the date/time of departure of the route *startDateTime* - custom: returns the list of items declared in the *customFormat* parameter.	yes	standard
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	

parameter	description	optional	default
tolerance	Tolerance distance (in metres) for the geometry simplification.	yes	
graphName (depreciated)	Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
startDateTime	Start date and time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a start date of 21 January 2014, at 9.00am in Paris. Caution: the + character can be misinterpreted by browsers, so in this case, it should be replaced by *%2B*.	yes	
profileId (depreciated)	Vehicle identifier (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
profileName (depreciated)	Vehicle profile (saved in the vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
exclusions	List of restriction rules to use, separated by the *;* character (Example: Toll, Tunnel, Bridge)	yes	
timeLine	List of intermediate durations to calculate positions along the route trajectory. Expressed in seconds, separated by the *;* character. The returned position corresponds to the previous node, on the path, connectable to the network. For example, on a highway, the returned position is the last exit or service area before the requested position.	yes	
snapMethod	Snap to graph method - standard: to the nearest connectable road section - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without any restriction rules - nodes: Snap directly to nodes supplied by the originNode, destinationNode and waypointNodes parameters, or, if these parameters have not been set, to the nearest nodes sourced by the origin, destination and waypoint parameters	yes	standard
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84: `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))` Care: WKT geometries must be closed.	yes	
configName	Name of the configuration to use (defined in Geoconcept Web - Administration / Tools / Road graphs) This replaces the use of graphName, profileId and profileName	yes	
computeOptions	List of options for the calculation,separated by *;* characters - trafficPatterns: uses routing statistics (you will need to supply a value for the startDateTime parameter and use a graph that includes traffic patterns information) - speedPattern (M18): uses a *speed pattern* as defined in the SmartRoutingVehicles.xml file, located in the *`'<GEOCONCEPT_WEB_HOME>''\smartrouting\jee\smartrouting\conf\* folder. Use as follows: "speedPattern:slow-speed" - length (M18): maximum authorised length in centimeters (you need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "length:950"	yes	

parameter	description	optional	default
	- width (M18): maximum authorised width in centimeters (You will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "width:255" - height (M18): maximum authorised height in centimeters (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "height:360" - weight (M18): maximum authorised weight in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weight:18000" - axles (M18): maximum authorised number of axles (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "axles:2" - weightPerAxle (M18): maximum authorised weight per axle in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weightPerAxle:9000" - fuelType: fuel type used for calculating the carbon footprint for the route. Values available are: "UndefinedFuelType", "NoFuel", "Diesel", "UnleadedFuel", "LGP" et "CustomFuelType". Usage : "fuelType:Diesel" - averageConsumption: average consumption in litres for 100 kilometers. Use as follows: "averageConsumption:6.45" - customAverageCO2UnitEmission: defines the carbon footprint in kilograms per litre. Use as follows: "customAverageCO2UnitEmission:2.724" - snapSpeed: snap-to-graph speed in kilometers per hour. Use as follows: "snapSpeed:10"		
maxCost	Maximum cost not to exceed in the calculation -1: no maximum cost to take into account 0: take the default value defined in the SmartRouting Server configuration if not: value in metres if method=distance or in seconds if method=time	yes	
timeOut	Time out for the calculation (in milliseconds)	yes	
formatItems	List of available format items, if the *format* parameter is set to custom - ROUTE_DISTANCE - ROUTE_DURATION - ROUTE_DISTANCE_FORMATTED - ROUTE_DURATION_FORMATTED - ROUTE_BOUNDS - SRS - ROUTE_CARBON_FOOTPRINT - START_FINISH_DATETIME - SUBROUTE_DISTANCE - SUBROUTE_DURATION - SUBROUTE_DISTANCE_FORMATTED - SUBROUTE_DURATION_FORMATTED - SEGMENT_DISTANCE - SEGMENT_DURATION - SEGMENT_DISTANCE_FORMATTED - SEGMENT_DURATION_FORMATTED - SEGMENT_NAME - SEGMENT_NAVIGATIONINSTRUCTION - SEGMENT_POINTSLIST - SEGMENT_FIELDSVALUES - SEGMENT_UNCONSOLIDATED - ROUTE_GEOMETRY_WKT - ROUTE_SIMPLIFIEDGEOMETRY_WKT	yes	

parameter	description	optional	default
	- ROUTE_GEOMETRY_COMPRESSED - ROUTE_SIMPLIFIEDGEOMETRY_COMPRESSED - SUBROUTE_GEOMETRY_WKT - SUBROUTE_GEOMETRY_COMPRESSED - TIMELINE - POINTS_GRAPHNODES - ROUTE_TOLLCOST - COMPUTATION_TIME		
fields	List of fields to be displayed as a result of each segment (for the standard and extended _formats) - Iso country code - Country - Postcode left - Postcode right - City name left - City name right - Ramp - Urban - Tunnel - Toll - Bicycles - Automobiles - Frontage - Bridge	yes	

(M18) Available from version M18 and later versions of graphs supplied by GEOCONCEPT SAS.

Output

Route (routeResultV6)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	double	0/1	Total route distance in metres.
durationSeconds	double	0/1	Total duration of the route in seconds.
waypoints	string	0/1	Waypoints (BoundingBox) for the route geometry.
wktGeometry	string	0/1	Geometry of the route in WKT format
wktSimplified	string	0/1	Simplified geometry in WKT format.
leg (or legs in JSON / JSON-P)	subRouteV5 (or array in JSON / JSON-P)	0/ unlimited	List of route segments.
startDateTime	string	0/1	Start date and time (format: ISO8601 without any area code: local time) Example: 2014-01-21T09:00:00.000+01:00 for a departure on 21 January 2014, at 9.00am in Paris
finishDateTime	string	0/1	Arrival date and time (format: ISO8601 without any area code: local time)

parameter	type	min/max	description
			Example: 2014-01-21T09:00:00.000+01:00 for an arrival time on 21 January 2014, at 9.00am in Paris
timeLineItem	timeLineItemV5 (or an array in JSON / JSON-P)	0/ unlimited	List of calculated positions.
srs	string	0/1	projection passed as input (EPSG code such as epsg:4326 or wgs84)
originNode	string	0/1	Start node identifier (entered if format=NODE).
destinationNode	string	0/1	Arrival node identifier (entered if format=NODE).
waypointNodes	waypointNodes array (string)	0/ unlimited	Step nodes identifiers (entered if format=NODE).
carbonFootprint	double	0/1	Carbon footprint (CO2 emissions in kilograms)
tollCost	Array de tollCost	0/ unlimited	Toll cost information (filled in if format=tollCost).
fieldsNames	Array de fields	0/ unlimited	list of requested fields with the parameter *field*.

Route section (subRouteV6)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	double	0/1	Distance covered by the route segment in metres.
durationSeconds	double	0/1	Duration of the route segment in seconds.
step or (or steps in JSON / JSON-P)	segmentV5 (or array in JSON / JSON-P)	0/ unlimited	List of component segments in the route segments

Route segment (segmentV6)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	double	0/1	Distance covered by the route segment in metres.
durationSeconds	double	0/1	Duration of the route segment in seconds.
navigationInstruction	string	0/1	Navigation instruction code: - F: Straight on - FR: Turn slightly to the right - FL: Turn slightly to the left - R: Turn right - L: Turn left - BR: Turn hard right - BL: Turn hard left

parameter	type	min/max	description
			- B: U-turn - round_about_entry: Enter roundabout - round_about_exit: Exit from roundabout
name	string	0/1	Segment street name.
point	string	0/ unlimited	List of coordinates separated by the , character

Route position (timeLineItemV6)

parameter	type	min/max	description
durationSeconds	double	0/1	Route duration in seconds up to this position.
message	string	0/1	Error message for this position.
status	string	0/1	Status of this position.
distanceMeters	double	0/1	Distance of the route in metres up to this position.
location	string	0/1	Coordinates of the position.

TollCost (tollCost)

parameter	type	min/max	description
amount	double	0/1	Total cost of tolls.
currency	string	0/1	Currency.
costsByCountry	Array de costsByCountry	0/ unlimited	List of toll costs by country.

Cost of tolls per country (costsByCountry)

parameter	type	min/max	description
amount	double	0/1	Cost of tolls.
country	string	0/1	Country code (3-letter ISO country code).

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/routeService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:routeV6>
         <!--Optional:-->
         <request>
            <origin>
               <x>-1.351448</x>
               <y>47.446923</y>
            </origin>
            <!--Optional:-->
            <originNode></originNode>
            <destination>
```

```
                <x>-1.34529</x>
                <y>47.4479931</y>
             </destination>
             <!--Optional:-->
             <destinationNode></destinationNode>
             <!--Optional:-->
             <waypoints>
                <!--Zero or more repetitions:-->
                <waypoint>
                   <x>-1.34981</x>
                   <y>47.44837</y>
                </waypoint>
             </waypoints>
             <!--Optional:-->
             <waypointNodes>
                <!--Zero or more repetitions:-->
                <waypointNode></waypointNode>
             </waypointNodes>
             <!--Optional:-->
             <srs>epsg:4326</srs>
             <!--Optional:-->
             <method>time</method>
             <!--Optional:-->
             <format>STANDARD</format>
             <!--Optional:-->
             <tolerance>0.</tolerance>
             <!--Optional:-->
             <graphName></graphName>
             <!--Optional:-->
             <startDateTime></startDateTime>
             <!--Optional:-->
             <profileId></profileId>
             <!--Optional:-->
             <profileName></profileName>
             <!--Optional:-->
             <exclusions>
                <!--Zero or more repetitions:-->
                <exclusion></exclusion>
             </exclusions>
             <!--Optional:-->
             <timeLine>
                <!--Zero or more repetitions:-->
                <timeLineItem></timeLineItem>
             </timeLine>
             <!--Optional:-->
             <snapMethod></snapMethod>
             <!--Optional:-->
             <avoidArea></avoidArea>
             <!--Optional:-->
             <computeOptions></computeOptions>
             <!--Optional:-->
             <timeOut></timeOut>
             <!--Optional:-->
             <configName></configName>
          </request>
       </sch:routeV6>
    </soapenv:Body>
 </soapenv:Envelope>
```

Response

```
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
```

```
      <ns2:routeV6Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <RouteResult>
            <status>OK</status>
            <distance>640 m</distance>
            <duration>0:02:06</duration>
            <distanceMeters>639.97</distanceMeters>
            <durationSeconds>126.88</durationSeconds>
            <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
            <wktGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, [...] ,  -1.34529
47.447993)</wktGeometry>
            <wktSimplifiedGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922,  [...] , -1.34529
47.447993)</wktSimplifiedGeometry>
            <leg>
               <distance>233 m</distance>
               <duration>0:00:46</duration>
               <distanceMeters>233.1</distanceMeters>
               <durationSeconds>46.28</durationSeconds>
               <step>
                  <distance>139 m</distance>
                  <duration>0:00:29</duration>
                  <distanceMeters>138.59</distanceMeters>
                  <durationSeconds>29.28</durationSeconds>
                  <name>RUE LA FONTAINE BRUNEAU</name>
               </step>
               <step>
                  <distance>91 m</distance>
                  <duration>0:00:16</duration>
                  <distanceMeters>91.33</distanceMeters>
                  <durationSeconds>16.43</durationSeconds>
                  <navigationInstruction>FR</navigationInstruction>
                  <name>RUE DES SAULES</name>
               </step>
               <step>
                  <distance>3 m</distance>
                  <duration>0:00:00</duration>
                  <distanceMeters>3.18</distanceMeters>
                  <durationSeconds>0.57</durationSeconds>
                  <navigationInstruction>round_about_entry</navigationInstruction>
                  <name />
               </step>
            </leg>
            <leg>
               <distance>407 m</distance>
               <duration>0:01:20</duration>
               <distanceMeters>406.87</distanceMeters>
               <durationSeconds>80.6</durationSeconds>
               <step>
                  <distance>18 m</distance>
                  <duration>0:00:03</duration>
                  <distanceMeters>17.64</distanceMeters>
                  <durationSeconds>3.17</durationSeconds>
                  <name />
               </step>
               <step>
                  <distance>67 m</distance>
                  <duration>0:00:15</duration>
                  <distanceMeters>66.62</distanceMeters>
                  <durationSeconds>15.05</durationSeconds>
                  <navigationInstruction>round_about_exit</navigationInstruction>
                  <name>RUE DES FOURS</name>
               </step>
               <step>
                  <distance>36 m</distance>
```

```
                    <duration>0:00:08</duration>
                    <distanceMeters>35.74</distanceMeters>
                    <durationSeconds>8.57</durationSeconds>
                    <navigationInstruction>F</navigationInstruction>
                    <name>PLACE DE L'ÉGLISE</name>
                 </step>
                 <step>
                    <distance>72 m</distance>
                    <duration>0:00:15</duration>
                    <distanceMeters>72.25</distanceMeters>
                    <durationSeconds>15.24</durationSeconds>
                    <navigationInstruction>F</navigationInstruction>
                    <name>RUE DES PRESSOIRS</name>
                 </step>
                 <step>
                    <distance>26 m</distance>
                    <duration>0:00:04</duration>
                    <distanceMeters>26.02</distanceMeters>
                    <durationSeconds>4.68</durationSeconds>
                    <navigationInstruction>round_about_entry</navigationInstruction>
                    <name />
                 </step>
                 <step>
                    <distance>183 m</distance>
                    <duration>0:00:32</duration>
                    <distanceMeters>182.84</distanceMeters>
                    <durationSeconds>32.91</durationSeconds>
                    <navigationInstruction>round_about_exit</navigationInstruction>
                    <name>RUE DU BOURG DRAPÉ</name>
                 </step>
                 <step>
                    <distance>6 m</distance>
                    <duration>0:00:00</duration>
                    <distanceMeters>5.76</distanceMeters>
                    <durationSeconds>0.98</durationSeconds>
                    <navigationInstruction>FR</navigationInstruction>
                    <name>RUE DE LA CURE</name>
                 </step>
              </leg>
              <srs>epsg:4326</srs>
              <carbonFootprint>0.0</carbonFootprint>
           </RouteResult>
        </ns2:routeV6Response>
     </soap:Body>
  </soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/route/v6.json?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/route/v6.json?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/route/v6.xml?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837
```

Response

The response is always in UTF-8 format.

JSON format

```
{
    "message": null,
    "status": "OK",
    "distance": "640 m",
    "duration": "0:02:09",
    "distanceMeters": 639.95,
    "durationSeconds": 129.48,
    "bounds": "-1.351448,47.446922;-1.345263,47.44848",
    "wktGeometry": "LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])",
    "wktSimplifiedGeometry":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
 [...])",
    "legs": [
        {
            "distance": "233 m",
            "duration": "0:00:46",
            "distanceMeters": 233.09,
            "durationSeconds": 46.84,
            "steps": [
                {
                    "distance": "139 m",
                    "duration": "0:00:29",
                    "distanceMeters": 138.59,
                    "durationSeconds": 29.83,
                    "navigationInstruction": null,
                    "name": "RUE LA FONTAINE BRUNEAU",
                    "points": []
                },
                {
                    [...]
                }
            ]
        },
        {
            "distance": "407 m",
            "duration": "0:01:22",
            "distanceMeters": 406.86,
            "durationSeconds": 82.64,
            "steps": [
                {
                    "distance": "18 m",
                    "duration": "0:00:03",
                    "distanceMeters": 17.63,
                    "durationSeconds": 3.18,
                    "navigationInstruction": null,
                    "name": "",
                    "points": []
                },
                {
                   [...]
                }
            ]
        }
    ],
    "startDateTime": null,
    "finishDateTime": null,
    "srs": null,
    "originNode": null,
    "waypointNodes": null,
    "destinationNode": null,
    "carbonFootprint": 0.195038864105688
}
```

JSON-P format

```
myCallback(
    {
        "message": null,
        "status": "OK",
        "distance": "640 m",
        "duration": "0:02:09",
        "distanceMeters": 639.95,
        "durationSeconds": 129.48,
        "bounds": "-1.351448,47.446922;-1.345263,47.44848",
        "wktGeometry": "LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])",
        "wktSimplifiedGeometry":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
 [...])",
        "legs": [
            {
                "distance": "233 m",
                "duration": "0:00:46",
                "distanceMeters": 233.09,
                "durationSeconds": 46.84,
                "steps": [
                    {
                        "distance": "139 m",
                        "duration": "0:00:29",
                        "distanceMeters": 138.59,
                        "durationSeconds": 29.83,
                        "navigationInstruction": null,
                        "name": "RUE LA FONTAINE BRUNEAU",
                        "points": []
                    },
                    {
                        [...]
                    }
                ]
            },
            {
                "distance": "407 m",
                "duration": "0:01:22",
                "distanceMeters": 406.86,
                "durationSeconds": 82.64,
                "steps": [
                    {
                        "distance": "18 m",
                        "duration": "0:00:03",
                        "distanceMeters": 17.63,
                        "durationSeconds": 3.18,
                        "navigationInstruction": null,
                        "name": "",
                        "points": []
                    },
                    {
                       [...]
                    }
                ]
            }
        ],
        "startDateTime": null,
        "finishDateTime": null,
        "srs": null,
        "originNode": null,
        "waypointNodes": null,
        "destinationNode": null,
        "carbonFootprint": 0.195038864105688
    }
);
```

XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<routeResultV6>
        <status>OK</status>
        <distance>640 m</distance>
        <duration>0:02:09</duration>
        <distanceMeters>639.95</distanceMeters>
        <durationSeconds>129.48</durationSeconds>
        <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
        <wktGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
wktGeometry>
        <wktSimplifiedGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
 [...])</wktSimplifiedGeometry>
        <leg>
                <distance>233 m</distance>
                <duration>0:00:46</duration>
                <distanceMeters>233.09</distanceMeters>
                <durationSeconds>46.84</durationSeconds>
                <step>
                        <distance>139 m</distance>
                        <duration>0:00:29</duration>
                        <distanceMeters>138.59</distanceMeters>
                        <durationSeconds>29.83</durationSeconds>
                        <name>RUE LA FONTAINE BRUNEAU</name>
                </step>
                <step>
            [...]
                </step>
        </leg>
        <leg>
                <distance>407 m</distance>
                <duration>0:01:22</duration>
                <distanceMeters>406.86</distanceMeters>
                <durationSeconds>82.64</durationSeconds>
                <step>
                        <distance>18 m</distance>
                        <duration>0:00:03</duration>
                        <distanceMeters>17.63</distanceMeters>
                        <durationSeconds>3.18</durationSeconds>
                        <name />
                </step>
                <step>
                        [...]
                </step>
        </leg>
        <carbonFootprint>0.195038864105688</carbonFootprint>
</routeResultV6>
```

JavaScript API

Include the JavaScript library

```
var routeCtrl = new GCUI.Control.Route();
routeCtrl.route({
    url:'http://<server>/<webapp>/api/lbs/route/v5.json',
    tolerance : 100,
    origin : new OpenLayers.LonLat(0.691012, 47.384813),
    destination : new OpenLayers.LonLat(0.691012, 47.384813),
    waypoints : [ new OpenLayers.LonLat(2.344408, 49.898798) ],
    callback : function(result, options) {
        console.log(result);
    }
});
```

The `result` variable is in JSON format as described above. The `callback` function passed to parameter is called at the end of the route calculation.

Possible responses

Case of a route found (routeResultV6/status is OK)

```
<?xml version="1.0" encoding="UTF-8"?>
<routeResultV6>
        <status>OK</status>
        <distance>640 m</distance>
        <duration>0:02:09</duration>
        <distanceMeters>639.95</distanceMeters>
        <durationSeconds>129.48</durationSeconds>
        <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
        <wktGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
wktGeometry>
        <wktSimplifiedGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
 [...])</wktSimplifiedGeometry>
        <leg>
                <distance>233 m</distance>
                <duration>0:00:46</duration>
                <distanceMeters>233.09</distanceMeters>
                <durationSeconds>46.84</durationSeconds>
                <step>
                        <distance>139 m</distance>
                        <duration>0:00:29</duration>
                        <distanceMeters>138.59</distanceMeters>
                        <durationSeconds>29.83</durationSeconds>
                        <name>RUE LA FONTAINE BRUNEAU</name>
                </step>
                <step>
            [...]
                </step>
        </leg>
        <leg>
                <distance>407 m</distance>
                <duration>0:01:22</duration>
                <distanceMeters>406.86</distanceMeters>
                <durationSeconds>82.64</durationSeconds>
                <step>
                        <distance>18 m</distance>
                        <duration>0:00:03</duration>
                        <distanceMeters>17.63</distanceMeters>
                        <durationSeconds>3.18</durationSeconds>
                        <name />
                </step>
                <step>
                        [...]
                </step>
        </leg>
        <carbonFootprint>0.195038864105688</carbonFootprint>
</routeResultV6>
```

Case of a forgotten specification of a start or finish point (routeResultV6/status is ERROR)

```
<routeResultV6>
   <message>Origin and destination must be not null</message>
   <status>ERROR</status>
   <distanceMeters>0.0</distanceMeters>
   <durationSeconds>0.0</durationSeconds>
</routeResultV6>
```

Case of a forgotten specification of a start or finish point (routeResultV6/status is ERROR)

```
<routeResultV6>
   <message>Origin, destination and waypoints should be represented by a couple of coordinates</message>
   <status>ERROR</status>
   <distanceMeters>0.0</distanceMeters>
   <durationSeconds>0.0</durationSeconds>
</routeResultV6>
```

Case of a faulty type (serviceResult/status is ERROR)

```
<serviceResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
</serviceResult>
```

Case of a snap-to-graph error (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in route computation
Error in smartrouting
Failed to execute calculateRoute
com.geoconcept.smartrouting.SmartRoutingNativeException: failed to connect waypoint { 146.691012, 47.384813,
 0.000000 }
failed to connect waypoint { 146.691012, 47.384813, 0.000000 }</message>
   <status>ERROR</status>
</serviceResult>
```

Case of a problem with the graph: absent file, faulty filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
    <message>ServiceException: Error in route computation
Error in smartrouting
datasource is null</message>
    <status>ERROR</status>
</serviceResult>
```

V5

Parameters / properties

Input

parameter	description	optional	default
origin	Start point coordinates. The longitude and latitude coordinates are separated by , characters.	no	
originNode	Start node. Care: a physical node does not have the same ID in another graph.	yes *	
destination	Finish point coordinates. The longitude and latitude coordinates are separated by the , character	no	
destinationNode	Arrival node. Care: a physical node does not have the same ID in another graph.	yes *	
waypoints	Coordinates for route stops. Each pair of coordinates for a stop is framed by the <waypoint> tag The longitude and latitude coordinates are separated by the , character	yes	
waypointNodes	Step nodes. Each pair of coordinates for a step is framed by the <waypointNode> tag Care: a physical node does not have the same ID in another graph.	yes *	
method	the shortest (distance) and fastest (time) route	yes	time
format	- standard: result = route summary / waypoints / geometry in wkt format / simplified geometry in wkt format / list of concatenated segments (without geometries) - extended: result = route summary / waypoints / simplified geometry in wkt format / list of non-concatenated segments (with geometries) - summary: result = route summary - geometry: result = route summary / waypoints / geometry in wkt format - simplifiedgeometry: result = route summary / waypoints / simplified geometry in wkt format - geometries: result = route summary / waypoints / geometry in wkt format / simplified geometry in wkt format - brief: result = route summary / list of segments with duration and distance - standardext: result = route summary / waypoints / list of concatenated segments with duration and distance and geometry - compressedgeometry: result = route summary / waypoints / compressed geometry with polyline encoding - compressedsimplifiedgeometry: result = route summary / waypoints / simplified compressed geometry with polyline encoding - node: result = route summary + Id of snapped nodes - completegeometry : result = route summary / waypoints / geometry in wkt format / list of concatenated segments with geometry in wkt format + - compresscompletegeometry : result = route summary / waypoints / compressed geometry with polyline encoding + / list of concatenated segments with compressed geometry with polyline encoding - tollcost: résultat = route summary / cost of tolls. The cost is calculated according to the type of vehicle defined *configName* as well as the date/time of departure of the route *startDateTime*.	yes	standard
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
tolerance	Tolerance distance (in metres) for the geometry simplification.	yes	

parameter	description	optional	default
graphName (depreciated)	Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
startDateTime	Start date and time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a start date of 21 January 2014, at 9.00am in Paris. Caution: the + character can be misinterpreted by browsers, so in this case, it should be replaced by *%2B*.	yes	
profileId (depreciated)	Vehicle identifier (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
profileName (depreciated)	Vehicle profile (saved in the vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
exclusions	List of restriction rules to use, separated by the *;* character (Example: Toll, Tunnel, Bridge)	yes	
timeLine	List of intermediate durations to calculate positions along the route trajectory. Expressed in seconds, separated by the *;* character. The returned position corresponds to the previous node, on the path, connectable to the network. For example, on a highway, the returned position is the last exit or service area before the requested position.	yes	
snapMethod	Snap to graph method - standard: to the nearest connectable road section - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without any restriction rules - nodes: Snap directly to nodes supplied by the originNode, destinationNode and waypointNodes parameters, or, if these parameters have not been set, to the nearest nodes sourced by the origin, destination and waypoint parameters	yes	standard
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84: `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))` Care: WKT geometries must be closed.	yes	
configName	Name of the configuration to use (defined in Geoconcept Web - Administration / Tools / Road graphs) This replaces the use of graphName, profileId and profileName	yes	
computeOptions	List of options for the calculation,separated by *;* characters - trafficPatterns: uses routing statistics (you will need to supply a value for the startDateTime parameter and use a graph that includes traffic patterns information) - speedPattern (M18): uses a *speed pattern* as defined in the SmartRoutingVehicles.xml file, located in the `*‘<GEOCONCEPT_WEB_HOME>’'\smartrouting\jee\smartrouting\conf\* folder. Use as follows: "speedPattern:slow-speed" - length (M18): maximum authorised length in centimeters (you need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "length:950"	yes	

parameter	description	optional	default
	- width (M18): maximum authorised width in centimeters (You will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "width:255" - height (M18): maximum authorised height in centimeters (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "height:360" - weight (M18): maximum authorised weight in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weight:18000" - axles (M18): maximum authorised number of axles (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "axles:2" - weightPerAxle (M18): maximum authorised weight per axle in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weightPerAxle:9000" - fuelType: fuel type used for calculating the carbon footprint for the route. Values available are: "UndefinedFuelType", "NoFuel", "Diesel", "UnleadedFuel", "LGP" et "CustomFuelType". Usage : "fuelType:Diesel" - averageConsumption: average consumption in litres for 100 kilometers. Use as follows: "averageConsumption:6.45" - customAverageCO2UnitEmission: defines the carbon footprint in kilograms per litre. Use as follows: "customAverageCO2UnitEmission:2.724" - snapSpeed: snap-to-graph speed in kilometers per hour. Use as follows: "snapSpeed:10"		
maxCost	Maximum cost not to exceed in the calculation -1: no maximum cost to take into account 0: take the default value defined in the SmartRouting Server configuration if not: value in metres if method=distance or in seconds if method=time	yes	
timeOut	Time out for the calculation (in milliseconds)	yes	

(M18) Available from version M18 and later versions of graphs supplied by GEOCONCEPT SAS.

Output

Route (routeResultV5)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	double	0/1	Total route distance in metres.
durationSeconds	double	0/1	Total duration of the route in seconds.
waypoints	string	0/1	Waypoints (BoundingBox) for the route geometry.
wktGeometry	string	0/1	Geometry of the route in WKT format
wktSimplified	string	0/1	Simplified geometry in WKT format.
leg (or legs in JSON / JSON-P)	subRouteV5 (or array in JSON / JSON-P)	0/ unlimited	List of route segments.

parameter	type	min/max	description
startDateTime	string	0/1	Start date and time (format: ISO8601 without any area code: local time) Example: 2014-01-21T09:00:00.000+01:00 for a departure on 21 January 2014, at 9.00am in Paris
finishDateTime	string	0/1	Arrival date and time (format: ISO8601 without any area code: local time) Example: 2014-01-21T09:00:00.000+01:00 for an arrival time on 21 January 2014, at 9.00am in Paris
timeLineItem	timeLineItemV5 (or an array in JSON / JSON-P)	0/ unlimited	List of calculated positions.
srs	string	0/1	projection passed as input (EPSG code such as epsg:4326 or wgs84)
originNode	string	0/1	Start node identifier (entered if format=NODE).
destinationNode	string	0/1	Arrival node identifier (entered if format=NODE).
waypointNodes	waypointNodes array (string)	0/ unlimited	Step nodes identifiers (entered if format=NODE).
carbonFootprint	double	0/1	Carbon footprint (CO2 emissions in kilograms)
tollCost	Array de tollCost	0/ unlimited	Toll cost information (filled in if format=tollCost).

Itinerary portion (subRouteV5)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	double	0/1	Distance covered by the route segment in metres.
durationSeconds	double	0/1	Duration of the route segment in seconds.
step or (or steps in JSON / JSON-P)	segmentV5 (or array in JSON / JSON-P)	0/ unlimited	List of component segments in the route segments

Route segment (segmentV5)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	double	0/1	Distance covered by the route segment in metres.
durationSeconds	double	0/1	Duration of the route segment in seconds.
navigationInstruction	string	0/1	Navigation instruction code: - F: Straight on - FR: Turn slightly to the right

parameter	type	min/max	description
			- FL: Turn slightly to the left - R: Turn right - L: Turn left - BR: Turn hard right - BL: Turn hard left - B: U-turn - round_about_entry: Enter roundabout - round_about_exit: Exit from roundabout
name	string	0/1	Segment street name.
point	string	0/ unlimited	List of coordinates separated by the , character

Route position (timeLineItemV5)

parameter	type	min/max	description
durationSeconds	double	0/1	Route duration in seconds up to this position.
message	string	0/1	Error message for this position.
status	string	0/1	Status of this position.
distanceMeters	double	0/1	Distance of the route in metres up to this position.
location	string	0/1	Coordinates of the position.

TollCost (tollCost)

parameter	type	min/max	description
amount	double	0/1	Total cost of tolls.
currency	string	0/1	Currency.
costsByCountry	Array de costsByCountry	0/ unlimited	List of toll costs by country.

Cost of tolls per country (costsByCountry)

parameter	type	min/max	description
amount	double	0/1	Cost of tolls.
country	string	0/1	Country code (3-letter ISO country code).

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/routeService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:routeV5>
         <!--Optional:-->
         <request>
            <origin>
```

```
               <x>-1.351448</x>
               <y>47.446923</y>
            </origin>
            <!--Optional:-->
            <originNode></originNode>
            <destination>
               <x>-1.34529</x>
               <y>47.4479931</y>
            </destination>
            <!--Optional:-->
            <destinationNode></destinationNode>
            <!--Optional:-->
            <waypoints>
               <!--Zero or more repetitions:-->
               <waypoint>
                  <x>-1.34981</x>
                  <y>47.44837</y>
               </waypoint>
            </waypoints>
            <!--Optional:-->
            <waypointNodes>
               <!--Zero or more repetitions:-->
               <waypointNode></waypointNode>
            </waypointNodes>
            <!--Optional:-->
            <srs>epsg:4326</srs>
            <!--Optional:-->
            <method>time</method>
            <!--Optional:-->
            <format>STANDARD</format>
            <!--Optional:-->
            <tolerance>0.</tolerance>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <timeLine>
               <!--Zero or more repetitions:-->
               <timeLineItem></timeLineItem>
            </timeLine>
            <!--Optional:-->
            <snapMethod></snapMethod>
            <!--Optional:-->
            <avoidArea></avoidArea>
            <!--Optional:-->
            <computeOptions></computeOptions>
            <!--Optional:-->
            <timeOut></timeOut>
            <!--Optional:-->
            <configName></configName>
         </request>
      </sch:routeV5>
   </soapenv:Body>
```

```
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:routeV5Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <RouteResult>
            <status>OK</status>
            <distance>640 m</distance>
            <duration>0:02:06</duration>
            <distanceMeters>639.97</distanceMeters>
            <durationSeconds>126.88</durationSeconds>
            <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
            <wktGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, [...] ,  -1.34529
 47.447993)</wktGeometry>
            <wktSimplifiedGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922,  [...] , -1.34529
 47.447993)</wktSimplifiedGeometry>
            <leg>
               <distance>233 m</distance>
               <duration>0:00:46</duration>
               <distanceMeters>233.1</distanceMeters>
               <durationSeconds>46.28</durationSeconds>
               <step>
                  <distance>139 m</distance>
                  <duration>0:00:29</duration>
                  <distanceMeters>138.59</distanceMeters>
                  <durationSeconds>29.28</durationSeconds>
                  <name>RUE LA FONTAINE BRUNEAU</name>
               </step>
               <step>
                  <distance>91 m</distance>
                  <duration>0:00:16</duration>
                  <distanceMeters>91.33</distanceMeters>
                  <durationSeconds>16.43</durationSeconds>
                  <navigationInstruction>FR</navigationInstruction>
                  <name>RUE DES SAULES</name>
               </step>
               <step>
                  <distance>3 m</distance>
                  <duration>0:00:00</duration>
                  <distanceMeters>3.18</distanceMeters>
                  <durationSeconds>0.57</durationSeconds>
                  <navigationInstruction>round_about_entry</navigationInstruction>
                  <name />
               </step>
            </leg>
            <leg>
               <distance>407 m</distance>
               <duration>0:01:20</duration>
               <distanceMeters>406.87</distanceMeters>
               <durationSeconds>80.6</durationSeconds>
               <step>
                  <distance>18 m</distance>
                  <duration>0:00:03</duration>
                  <distanceMeters>17.64</distanceMeters>
                  <durationSeconds>3.17</durationSeconds>
                  <name />
               </step>
               <step>
                  <distance>67 m</distance>
                  <duration>0:00:15</duration>
                  <distanceMeters>66.62</distanceMeters>
```

```
                     <durationSeconds>15.05</durationSeconds>
                     <navigationInstruction>round_about_exit</navigationInstruction>
                     <name>RUE DES FOURS</name>
                  </step>
                  <step>
                     <distance>36 m</distance>
                     <duration>0:00:08</duration>
                     <distanceMeters>35.74</distanceMeters>
                     <durationSeconds>8.57</durationSeconds>
                     <navigationInstruction>F</navigationInstruction>
                     <name>PLACE DE L'ÉGLISE</name>
                  </step>
                  <step>
                     <distance>72 m</distance>
                     <duration>0:00:15</duration>
                     <distanceMeters>72.25</distanceMeters>
                     <durationSeconds>15.24</durationSeconds>
                     <navigationInstruction>F</navigationInstruction>
                     <name>RUE DES PRESSOIRS</name>
                  </step>
                  <step>
                     <distance>26 m</distance>
                     <duration>0:00:04</duration>
                     <distanceMeters>26.02</distanceMeters>
                     <durationSeconds>4.68</durationSeconds>
                     <navigationInstruction>round_about_entry</navigationInstruction>
                     <name />
                  </step>
                  <step>
                     <distance>183 m</distance>
                     <duration>0:00:32</duration>
                     <distanceMeters>182.84</distanceMeters>
                     <durationSeconds>32.91</durationSeconds>
                     <navigationInstruction>round_about_exit</navigationInstruction>
                     <name>RUE DU BOURG DRAPÉ</name>
                  </step>
                  <step>
                     <distance>6 m</distance>
                     <duration>0:00:00</duration>
                     <distanceMeters>5.76</distanceMeters>
                     <durationSeconds>0.98</durationSeconds>
                     <navigationInstruction>FR</navigationInstruction>
                     <name>RUE DE LA CURE</name>
                  </step>
               </leg>
               <srs>epsg:4326</srs>
               <carbonFootprint>0.0</carbonFootprint>
            </RouteResult>
         </ns2:routeV5Response>
      </soap:Body>
   </soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/route/v5.json?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/route/v5.json?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/route/v5.xml?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837
```

Response

The response is always in UTF-8 format.

JSON format

```
{
    "message": null,
    "status": "OK",
    "distance": "640 m",
    "duration": "0:02:09",
    "distanceMeters": 639.95,
    "durationSeconds": 129.48,
    "bounds": "-1.351448,47.446922;-1.345263,47.44848",
    "wktGeometry": "LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])",
    "wktSimplifiedGeometry":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
 [...])",
    "legs": [
        {
            "distance": "233 m",
            "duration": "0:00:46",
            "distanceMeters": 233.09,
            "durationSeconds": 46.84,
            "steps": [
                {
                    "distance": "139 m",
                    "duration": "0:00:29",
                    "distanceMeters": 138.59,
                    "durationSeconds": 29.83,
                    "navigationInstruction": null,
                    "name": "RUE LA FONTAINE BRUNEAU",
                    "points": []
                },
                {
                    [...]
                }
            ]
        },
        {
            "distance": "407 m",
            "duration": "0:01:22",
            "distanceMeters": 406.86,
            "durationSeconds": 82.64,
            "steps": [
                {
                    "distance": "18 m",
                    "duration": "0:00:03",
                    "distanceMeters": 17.63,
                    "durationSeconds": 3.18,
                    "navigationInstruction": null,
                    "name": "",
                    "points": []
                },
                {
                   [...]
                }
            ]
        }
    ],
    "startDateTime": null,
    "finishDateTime": null,
    "srs": null,
    "originNode": null,
    "waypointNodes": null,
    "destinationNode": null,
    "carbonFootprint": 0.195038864105688
}
```

JSON-P format

```
myCallback(
    {
        "message": null,
        "status": "OK",
        "distance": "640 m",
        "duration": "0:02:09",
        "distanceMeters": 639.95,
        "durationSeconds": 129.48,
        "bounds": "-1.351448,47.446922;-1.345263,47.44848",
        "wktGeometry": "LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])",
        "wktSimplifiedGeometry":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
 [...])",
        "legs": [
            {
                "distance": "233 m",
                "duration": "0:00:46",
                "distanceMeters": 233.09,
                "durationSeconds": 46.84,
                "steps": [
                    {
                        "distance": "139 m",
                        "duration": "0:00:29",
                        "distanceMeters": 138.59,
                        "durationSeconds": 29.83,
                        "navigationInstruction": null,
                        "name": "RUE LA FONTAINE BRUNEAU",
                        "points": []
                    },
                    {
                        [...]
                    }
                ]
            },
            {
                "distance": "407 m",
                "duration": "0:01:22",
                "distanceMeters": 406.86,
                "durationSeconds": 82.64,
                "steps": [
                    {
                        "distance": "18 m",
                        "duration": "0:00:03",
                        "distanceMeters": 17.63,
                        "durationSeconds": 3.18,
                        "navigationInstruction": null,
                        "name": "",
                        "points": []
                    },
                    {
                       [...]
                    }
                ]
            }
        ],
        "startDateTime": null,
        "finishDateTime": null,
        "srs": null,
        "originNode": null,
        "waypointNodes": null,
        "destinationNode": null,
        "carbonFootprint": 0.195038864105688
    }
);
```

XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<routeResultV5>
        <status>OK</status>
        <distance>640 m</distance>
        <duration>0:02:09</duration>
        <distanceMeters>639.95</distanceMeters>
        <durationSeconds>129.48</durationSeconds>
        <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
        <wktGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
wktGeometry>
        <wktSimplifiedGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
 [...])</wktSimplifiedGeometry>
        <leg>
                <distance>233 m</distance>
                <duration>0:00:46</duration>
                <distanceMeters>233.09</distanceMeters>
                <durationSeconds>46.84</durationSeconds>
                <step>
                        <distance>139 m</distance>
                        <duration>0:00:29</duration>
                        <distanceMeters>138.59</distanceMeters>
                        <durationSeconds>29.83</durationSeconds>
                        <name>RUE LA FONTAINE BRUNEAU</name>
                </step>
                <step>
            [...]
                </step>
        </leg>
        <leg>
                <distance>407 m</distance>
                <duration>0:01:22</duration>
                <distanceMeters>406.86</distanceMeters>
                <durationSeconds>82.64</durationSeconds>
                <step>
                        <distance>18 m</distance>
                        <duration>0:00:03</duration>
                        <distanceMeters>17.63</distanceMeters>
                        <durationSeconds>3.18</durationSeconds>
                        <name />
                </step>
                <step>
```

```
                        [...]
                </step>
        </leg>
        <carbonFootprint>0.195038864105688</carbonFootprint>
</routeResultV5>
```

JavaScript API

Include the JavaScript library

```
var routeCtrl = new GCUI.Control.Route();
routeCtrl.route({
    url:'http://<server>/<webapp>/api/lbs/route/v5.json',
    tolerance : 100,
    origin : new OpenLayers.LonLat(0.691012, 47.384813),
    destination : new OpenLayers.LonLat(0.691012, 47.384813),
    waypoints : [ new OpenLayers.LonLat(2.344408, 49.898798) ],
    callback : function(result, options) {
        console.log(result);
    }
});
```

The `result` variable is in JSON format as described above. The `callback` function passed to parameter is called at the end of the route calculation.

Possible responses

Case of a route found (routeResultV5/status is OK)

```
<?xml version="1.0" encoding="UTF-8"?>
<routeResultV5>
        <status>OK</status>
        <distance>640 m</distance>
        <duration>0:02:09</duration>
        <distanceMeters>639.95</distanceMeters>
        <durationSeconds>129.48</durationSeconds>
        <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
        <wktGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
wktGeometry>
        <wktSimplifiedGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
 [...])</wktSimplifiedGeometry>
        <leg>
                <distance>233 m</distance>
                <duration>0:00:46</duration>
                <distanceMeters>233.09</distanceMeters>
                <durationSeconds>46.84</durationSeconds>
                <step>
                        <distance>139 m</distance>
                        <duration>0:00:29</duration>
                        <distanceMeters>138.59</distanceMeters>
                        <durationSeconds>29.83</durationSeconds>
                        <name>RUE LA FONTAINE BRUNEAU</name>
                </step>
                <step>
            [...]
                </step>
        </leg>
        <leg>
                <distance>407 m</distance>
                <duration>0:01:22</duration>
```

```
                  <distanceMeters>406.86</distanceMeters>
                  <durationSeconds>82.64</durationSeconds>
                  <step>
                          <distance>18 m</distance>
                          <duration>0:00:03</duration>
                          <distanceMeters>17.63</distanceMeters>
                          <durationSeconds>3.18</durationSeconds>
                          <name />
                  </step>
                  <step>
                          [...]
                  </step>
          </leg>
          <carbonFootprint>0.195038864105688</carbonFootprint>
  </routeResultV5>
```

Case of a forgotten specification of a start or finish point (routeResultV5/status is ERROR)

```
<routeResultV5>
   <message>Origin and destination must be not null</message>
   <status>ERROR</status>
   <distanceMeters>0.0</distanceMeters>
   <durationSeconds>0.0</durationSeconds>
</routeResultV5>
```

Case of a faulty format assigned to the start or finish point or to route stops (routeResultV5/status is ERROR)

```
<routeResultV5>
   <message>Origin, destination and waypoints should be represented by a couple of coordinates</message>
   <status>ERROR</status>
   <distanceMeters>0.0</distanceMeters>
   <durationSeconds>0.0</durationSeconds>
</routeResultV5>
```

Case of a faulty type (serviceResult/status is ERROR)

```
<serviceResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
</serviceResult>
```

Case of a snap-to-graph error (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in route computation
Error in smartrouting
Failed to execute calculateRoute
com.geoconcept.smartrouting.SmartRoutingNativeException: failed to connect waypoint { 146.691012, 47.384813,
 0.000000 }
failed to connect waypoint { 146.691012, 47.384813, 0.000000 }</message>
   <status>ERROR</status>
</serviceResult>
```

Case of a problem with the graph: absent file, faulty filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
    <message>ServiceException: Error in route computation
Error in smartrouting
```

```
datasource is null</message>
    <status>ERROR</status>
</serviceResult>
```

V4

Parameters / properties

Input

parameter	description	optional	default
origin	Start point coordinates. The longitude and latitude coordinates are separated by , characters.	no	
originNode	Start node. Care: a physical node does not have the same ID in another graph.	yes *	
destination	Finish point coordinates. The longitude and latitude coordinates are separated by the , character	no	
destinationNode	Arrival node. Care: a physical node does not have the same ID in another graph.	yes *	
waypoints	Coordinates for route stops. Each pair of coordinates for a stop is framed by the <waypoint> tag The longitude and latitude coordinates are separated by the , character	yes	
waypointNodes	Step nodes. Each pair of coordinates for a step is framed by the <waypointNode> tag Care: a physical node does not have the same ID in another graph.	yes *	
method	the shortest (distance) and fastest (time) route	yes	time
format	- standard: result = route summary / waypoints / geometry in wkt format / simplified geometry in wkt format / list of concatenated segments (without geometries) - extended: result = route summary / waypoints / simplified geometry in wkt format / list of non-concatenated segments (with geometries) - summary: result = route summary - geometry: result = route summary / waypoints / geometry in wkt format - simplifiedgeometry: result = route summary / waypoints / simplified geometry in wkt format - geometries: result = route summary / waypoints / geometry in wkt format / simplified geometry in wkt format - brief: result = route summary / List of segments with duration and distance - standardext: result = route summary / waypoints / list of concatenated segments with duration and distance and geometry - node: result = route summary + Id of snapped nodes	yes	standard
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
tolerance	Tolerance distance (in metres) for the geometry simplification.	yes	
graphName	Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
startDateTime	Start date and time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a start date of 21 January 2014, at 9.00am in Paris. Caution: the + character can be misinterpreted by browsers, so in this case, it should be replaced by *%2B*.	yes	

parameter	description	optional	default
profileId	Vehicle identifier (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
profileName	Vehicle profile (saved in the vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
exclusions	List of restriction rules to use, separated by the , or *;* character (Example: Toll, Tunnel, Bridge)	yes	
timeLine	List of intermediate durations to calculate positions along the route trajectory. Expressed in seconds, separated by the *;* character	yes	
snapMethod	Snap to graph method - standard: to the nearest connectable road segment - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without any restriction rules - nodes: Snap directly to nodes supplied by the originNode originNode, destinationNode and waypointNodes parameters	yes	standard
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84: `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))` Care: WKT geometries must be closed.	yes	
configName	Name of the configuration to use (defined in Geoconcept Web - Administration / Tools / Road graphs) This replaces the use of graphName, profileId and profileName	yes	

Output

Route (routeResultV4)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	double	0/1	Total route distance in metres.
durationSeconds	double	0/1	Total duration of the route in seconds.
waypoints	string	0/1	Waypoints (BoundingBox) for the route geometry.
wktGeometry	string	0/1	Geometry of the route in WKT format
wktSimplified	string	0/1	Simplified geometry in WKT format.
leg (or legs in JSON / JSON-P)	subRouteV4 (or array in JSON / JSON-P)	0/ unlimited	List of route segments.
startDateTime	string	0/1	Start date and time (format: ISO8601 without any area code: local time) Example: 2014-01-21T09:00:00.000+01:00 for a departure on 21 January 2014, at 9.00am in Paris

parameter	type	min/max	description
finishDateTime	string	0/1	Arrival date and time (format: ISO8601 without any area code: local time) Example: 2014-01-21T09:00:00.000+01:00 for an arrival time on 21 January 2014, at 9.00am in Paris
timeLineItem	timeLineItemV4 (or an array in JSON / JSON-P)	0/ unlimited	List of calculated positions.
srs	string	0/1	projection passed as input (EPSG code such as epsg:4326 or wgs84)
originNode	string	0/1	Start node identifier (entered if format=NODE).
destinationNode	string	0/1	Arrival node identifier (entered if format=NODE).
waypointNodes	waypointNodes array (string)	0/ unlimited	Step nodes identifiers (entered if format=NODE).
carbonFootprint	double	0/1	Carbon footprint (CO2 emissions in kilograms)

Itinerary portion (subRouteV4)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	double	0/1	Distance covered by the route segment in metres.
durationSeconds	double	0/1	Duration of the route segment in seconds.
step or (or steps in JSON / JSON-P)	segmentV4 (or array in JSON / JSON-P)	0/ unlimited	List of component segments in the route segments

Route segment (segmentV4)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	double	0/1	Distance covered by the route segment in metres.
durationSeconds	double	0/1	Duration of the route segment in seconds.
navigationInstruction	string	0/1	Navigation instruction code: - F: Straight on - FR: Turn slightly to the right - FL: Turn slightly to the left - R: Turn right - L: Turn left - BR: Turn hard right - BL: Turn hard left - B: U-turn - round_about_entry: Enter roundabout

parameter	type	min/max	description
			- round_about_exit: Exit from roundabout
name	string	0/1	Segment street name.
point	string	0/ unlimited	List of coordinates separated by the , character

Route position (timeLineItemV4)

parameter	type	min/max	description
durationSeconds	double	0/1	Route duration in seconds up to this position.
message	string	0/1	Error message for this position.
status	string	0/1	Status of this position.
distanceMeters	double	0/1	Distance of the route in metres up to this position.
location	string	0/1	Coordinates of the position.

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/routeService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:routeV4>
         <!--Optional:-->
         <request>
            <origin>
               <x>-1.351448</x>
               <y>47.446923</y>
            </origin>
            <!--Optional:-->
            <originNode></originNode>
            <destination>
               <x>-1.34529</x>
               <y>47.4479931</y>
            </destination>
            <!--Optional:-->
            <destinationNode></destinationNode>
            <!--Optional:-->
            <waypoints>
               <!--Zero or more repetitions:-->
               <waypoint>
                  <x>-1.34981</x>
                  <y>47.44837</y>
               </waypoint>
            </waypoints>
            <!--Optional:-->
            <waypointNodes>
               <!--Zero or more repetitions:-->
               <waypointNode></waypointNode>
            </waypointNodes>
            <!--Optional:-->
            <srs>epsg:4326</srs>
            <!--Optional:-->
            <method>time</method>
            <!--Optional:-->
            <format>STANDARD</format>
            <!--Optional:-->
            <tolerance>0.</tolerance>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <timeLine>
               <!--Zero or more repetitions:-->
               <timeLineItem></timeLineItem>
            </timeLine>
            <!--Optional:-->
            <snapMethod></snapMethod>
            <!--Optional:-->
            <avoidArea></avoidArea>
            <!--Optional:-->
            <configName></configName>
         </request>
      </sch:routeV4>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:routeV4Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <RouteResult>
            <status>OK</status>
            <distance>640 m</distance>
            <duration>0:02:06</duration>
            <distanceMeters>639.95</distanceMeters>
            <durationSeconds>126.88</durationSeconds>
            <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
            <wktGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922,  [...] )</wktGeometry>
            <wktSimplifiedGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922,  [...] )</
wktSimplifiedGeometry>
            <leg>
               <distance>233 m</distance>
               <duration>0:00:46</duration>
               <distanceMeters>233.09</distanceMeters>
               <durationSeconds>46.28</durationSeconds>
               <step>
                  <distance>139 m</distance>
                  <duration>0:00:29</duration>
                  <distanceMeters>138.59</distanceMeters>
                  <durationSeconds>29.28</durationSeconds>
                  <name>RUE LA FONTAINE BRUNEAU</name>
               </step>
               <step>
            <distance>91 m</distance>
            <duration>0:00:16</duration>
            <distanceMeters>91.33</distanceMeters>
            <durationSeconds>16.43</durationSeconds>
            <navigationInstruction>FR</navigationInstruction>
            <name>RUE DES SAULES</name>
         </step>
         <step>
            <distance>3 m</distance>
            <duration>0:00:00</duration>
            <distanceMeters>3.17</distanceMeters>
            <durationSeconds>0.57</durationSeconds>
            <navigationInstruction>round_about_entry</navigationInstruction>
            <name/>
         </step>
      </leg>
      <leg>
         <distance>407 m</distance>
         <duration>0:01:20</duration>
         <distanceMeters>406.86</distanceMeters>
         <durationSeconds>80.6</durationSeconds>
         <step>
            <distance>18 m</distance>
            <duration>0:00:03</duration>
            <distanceMeters>17.63</distanceMeters>
            <durationSeconds>3.17</durationSeconds>
            <name/>
         </step>
         <step>
            <distance>67 m</distance>
            <duration>0:00:15</duration>
            <distanceMeters>66.62</distanceMeters>
            <durationSeconds>15.05</durationSeconds>
            <navigationInstruction>round_about_exit</navigationInstruction>
            <name>RUE DES FOURS</name>
         </step>
         <step>
            <distance>36 m</distance>
            <duration>0:00:08</duration>
            <distanceMeters>35.74</distanceMeters>
            <durationSeconds>8.57</durationSeconds>
            <navigationInstruction>F</navigationInstruction>
            <name>PLACE DE L'ÉGLISE</name>
         </step>
         <step>
            <distance>72 m</distance>
            <duration>0:00:15</duration>
            <distanceMeters>72.25</distanceMeters>
            <durationSeconds>15.24</durationSeconds>
            <navigationInstruction>F</navigationInstruction>
            <name>RUE DES PRESSOIRS</name>
         </step>
         <step>
            <distance>26 m</distance>
            <duration>0:00:04</duration>
            <distanceMeters>26.02</distanceMeters>
            <durationSeconds>4.68</durationSeconds>
            <navigationInstruction>round_about_entry</navigationInstruction>
            <name/>
         </step>
         <step>
            <distance>183 m</distance>
            <duration>0:00:32</duration>
```

```
                  <distanceMeters>182.84</distanceMeters>
                  <durationSeconds>32.91</durationSeconds>
                  <navigationInstruction>round_about_exit</navigationInstruction>
                  <name>RUE DU BOURG DRAPÉ</name>
               </step>
               <step>
                  <distance>6 m</distance>
                  <duration>0:00:00</duration>
                  <distanceMeters>5.76</distanceMeters>
                  <durationSeconds>0.98</durationSeconds>
                  <navigationInstruction>FR</navigationInstruction>
                  <name>RUE DE LA CURE</name>
               </step>
            </leg>
            <timeLineItem>
               <durationSeconds>0.0</durationSeconds>
               <status>OK</status>
               <distanceMeters>0.0</distanceMeters>
               <location>-1.351448,47.446923</location>
            </timeLineItem>
            <srs>epsg:4326</srs>
            <carbonFootprint>0.0</carbonFootprint>
         </RouteResult>
      </ns2:routeV4Response>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/route/v4.json?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/route/v4.json?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/route/v4.xml?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837
```

Response

The response is always in UTF-8 format.

JSON format

```
{
    "message": null,
    "status": "OK",
    "distance": "640 m",
    "duration": "0:02:09",
    "distanceMeters": 639.95,
    "durationSeconds": 129.48,
    "bounds": "-1.351448,47.446922;-1.345263,47.44848",
    "wktGeometry": "LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])",
    "wktSimplifiedGeometry":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
 [...])",
    "legs": [
        {
            "distance": "233 m",
            "duration": "0:00:46",
            "distanceMeters": 233.09,
            "durationSeconds": 46.84,
            "steps": [
                {
                    "distance": "139 m",
                    "duration": "0:00:29",
                    "distanceMeters": 138.59,
                    "durationSeconds": 29.83,
                    "navigationInstruction": null,
                    "name": "RUE LA FONTAINE BRUNEAU",
                    "points": []
                },
                {
                    [...]
                }
            ]
        },
        {
            "distance": "407 m",
            "duration": "0:01:22",
            "distanceMeters": 406.86,
            "durationSeconds": 82.64,
            "steps": [
                {
                    "distance": "18 m",
                    "duration": "0:00:03",
                    "distanceMeters": 17.63,
                    "durationSeconds": 3.18,
                    "navigationInstruction": null,
                    "name": "",
                    "points": []
                },
                {
                   [...]
                }
            ]
        }
    ],
    "startDateTime": null,
    "finishDateTime": null,
    "srs": null,
    "originNode": null,
    "waypointNodes": null,
    "destinationNode": null,
    "carbonFootprint": 0.195038864105688
}
```

JSON-P format

```
myCallback(
    {
        "message": null,
        "status": "OK",
        "distance": "640 m",
        "duration": "0:02:09",
        "distanceMeters": 639.95,
        "durationSeconds": 129.48,
        "bounds": "-1.351448,47.446922;-1.345263,47.44848",
        "wktGeometry": "LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])",
        "wktSimplifiedGeometry":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
  [...])",
        "legs": [
            {
                "distance": "233 m",
                "duration": "0:00:46",
                "distanceMeters": 233.09,
                "durationSeconds": 46.84,
                "steps": [
                    {
                        "distance": "139 m",
                        "duration": "0:00:29",
                        "distanceMeters": 138.59,
                        "durationSeconds": 29.83,
                        "navigationInstruction": null,
                        "name": "RUE LA FONTAINE BRUNEAU",
                        "points": []
                    },
                    {
                        [...]
                    }
                ]
            },
            {
                "distance": "407 m",
                "duration": "0:01:22",
                "distanceMeters": 406.86,
                "durationSeconds": 82.64,
                "steps": [
                    {
                        "distance": "18 m",
                        "duration": "0:00:03",
                        "distanceMeters": 17.63,
                        "durationSeconds": 3.18,
                        "navigationInstruction": null,
                        "name": "",
                        "points": []
                    },
                    {
                       [...]
                    }
                ]
            }
        ],
        "startDateTime": null,
        "finishDateTime": null,
        "srs": null,
        "originNode": null,
        "waypointNodes": null,
        "destinationNode": null,
        "carbonFootprint": 0.195038864105688
    }
);
```

XML format

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<routeResultV4>
        <status>OK</status>
        <distance>640 m</distance>
        <duration>0:02:09</duration>
        <distanceMeters>639.95</distanceMeters>
        <durationSeconds>129.48</durationSeconds>
        <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
        <wktGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
wktGeometry>
        <wktSimplifiedGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
 [...])</wktSimplifiedGeometry>
        <leg>
                <distance>233 m</distance>
                <duration>0:00:46</duration>
                <distanceMeters>233.09</distanceMeters>
                <durationSeconds>46.84</durationSeconds>
                <step>
                        <distance>139 m</distance>
                        <duration>0:00:29</duration>
                        <distanceMeters>138.59</distanceMeters>
                        <durationSeconds>29.83</durationSeconds>
                        <name>RUE LA FONTAINE BRUNEAU</name>
                </step>
                <step>
            [...]
                </step>
        </leg>
        <leg>
                <distance>407 m</distance>
                <duration>0:01:22</duration>
                <distanceMeters>406.86</distanceMeters>
                <durationSeconds>82.64</durationSeconds>
                <step>
                        <distance>18 m</distance>
                        <duration>0:00:03</duration>
                        <distanceMeters>17.63</distanceMeters>
                        <durationSeconds>3.18</durationSeconds>
                        <name />
                </step>
                <step>
                        [...]
                </step>
        </leg>
        <carbonFootprint>0.195038864105688</carbonFootprint>
</routeResultV4>
```

JavaScript API

Include the JavaScript library

```
var routeCtrl = new GCUI.Control.Route();
routeCtrl.route({
    url:'http://<server>/<webapp>/api/lbs/route/v4.json',
    tolerance : 100,
    origin : new OpenLayers.LonLat(0.691012, 47.384813),
    destination : new OpenLayers.LonLat(0.691012, 47.384813),
    waypoints : [ new OpenLayers.LonLat(2.344408, 49.898798) ],
    callback : function(result, options) {
        console.log(result);
    }
});
```

The `result` variable is in JSON format as described above. The `callback` function passed to parameter is called at the end of the route calculation.

Possible responses

Case of a route found (routeResultV4/status is OK)

```
<?xml version="1.0" encoding="UTF-8"?>
<routeResultV4>
        <status>OK</status>
        <distance>640 m</distance>
        <duration>0:02:09</duration>
        <distanceMeters>639.95</distanceMeters>
        <durationSeconds>129.48</durationSeconds>
        <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
        <wktGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
wktGeometry>
        <wktSimplifiedGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
 [...])</wktSimplifiedGeometry>
        <leg>
                <distance>233 m</distance>
                <duration>0:00:46</duration>
                <distanceMeters>233.09</distanceMeters>
                <durationSeconds>46.84</durationSeconds>
                <step>
                        <distance>139 m</distance>
                        <duration>0:00:29</duration>
                        <distanceMeters>138.59</distanceMeters>
                        <durationSeconds>29.83</durationSeconds>
                        <name>RUE LA FONTAINE BRUNEAU</name>
                </step>
                <step>
            [...]
                </step>
        </leg>
        <leg>
                <distance>407 m</distance>
                <duration>0:01:22</duration>
                <distanceMeters>406.86</distanceMeters>
                <durationSeconds>82.64</durationSeconds>
                <step>
                        <distance>18 m</distance>
                        <duration>0:00:03</duration>
                        <distanceMeters>17.63</distanceMeters>
                        <durationSeconds>3.18</durationSeconds>
                        <name />
                </step>
                <step>
                        [...]
                </step>
        </leg>
        <carbonFootprint>0.195038864105688</carbonFootprint>
</routeResultV4>
```

Case of a forgotten specification of a start or finish point (routeResultV4/status is ERROR)

```
<routeResultV4>
   <message>Origin and destination must be not null</message>
   <status>ERROR</status>
   <distanceMeters>0.0</distanceMeters>
   <durationSeconds>0.0</durationSeconds>
```

```
</routeResultV4>
```

Case of a faulty format assigned to the start or finish point or to route stops (routeResultV4/status is ERROR)

```
<routeResultV4>
   <message>Origin, destination and waypoints should be represented by a couple of coordinates</message>
   <status>ERROR</status>
   <distanceMeters>0.0</distanceMeters>
   <durationSeconds>0.0</durationSeconds>
</routeResultV4>
```

Case of a faulty type (serviceResult/status is ERROR)

```
<serviceResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
</serviceResult>
```

Case of a snap-to-graph error (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in route computation
Error in smartrouting
Failed to execute calculateRoute
com.geoconcept.smartrouting.SmartRoutingNativeException: failed to connect waypoint { 146.691012, 47.384813,
 0.000000 }
failed to connect waypoint { 146.691012, 47.384813, 0.000000 }</message>
   <status>ERROR</status>
</serviceResult>
```

Case of a problem with the graph: absent file, faulty filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
    <message>ServiceException: Error in route computation
Error in smartrouting
datasource is null</message>
    <status>ERROR</status>
</serviceResult>
```

V3

Parameters / properties

Input

parameter	description	optional	default
origin	Start point coordinates. The longitude and latitude coordinates are separated by , characters.	no	
destination	Finish point coordinates. The longitude and latitude coordinates are separated by the , character	no	
waypoints	Coordinates for route stops. Each pair of coordinates for a stop is framed by the <waypoint> tag The longitude and latitude coordinates are separated by the , character	yes	
method	the shortest (distance) and fastest (time) route	yes	time

parameter	description	optional	default
format	- standard: result = route summary / waypoints / geometry in wkt format / simplified geometry in wkt format / list of concatenated segments (without geometries) - extended: result = route summary / waypoints / simplified geometry in wkt format / list of non-concatenated segments (with geometries) - summary: result = route summary - geometry: result = route summary / waypoints / geometry in wkt format - simplifiedgeometry: result = route summary / waypoints / simplified geometry in wkt format - geometries: result = route summary / waypoints / geometry in wkt format / simplified geometry in wkt format - brief: result = route summary / list of segments with duration and distance - standardext: result = route summary / waypoints / list of concatenated segments with duration distance and geometry.	yes	standard
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
tolerance	Tolerance distance (in metres) for the geometry simplification.	yes	
graphName	Name of the graph to use	yes	
startDateTime	Start date and time (format ISO8601: local time) example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a start on 21 January 2014, at 9.00am in Paris	yes	
profileId	Vehicle identifier (saved under vehicle profiles) to use	yes	
profileName	Vehicle profile (saved under vehicle profiles) to use	yes	
exclusions	List of restriction rules to use, separated by the , or ; character (Example: Toll, Tunnel, Bridge)	yes	
timeLine	List of intermediate durations to calculate positions along the route trajectory. Expressed in seconds, separated by the ; character	yes	
snapMethod	Snap to graph method - standard: to the nearest connectable road segment - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without any restriction rules	yes	standard

Output

Route (routeResultV3)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	double	0/1	Total route distance in metres.
durationSeconds	double	0/1	Total duration of the route in seconds.
waypoints	string	0/1	Waypoints (BoundingBox) for the route geometry.
wktGeometry	string	0/1	Geometry of the route in WKT format
wktSimplified	string	0/1	Simplified geometry in WKT format.

parameter	type	min/max	description
leg (or legs in JSON / JSON-P)	subRouteV3 (or array in JSON / JSON-P)	0/ unlimited	List of route segments.
startDateTime	string	0/1	Start date and time (format: ISO8601 without any area code: local time) Example: 2014-01-21T09:00:00.000+01:00 for a departure on 21 January 2014, at 9.00am in Paris
finishDateTime	string	0/1	Arrival date and time (format: ISO8601 without any area code: local time) Example: 2014-01-21T09:00:00.000+01:00 for an arrival time on 21 January 2014, at 9.00am in Paris
timeLineItem	timeLineItemV3 (or an array in JSON / JSON-P)	0/ unlimited	List of calculated positions.
srs	string	0/1	projection passed as input (EPSG code such as epsg:4326 or wgs84)

Route subsection (SubRouteV3)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	double	0/1	Distance covered by the route segment in metres.
durationSeconds	double	0/1	Duration of the route segment in seconds.
step or (or steps in JSON / JSON-P)	segmentV3 (or array in JSON / JSON-P)	0/ unlimited	List of component segments in the route segments

Route segment (segmentV3)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	double	0/1	Distance covered by the route segment in metres.
durationSeconds	double	0/1	Duration of the route segment in seconds.
navigationInstruction	string	0/1	Navigation instruction code: - F: Straight on - FR: Turn slightly to the right - FL: Turn slightly to the left - R: Turn right - L: Turn left - BR: Turn hard right - BL: Turn hard left - B: U-turn - round_about_entry: Enter roundabout

parameter	type	min/max	description
			- round_about_exit: Exit from roundabout
name	string	0/1	Segment street name.
point	string	0/ unlimited	List of coordinates separated by the , character

Route position (timeLineItemV3)

parameter	type	min/max	description
durationSeconds	double	0/1	Route duration in seconds up to this position.
message	string	0/1	Error message for this position.
status	string	0/1	Status of this position.
distanceMeters	double	0/1	Distance of the route in metres up to this position.
location	string	0/1	Coordinates of the position.

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/routeService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:routeV3>
         <!--Optional:-->
         <request>
            <origin>
               <x>-1.351448</x>
               <y>47.446923</y>
            </origin>
            <destination>
               <x>-1.34529</x>
               <y>47.4479931</y>
            </destination>
            <!--Optional:-->
            <waypoints>
               <!--Zero or more repetitions:-->
               <waypoint>
                  <x>-1.34981</x>
                  <y>47.44837</y>
               </waypoint>
            </waypoints>
            <!--Optional:-->
            <srs>epsg:4326</srs>
            <!--Optional:-->
            <method>time</method>
            <!--Optional:-->
            <format>STANDARD</format>
            <!--Optional:-->
            <tolerance>0.</tolerance>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <startDateTime>2015-07-29T09:00:00.000+01:00</startDateTime>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <timeLine>
               <!--Zero or more repetitions:-->
               <timeLineItem>5</timeLineItem>
               <timeLineItem>15</timeLineItem>
            </timeLine>
            <!--Optional:-->
            <snapMethod></snapMethod>
         </request>
      </sch:routeV3>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:routeV3Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <RouteResult>
            <status>OK</status>
            <distance>640 m</distance>
            <duration>0:02:06</duration>
            <distanceMeters>639.95</distanceMeters>
            <durationSeconds>126.88</durationSeconds>
            <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
            <wktGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922,  [...] )</wktGeometry>
            <wktSimplifiedGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922,  [...] )</
wktSimplifiedGeometry>
            <leg>
               <distance>233 m</distance>
               <duration>0:00:46</duration>
               <distanceMeters>233.09</distanceMeters>
               <durationSeconds>46.28</durationSeconds>
               <step>
                  <distance>139 m</distance>
                  <duration>0:00:29</duration>
                  <distanceMeters>138.59</distanceMeters>
                  <durationSeconds>29.28</durationSeconds>
                  <name>RUE LA FONTAINE BRUNEAU</name>
               </step>
               <step>
                  <distance>91 m</distance>
                  <duration>0:00:16</duration>
                  <distanceMeters>91.33</distanceMeters>
                  <durationSeconds>16.43</durationSeconds>
                  <navigationInstruction>FR</navigationInstruction>
                  <name>RUE DES SAULES</name>
               </step>
               <step>
                  <distance>3 m</distance>
                  <duration>0:00:00</duration>
                  <distanceMeters>3.17</distanceMeters>
                  <durationSeconds>0.57</durationSeconds>
```

```
                <navigationInstruction>round_about_entry</navigationInstruction>
                <name/>
            </step>
        </leg>
        <leg>
            <distance>407 m</distance>
            <duration>0:01:20</duration>
            <distanceMeters>406.86</distanceMeters>
            <durationSeconds>80.6</durationSeconds>
            <step>
                <distance>18 m</distance>
                <duration>0:00:03</duration>
                <distanceMeters>17.63</distanceMeters>
                <durationSeconds>3.17</durationSeconds>
                <name/>
            </step>
            <step>
                <distance>67 m</distance>
                <duration>0:00:15</duration>
                <distanceMeters>66.62</distanceMeters>
                <durationSeconds>15.05</durationSeconds>
                <navigationInstruction>round_about_exit</navigationInstruction>
                <name>RUE DES FOURS</name>
            </step>
            <step>
                <distance>36 m</distance>
                <duration>0:00:08</duration>
                <distanceMeters>35.74</distanceMeters>
                <durationSeconds>8.57</durationSeconds>
                <navigationInstruction>F</navigationInstruction>
                <name>PLACE DE L'ÉGLISE</name>
            </step>
            <step>
                <distance>72 m</distance>
                <duration>0:00:15</duration>
                <distanceMeters>72.25</distanceMeters>
                <durationSeconds>15.24</durationSeconds>
                <navigationInstruction>F</navigationInstruction>
                <name>RUE DES PRESSOIRS</name>
            </step>
            <step>
                <distance>26 m</distance>
                <duration>0:00:04</duration>
                <distanceMeters>26.02</distanceMeters>
                <durationSeconds>4.68</durationSeconds>
                <navigationInstruction>round_about_entry</navigationInstruction>
                <name/>
            </step>
            <step>
                <distance>183 m</distance>
                <duration>0:00:32</duration>
                <distanceMeters>182.84</distanceMeters>
                <durationSeconds>32.91</durationSeconds>
                <navigationInstruction>round_about_exit</navigationInstruction>
                <name>RUE DU BOURG DRAPÉ</name>
            </step>
            <step>
                <distance>6 m</distance>
                <duration>0:00:00</duration>
                <distanceMeters>5.76</distanceMeters>
                <durationSeconds>0.98</durationSeconds>
                <navigationInstruction>FR</navigationInstruction>
                <name>RUE DE LA CURE</name>
```

```
               </step>
            </leg>
            <startDateTime>2015-07-29T10:00:00.000+02:00</startDateTime>
            <finishDateTime>2015-07-29T10:02:06.880+02:00</finishDateTime>
            <timeLineItem>
               <durationSeconds>5.0</durationSeconds>
               <status>OK</status>
               <distanceMeters>23.66</distanceMeters>
               <location>-1.35137,47.447123</location>
            </timeLineItem>
            <timeLineItem>
               <durationSeconds>15.0</durationSeconds>
               <status>OK</status>
               <distanceMeters>71.0</distanceMeters>
               <location>-1.351204,47.447533</location>
            </timeLineItem>
            <srs>epsg:4326</srs>
         </RouteResult>
      </ns2:routeV3Response>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/route/v3.json?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/route/v3.json?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/route/v3.xml?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message":null,
   "status":"OK",
   "distance":"640 m",
   "duration":"0:01:25",
   "distanceMeters":640.0,
   "durationSeconds":85.64,
   "bounds":"-1.351448,47.446922;-1.345263,47.44848",
   "wktGeometry":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])",
   "wktSimplifiedGeometry":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
 [...])",
   "legs":[
      {
         "distance":"233 m",
         "duration":"0:00:36",
         "distanceMeters":233.1,
         "durationSeconds":36.33,
         "steps":[
            {
               "distance":"139 m",
               "duration":"0:00:25",
               "distanceMeters":138.59,
               "durationSeconds":25.0,
               "navigationInstruction":null,
               "name":"RUE LA FONTAINE BRUNEAU",
               "points":[

               ]
            },
            {
                [...]
            }
         ]
      },
      {
         "distance":"407 m",
         "duration":"0:00:49",
         "distanceMeters":406.90000000000003,
         "durationSeconds":49.31,
         "steps":[
            {
               "distance":"18 m",
               "duration":"0:00:02",
               "distanceMeters":17.62,
               "durationSeconds":2.11,
               "navigationInstruction":null,
               "name":"",
               "points":[

               ]
            },
            {
               [...]
            }
         ]
      }
   ],
   "startDateTime":null,
   "finishDateTime":null,
   "srs":"epsg:4326"
}
```

JSON-P format

```
myCallback(
    {
       "message":null,
       "status":"OK",
       "distance":"640 m",
       "duration":"0:01:25",
       "distanceMeters":640.0,
       "durationSeconds":85.64,
       "bounds":"-1.351448,47.446922;-1.345263,47.44848",
       "wktGeometry":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])",
       "wktSimplifiedGeometry":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699,
 [...])",
       "legs":[
          {
             "distance":"233 m",
             "duration":"0:00:36",
             "distanceMeters":233.1,
             "durationSeconds":36.33,
             "steps":[
                {
                   "distance":"139 m",
                   "duration":"0:00:25",
                   "distanceMeters":138.59,
                   "durationSeconds":25.0,
                   "navigationInstruction":null,
                   "name":"RUE LA FONTAINE BRUNEAU",
                   "points":[

                   ]
                },
                {
                    [...]
                }
             ]
          },
          {
             "distance":"407 m",
             "duration":"0:00:49",
             "distanceMeters":406.90000000000003,
             "durationSeconds":49.31,
             "steps":[
                {
                   "distance":"18 m",
                   "duration":"0:00:02",
                   "distanceMeters":17.62,
                   "durationSeconds":2.11,
                   "navigationInstruction":null,
                   "name":"",
                   "points":[

                   ]
                },
                {
                   [...]
                }
             ]
          }
       ],
       "startDateTime":null,
       "finishDateTime":null,
       "srs":"epsg:4326"
    }
);
```

XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<routeResultV3>
   <status>OK</status>
   <distance>640 m</distance>
   <duration>0:01:25</duration>
```

```
    <distanceMeters>640.0</distanceMeters>
    <durationSeconds>85.64</durationSeconds>
    <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
    <wktGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
wktGeometry>
    <wktSimplifiedGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
wktSimplifiedGeometry>
    <leg>
       <distance>233 m</distance>
       <duration>0:00:36</duration>
       <distanceMeters>233.1</distanceMeters>
       <durationSeconds>36.33</durationSeconds>
       <step>
          <distance>139 m</distance>
          <duration>0:00:25</duration>
          <distanceMeters>138.59</distanceMeters>
          <durationSeconds>25.0</durationSeconds>
          <name>RUE LA FONTAINE BRUNEAU</name>
       </step>
       <step>
             [...]
       </step>
    </leg>
    <leg>
       <distance>407 m</distance>
       <duration>0:00:49</duration>
       <distanceMeters>406.90000000000003</distanceMeters>
       <durationSeconds>49.31</durationSeconds>
       <step>
          <distance>18 m</distance>
          <duration>0:00:02</duration>
          <distanceMeters>17.62</distanceMeters>
          <durationSeconds>2.11</durationSeconds>
          <name />
       </step>
       <step>
             [...]
       </step>
    </leg>
    <srs>epsg:4326</srs>
</routeResultV3>
```

JavaScript API

Include the JavaScript library

```
var routeCtrl = new GCUI.Control.Route();
routeCtrl.route({
    url:'http://<server>/<webapp>/api/lbs/route/v3.json',
    tolerance : 100,
    origin : new OpenLayers.LonLat(0.691012, 47.384813),
    destination : new OpenLayers.LonLat(0.691012, 47.384813),
    waypoints : [ new OpenLayers.LonLat(2.344408, 49.898798) ],
    callback : function(result, options) {
        console.log(result);
    }
});
```

The `result` variable is in JSON format as described above. The `callback` function passed to parameter is called at the end of the route calculation.

Possible responses

Case of a route found (routeResultV3/status is OK)

```
<?xml version="1.0" encoding="UTF-8"?>
<routeResultV3>
   <status>OK</status>
   <distance>640 m</distance>
   <duration>0:01:25</duration>
   <distanceMeters>640.0</distanceMeters>
   <durationSeconds>85.64</durationSeconds>
   <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
   <wktGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
wktGeometry>
   <wktSimplifiedGeometry>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
wktSimplifiedGeometry>
   <leg>
      <distance>233 m</distance>
      <duration>0:00:36</duration>
      <distanceMeters>233.1</distanceMeters>
      <durationSeconds>36.33</durationSeconds>
      <step>
         <distance>139 m</distance>
         <duration>0:00:25</duration>
         <distanceMeters>138.59</distanceMeters>
         <durationSeconds>25.0</durationSeconds>
         <name>RUE LA FONTAINE BRUNEAU</name>
      </step>
      <step>
            [...]
      </step>
   </leg>
   <leg>
      <distance>407 m</distance>
      <duration>0:00:49</duration>
      <distanceMeters>406.90000000000003</distanceMeters>
      <durationSeconds>49.31</durationSeconds>
      <step>
         <distance>18 m</distance>
         <duration>0:00:02</duration>
         <distanceMeters>17.62</distanceMeters>
         <durationSeconds>2.11</durationSeconds>
         <name />
      </step>
      <step>
            [...]
      </step>
   </leg>
   <srs>epsg:4326</srs>
</routeResultV3>
```

Case of a forgotten specification of a start or finish point (routeResultV3/status is ERROR)

```
<routeResultV3>
   <message>Origin and destination must be not null</message>
   <status>ERROR</status>
   <distanceMeters>0.0</distanceMeters>
   <durationSeconds>0.0</durationSeconds>
</routeResultV3>
```

Case of a faulty format assigned to the start or finish point or to route stops (routeResultV3/status is ERROR)

```
<routeResultV3>
   <message>Origin, destination and waypoints should be represented by a couple of coordinates</message>
   <status>ERROR</status>
   <distanceMeters>0.0</distanceMeters>
   <durationSeconds>0.0</durationSeconds>
</routeResultV3>
```

Case of a faulty type (serviceResult/status is ERROR)

```
<serviceResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
</serviceResult>
```

Case of a snap-to-graph error (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in route computation
Error in smartrouting
Failed to execute calculateRoute
com.geoconcept.smartrouting.SmartRoutingNativeException: failed to connect waypoint { 146.691012, 47.384813,
 0.000000 }
failed to connect waypoint { 146.691012, 47.384813, 0.000000 }</message>
   <status>ERROR</status>
</serviceResult>
```

Case of a problem with the graph: absent file, faulty filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
    <message>ServiceException: Error in route computation
Error in smartrouting
datasource is null</message>
    <status>ERROR</status>
</serviceResult>
```

V2

Parameters / properties

Input

parameter	description	optional	default
origin	Start point coordinates. The longitude and latitude coordinates are separated by , characters.	no	
destination	Finish point coordinates. The longitude and latitude coordinates are separated by the , character	no	
waypoints	Coordinates for route stops. Each pair of coordinates for a stop is framed by the <waypoint> tag The longitude and latitude coordinates are separated by the , character	yes	

parameter	description	optional	default
method	the shortest (distance) and fastest (time) route	yes	time
format	- standard: result = route summary / waypoints / geometry in wkt format / simplified geometry in wkt format / list of concatenated segments (without geometries) - extended: result = route summary / waypoints / simplified geometry in wkt format / list of non-concatenated segments (with geometries) - summary: result = route summary - geometry: result = route summary / waypoints / geometry in wkt format - simplifiedgeometry: result = route summary / waypoints / simplified geometry in wkt format - geometries: result = route summary / waypoints / geometry in wkt format / simplified geometry in wkt format - brief: result = route summary / list of segments with duration and distance - standardext: result = route summary / waypoints / list of concatenated segments with duration distance and geometry.	yes	standard
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
tolerance	Tolerance distance (in metres) for the geometry simplification.	yes	
graphName	Name of the graph to use	yes	
startDateTime	Start date and time (format ISO8601: local time) example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a start on 21 January 2014, at 9.00am in Paris	yes	
profileId	Vehicle identifier (saved under vehicle profiles) to use	yes	
profileName	Vehicle profile (saved under vehicle profiles) to use	yes	
exclusions	List of restriction rules to use, separated by the , or *;* character (Example: Toll, Tunnel, Bridge)	yes	
timeLine	List of intermediate durations to calculate positions along the route trajectory. Expressed in seconds, separated by the *;* character	yes	

Output

Route (routeResult)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	string	0/1	Total route distance in metres.
durationSeconds	string	0/1	Total duration of the route in seconds.
waypoints	string	0/1	Waypoints (BoundingBox) for the route geometry.
geometryWkt	string	0/1	Geometry of the route in WKT format
simplifiedWkt	string	0/1	Simplified geometry in WKT format.
leg (or legs in JSON / JSON-P)	subRoute (or array in JSON / JSON-P)	0/ unlimited	List of route segments.
startDateTime	string	0/1	Start date and time (format: ISO8601 without any area code: local time)

parameter	type	min/max	description
			Example: 2014-01-21T09:00:00.000+01:00 for a departure on 21 January 2014, at 9.00am in Paris
finishDateTime	string	0/1	Arrival date and time (format: ISO8601 without any area code: local time) Example: 2014-01-21T09:00:00.000+01:00 for an arrival time on 21 January 2014, at 9.00am in Paris
timeLineItem	timellineItem (or array in JSON / JSON-P)	0/ unlimited	List of calculated positions.

Route segment (subRoute)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	string	0/1	Distance covered by the route segment in metres.
durationSeconds	string	0/1	Duration of the route segment in seconds.
step or (or steps in JSON / JSON-P)	segment (or array in JSON / JSON-P)	0/ unlimited	List of component segments in the route segments

Route segment (segment)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	string	0/1	Distance covered by the route segment in metres.
durationSeconds	string	0/1	Duration of the route segment in seconds.
navigationInstruction	string	0/1	Navigation instruction code: - F: Straight on - FR: Turn slightly to the right - FL: Turn slightly to the left - R: Turn right - L: Turn left - BR: Turn hard right - BL: Turn hard left - B: U-turn - round_about_entry: Enter roundabout - round_about_exit: Exit from roundabout
name	string	0/1	Segment street name.
point	string	0/ unlimited	List of coordinates separated by the , character

Route position (timeLineItem)

parameter	type	min/max	description
elapsedTimeSeconds	double	0/1	Route duration in seconds up to this position.
message	string	0/1	Error message for this position.
status	string	0/1	Status of this position.
distanceMeters	double	0/1	Distance of the route in metres up to this position.
coordinates	string	0/1	Coordinates of the position.

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/routeService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:routeV2>
         <!--Optional:-->
         <request>
            <origin>
              <x>-1.351448</x>
              <y>47.446923</y>
            </origin>
            <destination>
               <x>-1.34529</x>
                <y>47.4479931</y>
            </destination>
            <!--Optional:-->
            <waypoints>
               <!--Zero or more repetitions:-->
               <waypoint>
                <x>-1.34981</x>
                 <y>47.44837</y>
               </waypoint>
            </waypoints>
            <!--Optional:-->
            <srs>epsg:4326</srs>
            <!--Optional:-->
            <method>time</method>
            <!--Optional:-->
            <format>STANDARD</format>
            <!--Optional:-->
            <tolerance>0.</tolerance>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <startDateTime>2015-07-29T09:00:00.000+01:00</startDateTime>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Zero or more repetitions:-->
            <rejectFlags></rejectFlags>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <timeLine>
               <!--Zero or more repetitions:-->
               <timeLineItem>5</timeLineItem>
               <timeLineItem>15</timeLineItem>
            </timeLine>
         </request>
      </sch:routeV2>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:routeV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <RouteResult>
            <status>OK</status>
            <distance>640 m</distance>
            <duration>0:01:24</duration>
            <distanceMeters>640</distanceMeters>
            <durationSeconds>84.87</durationSeconds>
            <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
            <geometryWkt>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
geometryWkt>
            <simplifiedWkt>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
simplifiedWkt>
            <leg>
               <distance>233 m</distance>
               <duration>0:00:36</duration>
               <distanceMeters>233.1</distanceMeters>
               <durationSeconds>36.16</durationSeconds>
               <step>
                  <distance>139 m</distance>
                  <duration>0:00:24</duration>
                  <distanceMeters>138.59</distanceMeters>
                  <durationSeconds>24.83</durationSeconds>
                  <name>RUE LA FONTAINE BRUNEAU</name>
               </step>
               <step>
                    [...]
               </step>
            </leg>
            <leg>
               <distance>407 m</distance>
               <duration>0:00:48</duration>
               <distanceMeters>406.9</distanceMeters>
               <durationSeconds>48.71</durationSeconds>
               <step>
                  <distance>18 m</distance>
                  <duration>0:00:02</duration>
                  <distanceMeters>17.62</distanceMeters>
                  <durationSeconds>2.11</durationSeconds>
                  <name/>
               </step>
               <step>
                    [...]
               </step>
            </leg>
            <startDateTime>2015-07-29T13:30:00.000+05:30</startDateTime>
            <finishDateTime>2015-07-29T13:31:24.870+05:30</finishDateTime>
            <timeLineItem>
               <elapsedTimeSeconds>5.0</elapsedTimeSeconds>
               <status>OK</status>
               <distanceMeters>27.90894700354487</distanceMeters>
               <coordinates>-1.351356,47.447160</coordinates>
            </timeLineItem>
            <timeLineItem>
               <elapsedTimeSeconds>15.0</elapsedTimeSeconds>
               <status>OK</status>
               <distanceMeters>83.72684101063462</distanceMeters>
               <coordinates>-1.351156,47.447643</coordinates>
            </timeLineItem>
         </RouteResult>
      </ns2:routeV2Response>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/route/v2.json?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/route/v2.json?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/route/v2.xml?
origin=-1.351448,47.446923&destination=-1.34529,47.4479931&waypoints=-1.34981,47.44837
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message":null,
   "status":"OK",
   "distance":"640 m",
   "duration":"0:02:06",
   "distanceMeters":"639.97",
   "durationSeconds":"126.55",
   "bounds":"-1.351448,47.446922;-1.345263,47.44848",
   "geometryWkt":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])",
   "simplifiedWkt":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])",
   "legs":[
      {
         "distance":"233 m",
         "duration":"0:00:46",
         "distanceMeters":"233.09",
         "durationSeconds":"46.21",
         "steps":[
            {
               "distance":"139 m",
               "duration":"0:00:29",
               "distanceMeters":"138.59",
               "durationSeconds":"29.21",
               "navInstruction":null,
               "name":"RUE LA FONTAINE BRUNEAU",
               "points":[

               ]
            },
            {
                [...]
            }
         ]
      },
      {
         "distance":"407 m",
         "duration":"0:01:20",
         "distanceMeters":"406.88",
         "durationSeconds":"80.34",
         "steps":[
            {
               "distance":"18 m",
               "duration":"0:00:03",
               "distanceMeters":"17.63",
               "durationSeconds":"3.17",
               "navInstruction":null,
               "name":"",
               "points":[

               ]
            },
            {
                [...]
            }
         ]
      }
   ],
   "startDateTime":null,
   "finishDateTime":null
}
```

JSON-P format

```
myCallback(
    {
       "message":null,
       "status":"OK",
       "distance":"640 m",
       "duration":"0:02:06",
       "distanceMeters":"639.97",
       "durationSeconds":"126.55",
       "bounds":"-1.351448,47.446922;-1.345263,47.44848",
       "geometryWkt":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])",
       "simplifiedWkt":"LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])",
       "legs":[
          {
             "distance":"233 m",
             "duration":"0:00:46",
             "distanceMeters":"233.09",
             "durationSeconds":"46.21",
             "steps":[
                {
                   "distance":"139 m",
                   "duration":"0:00:29",
                   "distanceMeters":"138.59",
                   "durationSeconds":"29.21",
                   "navInstruction":null,
                   "name":"RUE LA FONTAINE BRUNEAU",
                   "points":[

                   ]
                },
                {
                    [...]
                }
             ]
          },
          {
             "distance":"407 m",
             "duration":"0:01:20",
             "distanceMeters":"406.88",
             "durationSeconds":"80.34",
             "steps":[
                {
                   "distance":"18 m",
                   "duration":"0:00:03",
                   "distanceMeters":"17.63",
                   "durationSeconds":"3.17",
                   "navInstruction":null,
                   "name":"",
                   "points":[

                   ]
                },
                {
                    [...]
                }
             ]
          }
       ],
       "startDateTime":null,
       "finishDateTime":null
    }
);
```

XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<routeResult>
   <status>OK</status>
   <distance>640 m</distance>
   <duration>0:02:06</duration>
   <distanceMeters>639.97</distanceMeters>
   <durationSeconds>126.55</durationSeconds>
   <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
   <geometryWkt>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
geometryWkt>
   <simplifiedWkt>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
simplifiedWkt>
   <leg>
      <distance>233 m</distance>
      <duration>0:00:46</duration>
```

```
         <distanceMeters>233.09</distanceMeters>
         <durationSeconds>46.21</durationSeconds>
         <step>
            <distance>139 m</distance>
            <duration>0:00:29</duration>
            <distanceMeters>138.59</distanceMeters>
            <durationSeconds>29.21</durationSeconds>
            <name>RUE LA FONTAINE BRUNEAU</name>
         </step>
         <step>
            [...]
         </step>
      </leg>
      <leg>
         <distance>407 m</distance>
         <duration>0:01:20</duration>
         <distanceMeters>406.88</distanceMeters>
         <durationSeconds>80.34</durationSeconds>
         <step>
            <distance>18 m</distance>
            <duration>0:00:03</duration>
            <distanceMeters>17.63</distanceMeters>
            <durationSeconds>3.17</durationSeconds>
            <name />
         </step>
         <step>
            [...]
         </step>
      </leg>
   </routeResult>
```

JavaScript API

Include the JavaScript library

```
var routeCtrl = new GCUI.Control.Route();
routeCtrl.route({
    url:'http://<server>/<webapp>/api/lbs/route/v2.json',
    tolerance : 100,
    origin : new OpenLayers.LonLat(0.691012, 47.384813),
    destination : new OpenLayers.LonLat(0.691012, 47.384813),
    waypoints : [ new OpenLayers.LonLat(2.344408, 49.898798) ],
    callback : function(result, options) {
        console.log(result);
    }
});
```

The `result` variable is in JSON format as described above. The `callback` function passed to parameter is called at the end of the route calculation.

Possible responses

Case of a route found (routeResult/status is OK)

```
<?xml version="1.0" encoding="UTF-8"?>
<routeResult>
   <status>OK</status>
   <distance>640 m</distance>
   <duration>0:02:06</duration>
   <distanceMeters>639.97</distanceMeters>
   <durationSeconds>126.55</durationSeconds>
   <bounds>-1.351448,47.446922;-1.345263,47.44848</bounds>
   <geometryWkt>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
geometryWkt>
   <simplifiedWkt>LINESTRING (-1.351448 47.446923, -1.351439 47.446922, -1.35142 47.44699, [...])</
simplifiedWkt>
   <leg>
      <distance>233 m</distance>
      <duration>0:00:46</duration>
      <distanceMeters>233.09</distanceMeters>
      <durationSeconds>46.21</durationSeconds>
      <step>
         <distance>139 m</distance>
         <duration>0:00:29</duration>
         <distanceMeters>138.59</distanceMeters>
         <durationSeconds>29.21</durationSeconds>
         <name>RUE LA FONTAINE BRUNEAU</name>
      </step>
      <step>
         [...]
      </step>
   </leg>
   <leg>
      <distance>407 m</distance>
      <duration>0:01:20</duration>
      <distanceMeters>406.88</distanceMeters>
      <durationSeconds>80.34</durationSeconds>
      <step>
         <distance>18 m</distance>
         <duration>0:00:03</duration>
         <distanceMeters>17.63</distanceMeters>
         <durationSeconds>3.17</durationSeconds>
         <name />
      </step>
      <step>
         [...]
      </step>
   </leg>
</routeResult>
```

Case of a forgotten specification of a start or finish point (routeResult/status is ERROR)

```
<routeResult>
   <message>Origin and destination must be not null</message>
   <status>ERROR</status>
</routeResult>
```

Case of a faulty format assigned to the start or finish point or to route stops (routeResult/status is ERROR)

```
<routeResult>
   <message>Origin, destination and waypoints should be represented by a couple of coordinates</message>
   <status>ERROR</status>
</routeResult>
```

Case of a faulty type (serviceResult/status is ERROR)

```
<serviceResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
```

```
</serviceResult>
```

Case of a snap-to-graph error (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in route computation
Error in smartrouting
Failed to execute calculateRoute
com.geoconcept.smartrouting.SmartRoutingNativeException: failed to connect waypoint { 146.691012, 47.384813,
 0.000000 }
failed to connect waypoint { 146.691012, 47.384813, 0.000000 }</message>
   <status>ERROR</status>
</serviceResult>
```

Case of a problem with the graph: absent file, faulty filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
    <message>ServiceException: Error in route computation
Error in smartrouting
datasource is null</message>
    <status>ERROR</status>
</serviceResult>
```

V1

Parameters / properties

Input

parameter	description	optional	default
origin	Start point coordinates. The longitude and latitude coordinates are separated by , characters.	no	
destination	Finish point coordinates. The longitude and latitude coordinates are separated by the , character	no	
waypoints	Coordinates for route stops. Each pair of coordinates for a stop is framed by the <waypoint> tag The longitude and latitude coordinates are separated by the , character	yes	
method	the shortest (distance) and fastest (time) route	yes	time
format	- standard: result = route summary / waypoints / geometry in wkt format / simplified geometry in wkt format / list of concatenated segments (without geometries) - extended: result = route summary / waypoints / simplified geometry in wkt format / list of non-concatenated segments (with geometries) - summary: result = route summary - geometry: result = route summary / waypoints / geometry in wkt format - simplifiedgeometry: result = route summary / waypoints / simplified geometry in wkt format - geometries: result = route summary / waypoints / geometry in wkt format / simplified geometry in wkt format - brief: result = route summary / list of segments with duration and distance - standardext: result = route summary / waypoints / list of concatenated segments with duration distance and geometry.	yes	standard
projection	Deprecated, replaced by srs	yes	

parameter	description	optional	default
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
tolerance	Tolerance distance (in metres) for the geometry simplification.	yes	
graphName	Name of the graph to use	yes	
map	Deprecated: Name of the map to use	yes	
applyMapPrecisionOut	Deprecated: Precision of the map	yes	
startDateTime	Start date and time (format ISO8601: local time) example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a start on 21 January 2014, at 9.00am in Paris	yes	
profileId	Vehicle identifier (saved under vehicle profiles) to use	yes	
profileName	Vehicle profile (saved under vehicle profiles) to use	yes	
rejectFlags	Deprecated, replaced by exclusions	yes	
exclusions	List of restriction rules to use, separated by the , or *;* character (Example: Toll, Tunnel, Bridge)	yes	

Output

Route (routeResult)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	string	0/1	Total route distance in metres.
durationSeconds	string	0/1	Total duration of the route in seconds.
waypoints	string	0/1	Waypoints (BoundingBox) for the route geometry.
geometryWkt	string	0/1	Geometry of the route in WKT format
simplifiedWkt	string	0/1	Simplified geometry in WKT format.
leg (or legs in JSON / JSON-P)	subRoute (or array in JSON / JSON-P)	0/ unlimited	List of route segments.
startDateTime	string	0/1	Start date and time (format: ISO8601 without any area code: local time) Example: 2014-01-21T09:00:00.000+01:00 for a departure on 21 January 2014, at 9.00am in Paris
finishDateTime	string	0/1	Arrival date and time (format: ISO8601 without any area code: local time) Example: 2014-01-21T09:00:00.000+01:00 for an arrival time on 21 January 2014, at 9.00am in Paris
timeLineItem	timellineItem (or array in JSON / JSON-P)	0/ unlimited	Not implemented in this version.

Route segment (subRoute)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted:

parameter	type	min/max	description
			- xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	string	0/1	Distance covered by the route segment in metres.
durationSeconds	string	0/1	Duration of the route segment in seconds.
step or (or steps in JSON / JSON-P)	segment (or array in JSON / JSON-P)	0/ unlimited	List of component segments in the route segments

Route segment (segment)

parameter	type	min/max	description
distance	string	0/1	Total route distance, formatted: - xx.xx Km - xx m (if the distance is less than 1 km)
duration	string	0/1	Total duration of the route, formatted: - HH:mm:ss (HH=hours, mm=minutes, ss=seconds)
distanceMeters	string	0/1	Distance covered by the route segment in metres.
durationSeconds	string	0/1	Duration of the route segment in seconds.
navigationInstruction	string	0/1	Navigation instruction code: - F: Straight on - FR: Turn slightly to the right - FL: Turn slightly to the left - R: Turn right - L: Turn left - BR: Turn hard right - BL: Turn hard left - B: U-turn - round_about_entry: Enter roundabout - round_about_exit: Exit from roundabout
name	string	0/1	Segment street name.
point	string	0/ unlimited	List of coordinates separated by the , character

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/routeService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:route>
         <!--Optional:-->
         <request>
            <origin>
```

```
               <x>0.691012</x>
               <y>47.384813</y>
            </origin>
            <destination>
               <x>0.693012</x>
               <y>47.385813</y>
            </destination>
            <!--Optional:-->
            <waypoints>
               <!--Zero or more repetitions:-->
               <waypoint>
                  <x>0.692012</x>
                  <y>47.384813</y>
               </waypoint>
            </waypoints>
            <!--Optional:-->
            <srs></srs>
            <!--Optional:-->
            <method></method>
            <!--Optional:-->
            <format></format>
            <!--Optional:-->
            <tolerance></tolerance>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Zero or more repetitions:-->
            <rejectFlags></rejectFlags>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
         </request>
      </sch:route>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:routeResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <RouteResult>
            <status>OK</status>
            <distance>577 m</distance>
            <duration>0:02:49</duration>
            <bounds>0.69089,47.38404;0.693012,47.386078</bounds>
            <geometryWkt>LINESTRING (0.6910119767724592 47.38481290982329, 0.6910138207911578
 47.384806371938815, ...)</geometryWkt>
            <simplifiedWkt>LINESTRING (0.6910119767724592 47.38481290982329, 0.6910138207911578
 47.384806371938815, ...)</simplifiedWkt>
            <leg>
               <distance>266 m</distance>
               <duration>0:01:08</duration>
               <step>
                  <distance>10</distance>
                  <duration>2</duration>
                  <name>RUE EUPATORIA</name>
               </step>
               <step>
                  <distance>86</distance>
                  <duration>18</duration>
                  <navInstruction>L</navInstruction>
                  <name>AVENUE DE GRAMMONT</name>
               </step>
               <step>
                  <distance>13</distance>
                  <duration>2</duration>
                  <navInstruction>L</navInstruction>
                  <name>PLACE VAILLANT</name>
               </step>
               <step>
                  <distance>66</distance>
                  <duration>12</duration>
                  <navInstruction>L</navInstruction>
                  <name>AVENUE DE GRAMMONT</name>
               </step>
               <step>
                  <distance>89</distance>
                  <duration>32</duration>
                  <navInstruction>R</navInstruction>
                  <name>RUE PARMENTIER</name>
               </step>
            </leg>
            <leg>
               <distance>311 m</distance>
               <duration>0:01:40</duration>
               <step>
                  <distance>90</distance>
                  <duration>33</duration>
                  <name>RUE PARMENTIER</name>
               </step>
               <step>
                  <distance>153</distance>
                  <duration>32</duration>
                  <navInstruction>L</navInstruction>
                  <name>RUE MICHELET</name>
               </step>
               <step>
                  <distance>66</distance>
                  <duration>35</duration>
                  <navInstruction>R</navInstruction>
                  <name>RUE DUPORTAL</name>
               </step>
            </leg>
         </RouteResult>
      </ns2:routeResponse>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/route.json?
origin=0.691012,47.384813&destination=0.693012,47.385813&waypoints=0.692012,47.384813
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/route.json?
origin=0.691012,47.384813&destination=0.693012,47.385813&waypoints=0.692012,47.384813&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/route.xml?
origin=0.691012,47.384813&destination=0.693012,47.385813&waypoints=0.692012,47.384813
```

Response

The response is always in UTF-8 format.

JSON format

```
{
        "message":null,"status":"OK",
        "distance":"577 m","duration":"0:02:49","bounds":"475757.89,2266229.66;475920.47,2266453.7",
        "geometryWkt":"LINESTRING (475767.17 2266316.12, 475767.3 2266315.39, ...)",
        "legs":
                [
                        {"distance":"266 m","duration":"0:01:08",
                                "steps":
                                        [

 {"distance":"10","duration":"2","navInstruction":null,"name":"RUE EUPATORIA","points":[]},

 {"distance":"86","duration":"18","navInstruction":"L","name":"AVENUE DE GRAMMONT","points":[]},

 {"distance":"13","duration":"2","navInstruction":"L","name":"PLACE VAILLANT","points":[]},

 {"distance":"66","duration":"12","navInstruction":"L","name":"AVENUE DE GRAMMONT","points":[]},

 {"distance":"89","duration":"32","navInstruction":"R","name":"RUE PARMENTIER","points":[]}
                                        ]
                        },
                        {"distance":"311 m","duration":"0:01:40",
                                "steps":
                                        [

 {"distance":"90","duration":"33","navInstruction":null,"name":"RUE PARMENTIER","points":[]},

 {"distance":"153","duration":"32","navInstruction":"L","name":"RUE MICHELET","points":[]},

 {"distance":"66","duration":"35","navInstruction":"R","name":"RUE DUPORTAL","points":[]}
                                        ]
                                }
                ],
        "startDateTime":null,"finishDateTime":null
}
```

JSON-P format

```
myCallback(
        {
                "message":null,"status":"OK",
                "distance":"577 m","duration":"0:02:49","bounds":"475757.89,2266229.66;475920.47,2266453.7",
                "geometryWkt":"LINESTRING (475767.17 2266316.12, 475767.3 2266315.39, ...)",
                "legs":
                        [
                                {"distance":"266 m","duration":"0:01:08",
                                "steps":
                                        [

 {"distance":"10","duration":"2","navInstruction":null,"name":"RUE EUPATORIA","points":[]},

 {"distance":"86","duration":"18","navInstruction":"L","name":"AVENUE DE GRAMMONT","points":[]},

 {"distance":"13","duration":"2","navInstruction":"L","name":"PLACE VAILLANT","points":[]},

 {"distance":"66","duration":"12","navInstruction":"L","name":"AVENUE DE GRAMMONT","points":[]},

 {"distance":"89","duration":"32","navInstruction":"R","name":"RUE PARMENTIER","points":[]}
                                        ]
                                },
                                {"distance":"311 m","duration":"0:01:40",
                                "steps":
                                        [

 {"distance":"90","duration":"33","navInstruction":null,"name":"RUE PARMENTIER","points":[]},

 {"distance":"153","duration":"32","navInstruction":"L","name":"RUE MICHELET","points":[]},

 {"distance":"66","duration":"35","navInstruction":"R","name":"RUE DUPORTAL","points":[]}
                                        ]
                                }
                        ],
                "startDateTime":null,"finishDateTime":null
        }
);
```

XML format

```
<routeResult>
        <status>OK</status>
        <distance>577 m</distance>
        <duration>0:02:49</duration>
        <bounds>475757.89,2266229.66;475920.47,2266453.7</bounds>
        <geometryWkt>LINESTRING (475767.17 2266316.12, 475767.3 2266315.39, ...)</geometryWkt>
        <simplifiedWkt>LINESTRING (475767.17 2266316.12, 475767.3 2266315.39, ...)</simplifiedWkt>
        <leg>
                <distance>266 m</distance>
                <duration>0:01:08</duration>
                <step>
                        <distance>10</distance>
                        <duration>2</duration>
                        <name>RUE EUPATORIA</name>
                </step>
                <step>
                        <distance>86</distance>
                        <duration>18</duration>
                        <navInstruction>L</navInstruction>
                        <name>AVENUE DE GRAMMONT</name>
                </step>
                <step>
                        <distance>13</distance>
                        <duration>2</duration>
                        <navInstruction>L</navInstruction>
                        <name>PLACE VAILLANT</name>
                </step>
                <step>
```

```
                    <distance>66</distance>
                    <duration>12</duration>
                    <navInstruction>L</navInstruction>
                    <name>AVENUE DE GRAMMONT</name>
            </step>
            <step>
                    <distance>89</distance>
                    <duration>32</duration>
                    <navInstruction>R</navInstruction>
                    <name>RUE PARMENTIER</name>
            </step>
    </leg>
    <leg>
            <distance>311 m</distance>
            <duration>0:01:40</duration>
            <step>
                    <distance>90</distance>
                    <duration>33</duration>
                    <name>RUE PARMENTIER</name>
            </step>
            <step>
                    <distance>153</distance>
                    <duration>32</duration>
                    <navInstruction>L</navInstruction>
                    <name>RUE MICHELET</name>
            </step>
            <step>
                    <distance>66</distance>
                    <duration>35</duration>
                    <navInstruction>R</navInstruction>
                    <name>RUE DUPORTAL</name>
            </step>
    </leg>
</routeResult>
```

JavaScript API

Include the JavaScript library

```
var routeCtrl = new GCUI.Control.Route();
routeCtrl.route({
    url:'http://<server>/<webapp>/api/lbs/route.json',
    tolerance : 100,
    origin : new OpenLayers.LonLat(0.691012, 47.384813),
    destination : new OpenLayers.LonLat(0.691012, 47.384813),
    waypoints : [ new OpenLayers.LonLat(2.344408, 49.898798) ],
    callback : function(result, options) {
        console.log(result);
    }
});
```

The `result` variable is in JSON format as described above. The `callback` function passed to parameter is called at the end of the route calculation.

Possible responses

Case of a route found (routeResult/status is OK)

```
<routeResult>
    <status>OK</status>
    <distance>1.66 Km</distance>
    <duration>0:04:53</duration>
    <bounds>2.423385,48.84452;2.43999,48.84741</bounds>
    <geometryWkt>LINESTRING (2.4233899276812516 48.84461991900793, 2.4233847309012826
 48.84462042192212, ...)</geometryWkt>
    <simplifiedWkt>LINESTRING (2.4233899276812516 48.84461991900793, 2.4233847309012826
 48.84462042192212, ...)</simplifiedWkt>
    <leg>
        <distance>1.66 Km</distance>
        <duration>0:04:53</duration>
        <step>
            <distance>147</distance>
            <duration>31</duration>
            <name>AVENUE PASTEUR</name>
        </step>
        <step>
            <distance>1108</distance>
            <duration>159</duration>
            <navInstruction>R</navInstruction>
            <name>AVENUE DE PARIS</name>
        </step>
                ...
        </step>
    </leg>
</routeResult>
```

Case of a forgotten specification of a start or finish point (routeResult/status is ERROR)

```
<routeResult>
    <message>Origin and destination must be not null</message>
    <status>ERROR</status>
</routeResult>
```

Case of a faulty type (serviceResult/status is ERROR)

```
<serviceResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
</serviceResult>
```

Case of a snap-to-graph error (serviceResult/status is ERROR)

```
<serviceResult>
    <message>ServiceException: Error in route computation
Error in smartrouting
Failed to execute calculateRoute
com.geoconcept.smartrouting.SmartRoutingNativeException: failed to connect waypoint { 243966.000000,
 48.847410, 0.000000 }
failed to connect waypoint { 243966.000000, 48.847410, 0.000000 }</message>
    <status>ERROR</status>
</serviceResult>
```

Case of a problem with the graph: absent file, faulty filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
    <message>ServiceException: Error in route computation
Error in smartrouting
datasource is null</message>
    <status>ERROR</status>
```

```
</serviceResult>
```

FAQ

1. Is it possible to give priority to either journey time or distance?

   Yes, by changing the method `method` distance or time.

2. Can I use aliases instead of .siti file names to call a datasource?

   Yes, refer to details in the FAQ of the reverse geocoding Web Service.

3. How can I use route statistics?

   Check that the graph does contain these statistics and use these parameters: `computeOptions` with a value of *trafficPatterns* and `startDateTime` to specify the departure date/time.

4. Is the order of points important?

   The points - whether they are start points, finish points, or way points - are read and used in that order. The first point declared is considered as a start point, the second as an arrival point, and the others as way points. The order of the points is therefore very important.

5. How can I perform a route calculation excluding toll roads?

   If the *Toll* constraint has been included in the graph, place an exclusion in `exclusions` :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:route>
         <!--Optional:-->
         <request>
            <origin>
               <x>0.691012</x>
               <y>47.384813</y>
            </origin>
            <destination>
               <x>0.693012</x>
               <y>47.385813</y>
            </destination>
            <!--Zero or more repetitions:-->
            <waypoints>
                                <waypoint>
                                        <x>0.692012</x>
                                        <y>47.384813</y>
                                </waypoint>
            </waypoints>
            <!--Optional:-->
            <srs></srs>
            <!--Optional:-->
            <method></method>
            <!--Optional:-->
            <format></format>
            <tolerance></tolerance>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <startDateTime></startDateTime>
                        <!--Optional:-->
            <profileId></profileId>
                        <!--Optional:-->
                        <profileName></profileName>
            <!--Zero or more repetitions:-->
            <exclusions>Toll</exclusions>
         </request>
      </sch:route>
   </soapenv:Body>
</soapenv:Envelope>
```

6. What are Speed Patterns? How can I use them?

   In order to propose journey times that are as accurate as possible and reflect changing traffic conditions, graphs supplied by GEOCONCEPT SAS include, from version M18 upwards, 5 speed profiles (Speed Patterns) for cars and lorries, to take into account the different levels of congestion during the course of one day:

   - standard *normal-speed* corresponds to the speed at a time when the roads have an average congestion level (eg 11.00am)
   - night *fast-speed* corresponds to a very fluid traffic situation, often experienced at night-time (3.00am)
   - congested *slow-speed* corresponds to the times when traffic is densest (8.00am)
   - rush hour *very-slow-speed* corresponds to times when traffic is densest in urban and built-up areas, and is slower than the speed above (8.00am)
   - default *default* corresponds to speeds averaged out over an entire day

   To use them, you need to pass to parameter, when calling the web service, the **`computeOptions`** parameter with the *speedPattern* option and specify the value of the Speed Pattern requested, without forgetting a vehicle profile
   Example:

   +

   ```
   http://<server>/<webapp>/api/lbs/route/v5.xml?
   origin=-1.519363,47.215945&destination=-1.559309,47.213465&computeOptions=speedPattern:fast-
   speed&profileId=1
   ```

   + Returns a journey of 707 seconds as against 817 seconds for a *slow-speed* speedPattern.

7. How to use Heavy Goods Vehicle attributes?

   The graph must include the attributes for Heavy Goods Vehicles (as standard in the graphs supplied by GEOCONCEPT SAS from version M18 upwards) and either calculate an itinerary using a vehicle profile using restrictions (cf. the catalogue of vehicles, editable, defined in the SmartRoutingVehicles.xml file, stored in the folder `*‘<GEOCONCEPT_WEB_HOME>”\smartrouting\jee\smartrouting\conf\* ), or overwrite the call to the web service by using the **`computeOptions`** parameter with the options *length*, *width*, *height*, *weight*, *axles* and/or *weightPerAxle*.
   Example for a journey with a vehicle 4.5 metres high:

   ```
   http://<server>/<webapp>/api/lbs/route/v5.xml?
   origin=-1.543363,47.215945&destination=-1.550309,47.213465&computeOptions=height:450
   ```

   Returns a journey of 1,529 metres as against 624 metres without any height restriction.

8. How is carbon footprint for a route calculated?

The carbon footprint calculation takes into account vehicle type, fuel type, and speed - road section by road section: note that 10 km in a built-up area does not have the same carbon footprint as 10 kms in the countryside.
The carbon footprint for the itinerary is calculated automatically EITHER using a vehicle profile that has pre-assigned restrictions as defined in the vehicle catalogue, which can be edited, and is located in the SmartRoutingVehicles.xml file in the folder `*‘<GEOCONCEPT_WEB_HOME>’’\smartrouting\jee\smartrouting\conf\* ), OR by overwriting the restriction values at the time of the call to the web service using `computeOptions` and the *fuelType*, *averageConsumption* and/or *customAverageCO2UnitEmission* options.
The available values are fuelType with possible values:

- NoFuel
- Diesel
- UnleadedFuel
- LGP
- CustomFuelType (allows you to personalise a value in kgs of CO2 per litre, to be entered under customAverageCO2UnitEmission)

averageConsumption allows you to indicate the average vehicle consumption for 100 kilometers
Example, for the following configuration:

+

```
                <vehicle id="10" name="Car">
                        <speedProfile>Cars</speedProfile>
                        <speedProfileId>3</speedProfileId>
                        <rejectFlag>Automobiles</rejectFlag>
                        <snapSpeed>4</snapSpeed>
                        <fuelType>UndefinedFuelType</fuelType>
                </vehicle>
                <vehicle id="11" name="Car UnleadedFuel">
                        <speedProfile>Cars</speedProfile>
                        <speedProfileId>3</speedProfileId>
                        <rejectFlag>Automobiles</rejectFlag>
                        <speedPatternsProfile>Cars SpeedPatterns</speedPatternsProfile>
                        <snapSpeed>4</snapSpeed>
                        <fuelType>UnleadedFuel</fuelType>
                        <averageConsumption>7.27</averageConsumption>
                </vehicle>
                <vehicle id="12" name="Car Diesel">
                        <speedProfile>Cars</speedProfile>
                        <speedProfileId>3</speedProfileId>
                        <rejectFlag>Automobiles</rejectFlag>
                        <speedPatternsProfile>Cars SpeedPatterns</speedPatternsProfile>
                        <snapSpeed>4</snapSpeed>
                        <fuelType>Diesel</fuelType>
                        <averageConsumption>6.06</averageConsumption>
                </vehicle>
                <vehicle id="30" name="Truck CustomFuelType">
                        <speedProfile>Trucks</speedProfile>
                        <speedProfileId>5</speedProfileId>
                        <rejectFlag>Trucks</rejectFlag>
                        <snapSpeed>4</snapSpeed>
```

```
                            <fuelType>CustomFuelType</fuelType>
                            <customAverageCO2UnitEmission>5</customAverageCO2UnitEmission>
                    </vehicle>
            <vehicle id="32" name="Truck Diesel">
                            <speedProfile>Trucks</speedProfile>
                            <speedProfileId>5</speedProfileId>
                            <rejectFlag>Trucks</rejectFlag>
                            <speedPatternsProfile>Trucks SpeedPatterns</speedPatternsProfile>
                            <snapSpeed>4</snapSpeed>
                            <fuelType>Diesel</fuelType>
                            <averageConsumption>34</averageConsumption>
                    </vehicle>
```

+ the following is obtained:

+ Car

+

```
http://<server>/<webapp>/api/lbs/route/v5.xml?
origin=-0.978536,47.443316&destination=-2.511388,47.303726&profileId=10
```

+ returns in kgs the CO2 equivalent:

+

```
<carbonFootprint>0.0</carbonFootprint>
```

+ Car UnleadedFuel

+

```
http://<server>/<webapp>/api/lbs/route/v5.xml?
origin=-0.978536,47.443316&destination=-2.511388,47.303726&profileId=11
```

+ returns in kgs the CO2 equivalent:

+

```
<carbonFootprint>31.151</carbonFootprint>
```

+ Car Diesel

+

```
http://<server>/<webapp>/api/lbs/route/v5.xml?
origin=-0.978536,47.443316&destination=-2.511388,47.303726&profileId=12
```

+ returns in kgs the CO2 equivalent:

+

```
<carbonFootprint>26.117</carbonFootprint>
```

+ Truck CustomFuelType

+

```
http://<server>/<webapp>/api/lbs/route/v5.xml?
origin=-0.978536,47.443316&destination=-2.511388,47.303726&profileId=30
```

+ returns in kgs the CO2 equivalent:

+

```
<carbonFootprint>43.914</carbonFootprint>
```

+ Truck Diesel

+

```
http://<server>/<webapp>/api/lbs/route/v5.xml?
origin=-0.978536,47.443316&destination=-2.511388,47.303726&profileId=32
```

+ returns in kgs the CO2 equivalent:

+

```
<carbonFootprint>153.081</carbonFootprint>
```

Route calculation (batch)

(fr) Cette page n’est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

This web service calculates an itineraries set between two points and returns a full route sheet. Adding intermediate steps is an option for the user, in the context of the configured graph or network, the name of which has been specified in the interface.

V1

Parameters / properties

Input

parameter	description	optional	default
routeRequests	RouteRequestV5 table	No	Route table
timeOut	string	Yes	Time out calculation

parameter	description	optional	default
			value (in milliseconds).

- Route request input (RouteRequestV5)**

Described under Route calculation web service.

Output"

parameter	type	min/max	description
result	RouteResultV5 table	0/ unlimited	Routes issued as output

- Route result output (RouteResultV5)**

Described under Route calculation web service.

Possible returns

Case of a route found (routeResultV5/status is OK)

```
{
    "message": null,
    "status": "OK",
    "results": [
        {
            "message": null,
            "status": "OK",
            "distance": "1.66 Km",
            "duration": "0:07:56",
            "distanceMeters": 1658.9,
            "durationSeconds": 476.69,
            "bounds": null,
            "wktGeometry": null,
            "wktSimplifiedGeometry": null,
            "compressedGeometry": null,
            "compressedSimplifiedGeometry": null,
            "legs": [
            ],
            "startDateTime": null,
            "finishDateTime": null,
            "srs": "epsg:4326",
            "originNode": null,
            "waypointNodes": null,
            "destinationNode": null,
            "carbonFootprint": null
        },
        {
            "message": null,
            "status": "OK",
            "distance": "159 m",
            "duration": "0:00:47",
            "distanceMeters": 158.76,
            "durationSeconds": 47.66,
            "bounds": null,
            "wktGeometry": null,
            "wktSimplifiedGeometry": null,
            "compressedGeometry": null,
            "compressedSimplifiedGeometry": null,
            "legs": [
            ],
            "startDateTime": null,
            "finishDateTime": null,
            "srs": "epsg:4326",
            "originNode": null,
            "waypointNodes": null,
            "destinationNode": null,
            "carbonFootprint": null
        }
    ]
}
```

See <<gcweb-ws-routing-returns,Calculate route> web service.

FAQ

See Calculate route web service.

Optimisation (simplified version)

(fr) Cette page n'est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

This web service allows you to define optimum routes while minimising travel times and distances. The ordering of points to visit is optimised on the basis of geographic and operational constraints (time slots, duration of the visit..;). It is based on the configured graph, the name of which has been specified in the Geoconcept Web administration interface.

Availability

This web service is a Geoconcept Web option: please contact us to know how to purchase this service.

Version change

Earlier versions of the web service are conserved in Geoconcept Web to ensure compatibility with earlier software versioning and development. We recommend using the most recent version.

Changes in relation to V3

- Addition of various input parameters: "graphName", "profileId", "profileName", "configName", "snapMethod", "exclusions", "startDateTime", "avoidArea", "computeOptions", "maxCost" and "timeOut".
- Addition in output of the list of "unreachableSteps", these being the route stops that could not be visited.

Changes in relation to V2

- Addition of the "timewindows/timewindow" parameter.

- The "totalDistance" and "totalTime" parameters have been renamed "distanceMeters" and "durationSeconds" respectively.
- The Web service is no longer available in REST GET.

V3

Parameters / properties

Input

parameter	description	optional	default
origin (optimRouteStepV3)	origin point (id,x,y)	yes	
destination (optimRouteStepV3)	destination point (id,x,y)	yes	
steps (optimRouteStepV3)	points to visit (id,x,y;id,x,y;…) Several slots for visiting times in milliseconds can be specified for each point: (id,x,y,start, finish)	no	
method	shortest (distance) or fastest (time) route	yes	time
graphName	"Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
profileId	Vehicle identifier (saved in the vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
profileName	Vehicle profile (saved in the vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
configName	Name of the configuration to use (defined in Geoconcept Web - Administration / Tools / Road graphs) This replaces the use of graphName, profileId and profileName	yes	
snapMethod	Snap to graph method - standard: to the nearest connectable road section - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without any restriction rules - nodes: Snap directly to nodes supplied by the originNode, destinationNode and waypointNodes parameters, or, if these parameters have not been set, to the nearest nodes sourced by the origin, destination and waypoint parameters	yes	standard
exclusions	List of restriction rules to use, separated by the *;* character (Example: Toll, Tunnel, Bridge)	yes	
startDateTime	Departure Date and Time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a departure date on 21 Januray 2014, at 9.00am in Paris. Caution: the + character may be misinterpreted by some browsers, and in this instance, you will need to replace it with *%2B*.	yes	
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84: `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...]`	yes	

parameter	description	optional	default
	`-1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))` Care: WKT geometries must be closed.		
computeOptions	List of options for the calculation,separated by *;* characters - trafficPatterns: uses routing statistics (you will need to supply a value for the startDateTime parameter and use a graph that includes traffic patterns information) - speedPattern (M18): uses a *speed pattern* as defined in the SmartRoutingVehicles.xml file, located in the `'*<GEOCONCEPT_WEB_HOME>"\smartrouting\jee\smartrouting\conf\* folder. Use as follows: "speedPattern:slow-speed" - length (M18): maximum authorised length in centimeters (you need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "length:950" - width (M18): maximum authorised width in centimeters (You will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "width:255" - height (M18): maximum authorised height in centimeters (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "height:360" - weight (M18): maximum authorised weight in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weight:18000" - axles (M18): maximum authorised number of axles (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "axles:2" - weightPerAxle (M18): maximum authorised weight per axle in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weightPerAxle:9000" - snapSpeed: snap-to-graph speed in kilometers per hour. Use as follows: "snapSpeed:10"	yes	
matrixProvider	matrix calculation provider globe: as the crow flies on the globe (the coordinates must be in long/lat) euclidean: as the crow flies on the map (the coordinates must be in metres) smartrouting: calculation on the road network by SmartRouting nokia: calculation on the road network via a Nokia service provided: matrix parameter	yes	smartrouting
matrix	time/distance matrix (id1,id2,temps,distance;…)	yes	
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	wgs84
directSpeed	Default speed during utilisation of globe or euclidean matrices	yes	50
maxCost	Maximum cost not to exceed in the calculation -1: no maximum cost to take into account 0: take the default value defined in the SmartRouting Server configuration if not: value in metres if method=distance or in seconds if method=time	yes	
timeOut	Time out for the calculation (in milliseconds)	yes	

(optimRouteStepV3) steps

parameter	type	min/max	description
id	string	0/1	point id
duration	long	1/1	duration of a visit, in milliseconds. O by default

parameter	type	min/max	description
timeWindows/ timeWindow (optimTimeWindow)	long	1/1	Time windows

Time windows (optimTimeWindow)

parameter	type	min/max	description
start	long	1/1	start of the time window, in milliseconds. 0 by default.
end	long	1/1	time window end, in milliseconds. Long.MAX_VALUE by default

Output

parameter	type	min/max	description
steps/step	array (optimRouteStepV3)	0/ unlimited	list of route stops in sequence
distanceMeters	long	1/1	total journey distance in meters
durationSeconds	long	1/1	total journey time in seconds
unreachableSteps	array (OptimUnreachableStepV3)	0/ unlimited	list of unreachable route stops.

(optimRouteStepV3) steps

parameter	type	min/max	description
id	string	0/1	route stop ID
x	number	1/1	route stop’s X coordinate
y	number	1/1	route stop’s Y coordinate
duration	long	1/1	duration of a visit, in milliseconds. O by default
effectiveStart	long	1/1	effective start time for a visit, in milliseconds
driveDistanceBefore	int	1/1	drive time before the step, in metres
driveDistanceAfter	int	1/1	drive distance after the step, in metres
driveTimeBefore	long	1/1	drive time before the step, in milliseconds
driveTimeAfter	long	1/1	drive time after the step, in milliseconds
timeWindows/ timeWindow (optimTimeWindow)	long	1/1	Time windows

Time windows (optimTimeWindow)

parameter	type	min/max	description
start	long	1/1	start of the time window, in milliseconds. 0 by default.
end	long	1/1	time window end, in milliseconds. Long.MAX_VALUE by default

List of unreachable route stops (OptimUnreachableStepV3)

parameter	type	min/max	description
id	string	1/1	route stop ID
x	double	1/1	Route stop’s Longitudinal position

parameter	type	min/max	description
y	double	1/1	Route stop’s Latitudinal position

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/optimService?wsdl

Query

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header />
   <soapenv:Body>
      <sch:optimRouteV3>
         <request>
            <origin>
               <x>-1.5291995940651142</x>
               <y>47.19158811606974</y>
               <id>1</id>
               <duration>3600000</duration>
               <effectiveStart>0</effectiveStart>
               <driveDistanceBefore>0</driveDistanceBefore>
               <driveDistanceAfter>0</driveDistanceAfter>
               <driveTimeBefore>0</driveTimeBefore>
               <driveTimeAfter>0</driveTimeAfter>
            </origin>
            <destination>
               <x>-1.5405963750884433</x>
               <y>47.19752774053115</y>
               <id>5</id>
               <duration>3600000</duration>
               <effectiveStart>0</effectiveStart>
               <driveDistanceBefore>0</driveDistanceBefore>
               <driveDistanceAfter>0</driveDistanceAfter>
               <driveTimeBefore>0</driveTimeBefore>
               <driveTimeAfter>0</driveTimeAfter>
            </destination>
            <steps>
               <step>
                  <x>-1.5315788375137436</x>
                  <y>47.209701524818236</y>
                  <id>2</id>
                  <duration>3600000</duration>
                  <effectiveStart>0</effectiveStart>
                  <driveDistanceBefore>0</driveDistanceBefore>
                  <driveDistanceAfter>0</driveDistanceAfter>
                  <driveTimeBefore>0</driveTimeBefore>
                  <driveTimeAfter>0</driveTimeAfter>
                  <timeWindows>
                     <!--Zero or more repetitions:-->
                     <timeWindow>
                        <start>0</start>
                        <end>9223372036854775807</end>
                     </timeWindow>
                  </timeWindows>
               </step>
               <step>
                  <x>-1.5391738246474214</x>
```

```
                    <y>47.193576663578824</y>
                    <id>3</id>
                    <duration>3600000</duration>
                    <effectiveStart>0</effectiveStart>
                    <driveDistanceBefore>0</driveDistanceBefore>
                    <driveDistanceAfter>0</driveDistanceAfter>
                    <driveTimeBefore>0</driveTimeBefore>
                    <driveTimeAfter>0</driveTimeAfter>
                    <timeWindows>
                       <!--Zero or more repetitions:-->
                       <timeWindow>
                          <start>0</start>
                          <end>9223372036854775807</end>
                       </timeWindow>
                    </timeWindows>
                 </step>
                 <step>
                    <x>-1.5562968681202867</x>
                    <y>47.21808294131743</y>
                    <id>4</id>
                    <duration>3600000</duration>
                    <effectiveStart>0</effectiveStart>
                    <driveDistanceBefore>0</driveDistanceBefore>
                    <driveDistanceAfter>0</driveDistanceAfter>
                    <driveTimeBefore>0</driveTimeBefore>
                    <driveTimeAfter>0</driveTimeAfter>
                 </step>
              </steps>
              <srs>epsg:4326</srs>
              <method>time</method>
           </request>
        </sch:optimRouteV3>
     </soapenv:Body>
  </soapenv:Envelope>
```

Response

```
  <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
     <soap:Body>
        <ns2:optimRouteV3Response xmlns:ns2="http://geoconcept.com/gc/schemas">
           <OptimRouteResult>
              <steps>
                 <step>
                    <x>-1.5291995940651142</x>
                    <y>47.19158811606974</y>
                    <id>1</id>
                    <duration>3600000</duration>
                    <effectiveStart>0</effectiveStart>
                    <driveDistanceBefore>0</driveDistanceBefore>
                    <driveDistanceAfter>1779</driveDistanceAfter>
                    <driveTimeBefore>0</driveTimeBefore>
                    <driveTimeAfter>372710</driveTimeAfter>
                    <timeWindows/>
                 </step>
                 <step>
                    <x>-1.5391738246474214</x>
                    <y>47.193576663578824</y>
                    <id>3</id>
                    <duration>3600000</duration>
                    <effectiveStart>0</effectiveStart>
                    <driveDistanceBefore>1779</driveDistanceBefore>
                    <driveDistanceAfter>3486</driveDistanceAfter>
                    <driveTimeBefore>372710</driveTimeBefore>
                    <driveTimeAfter>752870</driveTimeAfter>
                    <timeWindows>
                       <timeWindow>
                          <start>0</start>
                          <end>9223372036854775807</end>
                       </timeWindow>
                    </timeWindows>
                 </step>
                 <step>
                    <x>-1.5562968681202867</x>
                    <y>47.21808294131743</y>
                    <id>4</id>
                    <duration>3600000</duration>
                    <effectiveStart>0</effectiveStart>
                    <driveDistanceBefore>3486</driveDistanceBefore>
                    <driveDistanceAfter>3175</driveDistanceAfter>
                    <driveTimeBefore>752870</driveTimeBefore>
                    <driveTimeAfter>662230</driveTimeAfter>
                    <timeWindows/>
                 </step>
                 <step>
                    <x>-1.5315788375137436</x>
                    <y>47.209701524818236</y>
                    <id>2</id>
                    <duration>3600000</duration>
                    <effectiveStart>0</effectiveStart>
                    <driveDistanceBefore>3175</driveDistanceBefore>
                    <driveDistanceAfter>2713</driveDistanceAfter>
                    <driveTimeBefore>662230</driveTimeBefore>
                    <driveTimeAfter>520090</driveTimeAfter>
                    <timeWindows>
                       <timeWindow>
                          <start>0</start>
                          <end>9223372036854775807</end>
                       </timeWindow>
                    </timeWindows>
                 </step>
                 <step>
                    <x>-1.5405963750884433</x>
                    <y>47.19752774053115</y>
                    <id>5</id>
                    <duration>3600000</duration>
                    <effectiveStart>0</effectiveStart>
                    <driveDistanceBefore>2713</driveDistanceBefore>
                    <driveDistanceAfter>0</driveDistanceAfter>
                    <driveTimeBefore>520090</driveTimeBefore>
                    <driveTimeAfter>0</driveTimeAfter>
                    <timeWindows/>
                 </step>
              </steps>
              <distanceMeters>11153</distanceMeters>
              <durationSeconds>2307</durationSeconds>
           </OptimRouteResult>
        </ns2:optimRouteV3Response>
     </soap:Body>
  </soap:Envelope>
```

REST (POST)

Query

Query

```
http://<server>/<webapp>/api/lbs/optim/route.xml
```

Data (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<optimRouteRequestV3>
   <origin>
      <x>-1.5291995940651142</x>
      <y>47.19158811606974</y>
      <!--Optional:-->
      <id>1</id>
      <timeWindows>
         <timeWindow>
            <start>09:00</start>
            <end>20:00</end>
         </timeWindow>
      </timeWindows>
   </origin>
   <destination>
      <x>-1.5405963750884433</x>
      <y>47.19752774053115</y>
      <!--Optional:-->
      <id>5</id>
   </destination>
   <steps>
      <step>
         <x>-1.5315788375137436</x>
         <y>47.209701524818236</y>
         <id>2</id>
         <timeWindows>
            <timeWindow>
               <start>09:00</start>
               <end>20:00</end>
            </timeWindow>
         </timeWindows>
      </step>
      <step>
         <x>-1.5391738246474214</x>
         <y>47.193576663578824</y>
         <id>3</id>
      </step>
      <step>
         <x>-1.5562968681202867</x>
         <y>47.21808294131743</y>
         <id>4</id>
      </step>
   </steps>
   <method>time</method>
   <srs>epsg:4326</srs>
   <matrixProvider>SMARTROUTING</matrixProvider>
</optimRouteRequestV3>
```

Response

The response is always in UTF-8 format

XML Format

```
<optimRouteResultV3>
   <steps>
      <step>
```

```
            <x>-1.5291995940651142</x>
            <y>47.19158811606974</y>
            <id>1</id>
            <duration>0</duration>
            <effectiveStart>0</effectiveStart>
            <driveDistanceBefore>0</driveDistanceBefore>
            <driveDistanceAfter>1779</driveDistanceAfter>
            <driveTimeBefore>0</driveTimeBefore>
            <driveTimeAfter>372710</driveTimeAfter>
            <timeWindows>
               <timeWindow>
                  <start>0</start>
                  <end>0</end>
               </timeWindow>
            </timeWindows>
         </step>
         <step>
            <x>-1.5391738246474214</x>
            <y>47.193576663578824</y>
            <id>3</id>
            <duration>0</duration>
            <effectiveStart>0</effectiveStart>
            <driveDistanceBefore>1779</driveDistanceBefore>
            <driveDistanceAfter>3486</driveDistanceAfter>
            <driveTimeBefore>372710</driveTimeBefore>
            <driveTimeAfter>752870</driveTimeAfter>
            <timeWindows/>
         </step>
         <step>
            <x>-1.5562968681202867</x>
            <y>47.21808294131743</y>
            <id>4</id>
            <duration>0</duration>
            <effectiveStart>0</effectiveStart>
            <driveDistanceBefore>3486</driveDistanceBefore>
            <driveDistanceAfter>3175</driveDistanceAfter>
            <driveTimeBefore>752870</driveTimeBefore>
            <driveTimeAfter>662230</driveTimeAfter>
            <timeWindows/>
         </step>
         <step>
            <x>-1.5315788375137436</x>
            <y>47.209701524818236</y>
            <id>2</id>
            <duration>0</duration>
            <effectiveStart>0</effectiveStart>
            <driveDistanceBefore>3175</driveDistanceBefore>
            <driveDistanceAfter>2713</driveDistanceAfter>
            <driveTimeBefore>662230</driveTimeBefore>
            <driveTimeAfter>520090</driveTimeAfter>
            <timeWindows>
               <timeWindow>
                  <start>0</start>
                  <end>0</end>
               </timeWindow>
            </timeWindows>
         </step>
         <step>
            <x>-1.5405963750884433</x>
            <y>47.19752774053115</y>
            <id>5</id>
            <duration>0</duration>
            <effectiveStart>0</effectiveStart>
            <driveDistanceBefore>2713</driveDistanceBefore>
            <driveDistanceAfter>0</driveDistanceAfter>
            <driveTimeBefore>520090</driveTimeBefore>
            <driveTimeAfter>0</driveTimeAfter>
            <timeWindows/>
         </step>
      </steps>
      <distanceMeters>11153</distanceMeters>
      <durationSeconds>2307</durationSeconds>
   </optimRouteResultV3>
```

Possible responses

case of an optimization applied successfully

```
<optimRouteResultV3>
   <steps>
      <step>
         <x>-1.5291995940651142</x>
         <y>47.19158811606974</y>
         <id>1</id>
         <duration>0</duration>
         <effectiveStart>0</effectiveStart>
         <driveDistanceBefore>0</driveDistanceBefore>
         <driveDistanceAfter>1779</driveDistanceAfter>
         <driveTimeBefore>0</driveTimeBefore>
         <driveTimeAfter>372710</driveTimeAfter>
         <timeWindows>
            <timeWindow>
               <start>0</start>
               <end>0</end>
            </timeWindow>
         </timeWindows>
      </step>
      <step>
         [...]
      </step>
   </steps>
   <distanceMeters>11153</distanceMeters>
   <durationSeconds>2307</durationSeconds>
</optimRouteResultV3>
```

Case of a problem with the graph: absent file, faulty filepath, etc ⇒ error with faultstring containing the description

```
<soap:Fault>
    <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
    <faultstring>Error in matrix computation
        Error in smartrouting
        datasource is null</faultstring>
</soap:Fault>
```

V2

Parameters / properties

Input

parameter	description	optional	default
origin (optimRouteStepV2)	origin point (id,x,y)	yes	
destination (optimRouteStepV2)	destination point (id,x,y)	yes	
steps (optimRouteStepV2)	points to visit (id,x,y;id,x,y;…) Several slots for visiting times in milliseconds can be specified for each point: (id,x,y,start, finish)	no	
method	shortest (distance) or fastest (time) route	yes	time
matrixProvider	matrix calculation provider globe: as the crow flies on the globe (the coordinates must be in long/lat) euclidean: as the crow flies on the map (the coordinates must be in metres) smartrouting: calculation on the road network by SmartRouting nokia: calculation on the road network via a Nokia service provided: matrix parameter	yes	smartrouting
matrix	time/distance matrix (id1,id2,temps,distance;…)	yes	
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	wgs84
directSpeed	Default speed during utilisation of globe or euclidean matrices	yes	50

(optimRouteStepV2) steps

parameter	type	min/max	description
id	string	0/1	point id
duration	long	1/1	duration of a visit, in milliseconds. O by default
timeWindows/ timeWindow (optimTimeWindow)	long	1/1	Time windows

Time windows (optimTimeWindow)

parameter	type	min/max	description
start	long	1/1	start of the time window, in milliseconds. 0 by default.
end	long	1/1	time window end, in milliseconds. Long.MAX_VALUE by default

Output

parameter	type	min/max	description
steps/step	array (optimRouteStepV2)	0/ unlimited	ordered points to visit
distanceMeters	long	1/1	total journey distance in meters
durationSeconds	long	1/1	total journey time in seconds

(optimRouteStepV2) steps

parameter	type	min/max	description
id	string	0/1	point id
duration	long	1/1	duration of a visit, in milliseconds. O by default
effectiveStart	long	1/1	effective start time for a visit, in milliseconds

parameter	type	min/max	description
driveDistanceBefore	int	1/1	drive time before the step, in metres
driveDistanceAfter	int	1/1	drive distance after the step, in metres
driveTimeBefore	long	1/1	drive time before the step, in milliseconds
driveTimeAfter	long	1/1	drive time after the step, in milliseconds
timeWindows/ timeWindow (optimTimeWindow)	long	1/1	Time windows

Time windows (optimTimeWindow)

parameter	type	min/max	description
start	long	1/1	start of the time window, in milliseconds. 0 by default.
end	long	1/1	time window end, in milliseconds. Long.MAX_VALUE by default

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/optimService?wsdl

Query

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header />
   <soapenv:Body>
      <sch:optimRouteV2>
         <request>
            <origin>
               <x>-1.5291995940651142</x>
               <y>47.19158811606974</y>
               <id>1</id>
               <duration>3600000</duration>
               <effectiveStart>0</effectiveStart>
               <driveDistanceBefore>0</driveDistanceBefore>
               <driveDistanceAfter>0</driveDistanceAfter>
               <driveTimeBefore>0</driveTimeBefore>
               <driveTimeAfter>0</driveTimeAfter>
            </origin>
            <destination>
               <x>-1.5405963750884433</x>
               <y>47.19752774053115</y>
               <id>5</id>
               <duration>3600000</duration>
               <effectiveStart>0</effectiveStart>
               <driveDistanceBefore>0</driveDistanceBefore>
               <driveDistanceAfter>0</driveDistanceAfter>
               <driveTimeBefore>0</driveTimeBefore>
               <driveTimeAfter>0</driveTimeAfter>
            </destination>
            <steps>
               <step>
                  <x>-1.5315788375137436</x>
                  <y>47.209701524818236</y>
                  <id>2</id>
```

```
                    <duration>3600000</duration>
                    <effectiveStart>0</effectiveStart>
                    <driveDistanceBefore>0</driveDistanceBefore>
                    <driveDistanceAfter>0</driveDistanceAfter>
                    <driveTimeBefore>0</driveTimeBefore>
                    <driveTimeAfter>0</driveTimeAfter>
                    <timeWindows>
                       <!--Zero or more repetitions:-->
                       <timeWindow>
                          <start>0</start>
                          <end>9223372036854775807</end>
                       </timeWindow>
                    </timeWindows>
                 </step>
                 <step>
                    <x>-1.5391738246474214</x>
                    <y>47.193576663578824</y>
                    <id>3</id>
                    <duration>3600000</duration>
                    <effectiveStart>0</effectiveStart>
                    <driveDistanceBefore>0</driveDistanceBefore>
                    <driveDistanceAfter>0</driveDistanceAfter>
                    <driveTimeBefore>0</driveTimeBefore>
                    <driveTimeAfter>0</driveTimeAfter>
                    <timeWindows>
                       <!--Zero or more repetitions:-->
                       <timeWindow>
                          <start>0</start>
                          <end>9223372036854775807</end>
                       </timeWindow>
                    </timeWindows>
                 </step>
                 <step>
                    <x>-1.5562968681202867</x>
                    <y>47.21808294131743</y>
                    <id>4</id>
                    <duration>3600000</duration>
                    <effectiveStart>0</effectiveStart>
                    <driveDistanceBefore>0</driveDistanceBefore>
                    <driveDistanceAfter>0</driveDistanceAfter>
                    <driveTimeBefore>0</driveTimeBefore>
                    <driveTimeAfter>0</driveTimeAfter>
                 </step>
              </steps>
              <srs>epsg:4326</srs>
              <method>time</method>
           </request>
        </sch:optimRouteV2>
     </soapenv:Body>
  </soapenv:Envelope>
```

Response

```
  <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
     <soap:Body>
        <ns2:optimRouteV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
           <OptimRouteResult>
              <steps>
                 <step>
                    <x>-1.5291995940651142</x>
                    <y>47.19158811606974</y>
                    <id>1</id>
                    <duration>3600000</duration>
```

```
               <effectiveStart>0</effectiveStart>
               <driveDistanceBefore>0</driveDistanceBefore>
               <driveDistanceAfter>1779</driveDistanceAfter>
               <driveTimeBefore>0</driveTimeBefore>
               <driveTimeAfter>372710</driveTimeAfter>
               <timeWindows/>
            </step>
            <step>
               <x>-1.5391738246474214</x>
               <y>47.193576663578824</y>
               <id>3</id>
               <duration>3600000</duration>
               <effectiveStart>0</effectiveStart>
               <driveDistanceBefore>1779</driveDistanceBefore>
               <driveDistanceAfter>3486</driveDistanceAfter>
               <driveTimeBefore>372710</driveTimeBefore>
               <driveTimeAfter>752870</driveTimeAfter>
               <timeWindows>
                  <timeWindow>
                     <start>0</start>
                     <end>9223372036854775807</end>
                  </timeWindow>
               </timeWindows>
            </step>
            <step>
               <x>-1.5562968681202867</x>
               <y>47.21808294131743</y>
               <id>4</id>
               <duration>3600000</duration>
               <effectiveStart>0</effectiveStart>
               <driveDistanceBefore>3486</driveDistanceBefore>
               <driveDistanceAfter>3175</driveDistanceAfter>
               <driveTimeBefore>752870</driveTimeBefore>
               <driveTimeAfter>662230</driveTimeAfter>
               <timeWindows/>
            </step>
            <step>
               <x>-1.5315788375137436</x>
               <y>47.209701524818236</y>
               <id>2</id>
               <duration>3600000</duration>
               <effectiveStart>0</effectiveStart>
               <driveDistanceBefore>3175</driveDistanceBefore>
               <driveDistanceAfter>2713</driveDistanceAfter>
               <driveTimeBefore>662230</driveTimeBefore>
               <driveTimeAfter>520090</driveTimeAfter>
               <timeWindows>
                  <timeWindow>
                     <start>0</start>
                     <end>9223372036854775807</end>
                  </timeWindow>
               </timeWindows>
            </step>
            <step>
               <x>-1.5405963750884433</x>
               <y>47.19752774053115</y>
               <id>5</id>
               <duration>3600000</duration>
               <effectiveStart>0</effectiveStart>
               <driveDistanceBefore>2713</driveDistanceBefore>
               <driveDistanceAfter>0</driveDistanceAfter>
               <driveTimeBefore>520090</driveTimeBefore>
               <driveTimeAfter>0</driveTimeAfter>
```

```
                    <timeWindows/>
                 </step>
              </steps>
              <distanceMeters>11153</distanceMeters>
              <durationSeconds>2307</durationSeconds>
           </OptimRouteResult>
        </ns2:optimRouteV2Response>
     </soap:Body>
  </soap:Envelope>
```

REST (POST)

Query

Query

```
http://<server>/<webapp>/api/lbs/optim/route.xml
```

Data (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<optimRouteRequestV2>
   <origin>
      <x>-1.5291995940651142</x>
      <y>47.19158811606974</y>
      <!--Optional:-->
      <id>1</id>
      <timeWindows>
         <timeWindow>
            <start>09:00</start>
            <end>20:00</end>
         </timeWindow>
      </timeWindows>
   </origin>
   <destination>
      <x>-1.5405963750884433</x>
      <y>47.19752774053115</y>
      <!--Optional:-->
      <id>5</id>
   </destination>
   <steps>
      <step>
         <x>-1.5315788375137436</x>
         <y>47.209701524818236</y>
         <id>2</id>
         <timeWindows>
            <timeWindow>
               <start>09:00</start>
               <end>20:00</end>
            </timeWindow>
         </timeWindows>
      </step>
      <step>
         <x>-1.5391738246474214</x>
         <y>47.193576663578824</y>
         <id>3</id>
      </step>
      <step>
         <x>-1.5562968681202867</x>
         <y>47.21808294131743</y>
         <id>4</id>
```

```
        </step>
    </steps>
    <method>time</method>
    <srs>epsg:4326</srs>
    <matrixProvider>SMARTROUTING</matrixProvider>
</optimRouteRequestV2>
```

Response

The response is always in UTF-8 format

XML Format

```
<optimRouteResultV2>
    <steps>
        <step>
            <x>-1.5291995940651142</x>
            <y>47.19158811606974</y>
            <id>1</id>
            <duration>0</duration>
            <effectiveStart>0</effectiveStart>
            <driveDistanceBefore>0</driveDistanceBefore>
            <driveDistanceAfter>1779</driveDistanceAfter>
            <driveTimeBefore>0</driveTimeBefore>
            <driveTimeAfter>372710</driveTimeAfter>
            <timeWindows>
                <timeWindow>
                    <start>0</start>
                    <end>0</end>
                </timeWindow>
            </timeWindows>
        </step>
        <step>
            <x>-1.5391738246474214</x>
            <y>47.193576663578824</y>
            <id>3</id>
            <duration>0</duration>
            <effectiveStart>0</effectiveStart>
            <driveDistanceBefore>1779</driveDistanceBefore>
            <driveDistanceAfter>3486</driveDistanceAfter>
            <driveTimeBefore>372710</driveTimeBefore>
            <driveTimeAfter>752870</driveTimeAfter>
            <timeWindows/>
        </step>
        <step>
            <x>-1.5562968681202867</x>
            <y>47.21808294131743</y>
            <id>4</id>
            <duration>0</duration>
            <effectiveStart>0</effectiveStart>
            <driveDistanceBefore>3486</driveDistanceBefore>
            <driveDistanceAfter>3175</driveDistanceAfter>
            <driveTimeBefore>752870</driveTimeBefore>
            <driveTimeAfter>662230</driveTimeAfter>
            <timeWindows/>
        </step>
        <step>
            <x>-1.5315788375137436</x>
            <y>47.209701524818236</y>
            <id>2</id>
            <duration>0</duration>
            <effectiveStart>0</effectiveStart>
            <driveDistanceBefore>3175</driveDistanceBefore>
            <driveDistanceAfter>2713</driveDistanceAfter>
            <driveTimeBefore>662230</driveTimeBefore>
            <driveTimeAfter>520090</driveTimeAfter>
            <timeWindows>
               <timeWindow>
                  <start>0</start>
                  <end>0</end>
               </timeWindow>
            </timeWindows>
         </step>
         <step>
            <x>-1.5405963750884433</x>
            <y>47.19752774053115</y>
            <id>5</id>
            <duration>0</duration>
            <effectiveStart>0</effectiveStart>
            <driveDistanceBefore>2713</driveDistanceBefore>
            <driveDistanceAfter>0</driveDistanceAfter>
            <driveTimeBefore>520090</driveTimeBefore>
            <driveTimeAfter>0</driveTimeAfter>
            <timeWindows/>
         </step>
      </steps>
      <distanceMeters>11153</distanceMeters>
      <durationSeconds>2307</durationSeconds>
   </optimRouteResultV2>
```

Possible responses

case of an optimization applied successfully

```
   <optimRouteResultV2>
      <steps>
         <step>
            <x>-1.5291995940651142</x>
            <y>47.19158811606974</y>
            <id>1</id>
            <duration>0</duration>
            <effectiveStart>0</effectiveStart>
            <driveDistanceBefore>0</driveDistanceBefore>
            <driveDistanceAfter>1779</driveDistanceAfter>
            <driveTimeBefore>0</driveTimeBefore>
            <driveTimeAfter>372710</driveTimeAfter>
            <timeWindows>
               <timeWindow>
                  <start>0</start>
                  <end>0</end>
               </timeWindow>
            </timeWindows>
         </step>
         <step>
            [...]
         </step>
      </steps>
      <distanceMeters>11153</distanceMeters>
      <durationSeconds>2307</durationSeconds>
   </optimRouteResultV2>
```

Case of a problem with the graph: absent file, faulty filepath, etc ⇒ error with faultstring containing the description

```
<soap:Fault>
    <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
    <faultstring>Error in matrix computation
        Error in smartrouting
        datasource is null</faultstring>
</soap:Fault>
```

V1

Parameters / properties

Input

parameter	description	optional	default
origin	origin point (id,x,y)	yes	
destination	destination point (id,x,y)	yes	
steps	points to visit (id,x,y;id,x,y;…) Visiting times in milliseconds can be specified for each point: (id,x,y,start, finish)	no	
method	shortest (distance) or fastest (time) route	yes	time
matrixProvider	matrix calculation provider globe: as the crow flies on the globe (the coordinates must be in long/lat) euclidean: as the crow flies on the map (the coordinates must be in metres) smartrouting: calculation on the road network by SmartRouting nokia: calculation on the road network via a Nokia service provided: matrix parameter	yes	smartrouting
matrix	time/distance matrix (id1,id2,temps,distance;…)	yes	
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	wgs84
directSpeed	Default speed during utilisation of globe or euclidean matrices	yes	50

Output

parameter	type	min/max	description
steps/step	array (optimRouteStep)	0/ unlimited	ordered points to visit
totalDistance	long	1/1	total journey distance in meters
totalTime	long	1/1	total journey time in minutes

Steps (optimRouteStep)

parameter	type	min/max	description
id	string	0/1	point id
duration	long	1/1	duration of a visit, in milliseconds. O by default
timeWindowStart	long	1/1	start of the time window, in milliseconds. 0 by default.
timeWindowEnd	long	1/1	time window end, in milliseconds. Long.MAX_VALUE by default
effectiveStart	long	1/1	effective start time for a visit, in milliseconds
driveDistanceBefore	int	1/1	drive time before the step, in metres
driveDistanceAfter	int	1/1	drive distance after the step, in metres

parameter	type	min/max	description
driveTimeBefore	long	1/1	drive time before the step, in milliseconds
driveTimeAfter	long	1/1	drive time after the step, in milliseconds

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/optimService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:optimRoute>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <origin>
               <x>-1.55306</x>
               <y>47.21812</y>
               <!--Optional:-->
               <id>0</id>
               <duration>0</duration>
               <timeWindowStart>0</timeWindowStart>
               <timeWindowEnd>0</timeWindowEnd>
               <effectiveStart>0</effectiveStart>
               <driveDistanceBefore>0</driveDistanceBefore>
               <driveDistanceAfter>0</driveDistanceAfter>
               <driveTimeBefore>0</driveTimeBefore>
               <driveTimeAfter>0</driveTimeAfter>
            </origin>
            <!--Optional:-->
            <destination>
               <x>-2.2164099</x>
               <y>47.2806206</y>
               <!--Optional:-->
               <id>1</id>
               <duration>0</duration>
               <timeWindowStart>0</timeWindowStart>
               <timeWindowEnd>0</timeWindowEnd>
               <effectiveStart>0</effectiveStart>
               <driveDistanceBefore>0</driveDistanceBefore>
               <driveDistanceAfter>0</driveDistanceAfter>
               <driveTimeBefore>0</driveTimeBefore>
               <driveTimeAfter>0</driveTimeAfter>
            </destination>
            <!--Optional:-->
            <steps>
               <!--Zero or more repetitions:-->
               <step>
                  <x>-2.02616</x>
                  <y>47.2835503</y>
                  <!--Optional:-->
               <id>2</id>
               <duration>0</duration>
               <timeWindowStart>0</timeWindowStart>
               <timeWindowEnd>0</timeWindowEnd>
```

```
                <effectiveStart>0</effectiveStart>
                <driveDistanceBefore>0</driveDistanceBefore>
                <driveDistanceAfter>0</driveDistanceAfter>
                <driveTimeBefore>0</driveTimeBefore>
                <driveTimeAfter>0</driveTimeAfter>
                </step>
             </steps>
             <!--Optional:-->
             <srs>wgs84</srs>
             <!--Optional:-->
             <method>time</method>
             <!--Optional:-->
             <matrixProvider>SMARTROUTING</matrixProvider>
             <!--Optional:-->
             <matrix>
                <!--Optional:-->
                <elements>
                   <!--Zero or more repetitions:-->
                   <element>
                      <!--Optional:-->
                      <origin></origin>
                      <!--Optional:-->
                      <destination></destination>
                      <time>1</time>
                      <distance>1</distance>
                   </element>
                </elements>
             </matrix>
             <directSpeed>0</directSpeed>
          </request>
       </sch:optimRoute>
    </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:optimRouteResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <OptimRouteResult>
            <steps>
               <step>
                  <x>-1.55306</x>
                  <y>47.21812</y>
                  <id>0</id>
                  <duration>0</duration>
                  <timeWindowStart>0</timeWindowStart>
                  <timeWindowEnd>0</timeWindowEnd>
                  <effectiveStart>0</effectiveStart>
                  <driveDistanceBefore>0</driveDistanceBefore>
                  <driveDistanceAfter>47085</driveDistanceAfter>
                  <driveTimeBefore>0</driveTimeBefore>
                  <driveTimeAfter>3139890</driveTimeAfter>
               </step>
               <step>
                  <x>-2.02616</x>
                  <y>47.2835503</y>
                  <id>2</id>
                  <duration>0</duration>
                  <timeWindowStart>0</timeWindowStart>
                  <timeWindowEnd>0</timeWindowEnd>
                  <effectiveStart>0</effectiveStart>
                  <driveDistanceBefore>47085</driveDistanceBefore>
```

```
                    <driveDistanceAfter>22124</driveDistanceAfter>
                    <driveTimeBefore>3139890</driveTimeBefore>
                    <driveTimeAfter>1488050</driveTimeAfter>
                 </step>
                 <step>
                    <x>-2.2164099</x>
                    <y>47.2806206</y>
                    <id>1</id>
                    <duration>0</duration>
                    <timeWindowStart>0</timeWindowStart>
                    <timeWindowEnd>0</timeWindowEnd>
                    <effectiveStart>0</effectiveStart>
                    <driveDistanceBefore>22124</driveDistanceBefore>
                    <driveDistanceAfter>0</driveDistanceAfter>
                    <driveTimeBefore>1488050</driveTimeBefore>
                    <driveTimeAfter>0</driveTimeAfter>
                 </step>
              </steps>
              <totalDistance>69209</totalDistance>
              <totalTime>77</totalTime>
           </OptimRouteResult>
        </ns2:optimRouteResponse>
     </soap:Body>
  </soap:Envelope>
```

REST (POST)

Query

Query

```
http://<server>/<webapp>/api/lbs/optim/route.xml
```

Data (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<optimRouteRequest>
        <steps>
                <step>
                        <x>-1.55306</x>
                        <y>47.21812</y>
                        <id>0</id>
                </step>
                <step>
                        <x>-2.2164099</x>
                        <y>47.2806206</y>
                        <id>1</id>
                </step>
                <step>
                        <x>-2.02616</x>
                        <y>47.2835503</y>
                        <id>2</id>
                </step>
        </steps>
        <method>time</method>
        <srs>wgs84</srs>
        <matrixProvider>SMARTROUTING</matrixProvider>
</optimRouteRequest>
```

Response

The response is always in UTF-8 format

XML Format

```
<optimRouteResult>
   <steps>
      <step>
         <x>-2.02616</x>
         <y>47.2835503</y>
         <id>2</id>
         <duration>0</duration>
         <timeWindowStart>0</timeWindowStart>
         <timeWindowEnd>9223372036854775807</timeWindowEnd>
         <effectiveStart>0</effectiveStart>
         <driveDistanceBefore>0</driveDistanceBefore>
         <driveDistanceAfter>22124</driveDistanceAfter>
         <driveTimeBefore>0</driveTimeBefore>
         <driveTimeAfter>1488050</driveTimeAfter>
      </step>
      <step>
         <x>-2.2164099</x>
         <y>47.2806206</y>
         <id>1</id>
         <duration>0</duration>
         <timeWindowStart>0</timeWindowStart>
         <timeWindowEnd>9223372036854775807</timeWindowEnd>
         <effectiveStart>0</effectiveStart>
         <driveDistanceBefore>22124</driveDistanceBefore>
         <driveDistanceAfter>63139</driveDistanceAfter>
         <driveTimeBefore>1488050</driveTimeBefore>
         <driveTimeAfter>2992570</driveTimeAfter>
      </step>
      <step>
         <x>-1.55306</x>
         <y>47.21812</y>
         <id>0</id>
         <duration>0</duration>
         <timeWindowStart>0</timeWindowStart>
         <timeWindowEnd>9223372036854775807</timeWindowEnd>
         <effectiveStart>0</effectiveStart>
         <driveDistanceBefore>63139</driveDistanceBefore>
         <driveDistanceAfter>0</driveDistanceAfter>
         <driveTimeBefore>2992570</driveTimeBefore>
         <driveTimeAfter>0</driveTimeAfter>
      </step>
   </steps>
   <totalDistance>85263</totalDistance>
   <totalTime>74</totalTime>
</optimRouteResult>
```

REST (GET)

Query

JSON query

```
http://<server>/<webapp>/api/lbs/optim/route.json?
steps=0,-1.55306,47.21812;1,-2.2164099,47.2806206;2,-2.02616,47.2835503&method=time&srs=wgs84
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/optim/route.json?
steps=0,-1.55306,47.21812;1,-2.2164099,47.2806206;2,-2.02616,47.2835503&method=time&srs=wgs84&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/optim/route.xml?
steps=0,-1.55306,47.21812;1,-2.2164099,47.2806206;2,-2.02616,47.2835503&method=time&srs=wgs84
```

Response

The response is always in UTF-8 format

JSON format

```
{
   "steps":    [
            {
         "x": -2.02616, "y": 47.2835503, "id": "2",
         "duration": 0, "timeWindowStart": 0, "timeWindowEnd": 9223372036854775807, "effectiveStart": 0,
         "driveDistanceBefore": 0, "driveDistanceAfter": 22124, "driveTimeBefore": 0, "driveTimeAfter":
 1488050
      },
            {
         "x": -2.2164099, "y": 47.2806206, "id": "1",
         "duration": 0, "timeWindowStart": 0, "timeWindowEnd": 9223372036854775807, "effectiveStart": 0,
         "driveDistanceBefore": 22124, "driveDistanceAfter": 63139, "driveTimeBefore": 1488050,
 "driveTimeAfter": 2992570
      },
            {
         "x": -1.55306, "y": 47.21812, "id": "0",
         "duration": 0, "timeWindowStart": 0, "timeWindowEnd": 9223372036854775807, "effectiveStart": 0,
         "driveDistanceBefore": 63139, "driveDistanceAfter": 0, "driveTimeBefore": 2992570,
 "driveTimeAfter": 0
      }
   ],
   "totalDistance": 85263,
   "totalTime": 74
}
```

JSON-P format

```
myCallback({
   "steps":    [
            {
         "x": -2.02616, "y": 47.2835503, "id": "2",
         "duration": 0, "timeWindowStart": 0, "timeWindowEnd": 9223372036854775807, "effectiveStart": 0,
         "driveDistanceBefore": 0, "driveDistanceAfter": 22124, "driveTimeBefore": 0, "driveTimeAfter":
 1488050
      },
            {
         "x": -2.2164099, "y": 47.2806206, "id": "1",
         "duration": 0, "timeWindowStart": 0, "timeWindowEnd": 9223372036854775807, "effectiveStart": 0,
         "driveDistanceBefore": 22124, "driveDistanceAfter": 63139, "driveTimeBefore": 1488050,
 "driveTimeAfter": 2992570
      },
            {
         "x": -1.55306, "y": 47.21812, "id": "0",
         "duration": 0, "timeWindowStart": 0, "timeWindowEnd": 9223372036854775807, "effectiveStart": 0,
         "driveDistanceBefore": 63139, "driveDistanceAfter": 0, "driveTimeBefore": 2992570,
 "driveTimeAfter": 0
```

```
      }
    ],
    "totalDistance": 85263,
    "totalTime": 74
});
```

XML Format

```
<optimRouteResult>
   <steps>
      <step>
         <x>-2.02616</x>
         <y>47.2835503</y>
         <id>2</id>
         <duration>0</duration>
         <timeWindowStart>0</timeWindowStart>
         <timeWindowEnd>9223372036854775807</timeWindowEnd>
         <effectiveStart>0</effectiveStart>
         <driveDistanceBefore>0</driveDistanceBefore>
         <driveDistanceAfter>22124</driveDistanceAfter>
         <driveTimeBefore>0</driveTimeBefore>
         <driveTimeAfter>1488050</driveTimeAfter>
      </step>
      <step>
         <x>-2.2164099</x>
         <y>47.2806206</y>
         <id>1</id>
         <duration>0</duration>
         <timeWindowStart>0</timeWindowStart>
         <timeWindowEnd>9223372036854775807</timeWindowEnd>
         <effectiveStart>0</effectiveStart>
         <driveDistanceBefore>22124</driveDistanceBefore>
         <driveDistanceAfter>63139</driveDistanceAfter>
         <driveTimeBefore>1488050</driveTimeBefore>
         <driveTimeAfter>2992570</driveTimeAfter>
      </step>
      <step>
         <x>-1.55306</x>
         <y>47.21812</y>
         <id>0</id>
         <duration>0</duration>
         <timeWindowStart>0</timeWindowStart>
         <timeWindowEnd>9223372036854775807</timeWindowEnd>
         <effectiveStart>0</effectiveStart>
         <driveDistanceBefore>63139</driveDistanceBefore>
         <driveDistanceAfter>0</driveDistanceAfter>
         <driveTimeBefore>2992570</driveTimeBefore>
         <driveTimeAfter>0</driveTimeAfter>
      </step>
   </steps>
   <totalDistance>85263</totalDistance>
   <totalTime>74</totalTime>
</optimRouteResult>
```

Possible responses

case of an optimization applied successfully

```
<ns2:optimRouteResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
 <OptimRouteResult>
    <steps>
```

```
        <step>
           <x>-1.55306</x>
           <y>47.21812</y>
           <id>0</id>
           <duration>0</duration>
           <timeWindowStart>0</timeWindowStart>
           <timeWindowEnd>0</timeWindowEnd>
           <effectiveStart>0</effectiveStart>
           <driveDistanceBefore>0</driveDistanceBefore>
           <driveDistanceAfter>47085</driveDistanceAfter>
           <driveTimeBefore>0</driveTimeBefore>
           <driveTimeAfter>3139890</driveTimeAfter>
        </step>
        <step>
             ...
        </step>
     </steps>
     <totalDistance>69209</totalDistance>
     <totalTime>77</totalTime>
  </OptimRouteResult>
</ns2:optimRouteResponse>
```

Case of a problem with the graph: absent file, faulty filepath, etc ⇒ error with faultstring containing the description

```
<soap:Fault>
    <faultcode xmlns:ns1="geoconcept.com">ns1:8</faultcode>
    <faultstring>Error in matrix computation
        Error in smartrouting
        datasource is null</faultstring>
</soap:Fault>
```

FAQ

1. Can you force a start and/or arrival step in the calculation of a route?

   Yes, you just have to use the origin and / or destination parameters respectively. The journey between the steps route stops is optimised taking into account any operational constraints, and without taking into account any logistical constraint (departure from home, arrival in a depot, ..;).

2. How can you specify, for each step, a visit duration and a visit time window?

   using the duration parameter (to indicate the duration set aside for a visit) and the timeWindowStart and timeWindowEnd parameters (to specify the start and finish time for a visit).

Calculating Isochrones/Isodistances

(fr) Cette page n’est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

This web service calculates an isochrone/isodistance from a point and returns the geometry of the zone calculated. It bases its calculations on the configured graph, the name of which has been specified in the administration interface of Geoconcept Web.

Availability

This web service is available at all times with Geoconcept Web and one road network.

Version change

Earlier versions of the web service are conserved in Geoconcept Web to ensure compatibility with previous software developments. We recommend using the most recent version.

Changes in relation to V4

- Addition of "timeOut" and "computeOptions" parameters.

Changes in relation to V3

- Addition of "avoidArea" and "configName" parameters.

Changes in relation to V2

- Addition of "startDateTime" and "snapMethod" parameters

V4

Parameters / properties

Input

parameter	description	optional	default
id	Isochrone identifier	yes	
location	Start (or arrival point, if the `reverse` parameter is set to True). The coordinates are separated by , characters.	no	
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	epsg:4326
graphName (depreciated)	Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
profileId (depreciated)	Vehicle identifier (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
profileName (depreciated)	Vehicle profile (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
exclusions	List of restriction rules to use, separated by the *;* character (Example: Toll, Tunnel, Bridge)	yes	
method	"time" for isochrone or "distance" for isodistance	no	time
time	Maximum access time, in seconds	yes	
distance	Maximum access distance, in metres	yes	
reverse	if *true*, the `location` is considered as an arrival point	yes	false
smoothing	Smoothing	yes	false
holes	Display holes in the result zone (the geometry returned is more voluminous when this parameter is set to *true*)	yes	false
startDateTime	Departure Date and Time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00)	yes	

parameter	description	optional	default
	for a departure on 21 Januray 2014, at 9.00am in Paris. Caution: the + character may be misinterpreted by browsers, and so in this case should be replaced by *%2B*.		
snapMethod	Method for snapping to the graph - standard: at the nearest connectable road section - extended: via restricted road sections (pedestrian routes….) - nearest: only to the nearest road section - unrestricted: without any restriction rules	yes	standard
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84: `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))` NOTE: WKT geometries must be closed.	yes	
configName	Configuration name to use (defined in Geoconcept Web - Administration / Tools / Road networks) This replaces the use of graphName, profileId and profileName	yes	
timeOut	Time out for the calculation (in milliseconds)	yes	
computeOptions	List of options for the calculation,separated by *;* characters - trafficPatterns: uses routing statistics (you will need to supply a value for the startDateTime parameter and use a graph that includes traffic patterns information) - speedPattern (M18): uses a *speed pattern* as defined in the SmartRoutingVehicles.xml file, located in the *`‘<GEOCONCEPT_WEB_HOME>’'\smartrouting\jee\smartrouting\conf\* folder. Use as follows: "speedPattern:slow-speed" - length (M18): maximum authorised length in centimeters (you need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "length:950" - width (M18): maximum authorised width in centimeters (You will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "width:255" - height (M18): maximum authorised height in centimeters (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "height:360" - weight (M18): maximum authorised weight in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weight:18000" - axles (M18): maximum authorised number of axles (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "axles:2" - weightPerAxle (M18): maximum authorised weight per axle in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weightPerAxle:9000" - snapSpeed: snap-to-graph speed in kilometers per hour. Use as follows: "snapSpeed:10"	yes	

(M18) Available from version M18 and later versions of graphs supplied by GEOCONCEPT SAS.

Output

Itinerary (isochroneResult)

parameter	type	min/max	description
id	string	0/1	Isochrone identifier
location	string	0/1	Start (or Finish if the **`reverse`** is set to true).
srs	string	0/1	projection
time	string	0/1	Maximum access time, in seconds
distance	string	0/1	Maximum access distance, in metres
wktGeometry	string	0/1	Geometry of the isochrone, in wkt format

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/isochroneService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:isochroneV4>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <id>1</id>
            <location>
               <x>-1.557189</x>
               <y>47.217122</y>
            </location>
            <!--Optional:-->
            <srs></srs>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <method>distance</method>
            <!--Optional:-->
            <time></time>
            <!--Optional:-->
            <distance>5000</distance>
            <!--Optional:-->
            <reverse></reverse>
            <!--Optional:-->
            <smoothing></smoothing>
            <!--Optional:-->
            <holes></holes>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <snapMethod></snapMethod>
            <!--Optional:-->
            <avoidArea></avoidArea>
            <!--Optional:-->
            <configName></configName>
            <!--Optional:-->
            <computeOptions></computeOptions>
            <!--Optional:-->
            <timeOut></timeOut>
            </request>
      </sch:isochroneV4>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:isochroneV4Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <IsochroneResult>
            <status>OK</status>
            <id>1</id>
            <location>-1.557189,47.217122</location>
            <srs />
            <distance>5000</distance>
            <wktGeometry>POLYGON ((-1.544603 47.179044, -1.545261 47.179044, [...], -1.544603 47.179044))</
wktGeometry>
         </IsochroneResult>
      </ns2:isochroneV4Response>
   </soap:Body>
</soap:Envelope>
```

REST (POST)

Query

Query

```
http://<server>/<webapp>/api/lbs/isochrone/v4.xml
```

Data (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<isochroneRequestV4>
  <id></id>
  <location>
    <x>-1.557189</x>
    <y>47.217122</y>
  </location>
  <srs></srs>
  <graphName></graphName>
  <profileId></profileId>
  <profileName></profileName>
  <exclusions>
    <exclusion></exclusion>
    <exclusion></exclusion>
    <!--...more "exclusion" elements...-->
  </exclusions>
```

```
    <method></method>
    <time>50</time>
    <distance></distance>
    <reverse></reverse>
    <smoothing></smoothing>
    <holes></holes>
    <startDateTime></startDateTime>
    <snapMethod></snapMethod>
    <avoidArea></avoidArea>
    <configName></configName>
</isochroneRequestV4>
```

Response

The response is always coded in UTF-8.

XML format

```
<isochroneResultV4>
   <status>OK</status>
   <id>1</id>
   <location>-1.557189,47.217122</location>
   <srs/>
   <time>50</time>
   <wktGeometry>POLYGON ((-1.556864 47.216487, -1.556864 47.216948, [...]))</wktGeometry>
</isochroneResultV4>
```

REST (GET)

Query

JSON query

```
http://<server>/<webapp>/api/lbs/isochrone/v4.json?
location=-1.557189,47.217122&method=distance&distance=5000
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/isochrone/v4.json?
location=-1.557189,47.217122&method=distance&distance=5000&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/isochrone/v4.xml?
location=-1.557189,47.217122&method=distance&distance=2000&holes=true
```

Response

The response is always coded in UTF-8.

JSON format

```
{
   "message": null,
   "status": "OK",
   "id": null,
   "location": "-1.557189,47.217122",
   "srs": "epsg:4326",
   "time": null,
   "distance": "5000",
   "wktGeometry": "POLYGON ((-1.545312 47.178178, -1.545312 47.180418, ...))"
}
```

JSON-P format

```
myCallback(
    {
       "message": null,
       "status": "OK",
       "id": null,
       "location": "-1.557189,47.217122",
       "srs": "EPSG:4326",
       "time": null,
       "distance": "5000",
       "wktGeometry": "POLYGON ((-1.545312 47.178178, -1.545312 47.180418, ...))"
    }
);
```

XML format

```
<isochroneResultV4>
   <status>OK</status>
   <location>-1.557189,47.217122</location>
   <srs>EPSG:4326</srs>
   <distance>2000</distance>
   <wktGeometry>POLYGON ((-1.546926 47.202309, -1.546926 47.202752, ...))</wktGeometry>
</isochroneResultV4>
```

Possible responses

Case of an isochrone/isodistance that has been found (isochroneResultV2/status is OK)

```
<isochroneResultV4>
   <status>OK</status>
   <location>-1.557189,47.217122</location>
   <srs>EPSG:4326</srs>
   <distance>2000</distance>
   <wktGeometry>POLYGON ((-1.546926 47.202309, -1.546926 47.202752, ...))</wktGeometry>
</isochroneResultV4>
```

Case whereby method = distance and distance is not supplied (isochroneResult/status is ERROR)

```
<isochroneResultV4>
    <message>distance parameter must not be null or 0 !</message>
    <status>ERROR</status>
</isochroneResultV4>
```

Case whereby method + time are not supplied (isochroneResult/status is ERROR)

```
<isochroneResultV4>
    <message>time parameter must not be null or 0 !</message>
    <status>ERROR</status>
</isochroneResultV4>
```

Case whereby the location tag is missing (isochroneResult/status is ERROR)

```
<isochroneResultV4>
    <message>Location must be not null</message>
    <status>ERROR</status>
</isochroneResultV4>
```

Case whereby the start point is supplied with errors (isochroneResult/status is ERROR)

```
<isochroneResultV4>
   <message>Location point must have 2 components separated with a ,</message>
   <status>ERROR</status>
</isochroneResultV4>
```

Case of a snap-to-graph error (serviceResult/status is ERROR)

```
<serviceResult>
        <message>
                ServiceException: Error in isochrone computation Error in smartrouting Failed to execute
 calculateConcentricReachableAreas com.geoconcept.smartrouting.SmartRoutingNativeException: failed to
 connect isochrone orign { 52.324230, 48.803256, 0.000000 } failed to connect isochrone orign { 52.324230,
 48.803256, 0.000000 }
        </message>
        <status>ERROR</status>
</serviceResult>
```

Case of a problem with the graph: absent file, faulty filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in isochrone computation
        Error in smartrouting
        datasource is null
    </message>
   <status>ERROR</status>
</serviceResult>
```

V3

Parameters / properties

Input

parameter	description	optional	default
id	Isochrone identifier	yes	
location	Start (or arrival point, if the **`reverse`** parameter is set to True). The coordinates are separated by , characters.	no	
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	epsg:4326
graphName	Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
profileId	Vehicle identifier (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
profileName	Vehicle profile (saved under vehicle profiles)	yes	

parameter	description	optional	default
	This parameter is omitted if the configName parameter is used.		
exclusions	List of restriction rules to use, separated by the , or *;* character (Example: "Toll", "Tunnel", "Bridge")	yes	
method	"time" for isochrone or "distance" for isodistance	no	time
time	Maximum access time, in seconds	yes	
distance	Maximum access distance, in metres	yes	
reverse	if *true*, the **`location`** is considered as an arrival point	yes	false
smoothing	Smoothing	yes	false
holes	Display holes in the result zone (the geometry returned is more voluminous when this parameter is set to *true*)	yes	false
startDateTime	Departure Date and Time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a departure on 21 Januray 2014, at 9.00am in Paris. Caution: the + character may be misinterpreted by browsers, and so in this case should be replaced by *%2B*.	yes	
snapMethod	Method for snapping to the graph - standard: at the nearest connectable road section - extended: via restricted road sections (pedestrian routes….) - nearest: only to the nearest road section - unrestricted: without any restriction rules	yes	standard
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84: `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))` NOTE: WKT geometries must be closed.	yes	
configName	Configuration name to use (defined in Geoconcept Web - Administration / Tools / Road networks) This replaces the use of graphName, profileId and profileName	yes	

Output

Itinerary (isochroneResult)

parameter	type	min/max	description
id	string	0/1	Isochrone identifier
location	string	0/1	Start (or Finish if the **`reverse`** is set to true).
srs	string	0/1	projection
time	string	0/1	Maximum access time, in seconds
distance	string	0/1	Maximum access distance, in metres
wktGeometry	string	0/1	Geometry of the isochrone, in wkt format

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/isochroneService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:isochroneV3>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <id>1</id>
            <location>
               <x>-1.557189</x>
               <y>47.217122</y>
            </location>
            <!--Optional:-->
            <srs></srs>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <method>distance</method>
            <!--Optional:-->
            <time></time>
            <!--Optional:-->
            <distance>5000</distance>
            <!--Optional:-->
            <reverse></reverse>
            <!--Optional:-->
            <smoothing></smoothing>
            <!--Optional:-->
            <holes></holes>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <snapMethod></snapMethod>
            <!--Optional:-->
            <avoidArea></avoidArea>
            <!--Optional:-->
            <configName></configName>
         </request>
      </sch:isochroneV3>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:isochroneV3Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <IsochroneResult>
            <status>OK</status>
            <id>1</id>
            <location>-1.557189,47.217122</location>
            <srs/>
            <distance>5000</distance>
            <wktGeometry>POLYGON ((-1.544603 47.179044, -1.545261 47.179044, [...]))</wktGeometry>
         </IsochroneResult>
      </ns2:isochroneV3Response>
   </soap:Body>
</soap:Envelope>
```

REST (POST)

Query

Query

```
http://<server>/<webapp>/api/lbs/isochrone/v3.xml
```

Data (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<isochroneRequestV3>
  <id></id>
  <location>
    <x>-1.557189</x>
    <y>47.217122</y>
  </location>
  <srs></srs>
  <graphName></graphName>
  <profileId></profileId>
  <profileName></profileName>
  <exclusions>
    <exclusion></exclusion>
    <exclusion></exclusion>
    <!--...more "exclusion" elements...-->
  </exclusions>
  <method></method>
  <time>50</time>
  <distance></distance>
  <reverse></reverse>
  <smoothing></smoothing>
  <holes></holes>
  <startDateTime></startDateTime>
  <snapMethod></snapMethod>
  <avoidArea></avoidArea>
  <configName></configName>
</isochroneRequestV3>
```

Response

The response is always coded in UTF-8.

XML format

```
<isochroneResultV3>
   <status>OK</status>
   <id>1</id>
   <location>-1.557189,47.217122</location>
   <srs/>
   <time>50</time>
   <wktGeometry>POLYGON ((-1.556864 47.216487, -1.556864 47.216948, [...]))</wktGeometry>
</isochroneResultV3>
```

REST (GET)

Query

JSON query

```
http://<server>/<webapp>/api/lbs/isochrone/v3.json?
location=-1.557189,47.217122&method=distance&distance=5000
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/isochrone/v3.json?
location=-1.557189,47.217122&method=distance&distance=5000&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/isochrone/v3.xml?
location=-1.557189,47.217122&method=distance&distance=2000&holes=true
```

Response

The response is always coded in UTF-8.

JSON format

```
{
   "message": null,
   "status": "OK",
   "id": null,
   "location": "-1.557189,47.217122",
   "srs": "EPSG:4326",
   "time": null,
   "distance": "5000",
   "wktGeometry": "POLYGON ((-1.545312 47.178178, -1.545312 47.180418, ...))"
}
```

JSON-P format

```
myCallback(
    {
       "message": null,
       "status": "OK",
       "id": null,
       "location": "-1.557189,47.217122",
       "srs": "EPSG:4326",
       "time": null,
       "distance": "5000",
       "wktGeometry": "POLYGON ((-1.545312 47.178178, -1.545312 47.180418, ...))"
    }
);
```

XML format

```
<isochroneResultV3>
   <status>OK</status>
   <location>-1.557189,47.217122</location>
   <srs>EPSG:4326</srs>
   <distance>2000</distance>
   <wktGeometry>POLYGON ((-1.546926 47.202309, -1.546926 47.202752, ...))</wktGeometry>
</isochroneResultV3>
```

Possible responses

Case of an isochrone/isodistance that has been found (isochroneResultV2/status is OK)

```
<isochroneResultV3>
   <status>OK</status>
   <location>-1.557189,47.217122</location>
   <srs>EPSG:4326</srs>
   <distance>2000</distance>
   <wktGeometry>POLYGON ((-1.546926 47.202309, -1.546926 47.202752, ...))</wktGeometry>
</isochroneResultV3>
```

Case whereby method = distance and distance is not supplied (isochroneResult/status is ERROR)

```
<isochroneResultV3>
    <message>distance parameter must not be null or 0 !</message>
    <status>ERROR</status>
</isochroneResultV3>
```

Case whereby method + time are not supplied (isochroneResult/status is ERROR)

```
<isochroneResultV3>
    <message>time parameter must not be null or 0 !</message>
    <status>ERROR</status>
</isochroneResultV3>
```

Case whereby the location tag is missing (isochroneResult/status is ERROR)

```
<isochroneResultV3>
    <message>Location must be not null</message>
    <status>ERROR</status>
</isochroneResultV3>
```

Case whereby the start point is supplied with errors (isochroneResult/status is ERROR)

```
<isochroneResultV3>
   <message>Location point must have 2 components separated with a ,</message>
   <status>ERROR</status>
</isochroneResultV3>
```

Case of a snap-to-graph error (serviceResult/status is ERROR)

```
<serviceResult>
        <message>
                ServiceException: Error in isochrone computation Error in smartrouting Failed to execute
 calculateConcentricReachableAreas com.geoconcept.smartrouting.SmartRoutingNativeException: failed to
```

```
  connect isochrone orign { 52.324230, 48.803256, 0.000000 } failed to connect isochrone orign { 52.324230,
  48.803256, 0.000000 }
         </message>
         <status>ERROR</status>
</serviceResult>
```

Case of a problem with the graph: absent file, faulty filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in isochrone computation
        Error in smartrouting
        datasource is null
    </message>
   <status>ERROR</status>
</serviceResult>
```

V2

Parameters / properties

Input

parameter	description	optional	default
id	Isochrone identifier	yes	
location	Start (or arrival point, if the **`reverse`** parameter is set to True). The coordinates are separated by , characters.	no	
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	epsg:4326
graphName	Name of the graph to use	yes	
profileId	Vehicle identifier (saved under vehicle profiles) to use	yes	
profileName	Vehicle profile (saved under vehicle profiles) to use	yes	
exclusions	List of restriction rules to use, separated by the , or ; character (Example: "Toll", "Tunnel", "Bridge")	yes	
method	"time" for isochrone or "distance" for isodistance	no	time
time	Maximum access time, in seconds	yes	
distance	Maximum access distance, in metres	yes	
reverse	if *true*, the **`location`** is considered as an arrival point	yes	false
smoothing	Smoothing	yes	false
holes	Display holes in the result zone (the geometry returned is more voluminous when this parameter is set to *true*)	yes	false
startDateTime	Departure Date and Time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a departure on 21 Januray 2014, at 9.00am in Paris	yes	
snapMethod	Method for snapping to the graph - standard: at the nearest connectable road section - extended: via restricted road sections (pedestrian routes….) - nearest: only to the nearest road section - unrestricted: without any restriction rules	yes	standard

Output

Itinerary (isochroneResult)

parameter	type	min/max	description
id	string	0/1	Isochrone identifier
location	string	0/1	Start (or Finish if the **`reverse`** is set to true).
srs	string	0/1	projection
time	string	0/1	Maximum access time, in seconds
distance	string	0/1	Maximum access distance, in metres
wktGeometry	string	0/1	Geometry of the isochrone, in wkt format

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/isochroneService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:isochroneV2>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <id>1</id>
            <location>
               <x>-1.557189</x>
               <y>47.217122</y>
            </location>
            <!--Optional:-->
            <srs></srs>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <method>distance</method>
            <!--Optional:-->
            <time></time>
            <!--Optional:-->
            <distance>5000</distance>
            <!--Optional:-->
            <reverse></reverse>
            <!--Optional:-->
            <smoothing></smoothing>
            <!--Optional:-->
            <holes></holes>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <snapMethod></snapMethod>
         </request>
      </sch:isochroneV2>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:isochroneV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <IsochroneResult>
            <status>OK</status>
            <id/>
            <location>-1.557189,47.217122</location>
            <srs/>
            <distance>5000</distance>
            <wktGeometry>POLYGON ((-1.544603 47.179044, -1.545261 47.179044, [...]))</wktGeometry>
         </IsochroneResult>
      </ns2:isochroneV2Response>
   </soap:Body>
</soap:Envelope>
```

REST (POST)

Query

Query

```
http://<server>/<webapp>/api/lbs/isochrone/v2.xml
```

Data (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<isochroneRequestV2>
  <id></id>
  <location>
    <x>-1.557189</x>
    <y>47.217122</y>
  </location>
  <srs></srs>
  <graphName></graphName>
  <profileId></profileId>
  <profileName></profileName>
  <exclusions>
    <exclusion></exclusion>
    <exclusion></exclusion>
    <!--...more "exclusion" elements...-->
  </exclusions>
  <method></method>
  <time>50</time>
  <distance></distance>
  <reverse></reverse>
  <smoothing></smoothing>
  <holes></holes>
  <startDateTime></startDateTime>
  <snapMethod></snapMethod>
```

```
</isochroneRequestV2>
```

Response

The response is always coded in UTF-8.

XML format

```
<isochroneResultV2>
   <status>OK</status>
   <id>1</id>
   <location>-1.557189,47.217122</location>
   <srs/>
   <time>50</time>
   <wktGeometry>POLYGON ((-1.556864 47.216487, -1.556864 47.216948, [...]))</wktGeometry>
</isochroneResultV2>
```

REST (GET)

Query

JSON query

```
http://<server>/<webapp>/api/lbs/isochrone/v2.json?
location=-1.557189,47.217122&method=distance&distance=5000
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/isochrone/v2.json?
location=-1.557189,47.217122&method=distance&distance=5000&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/isochrone/v2.xml?
location=-1.557189,47.217122&method=distance&distance=2000&holes=true
```

Response

The response is always coded in UTF-8.

JSON format

```
{
   "message": null,
   "status": "OK",
   "id": null,
   "location": "-1.557189,47.217122",
   "srs": "EPSG:4326",
   "time": null,
   "distance": "5000",
   "wktGeometry": "POLYGON ((-1.545312 47.178178, -1.545312 47.180418, ...))"
}
```

JSON-P format

```
myCallback(
    {
       "message": null,
       "status": "OK",
       "id": null,
       "location": "-1.557189,47.217122",
       "srs": "EPSG:4326",
       "time": null,
       "distance": "5000",
       "wktGeometry": "POLYGON ((-1.545312 47.178178, -1.545312 47.180418, ...))"
    }
);
```

XML format

```
<isochroneResultV2>
   <status>OK</status>
   <location>-1.557189,47.217122</location>
   <srs>EPSG:4326</srs>
   <distance>2000</distance>
   <wktGeometry>POLYGON ((-1.546926 47.202309, -1.546926 47.202752, ...))</wktGeometry>
</isochroneResultV2>
```

Possible responses

Case of an isochrone/isodistance that has been found (isochroneResultV2/status is OK)

```
<isochroneResultV2>
   <status>OK</status>
   <location>-1.557189,47.217122</location>
   <srs>EPSG:4326</srs>
   <distance>2000</distance>
   <wktGeometry>POLYGON ((-1.546926 47.202309, -1.546926 47.202752, ...))</wktGeometry>
</isochroneResultV2>
```

Case whereby method = distance and distance is not supplied (isochroneResult/status is ERROR)

```
<isochroneResultV2>
    <message>distance parameter must not be null or 0 !</message>
    <status>ERROR</status>
</isochroneResultV2>
```

Case whereby method + time are not supplied (isochroneResult/status is ERROR)

```
<isochroneResultV2>
    <message>time parameter must not be null or 0 !</message>
    <status>ERROR</status>
</isochroneResultV2>
```

Case whereby the location tag is missing (isochroneResult/status is ERROR)

```
<isochroneResultV2>
    <message>Location must be not null</message>
    <status>ERROR</status>
</isochroneResultV2>
```

Case whereby the start point is supplied with errors (isochroneResult/status is ERROR)

```
<isochroneResultV2>
   <message>Location point must have 2 components separated with a ,</message>
   <status>ERROR</status>
</isochroneResultV2>
```

Case of a snap-to-graph error (serviceResult/status is ERROR)

```
<serviceResult>
        <message>
                ServiceException: Error in isochrone computation Error in smartrouting Failed to execute
 calculateConcentricReachableAreas com.geoconcept.smartrouting.SmartRoutingNativeException: failed to
 connect isochrone orign { 52.324230, 48.803256, 0.000000 } failed to connect isochrone orign { 52.324230,
 48.803256, 0.000000 }
        </message>
        <status>ERROR</status>
</serviceResult>
```

Case of a problem with the graph: absent file, faulty filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in isochrone computation
        Error in smartrouting
        datasource is null
    </message>
   <status>ERROR</status>
</serviceResult>
```

V1

Parameters / properties

Input

parameter	description	optional	default
id	Isochrone identifier	yes	
location	Start (or arrival point, if the **`reverse`** parameter is set to True). The coordinates are separated by , characters.	no	
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	epsg:4326
graphName	Name of the graph to use	yes	
profileId	Vehicle identifier (saved under vehicle profiles) to use	yes	
profileName	Vehicle profile (saved under vehicle profiles) to use	yes	
exclusions	List of restriction rules to use, separated by the , or ; character (Example: "Toll", "Tunnel", "Bridge")	yes	
method	"time" for isochrone or "distance" for isodistance	no	time
time	Maximum access time, in seconds	yes	
distance	Maximum access distance, in metres	yes	
reverse	if *true*, the **`location`** is considered as an arrival point	yes	false
smoothing	Smoothing	yes	false

parameter	description	optional	default
holes	Display holes in the result zone (the geometry returned is more voluminous when this parameter is set to *true*)	yes	false

Output

Itinerary (isochroneResult)

parameter	type	min/max	description
id	string	0/1	Isochrone identifier
location	string	0/1	Start (or Finish if the **`reverse`** is set to true).
srs	string	0/1	projection
time	string	0/1	Maximum access time, in seconds
distance	string	0/1	Maximum access distance, in metres
wktGeometry	string	0/1	Geometry of the isochrone, in wkt format

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/isochroneService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:isochrone>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <id/>
            <location>
               <x>-1.557189</x>
               <y>47.217122</y>
            </location>
            <!--Optional:-->
            <srs/>
            <!--Optional:-->
            <graphName/>
            <!--Optional:-->
            <profileId/>
            <!--Optional:-->
            <profileName/>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion/>
            </exclusions>
            <!--Optional:-->
            <method>distance</method>
            <!--Optional:-->
            <distance>5000</distance>
            <!--Optional:-->
            <reverse/>
            <!--Optional:-->
            <smoothing/>
            <!--Optional:-->
            <holes/>
         </request>
      </sch:isochrone>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:isochroneResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <IsochroneResult>
            <status>OK</status>
            <id/>
            <location>-1.557189,47.217122</location>
            <srs/>
            <distance>5000</distance>
            <wktGeometry>POLYGON ((-1.544603 47.179044, -1.545261 47.179044, [...]))</wktGeometry>
         </IsochroneResult>
      </ns2:isochroneResponse>
   </soap:Body>
</soap:Envelope>
```

REST (POST)

Query

Query

```
http://<server>/<webapp>/api/lbs/isochrone.xml
```

Data (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<isochroneRequest>
  <id>1</id>
  <location>
    <x>-1.557189</x>
    <y>47.217122</y>
  </location>
  <srs></srs>
  <graphName></graphName>
  <profileId></profileId>
  <profileName></profileName>
  <exclusions>
    <exclusion></exclusion>
    <exclusion></exclusion>
    <!--...more "exclusion" elements...-->
  </exclusions>
  <method></method>
  <time>50</time>
  <distance></distance>
  <reverse></reverse>
  <smoothing></smoothing>
  <holes></holes>
</isochroneRequest>
```

Response

The response is always coded in UTF-8.

XML format

```
<isochroneResult>
   <status>OK</status>
   <id>1</id>
   <location>-1.557189,47.217122</location>
   <srs/>
   <time>50</time>
   <wktGeometry>POLYGON ((-1.556864 47.216487, -1.556864 47.216948, ...))</wktGeometry>
</isochroneResult>
```

REST (GET)

Query

JSON query

```
http://<server>/<webapp>/api/lbs/isochrone.json?location=-1.557189,47.217122&method=distance&distance=5000
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/isochrone.json?
location=-1.557189,47.217122&method=distance&distance=5000&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/isochrone.xml??
location=-1.557189,47.217122&method=distance&distance=2000&holes=true
```

Response

The response is always coded in UTF-8.

JSON format

```
{
   "message": null,
   "status": "OK",
   "id": null,
   "location": "-1.557189,47.217122",
   "srs": "EPSG:4326",
   "time": null,
   "distance": "5000",
   "wktGeometry": "POLYGON ((-1.545312 47.178178, -1.545312 47.180418, ...))"
}
```

JSON-P format

```
myCallback(
    {
       "message": null,
       "status": "OK",
       "id": null,
       "location": "-1.557189,47.217122",
       "srs": "EPSG:4326",
       "time": null,
       "distance": "5000",
       "wktGeometry": "POLYGON ((-1.545312 47.178178, -1.545312 47.180418, ...))"
    }
);
```

XML format

```
<isochroneResult>
   <status>OK</status>
   <location>-1.557189,47.217122</location>
   <srs>EPSG:4326</srs>
   <distance>2000</distance>
   <wktGeometry>POLYGON ((-1.546926 47.202309, -1.546926 47.202752, ...))</wktGeometry>
</isochroneResult>
```

Possible responses

Case of an isochrone/isodistance that has been found (isochroneResult/status is OK)

```
<isochroneResult>
        <status>OK</status>
        <location>2.32423,48.803256</location>
        <srs>epsg:4326</srs>
        <distance>2000,000000</distance>
        <wktGeometry>
                POLYGON ((2.317525987661223 48.786657549807174, 2.317525987661223 48.787103131779986, ...))
        </wktGeometry>
</isochroneResult>
```

Case where the start point is empty (isochroneResult/status is ERROR)

```
<isochroneResult>
    <message>Location must be not null</message>
    <status>ERROR</status>
</isochroneResult>
```

Case whereby the start point is supplied with errors (isochroneResult/status is ERROR)

```
<isochroneResult>
   <message>Location point must have 2 components separated with a ,</message>
   <status>ERROR</status>
</isochroneResult>
```

Case of a typing error

```
<isochroneResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
</isochroneResult>
```

Case of a snap-to-graph error (isochroneResult/status is ERROR)

```
<serviceResult>
        <message>
                ServiceException: Error in isochrone computation Error in smartrouting Failed to execute
 calculateConcentricReachableAreas com.geoconcept.smartrouting.SmartRoutingNativeException: failed to
 connect isochrone orign { 52.324230, 48.803256, 0.000000 } failed to connect isochrone origin { 52.324230,
 48.803256, 0.000000 }
        </message>
        <status>ERROR</status>
</serviceResult>
```

Case of a problem with the graph: file missing, faulty filepath, etc … (isochroneResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in isochrone computation
        Error in smartrouting
        datasource is null
    </message>
   <status>ERROR</status>
</serviceResult>
```

FAQ

1. Is it possible to give priority to either journey time or distance?

   Yes, by changing the method to "time" `method` for isochrone, or "distance" for isodistance

2. Can I use aliases instead of .siti file names to call a datasource?

   Yes, refer to details in the FAQ of the reverse geocoding Web Service.

3. How can I use route statistics?

   Check that the graph does contain these statistics and use these two parameters: `computeOptions` with a value of *trafficPatterns* and `startDateTime` to specify the departure date/time.

4. How do you perform an isochrone/isodistance calculation without any tolls?

   If the *Toll* constraint has been included in the graph, place an exclusion in `exclusions` :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:isochroneV2>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <id>1</id>
            <location>
               <x>-1.557189</x>
               <y>47.217122</y>
            </location>
            <!--Optional:-->
            <srs></srs>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
```

```
               <!--Optional:-->
               <exclusions>
                  <!--Zero or more repetitions:-->
                  <exclusions>Toll</exclusions>
               </exclusions>
               <!--Optional:-->
               <method>distance</method>
               <!--Optional:-->
               <time></time>
               <!--Optional:-->
               <distance>5000</distance>
               <!--Optional:-->
               <reverse></reverse>
               <!--Optional:-->
               <smoothing></smoothing>
               <!--Optional:-->
               <holes></holes>
               <!--Optional:-->
               <startDateTime></startDateTime>
               <!--Optional:-->
               <snapMethod></snapMethod>
            </request>
         </sch:isochroneV2>
      </soapenv:Body>
   </soapenv:Envelope>
```

5. What are Speed Patterns? How can I use them?

   In order to propose journey times that are as accurate as possible and reflect changing traffic conditions, graphs supplied by GEOCONCEPT SAS include, from version M18 upwards, 5 speed profiles (Speed Patterns) for cars and lorries, to take into account the different levels of congestion during the course of one day:

   - standard *normal-speed* corresponds to the speed at a time when the roads have an average congestion level (eg 11.00am)
   - night *fast-speed* corresponds to a very fluid traffic situation, often experienced at night-time (3.00am)
   - congested *slow-speed* corresponds to the times when traffic is densest (8.00am)
   - rush hour *very-slow-speed* corresponds to times when traffic is densest in urban and built-up areas, and is slower than the speed above (8.00am)
   - default *default* corresponds to speeds averaged out over an entire day

   To use them, you need to pass to parameter, when calling the web service, the `computeOptions` parameter with the *speedPattern* option and specify the value of the Speed Pattern requested, without forgetting a vehicle profile
   Example:

+

```
&computeOptions=speedPattern:fast-speed&profileId=1
```

How to use Heavy Goods Vehicle attributes?

The graph must include the attributes for Heavy Goods Vehicles (as standard in the graphs supplied by GEOCONCEPT SAS from version M18 upwards) and either calculate an itinerary

using a vehicle profile using restrictions (cf. the catalogue of vehicles, editable, defined in the SmartRoutingVehicles.xml file, stored in the folder `‘*<GEOCONCEPT_WEB_HOME>”\smartrouting \jee\smartrouting\conf\* ), or overwrite the call to the web service by using the **`computeOptions`** parameter with the options *length*, *width*, *height*, *weight*, *axles* and/or *weightPerAxle*.
Example for a journey with a vehicle 4.5 metres high:

```
&computeOptions=height:450
```

Matrix calculation

(fr) Cette page n’est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

This web service calculates a route matrix for a series of points, and returns a distance matrix. Calculations are based on a configured graph, the name of which has been specified in the Geoconcept Web administration interface. See also, compact matrix calculation web service.

Availability

This web service is available at all times when a road network or graph is present with Geoconcept Web.

Version change

Earlier versions of the web service are conserved in Geoconcept Web to ensure compatibility with previous developments. We recommend using the most recent version.

Changes in relation to V4

- Addition of "timeOut" and "computeOptions" parameters.
- Addition in the "nodes" snapMethod of snap to the nearest nodes.

Changes in relation to V3

- Addition of the notion of node, which is faster, to snap to graph nodes rather than to geographic coordinates. Addition of the following items: "originNodes", "destinationNodes" and "nodes" in the snapMethod.
- Addition of the "maxCost" parameter

Changes in relation to V2

- Addition of the "snapMethod" parameter
- Addition of the "startDateTime" parameter

- Deletion of the "RejectFlags" parameter, replaced by "exclusions"
- The "distance" and "duration" parameters have been renamed "distanceMeters" and "durationSeconds" respectively

V4

Parameters / properties

Input

parameter	description	optional	default
srs	projection (ESPG code such as epsg:4326 or wgs84)	yes	
origins	List of coordinates of points of origin. The longitude and latitude coordinates are separated by the ,, character	yes *	
originNodes	List of origin ids nodes. ids nodes are separated by , characters. Care: a physical node does not have the same ID in another graph.	yes *	
destinations	List of coordinates of destination points. The latitude and longitude cooordinates are separated by the , character.	yes *	
destinationNodes	List of destination ids nodes. ids nodes are separated by , characters. Care: a physical node does not have the same ID in another graph.	yes *	
graphName (depreciated)	Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
method	shortest (distance) or fastest (time) route	yes	time
profileId (depreciated)	Vehicle identifier (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
profileName (depreciated)	Vehicle profile (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
exclusions	List of restriction rules to use, separated by the *;* character (Example: Toll, Tunnel, Bridge)	yes	
startDateTime	Departure date and time (format ISO8601: local time). Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a start on 21 January 2014, at 9.00am in Paris. Caution: the + character may be misinterpreted by browsers, and so it is best replaced by *%2B*.	yes	
snapMethod	Snap to graph method - standard: to the nearest connectable road section - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without any restriction rules - nodes: Snap directly to nodes supplied by the originNode, destinationNode and waypointNodes parameters, or, if these parameters have not been set, to the nearest nodes sourced by the origin, destination and waypoint parameters	yes	standard
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84 : `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))`	yes	

parameter	description	optional	default
	Care: WKT geometries must be closed.		
configName	Name of the configuration to use (defined in Geoconcept Web - Administration / Tools / Road graphs) This replaces use of graphName, profileId and profileName	yes	
computeOptions	List of options for the calculation,separated by *;* characters - trafficPatterns: uses routing statistics (you will need to supply a value for the startDateTime parameter and use a graph that includes traffic patterns information) - speedPattern (M18): uses a *speed pattern* as defined in the SmartRoutingVehicles.xml file, located in the `*'<GEOCONCEPT_WEB_HOME>''\smartrouting\jee\smartrouting\conf\* folder. Use as follows: "speedPattern:slow-speed" - length (M18): maximum authorised length in centimeters (you need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "length:950" - width (M18): maximum authorised width in centimeters (You will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "width:255" - height (M18): maximum authorised height in centimeters (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "height:360" - weight (M18): maximum authorised weight in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weight:18000" - axles (M18): maximum authorised number of axles (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "axles:2" - weightPerAxle (M18): maximum authorised weight per axle in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weightPerAxle:9000" - snapSpeed: snap-to-graph speed in kilometers per hour. Use as follows: "snapSpeed:10"	yes	
maxCost	Maximum cost not to exceed in the results -1: no maximum cost to take into account 0: take the default value defined in the SmartRouting Server configuration if not: value in metres if method=distance or in seconds if method=time	yes	
timeOut	Time out for the calculation (in milliseconds)	yes	

(*) At least one of the two pairs of parameters origins/destinations or originNodes/destinationsNodes must have values assigned.

(M18) Available from version M18 and later versions of graphs supplied by GEOCONCEPT SAS.

Output

Matrix (matrixResultV3)

parameter	type	min/max	description
origins/origin (or origins in JSON / JSON-P)	string (or array of origins/origin in JSON / JSON-P)	0/ unlimited	origin positions.

parameter	type	min/max	description
destinations/destination (or destinations in JSON / JSON-P)	string (or array of destinations/destination in JSON / JSON-P)	0/ unlimited	destination positions.
rows/row (or rows in JSON / JSON-P)	matrixRowV3 (or array of rows/row (matrixRow) in JSON / JSON-P)	0/ unlimited	Matrix line

Matrix line (matrixRowV3)

parameter	type	min/max	description
cells/cell (or cells in JSON / JSON-P)	matrixCellV3 (or array of cells/cell (matrixCellV3) in JSON / JSON-P)	0/ unlimited	Matrix cell

Matrix cell (matrixCellV3)

parameter	type	min/max	description
distanceMeters	double	1/1	Matrix cell distance (in metres)
durationSeconds	double	1/1	Matrix cell duration (in seconds)
status	string	0/1	Matrix cell status: - OK: the itinerary has been found for this cell - KO: no itinerary found for this cell

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/matrixService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:matrixV4>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <srs>epsg:27572</srs>
            <!--Optional:-->
            <origins>
               <!--1 or more repetitions:-->
               <origin>
                  <x>315846.96</x>
                  <y>2254268.35</y>
               </origin>
                <origin>
                              <x>313584.77</x>
                  <y>2251648.97</y>
               </origin>
            </origins>
            <!--Optional:-->
            <originNodes>
               <!--1 or more repetitions:-->
               <originNode></originNode>
            </originNodes>
            <!--Optional:-->
            <destinations>
               <!--1 or more repetitions:-->
               <destination>
                  <x>321442.89</x>
                  <y>2251013.98</y>
               </destination>
                <destination>
                  <x>318982.27</x>
                  <y>2248315.22</y>
               </destination>
            </destinations>
            <!--Optional:-->
            <destinationNodes>
               <!--1 or more repetitions:-->
               <destinationNode></destinationNode>
            </destinationNodes>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <method>time</method>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <snapMethod></snapMethod>
            <!--Optional:-->
            <avoidArea></avoidArea>
            <!--Optional:-->
            <configName></configName>
            <!--Optional:-->
            <computeOptions></computeOptions>
            <!--Optional:-->
            <timeOut></timeOut>
                        <!--Optional:-->
                        <maxCost>0</maxCost>
         </request>
      </sch:matrixV4>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:matrixV4Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <MatrixResult>
            <status>OK</status>
```

```
               <origins>
                  <origin>315846.96,2254268.35</origin>
                  <origin>313584.77,2251648.97</origin>
               </origins>
               <destinations>
                  <destination>321442.89,2251013.98</destination>
                  <destination>318982.27,2248315.22</destination>
               </destinations>
               <srs>epsg:27572</srs>
               <rows>
                  <row>
                     <cells>
                        <cell>
                           <distanceMeters>8109.0</distanceMeters>
                           <durationSeconds>564.66</durationSeconds>
                           <status>OK</status>
                        </cell>
                        <cell>
                           <distanceMeters>11249.39</distanceMeters>
                           <durationSeconds>816.38</durationSeconds>
                           <status>OK</status>
                        </cell>
                     </cells>
                  </row>
                  <row>
                     <cells>
                        <cell>
                           <distanceMeters>11736.89</distanceMeters>
                           <durationSeconds>856.77</durationSeconds>
                           <status>OK</status>
                        </cell>
                        <cell>
                           <distanceMeters>7636.69</distanceMeters>
                           <durationSeconds>526.1</durationSeconds>
                           <status>OK</status>
                        </cell>
                     </cells>
                  </row>
               </rows>
            </MatrixResult>
         </ns2:matrixV4Response>
      </soap:Body>
   </soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/matrix/v4.json?
origins=315846.96,2254268.35;313584.77,2251648.97&destinations=321442.89,2251013.98;318982.27,2248315.22&srs=epsg:27572&me
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/matrix/v4.json?
origins=315846.96,2254268.35;313584.77,2251648.97&destinations=321442.89,2251013.98;318982.27,2248315.22&srs=epsg:27572&me
```

XML query

```
http://<server>/<webapp>/api/lbs/matrix/v4.xml?
origins=315846.96,2254268.35;313584.77,2251648.97&destinations=321442.89,2251013.98;318982.27,2248315.22&srs=epsg:27572&metho
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message":null,
   "status":"OK",
   "origins":[
      "315846.96,2254268.35",
      "313584.77,2251648.97"
   ],
   "destinations":[
      "321442.89,2251013.98",
      "318982.27,2248315.22"
   ],
   "srs":"epsg:27572",
   "rows":[
      {
         "cells":[
            {
               "distanceMeters":8109.0,
               "durationSeconds":656.74,
               "status":"OK"
            },
            {
               "distanceMeters":11249.37,
               "durationSeconds":888.57,
               "status":"OK"
            }
         ]
      },
      {
         "cells":[
            {
               "distanceMeters":11736.89,
               "durationSeconds":959.9,
               "status":"OK"
            },
            {
               "distanceMeters":7636.67,
               "durationSeconds":609.35,
               "status":"OK"
            }
         ]
      }
   ]
}
```

JSON-P format

```
MyCallBack({
   "message":null,
   "status":"OK",
   "origins":[
      "315846.96,2254268.35",
      "313584.77,2251648.97"
   ],
   "destinations":[
      "321442.89,2251013.98",
      "318982.27,2248315.22"
   ],
   "srs":"epsg:27572",
   "rows":[
      {
         "cells":[
            {
               "distanceMeters":8109.0,
               "durationSeconds":656.74,
               "status":"OK"
            },
            {
               "distanceMeters":11249.37,
               "durationSeconds":888.57,
               "status":"OK"
            }
         ]
      },
      {
         "cells":[
            {
               "distanceMeters":11736.89,
               "durationSeconds":959.9,
               "status":"OK"
            },
            {
               "distanceMeters":7636.67,
               "durationSeconds":609.35,
               "status":"OK"
            }
         ]
      }
   ]
});
```

XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<matrixResultV4>
   <status>OK</status>
   <origins>
      <origin>315846.96,2254268.35</origin>
      <origin>313584.77,2251648.97</origin>
   </origins>
   <destinations>
      <destination>321442.89,2251013.98</destination>
      <destination>318982.27,2248315.22</destination>
   </destinations>
   <srs>epsg:27572</srs>
   <rows>
      <row>
         <cells>
            <cell>
               <distanceMeters>8109.0</distanceMeters>
               <durationSeconds>656.74</durationSeconds>
               <status>OK</status>
            </cell>
            <cell>
               <distanceMeters>11249.37</distanceMeters>
```

```
                <durationSeconds>888.57</durationSeconds>
                <status>OK</status>
            </cell>
         </cells>
      </row>
      <row>
         <cells>
            <cell>
               <distanceMeters>11736.89</distanceMeters>
               <durationSeconds>959.9</durationSeconds>
               <status>OK</status>
            </cell>
            <cell>
               <distanceMeters>7636.67</distanceMeters>
               <durationSeconds>609.35</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
   </rows>
</matrixResultV4>
```

Possible responses

Case of an itinerary that has been found (matrixResultV3/status is OK)

```
<?xml version="1.0" encoding="UTF-8"?>
<matrixResultV4>
   <status>OK</status>
   <origins>
      <origin>315846.96,2254268.35</origin>
      <origin>313584.77,2251648.97</origin>
   </origins>
   <destinations>
      <destination>321442.89,2251013.98</destination>
      <destination>318982.27,2248315.22</destination>
   </destinations>
   <srs>epsg:27572</srs>
   <rows>
      <row>
         <cells>
            <cell>
               <distanceMeters>8109.0</distanceMeters>
               <durationSeconds>656.74</durationSeconds>
               <status>OK</status>
            </cell>
            <cell>
               <distanceMeters>11249.37</distanceMeters>
               <durationSeconds>888.57</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
      <row>
         <cells>
            <cell>
               <distanceMeters>11736.89</distanceMeters>
               <durationSeconds>959.9</durationSeconds>
               <status>OK</status>
            </cell>
            <cell>
               <distanceMeters>7636.67</distanceMeters>
```

```
                <durationSeconds>609.35</durationSeconds>
                <status>OK</status>
             </cell>
          </cells>
       </row>
    </rows>
</matrixResultV4>
```

Case of forgotten origin and destination specification(matrixResultV3/status is ERROR)

```
<matrixResultV4>
   <message>Origins and destinations must be not null</message>
   <status>ERROR</status>
</matrixResultV4>
```

Case of a faulty type (serviceResult/status is ERROR)

```
<serviceResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
</serviceResult>
```

Case of a snap-to-graph error (matrixResultV3/status is OK) / (cell/status est KO)

```
<matrixResultV4>
   <status>OK</status>
   <origins>
      <origin>-0.36147,49.18392</origin>
      <origin>7.26132,43.70629</origin>
   </origins>
   <destinations>
      <destination>146.08821,48.43253</destination>
      <destination>2.34878,48.86473</destination>
   </destinations>
   <rows>
      <row>
         <cells>
            <cell>
               <distanceMeters>0.0</distanceMeters>
               <durationSeconds>0.0</durationSeconds>
               <status>KO</status>
            </cell>
            <cell>
               <distanceMeters>234196.77000000002</distanceMeters>
               <durationSeconds>9451.57</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
      <row>
         <cells>
            <cell>
               <distanceMeters>0.0</distanceMeters>
               <durationSeconds>0.0</durationSeconds>
               <status>KO</status>
            </cell>
            <cell>
               <distanceMeters>930934.64</distanceMeters>
               <durationSeconds>31773.440000000002</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
   </rows>
</matrixResultV4>
```

Case of a problem with the graph: absent file, faulty filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in matrix computation
Error in smartrouting
datasource is null</message>
   <status>ERROR</status>
</serviceResult>
```

V3

Parameters / properties

Input

parameter	description	optional	default
srs	projection (ESPG code such as epsg:4326 or wgs84)	yes	
origins	List of coordinates of points of origin. The longitude and latitude coordinates are separated by the ,, character	yes *	
originNodes	List of origin ids nodes. ids nodes are separated by , characters. Care: a physical node does not have the same ID in another graph.	yes *	
destinations	List of coordinates of destination points. The latitude and longitude cooordinates are separated by the , character.	yes *	
destinationNodes	List of destination ids nodes. ids nodes are separated by , characters. Care: a physical node does not have the same ID in another graph.	yes *	
graphName	Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
method	shortest (distance) or fastest (time) route	yes	time
profileId	Vehicle identifier (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
profileName	Vehicle profile (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
exclusions	List of restriction rules to use, separated by the , or ; character (Example: "Toll", "Tunnel", "Bridge")	yes	
startDateTime	Departure date and time (format ISO8601: local time). Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a start on 21 January 2014, at 9.00am in Paris. Caution: the + character may be misinterpreted by browsers, and so it is best replaced by *%2B*.	yes	
snapMethod	Snap-to-graph method - standard: to the nearest connectable road section - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without any restriction rules - nodes: snap directly to nodes indicated by locationNode and the *node* parameters of resources	yes	standard

parameter	description	optional	default
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84 : `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))` Care: WKT geometries must be closed.	yes	
configName	Name of the configuration to use (defined in Geoconcept Web - Administration / Tools / Road graphs) This replaces use of graphName, profileId and profileName	yes	
maxCost	Maximum cost not to exceed in the results -1: no maximum cost to take into account 0: take the default value defined in the SmartRouting Server configuration if not: value in metres if method=distance or in seconds if method=time	yes	

(*) At least one of the two pairs of parameters origins/destinations or originNodes/destinationsNodes must have values assigned.

Output

Matrix (matrixResultV3)

parameter	type	min/max	description
origins/origin (or origins in JSON / JSON-P)	string (or array of origins/origin in JSON / JSON-P)	0/ unlimited	origin positions.
destinations/destination (or destinations in JSON / JSON-P)	string (or array of destinations/destination in JSON / JSON-P)	0/ unlimited	destination positions.
rows/row (or rows in JSON / JSON-P)	matrixRowV3 (or array of rows/row (matrixRow) in JSON / JSON-P)	0/ unlimited	Matrix line

Matrix line (matrixRowV3)

parameter	type	min/max	description
cells/cell (or cells in JSON / JSON-P)	matrixCellV3 (or array of cells/cell (matrixCellV3) in JSON / JSON-P)	0/ unlimited	Matrix cell

Matrix cell (matrixCellV3)

parameter	type	min/max	description
distanceMeters	double	1/1	Matrix cell distance (in metres)
durationSeconds	double	1/1	Matrix cell duration (in seconds)
status	string	0/1	Matrix cell status: - OK: the itinerary has been found for this cell - KO: no itinerary found for this cell

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/matrixService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:matrixV3>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <srs>epsg:27572</srs>
            <!--Optional:-->
            <origins>
               <!--1 or more repetitions:-->
               <origin>
                  <x>315846.96</x>
                  <y>2254268.35</y>
               </origin>
                <origin>
                              <x>313584.77</x>
                  <y>2251648.97</y>
               </origin>
            </origins>
            <!--Optional:-->
            <originNodes>
               <!--1 or more repetitions:-->
               <originNode></originNode>
            </originNodes>
            <!--Optional:-->
            <destinations>
               <!--1 or more repetitions:-->
               <destination>
                  <x>321442.89</x>
                  <y>2251013.98</y>
               </destination>
                <destination>
                  <x>318982.27</x>
                  <y>2248315.22</y>
               </destination>
            </destinations>
            <!--Optional:-->
            <destinationNodes>
               <!--1 or more repetitions:-->
               <destinationNode></destinationNode>
            </destinationNodes>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <method>time</method>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <snapMethod></snapMethod>
            <!--Optional:-->
            <avoidArea></avoidArea>
            <!--Optional:-->
            <configName></configName>
            <!--Optional:-->
            <maxCost>0</maxCost>
         </request>
      </sch:matrixV3>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:matrixV3Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <MatrixResult>
            <status>OK</status>
            <origins>
               <origin>315846.96,2254268.35</origin>
               <origin>313584.77,2251648.97</origin>
            </origins>
            <destinations>
               <destination>321442.89,2251013.98</destination>
               <destination>318982.27,2248315.22</destination>
            </destinations>
            <srs>epsg:27572</srs>
            <rows>
               <row>
                  <cells>
                     <cell>
                        <distanceMeters>8109.0</distanceMeters>
                        <durationSeconds>564.66</durationSeconds>
                        <status>OK</status>
                     </cell>
                     <cell>
                        <distanceMeters>11249.37</distanceMeters>
                        <durationSeconds>816.38</durationSeconds>
                        <status>OK</status>
                     </cell>
                  </cells>
               </row>
               <row>
                  <cells>
                     <cell>
                        <distanceMeters>11736.89</distanceMeters>
                        <durationSeconds>856.77</durationSeconds>
                        <status>OK</status>
                     </cell>
                     <cell>
                        <distanceMeters>7636.67</distanceMeters>
                        <durationSeconds>526.1</durationSeconds>
                        <status>OK</status>
                     </cell>
                  </cells>
               </row>
            </rows>
```

```
        </MatrixResult>
      </ns2:matrixV3Response>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/matrix/v3.json?
origins=315846.96,2254268.35;313584.77,2251648.97&destinations=321442.89,2251013.98;318982.27,2248315.22&srs=epsg:27572&metho
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/matrix/v3.json?
origins=315846.96,2254268.35;313584.77,2251648.97&destinations=321442.89,2251013.98;318982.27,2248315.22&srs=epsg:27572&metho
```

XML query

```
http://<server>/<webapp>/api/lbs/matrix/v3.xml?
origins=315846.96,2254268.35;313584.77,2251648.97&destinations=321442.89,2251013.98;318982.27,2248315.22&srs=epsg:27572&metho
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message":null,
   "status":"OK",
   "origins":[
      "315846.96,2254268.35",
      "313584.77,2251648.97"
   ],
   "destinations":[
      "321442.89,2251013.98",
      "318982.27,2248315.22"
   ],
   "srs":"epsg:27572",
   "rows":[
      {
         "cells":[
            {
               "distanceMeters":8109.0,
               "durationSeconds":656.74,
               "status":"OK"
            },
            {
               "distanceMeters":11249.37,
               "durationSeconds":888.57,
               "status":"OK"
            }
         ]
      },
      {
         "cells":[
            {
               "distanceMeters":11736.89,
               "durationSeconds":959.9,
               "status":"OK"
            },
            {
               "distanceMeters":7636.67,
               "durationSeconds":609.35,
               "status":"OK"
            }
         ]
      }
   ]
}
```

JSON-P format

```
MyCallBack({
   "message":null,
   "status":"OK",
   "origins":[
      "315846.96,2254268.35",
      "313584.77,2251648.97"
   ],
   "destinations":[
      "321442.89,2251013.98",
      "318982.27,2248315.22"
   ],
   "srs":"epsg:27572",
   "rows":[
      {
         "cells":[
            {
               "distanceMeters":8109.0,
               "durationSeconds":656.74,
               "status":"OK"
            },
            {
               "distanceMeters":11249.37,
               "durationSeconds":888.57,
               "status":"OK"
            }
         ]
      },
      {
         "cells":[
            {
               "distanceMeters":11736.89,
               "durationSeconds":959.9,
               "status":"OK"
            },
            {
               "distanceMeters":7636.67,
               "durationSeconds":609.35,
               "status":"OK"
            }
         ]
      }
   ]
});
```

XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<matrixResultV3>
   <status>OK</status>
   <origins>
      <origin>315846.96,2254268.35</origin>
      <origin>313584.77,2251648.97</origin>
   </origins>
   <destinations>
      <destination>321442.89,2251013.98</destination>
      <destination>318982.27,2248315.22</destination>
   </destinations>
   <srs>epsg:27572</srs>
   <rows>
      <row>
         <cells>
            <cell>
               <distanceMeters>8109.0</distanceMeters>
               <durationSeconds>656.74</durationSeconds>
               <status>OK</status>
            </cell>
            <cell>
               <distanceMeters>11249.37</distanceMeters>
               <durationSeconds>888.57</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
      <row>
         <cells>
            <cell>
               <distanceMeters>11736.89</distanceMeters>
               <durationSeconds>959.9</durationSeconds>
               <status>OK</status>
            </cell>
            <cell>
               <distanceMeters>7636.67</distanceMeters>
               <durationSeconds>609.35</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
   </rows>
</matrixResultV3>
```

Possible responses

Case of an itinerary that has been found (matrixResultV3/status is OK)

```
<?xml version="1.0" encoding="UTF-8"?>
<matrixResultV3>
   <status>OK</status>
   <origins>
      <origin>315846.96,2254268.35</origin>
      <origin>313584.77,2251648.97</origin>
   </origins>
   <destinations>
      <destination>321442.89,2251013.98</destination>
      <destination>318982.27,2248315.22</destination>
   </destinations>
   <srs>epsg:27572</srs>
   <rows>
```

```
      <row>
         <cells>
            <cell>
               <distanceMeters>8109.0</distanceMeters>
               <durationSeconds>656.74</durationSeconds>
               <status>OK</status>
            </cell>
            <cell>
               <distanceMeters>11249.37</distanceMeters>
               <durationSeconds>888.57</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
      <row>
         <cells>
            <cell>
               <distanceMeters>11736.89</distanceMeters>
               <durationSeconds>959.9</durationSeconds>
               <status>OK</status>
            </cell>
            <cell>
               <distanceMeters>7636.67</distanceMeters>
               <durationSeconds>609.35</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
   </rows>
</matrixResultV3>
```

Case of forgotten origin and destination specification(matrixResultV3/status is ERROR)

```
<matrixResultV3>
   <message>Origins and destinations must be not null</message>
   <status>ERROR</status>
</matrixResultV3>
```

Case of a faulty type (serviceResult/status is ERROR)

```
<serviceResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
</serviceResult>
```

Case of a snap-to-graph error (matrixResultV3/status is OK) / (cell/status est KO)

```
<matrixResultV3>
   <status>OK</status>
   <origins>
      <origin>-0.36147,49.18392</origin>
      <origin>7.26132,43.70629</origin>
   </origins>
   <destinations>
      <destination>146.08821,48.43253</destination>
      <destination>2.34878,48.86473</destination>
   </destinations>
   <rows>
      <row>
         <cells>
            <cell>
```

```
                <distanceMeters>0.0</distanceMeters>
                <durationSeconds>0.0</durationSeconds>
                <status>KO</status>
             </cell>
             <cell>
                <distanceMeters>234196.77000000002</distanceMeters>
                <durationSeconds>9451.57</durationSeconds>
                <status>OK</status>
             </cell>
          </cells>
       </row>
       <row>
          <cells>
             <cell>
                <distanceMeters>0.0</distanceMeters>
                <durationSeconds>0.0</durationSeconds>
                <status>KO</status>
             </cell>
             <cell>
                <distanceMeters>930934.64</distanceMeters>
                <durationSeconds>31773.440000000002</durationSeconds>
                <status>OK</status>
             </cell>
          </cells>
       </row>
    </rows>
</matrixResultV3>
```

Case of a problem with the graph: absent file, faulty filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in matrix computation
Error in smartrouting
datasource is null</message>
   <status>ERROR</status>
</serviceResult>
```

V2

Parameters / properties

Input

parameter	description	optional	default
srs	projection (ESPG code such as epsg:4326 or wgs84)	yes	
origins	List of coordinates of points of origin. The longitude and latitude coordinates are separated by the ,, character	yes	
destinations	List of coordinates of destination points. The latitude and longitude cooordinates are separated by the , character.	yes	
graphName	Name of the graph to use	yes	
method	shortest (distance) or fastest (time) route	yes	time
profileId	Vehicle identifier (saved under vehicle profiles) to use	yes	
profileName	Vehicle profile (saved under vehicle profiles) to use	yes	
exclusions	List of restriction rules to use, separated by the , or ; character (Example: "Toll", "Tunnel", "Bridge")	yes	

parameter	description	optional	default
startDateTime	Start Date and Time (format ISO8601: local time). Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a start on 21 January 2014, at 9.00am in Paris	yes	
snapMethod	Snap-to-graph method - standard: to the nearest connectable road section - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without any restriction rules	yes	standard

Output

Matrix (matrixResultV2)

parameter	type	min/max	description
origins/origin (or origins in JSON / JSON-P)	string (or array of origins/origin in JSON / JSON-P)	0/ unlimited	origin positions.
destinations/destination (or destinations in JSON / JSON-P)	string (or array of destinations/destination in JSON / JSON-P)	0/ unlimited	destination positions.
rows/row (or rows in JSON / JSON-P)	matrixRowV2 (or array of rows/row (matrixRow) in JSON / JSON-P)	0/ unlimited	Matrix line

Matrix line (matrixRowV2)

parameter	type	min/max	description
cells/cell (or cells in JSON / JSON-P)	matrixCellV2 (or array of cells/cell (matrixCellV2) in JSON / JSON-P)	0/ unlimited	Matrix cell

Matrix cell (matrixCellV2)

parameter	type	min/max	description
distanceMeters	double	1/1	Matrix cell distance (in metres)
durationSeconds	double	1/1	Matrix cell duration (in seconds)
status	string	0/1	Matrix cell status: - OK: the itinerary has been found for this cell - KO: no itinerary found for this cell

SOAP

WSDL

http://`<server>`/`<webapp>`/api/ws/matrixService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:matrixV2>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <srs>epsg:27572</srs>
            <!--Optional:-->
            <origins>
               <!--1 or more repetitions:-->
               <origin>
                  <x>315846.96</x>
                  <y>2254268.3500000001</y>
               </origin>
                <origin>
                           <x>313584.77</x>
                  <y>2251648.9700000002</y>
               </origin>
            </origins>
            <!--Optional:-->
            <destinations>
               <!--1 or more repetitions:-->
               <destination>
                  <x>321442.89</x>
                  <y>2251013.98</y>
               </destination>
                <destination>
                  <x>318982.27</x>
                  <y>2248315.2200000002</y>
               </destination>
            </destinations>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <method>time</method>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <snapMethod></snapMethod>
         </request>
      </sch:matrixV2>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:matrixV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <MatrixResult>
            <status>OK</status>
```

```
               <origins>
                  <origin>315846.96,2254268.35</origin>
                  <origin>313584.77,2251648.97</origin>
               </origins>
               <destinations>
                  <destination>321442.89,2251013.98</destination>
                  <destination>318982.27,2248315.22</destination>
               </destinations>
               <srs>epsg:27572</srs>
               <rows>
                  <row>
                     <cells>
                        <cell>
                           <distanceMeters>8109.04</distanceMeters>
                           <durationSeconds>894.48</durationSeconds>
                           <status>OK</status>
                        </cell>
                        <cell>
                           <distanceMeters>11127.53</distanceMeters>
                           <durationSeconds>873.21</durationSeconds>
                           <status>OK</status>
                        </cell>
                     </cells>
                  </row>
                  <row>
                     <cells>
                        <cell>
                           <distanceMeters>11736.87</distanceMeters>
                           <durationSeconds>1158.04</durationSeconds>
                           <status>OK</status>
                        </cell>
                        <cell>
                           <distanceMeters>7514.82</distanceMeters>
                           <durationSeconds>615.79</durationSeconds>
                           <status>OK</status>
                        </cell>
                     </cells>
                  </row>
               </rows>
            </MatrixResult>
         </ns2:matrixV2Response>
      </soap:Body>
   </soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/matrix/v2.json?
origins=315846.96,2254268.3500000001;313584.77,2251648.9700000002&destinations=321442.89,2251013.98;318982.27,2248315.2200
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/matrix/v2.json?
origins=315846.96,2254268.3500000001;313584.77,2251648.9700000002&destinations=321442.89,2251013.98;318982.27,2248315.2200
```

XML query

```
http://<server>/<webapp>/api/lbs/matrix/v2.xml?
origins=315846.96,2254268.3500000001;313584.77,2251648.9700000002&destinations=321442.89,2251013.98;318982.27,2248315.2200000
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message":null,
   "status":"OK",
   "origins":[
      "315846.96,2254268.35",
      "313584.77,2251648.97"
   ],
   "destinations":[
      "321442.89,2251013.98",
      "318982.27,2248315.22"
   ],
   "srs":"epsg:27572",
   "rows":[
      {
         "cells":[
            {
               "distanceMeters":8109.04,
               "durationSeconds":894.48,
               "status":"OK"
            },
            {
               "distanceMeters":11127.53,
               "durationSeconds":873.21,
               "status":"OK"
            }
         ]
      },
      {
         "cells":[
            {
               "distanceMeters":11736.87,
               "durationSeconds":1158.04,
               "status":"OK"
            },
            {
               "distanceMeters":7514.82,
               "durationSeconds":615.79,
               "status":"OK"
            }
         ]
      }
   ]
}
```

JSON-P format

```
MyCallBack({
   "message":null,
   "status":"OK",
   "origins":[
      "315846.96,2254268.35",
      "313584.77,2251648.97"
   ],
   "destinations":[
      "321442.89,2251013.98",
      "318982.27,2248315.22"
   ],
   "srs":"epsg:27572",
   "rows":[
      {
         "cells":[
            {
               "distanceMeters":8109.04,
               "durationSeconds":894.48,
               "status":"OK"
            },
            {
               "distanceMeters":11127.53,
               "durationSeconds":873.21,
               "status":"OK"
            }
         ]
      },
      {
         "cells":[
            {
               "distanceMeters":11736.87,
               "durationSeconds":1158.04,
               "status":"OK"
            },
            {
               "distanceMeters":7514.82,
               "durationSeconds":615.79,
               "status":"OK"
            }
         ]
      }
   ]
});
```

XML format

```
<matrixResultV2>
   <status>OK</status>
   <origins>
      <origin>-0.36147,49.18392</origin>
      <origin>7.26132,43.70629</origin>
   </origins>
   <destinations>
      <destination>0.08821,48.43253</destination>
      <destination>2.34878,48.86473</destination>
   </destinations>
   <rows>
      <row>
         <cells>
            <cell>
               <distanceMeters>107526.06</distanceMeters>
               <durationSeconds>4906.25</durationSeconds>
               <status>OK</status>
            </cell>
            <cell>
               <distanceMeters>234196.77000000002</distanceMeters>
               <durationSeconds>9451.57</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
      <row>
         <cells>
            <cell>
               <distanceMeters>1103695.95</distanceMeters>
               <durationSeconds>37887.43</durationSeconds>
               <status>OK</status>
            </cell>
            <cell>
               <distanceMeters>930934.64</distanceMeters>
               <durationSeconds>31773.440000000002</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
   </rows>
</matrixResultV2>
```

Possible responses

Case of a found itinerary (matrixResultV2/status is OK)

```
<?xml version="1.0" encoding="UTF-8"?>
<matrixResultV2>
   <status>OK</status>
   <origins>
      <origin>315846.96,2254268.35</origin>
      <origin>313584.77,2251648.97</origin>
   </origins>
   <destinations>
      <destination>321442.89,2251013.98</destination>
      <destination>318982.27,2248315.22</destination>
   </destinations>
   <srs>epsg:27572</srs>
   <rows>
      <row>
         <cells>
            <cell>
               <distanceMeters>8109.04</distanceMeters>
               <durationSeconds>894.48</durationSeconds>
               <status>OK</status>
            </cell>
            <cell>
               <distanceMeters>11127.53</distanceMeters>
               <durationSeconds>873.21</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
      <row>
         <cells>
            <cell>
               <distanceMeters>11736.87</distanceMeters>
               <durationSeconds>1158.04</durationSeconds>
               <status>OK</status>
            </cell>
            <cell>
               <distanceMeters>7514.82</distanceMeters>
               <durationSeconds>615.79</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
   </rows>
</matrixResultV2>
```

Case of forgotten origin or destination specification (matrixResultV2/status is ERROR)

```
<matrixResultV2>
   <message>Origins and destinations must be not null</message>
   <status>ERROR</status>
</matrixResultV2>
```

Case of a faulty type (serviceResult/status is ERROR)

```
<serviceResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
</serviceResult>
```

Case of a snap-to-graph error (matrixResultV2/status is OK) / (cell/status est KO)

```
<matrixResultV2>
   <status>OK</status>
   <origins>
      <origin>-0.36147,49.18392</origin>
      <origin>7.26132,43.70629</origin>
   </origins>
   <destinations>
      <destination>146.08821,48.43253</destination>
      <destination>2.34878,48.86473</destination>
   </destinations>
   <rows>
      <row>
         <cells>
            <cell>
               <distanceMeters>0.0</distanceMeters>
               <durationSeconds>0.0</durationSeconds>
               <status>KO</status>
            </cell>
            <cell>
               <distanceMeters>234196.77000000002</distanceMeters>
               <durationSeconds>9451.57</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
      <row>
         <cells>
            <cell>
               <distanceMeters>0.0</distanceMeters>
               <durationSeconds>0.0</durationSeconds>
               <status>KO</status>
            </cell>
            <cell>
               <distanceMeters>930934.64</distanceMeters>
               <durationSeconds>31773.440000000002</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
   </rows>
</matrixResultV2>
```

Case of a problem with the graph: absent file, faulty filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in matrix computation
Error in smartrouting
datasource is null</message>
   <status>ERROR</status>
</serviceResult>
```

V1

Parameters / properties

Input

parameter	description	optional	default
srs	projection (ESPG code such as epsg:4326 or wgs84)	yes	
origins	List of coordinates of points of origin. The longitude and latitude coordinates are separated by the ,, character	yes	
destinations	List of coordinates of destination points. The latitude and longitude cooordinates are separated by the , character.	yes	
graphName	Name of the graph to use	yes	
method	shortest (distance) or fastest (time) route	yes	time
profileId	Vehicle identifier (saved under vehicle profiles) to use	yes	
profileName	Vehicle profile (saved under vehicle profiles) to use	yes	
rejectFlags	Deprecated, replaced by exclusions		
exclusions	List of restriction rules to use, separated by the , or ; character (Example: "Toll", "Tunnel", "Bridge")	yes	

Output

Matrix (matrixResult)

parameter	type	min/max	description
originLocations/ originLocation (or originLocations in JSON / JSON-P)	string (or array of originLocations/ originLocation in JSON / JSON-P)	0/ unlimited	origin positions
destinationLocations/ destinationLocation (or destinationLocations in JSON / JSON-P)	string (or array of destinationLocations/ destinationLocation in JSON / JSON-P)	0/ unlimited	destination positions
rows/row (or rows in JSON / JSON-P)	matrixRow (or array of rows/row (matrixRow) in JSON / JSON-P)	0/ unlimited	Matrix line

Matrix line (matrixRow)

parameter	type	min/max	description
cells/cell (or cells in JSON / JSON-P)	matrixCell (or array of cells/cell (matrixCell) in JSON / JSON-P)	0/ unlimited	Matrix cell

Matrix cell (matrixCell)

parameter	type	min/max	description
distance	double	1/1	Matrix cell distance (in metres)
duration	double	1/1	Matrix cell duration (in seconds)
status	string	0/1	Matrix cell status: - OK: the itinerary has been found for this cell - KO: no itinerary found for this cell

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/matrixService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:matrix>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <srs></srs>
            <!--Optional:-->
            <origins>
               <!--1 or more repetitions:-->
               <origin>
                  <x>-0.361470</x>
                  <y>49.183920</y>
               </origin>
               <origin>
                  <x>7.261320</x>
                  <y>43.706290</y>
               </origin>
            </origins>
            <!--Optional:-->
            <destinations>
               <!--1 or more repetitions:-->
               <destination>
                  <x>0.088210</x>
                  <y>48.432530</y>
               </destination>
               <destination>
                  <x>2.348780</x>
                  <y>48.864730</y>
               </destination>
            </destinations>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <method></method>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <exclusions></exclusions>
         </request>
      </sch:matrix>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:matrixResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <MatrixResult>
            <status>OK</status>
            <originLocations>
                                    <originLocation>-0,361470;49,183920</originLocation>
                                    <originLocation>7,261320;43,706290</originLocation>
            </originLocations>
            <destinationLocations>
                                    <destinationLocation>0,088210;48,432530</destinationLocation>
                                    <destinationLocation>2,348780;48,864730</destinationLocation>
            </destinationLocations>
            <rows>
               <row>
                  <cells>
                     <cell>
                        <distance>107898.07</distance>
                        <duration>3592.48</duration>
                        <status>OK</status>
                     </cell>
                     <cell>
                        <distance>233393.72</distance>
                        <duration>7063.8</duration>
                        <status>OK</status>
                     </cell>
                  </cells>
               </row>
               <row>
                  <cells>
                     <cell>
                        <distance>1105102.84</distance>
                        <duration>32465.86</duration>
                        <status>OK</status>
                     </cell>
                     <cell>
                        <distance>931394.85</distance>
                        <duration>27277.25</duration>
                        <status>OK</status>
                     </cell>
                  </cells>
               </row>
            </rows>
         </MatrixResult>
      </ns2:matrixResponse>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/matrix.json?
origins=-0.36147,49.18392;7.26132,43.70629&destinations=0.08821,48.43253;2.34878,48.86473
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/matrix.json?
origins=-0.36147,49.18392;7.26132,43.70629&destinations=0.08821,48.43253;2.34878,48.86473&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/matrix.xml?
origins=-0.36147,49.18392;7.26132,43.70629&destinations=0.08821,48.43253;2.34878,48.86473
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message":null,
   "status":"OK",
   "originLocations":[
      "-0,361470;49,183920",
      "7,261320;43,706290"
   ],
   "destinationLocations":[
      "0,088210;48,432530",
      "2,348780;48,864730"
   ],
   "rows":[
      {
         "cells":[
            {
               "distance":107528.37,
               "duration":5139.2300000000005,
               "status":"OK"
            },
            {
               "distance":234732.19,
               "duration":9880.54,
               "status":"OK"
            }
         ]
      },
      {
         "cells":[
            {
               "distance":1103920.65,
               "duration":38946.42,
               "status":"OK"
            },
            {
               "distance":930731.03,
               "duration":32689.65,
               "status":"OK"
            }
         ]
      }
   ]
}
```

JSON-P format

```
myCallback(
        {
           "message":null,
           "status":"OK",
           "originLocations":[
                  "-0,361470;49,183920",
                  "7,261320;43,706290"
           ],
           "destinationLocations":[
                  "0,088210;48,432530",
                  "2,348780;48,864730"
           ],
           "rows":[
                  {
                         "cells":[
                                {
                                   "distance":107528.37,
                                   "duration":5139.2300000000005,
                                   "status":"OK"
                                },
                                {
                                   "distance":234732.19,
                                   "duration":9880.54,
                                   "status":"OK"
                                }
                         ]
                  },
                  {
                         "cells":[
                                {
                                   "distance":1103920.65,
                                   "duration":38946.42,
                                   "status":"OK"
                                },
                                {
                                   "distance":930731.03,
                                   "duration":32689.65,
                                   "status":"OK"
                                }
                         ]
                  }
           ]
        }
);
```

XML format

```
<matrixResult>
        <status>OK</status>
        <originLocations>
                <originLocation>-0,361470;49,183920</originLocation>
                <originLocation>7,261320;43,706290</originLocation>
        </originLocations>
        <destinationLocations>
                <destinationLocation>0,088210;48,432530</destinationLocation>
                <destinationLocation>2,348780;48,864730</destinationLocation>
        </destinationLocations>
        <rows>
                <row>
                        <cells>
                                <cell>
                                        <distance>107528.37</distance>
                                        <duration>5139.2300000000005</duration>
                                        <status>OK</status>
                                </cell>
                                <cell>
                                        <distance>234732.19</distance>
                                        <duration>9880.54</duration>
                                        <status>OK</status>
                                </cell>
                        </cells>
                </row>
                <row>
                        <cells>
                                <cell>
                                        <distance>1103920.65</distance>
                                        <duration>38946.42</duration>
                                        <status>OK</status>
                                </cell>
                                <cell>
                                        <distance>930731.03</distance>
                                        <duration>32689.65</duration>
                                        <status>OK</status>
                                </cell>
                        </cells>
                </row>
        </rows>
</matrixResult>
```

Possible responses

Case of an itinerary that has been found (matrixResult/status is OK)

```
<matrixResult>
        <status>OK</status>
        <originLocations>
                <originLocation>-0,361470;49,183920</originLocation>
                <originLocation>7,261320;43,706290</originLocation>
        </originLocations>
        <destinationLocations>
                <destinationLocation>0,088210;48,432530</destinationLocation>
                <destinationLocation>2,348780;48,864730</destinationLocation>
        </destinationLocations>
        <rows>
                <row>
                        <cells>
                                <cell>
                                        <distance>107528.37</distance>
                                        <duration>5139.2300000000005</duration>
                                        <status>OK</status>
                                </cell>
                                <cell>
                                        <distance>234732.19</distance>
                                        <duration>9880.54</duration>
                                        <status>OK</status>
                                </cell>
                        </cells>
                </row>
                <row>
                        <cells>
                                <cell>
                                        <distance>1103920.65</distance>
                                        <duration>38946.42</duration>
                                        <status>OK</status>
                                </cell>
                                <cell>
                                        <distance>930731.03</distance>
                                        <duration>32689.65</duration>
                                        <status>OK</status>
                                </cell>
                        </cells>
                </row>
        </rows>
</matrixResult>
```

Case of forgetting to specify the origin and destination (matrixResult/status is OK)

```
<matrixResult>
        <status>OK</status>
        <originLocations/>
        <destinationLocations/>
</matrixResult>
```

Case of a faulty type (serviceResult/status is ERROR)

```
<serviceResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
</serviceResult>
```

Case of a snap-to-graph error (matrixResult/status is OK) / (matrixCell/status is KO)

```
<matrixResult>
        <status>OK</status>
        <originLocations>
                <originLocation>-0,361470;49,183920</originLocation>
                <originLocation>57,261320;43,706290</originLocation>
        </originLocations>
        <destinationLocations>
                <destinationLocation>0,088210;48,432530</destinationLocation>
                <destinationLocation>2,348780;48,864730</destinationLocation>
        </destinationLocations>
        <rows>
                <row>
                        <cells>
                                <cell>
                                        <distance>107528.37</distance>
                                        <duration>5139.2300000000005</duration>
                                        <status>OK</status>
                                </cell>
                                <cell>
                                        <distance>234732.19</distance>
                                        <duration>9880.54</duration>
                                        <status>OK</status>
                                </cell>
                        </cells>
                </row>
                <row>
                        <cells>
                                <cell>
                                        <distance>0.0</distance>
                                        <duration>0.0</duration>
                                        <status>KO</status>
                                </cell>
                                <cell>
                                        <distance>0.0</distance>
                                        <duration>0.0</duration>
                                        <status>KO</status>
                                </cell>
                        </cells>
                </row>
        </rows>
</matrixResult>
```

Case of a problem with the graph: absent file, faulty filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in matrix computation
Error in smartrouting
datasource is null</message>
   <status>ERROR</status>
</serviceResult>
```

FAQ

1. Can I use aliases instead of .siti file names to call a datasource?

   Yes, refer to details in the FAQ of the reverse geocoding Web Service.

2. How can I use route statistics?

   Check that the graph does contain these statistics and use these two parameters: **`computeOptions`** with a value of *trafficPatterns* and **`startDateTime`** to specify the departure date/time.

3. How can I perform a route matrix calculation excluding toll roads?

   If the *Toll* constraint has been included in the graph, place an exclusion in **`exclusions`** :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:matrix>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <srs></srs>
            <!--Optional:-->
            <origins>
```

```
                  <!--1 or more repetitions:-->
                  <origin>
                     <x>-0.361470</x>
                     <y>49.183920</y>
                  </origin>
                  <origin>
                     <x>7.261320</x>
                     <y>43.706290</y>
                  </origin>
               </origins>
               <!--Optional:-->
               <destinations>
                  <!--1 or more repetitions:-->
                  <destination>
                     <x>0.088210</x>
                     <y>48.432530</y>
                  </destination>
                  <destination>
                     <x>2.348780</x>
                     <y>48.864730</y>
                  </destination>
               </destinations>
               <!--Optional:-->
               <graphName></graphName>
               <!--Optional:-->
               <method></method>
               <!--Optional:-->
               <profileId></profileId>
               <!--Optional:-->
               <profileName></profileName>
               <!--Optional:-->
               <exclusions>Toll</exclusions>
            </request>
         </sch:matrix>
      </soapenv:Body>
   </soapenv:Envelope>
```

4. What are Speed Patterns? How can I use them?

   In order to propose journey times that are as accurate as possible and reflect changing traffic conditions, graphs supplied by GEOCONCEPT SAS include, from version M18 upwards, 5 speed profiles (Speed Patterns) for cars and lorries, to take into account the different levels of congestion during the course of one day:

   - standard *normal-speed* corresponds to the speed at a time when the roads have an average congestion level (eg 11.00am)
   - night *fast-speed* corresponds to a very fluid traffic situation, often experienced at night-time (3.00am)
   - congested *slow-speed* corresponds to the times when traffic is densest (8.00am)
   - rush hour *very-slow-speed* corresponds to times when traffic is densest in urban and built-up areas, and is slower than the speed above (8.00am)
   - default *default* corresponds to speeds averaged out over an entire day

   To use them, you need to pass to parameter, when calling the web service, the `computeOptions` parameter with the *speedPattern* option and specify the value of the Speed Pattern requested, without forgetting a vehicle profile
   Example:

+

```
&computeOptions=speedPattern:fast-speed&profileId=1
```

How to use Heavy Goods Vehicle attributes?

> The graph must include the attributes for Heavy Goods Vehicles (as standard in the graphs supplied by GEOCONCEPT SAS from version M18 upwards) and either calculate an itinerary using a vehicle profile using restrictions (cf. the catalogue of vehicles, editable, defined in the SmartRoutingVehicles.xml file, stored in the folder `‘*<GEOCONCEPT_WEB_HOME>”\smartrouting \jee\smartrouting\conf\* ), or overwrite  the call to the web service by using the **`computeOptions`** parameter with the options *length*, *width*, *height*, *weight*, *axles* and/or *weightPerAxle*.
> Example for a journey with a vehicle 4.5 metres high:

```
&computeOptions=height:450
```

Compact matrix calculation

(fr) Cette page n’est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

This web service calculates a route matix for a series of points and returns a distance matrix. It draws on the configured graph the name of which has been specified in the Geoconcept Web administration interface.

This web service is a variation of the matrix calculation web service that returns exactly the same results, but in another format.

Availability

This web service is available at all times with Geoconcept Web in tandem with a graph.

Version change

Earlier versions of the web service are conserved in Geoconcept Web to ensure compatibility with previous software developments. We recommend using the most recent version.

Changes in relation to V3

- Addition of "timeOut" and "computeOptions" parameters.
- Addition to the "nodes" snapMethod of snap to nearest nodes.

Changes in relation to V2

- Addition of the notion of the node, faster, to snap to graph nodes rather than to geographic coordinates. Addition of the following items: "originNodes", "destinationNodes" and "nodes" in the snapMethod.

- Addition of the "maxCost" parameter.

V3

Parameters / properties

Input

parameter	description	optional	default
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
origins	List of the coordinates for points of origin. Longitude and Latitude coordinates are separated by , characters.	yes	
originNodes	List of origin ids nodes. Ids nodes are separated by the . character. Note: a physical node does not have the same ID in another graph.	yes	
destinations	List of coordinates for destination points. Longitude and Latitude coordinates are separaged by . characters.	yes	
destinationNodes	List of destination ids nodes. ids nodes are separated by . characters. Note: a physical node does not have the same ID in another graph.	yes	
graphName (depreciated)	Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
method	shortest route (distance) or fastest route (time)	yes	time
profileId (depreciated)	Vehicle identifier (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
profileName (depreciated)	Vehicle profile (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
exclusions	List of restriction regulations to use, separated by *;* character (Example: Toll, Tunnel, Bridge)	yes	
startDateTime	Departure Date and Time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a departure on 21 Januray 2014, at 9.00am in Paris. Caution: the + character may be misinterpreted by browsers, so in this case it should be replaced by *%2B*.	yes	
snapMethod	Snap to graph method - standard: to the nearest connectable road section - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without any restriction rules - nodes: Snap directly to nodes supplied by the originNode, destinationNode and waypointNodes parameters, or, if these parameters have not been set, to the nearest nodes sourced by the origin, destination and waypoint parameters	yes	standard
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84: `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))` NOTE: WKT geometries must be closed.	yes	

parameter	description	optional	default
configName	Name of the configuration to use (defined in Geoconcept Web - Administration / Tools / Road graphs) This replaces the use of graphName, profileId and profileName	yes	
computeOptions	List of options for the calculation,separated by *;* characters - trafficPatterns: uses routing statistics (you will need to supply a value for the startDateTime parameter and use a graph that includes traffic patterns information) - speedPattern (M18): uses a *speed pattern* as defined in the SmartRoutingVehicles.xml file, located in the *`'<GEOCONCEPT_WEB_HOME>''\smartrouting\jee\smartrouting\conf\* folder. Use as follows: "speedPattern:slow-speed" - length (M18): maximum authorised length in centimeters (you need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "length:950" - width (M18): maximum authorised width in centimeters (You will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "width:255" - height (M18): maximum authorised height in centimeters (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "height:360" - weight (M18): maximum authorised weight in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weight:18000" - axles (M18): maximum authorised number of axles (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "axles:2" - weightPerAxle (M18): maximum authorised weight per axle in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weightPerAxle:9000" - snapSpeed: snap-to-graph speed in kilometers per hour. Use as follows: "snapSpeed:10"	yes	
maxCost	Maximum cost not to exceed in the results -1: no maximum cost to take into account 0: take the default value defined in the SmartRouting Server configuration ifnot: value in metres if method=distance or in seconds if method=time	yes	
timeOut	Time out for the calculation (in milliseconds)	yes	

(*) At least one of the two parameter pairs origins/destinations or originNodes/destinationsNodes must be defined.

(M18) Available from version M18 and later versions of graphs supplied by GEOCONCEPT SAS.

Output

Matrix (compactMatrixResultV3)

parameter	type	min/max	description
distanceMeters/ distanceMeter	array (compactRowV3)	0/ unlimited	Distance matrix result (in metres)
durationSeconds/ durationSecond	array (compactRowV3)	0/ unlimited	Time matrix result (in seconds)

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/compactMatrixService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:compactMatrixV3>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <srs>epsg:27572</srs>
            <!--Optional:-->
            <origins>
               <!--1 or more repetitions:-->
               <origin>
                  <x>315846.96</x>
                  <y>2254268.35</y>
               </origin>
                <origin>
                  <x>313584.77</x>
                  <y>2251648.97</y>
               </origin>
            </origins>
            <!--Optional:-->
            <originNodes>
               <!--1 or more repetitions:-->
               <originNode></originNode>
            </originNodes>
            <!--Optional:-->
            <destinations>
               <!--1 or more repetitions:-->
               <destination>
                  <x>321442.89</x>
                  <y>2251013.98</y>
               </destination>
                <destination>
                  <x>318982.27</x>
                  <y>2248315.22</y>
               </destination>
            </destinations>
            <!--Optional:-->
            <destinationNodes>
               <!--1 or more repetitions:-->
               <destinationNode></destinationNode>
            </destinationNodes>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <method>time</method>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
```

```
                <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <snapMethod></snapMethod>
            <!--Optional:-->
            <avoidArea></avoidArea>
            <!--Optional:-->
            <configName></configName>
            <!--Optional:-->
            <computeOptions></computeOptions>
            <!--Optional:-->
            <timeOut></timeOut>
            <!--Optional:-->
            <maxCost></maxCost>
         </request>
      </sch:compactMatrixV3>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:compactMatrixV3Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <CompactMatrixResult>
            <status>OK</status>
            <distanceMeters>
               <distanceMeter>8109,11249</distanceMeter>
               <distanceMeter>11737,7637</distanceMeter>
            </distanceMeters>
            <durationSeconds>
               <durationSecond>565,816</durationSecond>
               <durationSecond>857,526</durationSecond>
            </durationSeconds>
         </CompactMatrixResult>
      </ns2:compactMatrixV3Response>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/compactMatrix/v3.json?
origins=315846.96,2254268.35;313584.77,2251648.97&destinations=321442.89,2251013.98;318982.27,2248315.22&srs=epsg:27572&me
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/compactMatrix/v3.json?
origins=315846.96,2254268.35;313584.77,2251648.97&destinations=321442.89,2251013.98;318982.27,2248315.22&srs=epsg:27572&me
```

Response

The response is always in UTF-8 format.

JSON format

```
{
        "message": null,
        "status": "OK",
        "distanceMeters": [
                [
                        8109,
                        11249
                ],
                [
                        11737,
                        7637
                ]
        ],
        "durationSeconds": [
                [
                        657,
                        889
                ],
                [
                        960,
                        609
                ]
        ]
}
```

JSON-P format

```
myCallback(
    {
        "message": null,
        "status": "OK",
        "distanceMeters": [
            [
                8109,
                11249
            ],
            [
                11737,
                7637
            ]
        ],
        "durationSeconds": [
            [
                657,
                889
            ],
            [
                960,
                609
            ]
        ]
    }
);
```

Possible responses

Case of a found itinerary (compactMatrixResultV2/status is OK)

```
<compactMatrixResultV3>
```

```
   <status>OK</status>
   <distanceMeters>
      <distanceMeter>8109,11249</distanceMeter>
      <distanceMeter>11737,7637</distanceMeter>
   </distanceMeters>
   <durationSeconds>
      <durationSecond>657,889</durationSecond>
      <durationSecond>960,609</durationSecond>
   </durationSeconds>
</compactMatrixResultV3>
```

Case of a forgotten origin or destination specification tool (compactMatrixResultV2/status is ERROR)

```
<compactMatrixResultV3>
   <message>Origins and destinations must be not null</message>
   <status>ERROR</status>
</compactMatrixResultV3>
```

Case of a faulty type (serviceResult/status is ERROR)

```
<serviceResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
</serviceResult>
```

Case of a snap to graph error (compactMatrixResultV2/status is OK) / (cell/status is KO)

```
<compactMatrixResultV3>
   <status>OK</status>
   <origins>
      <origin>-0.36147,49.18392</origin>
      <origin>7.26132,43.70629</origin>
   </origins>
   <destinations>
      <destination>146.08821,48.43253</destination>
      <destination>2.34878,48.86473</destination>
   </destinations>
   <rows>
      <row>
         <cells>
            <cell>
               <distanceMeters>0.0</distanceMeters>
               <durationSeconds>0.0</durationSeconds>
               <status>KO</status>
            </cell>
            <cell>
               <distanceMeters>234196.77000000002</distanceMeters>
               <durationSeconds>9451.57</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
      <row>
         <cells>
            <cell>
               <distanceMeters>0.0</distanceMeters>
               <durationSeconds>0.0</durationSeconds>
               <status>KO</status>
            </cell>
            <cell>
```

```
                 <distanceMeters>930934.64</distanceMeters>
                 <durationSeconds>31773.440000000002</durationSeconds>
                 <status>OK</status>
              </cell>
           </cells>
        </row>
     </rows>
  </compactMatrixResultV3>
```

Case of a problem with the graph: absent file, bad filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in matrix computation
Error in smartrouting
datasource is null</message>
   <status>ERROR</status>
</serviceResult>
```

V2

Parameters / properties

Input

parameter	description	optional	default
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
origins	List of the coordinates for points of origin. Longitude and Latitude coordinates are separated by , characters.	yes	
originNodes	List of origin ids nodes. Ids nodes are separated by the . character. Note: a physical node does not have the same ID in another graph.	yes	
destinations	List of coordinates for destination points. Longitude and Latitude coordinates are separaged by . characters.	yes	
destinationNodes	List of destination ids nodes. ids nodes are separated by . characters. Note: a physical node does not have the same ID in another graph.	yes	
graphName	Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
method	shortest route (distance) or fastest route (time)	yes	time
profileId	Vehicle identifier (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
profileName	Vehicle profile (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
exclusions	List of restriction rules to use, separated by the , or ; character (Example: "Toll", "Tunnel", "Bridge")	yes	
startDateTime	Departure Date and Time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a departure on 21 Januray 2014, at 9.00am in Paris. Caution: the + character may be misinterpreted by browsers, so in this case it should be replaced by *%2B*.	yes	
snapMethod	Snap to graph method - standard: to the nearest connectable road section - extended: via restricted road sections (pedestrian routes…)	yes	standard

parameter	description	optional	default
	- nearest: only to the nearest road section - unrestricted: without any restriction rules - nodes: direct snap-to nodes indicated by locationNode and *node* parameters for resources		
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84: `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))` NOTE: WKT geometries must be closed.	yes	
configName	Name of the configuration to use (defined in Geoconcept Web - Administration / Tools / Road graphs) This replaces the use of graphName, profileId and profileName	yes	
maxCost	Maximum cost not to exceed in the results -1: no maximum cost to take into account 0: take the default value defined in the SmartRouting Server configuration ifnot: value in metres if method=distance or in seconds if method=time	yes	

(*) At least one of the two parameter pairs origins/destinations or originNodes/destinationsNodes must be defined.

Output

Matrix (compactMatrixResultV2)

parameter	type	min/max	description
distanceMeters/ distanceMeter	array (compactRowV2)	0/ unlimited	Distance matrix result (in metres)
durationSeconds/ durationSecond	array (compactRowV2)	0/ unlimited	Time matrix result (in seconds)

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/compactMatrixService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:compactMatrixV2>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <srs>epsg:27572</srs>
            <!--Optional:-->
            <origins>
```

```
                  <!--1 or more repetitions:-->
                  <origin>
                     <x>315846.96</x>
                     <y>2254268.35</y>
                  </origin>
                   <origin>
                     <x>313584.77</x>
                     <y>2251648.97</y>
                  </origin>
               </origins>
               <!--Optional:-->
               <originNodes>
                  <!--1 or more repetitions:-->
                  <originNode></originNode>
               </originNodes>
               <!--Optional:-->
               <destinations>
                  <!--1 or more repetitions:-->
                  <destination>
                     <x>321442.89</x>
                     <y>2251013.98</y>
                  </destination>
                   <destination>
                     <x>318982.27</x>
                     <y>2248315.22</y>
                  </destination>
               </destinations>
               <!--Optional:-->
               <destinationNodes>
                  <!--1 or more repetitions:-->
                  <destinationNode></destinationNode>
               </destinationNodes>
               <!--Optional:-->
               <graphName></graphName>
               <!--Optional:-->
               <method>time</method>
               <!--Optional:-->
               <profileId></profileId>
               <!--Optional:-->
               <profileName></profileName>
               <!--Optional:-->
               <exclusions>
                  <!--Zero or more repetitions:-->
                  <exclusion></exclusion>
               </exclusions>
               <!--Optional:-->
               <startDateTime></startDateTime>
               <!--Optional:-->
               <snapMethod></snapMethod>
               <!--Optional:-->
               <avoidArea></avoidArea>
               <!--Optional:-->
               <configName></configName>
               <!--Optional:-->
               <maxCost></maxCost>
            </request>
         </sch:compactMatrixV2>
      </soapenv:Body>
   </soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

```
   <soap:Body>
      <ns2:compactMatrixV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <CompactMatrixResult>
            <status>OK</status>
            <distanceMeters>
               <distanceMeter>8109,11249</distanceMeter>
               <distanceMeter>11737,7637</distanceMeter>
            </distanceMeters>
            <durationSeconds>
               <durationSecond>565,816</durationSecond>
               <durationSecond>857,526</durationSecond>
            </durationSeconds>
         </CompactMatrixResult>
      </ns2:compactMatrixV2Response>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/compactMatrix/v2.json?
origins=315846.96,2254268.35;313584.77,2251648.97&destinations=321442.89,2251013.98;318982.27,2248315.22&srs=epsg:27572&me
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/compactMatrix/v2.json?
origins=315846.96,2254268.35;313584.77,2251648.97&destinations=321442.89,2251013.98;318982.27,2248315.22&srs=epsg:27572&me
```

XML query

```
http://<server>/<webapp>/api/lbs/compactMatrix/v2.xml?
origins=315846.96,2254268.35;313584.77,2251648.97&destinations=321442.89,2251013.98;318982.27,2248315.22&srs=epsg:27572&me
```

Response

The response is always in UTF-8 format.

JSON format

```
{
        "message": null,
        "status": "OK",
        "distanceMeters": [
                [
                        8109,
                        11249
                ],
                [
                        11737,
                        7637
                ]
        ],
        "durationSeconds": [
                [
                        657,
                        889
                ],
                [
                        960,
                        609
                ]
        ]
}
```

JSON-P format

```
myCallback(
    {
        "message": null,
        "status": "OK",
        "distanceMeters": [
            [
                8109,
                11249
            ],
            [
                11737,
                7637
            ]
        ],
        "durationSeconds": [
            [
                657,
                889
            ],
            [
                960,
                609
            ]
        ]
    }
);
```

XML format

```
<compactMatrixResultV2>
   <status>OK</status>
   <distanceMeters>
      <distanceMeter>8109,11249</distanceMeter>
      <distanceMeter>11737,7637</distanceMeter>
   </distanceMeters>
   <durationSeconds>
      <durationSecond>657,889</durationSecond>
      <durationSecond>960,609</durationSecond>
   </durationSeconds>
</compactMatrixResultV2>
```

Possible responses

Case of a found itinerary (compactMatrixResultV2/status is OK)

```
<compactMatrixResultV2>
   <status>OK</status>
   <distanceMeters>
```

```
        <distanceMeter>8109,11249</distanceMeter>
        <distanceMeter>11737,7637</distanceMeter>
    </distanceMeters>
    <durationSeconds>
        <durationSecond>657,889</durationSecond>
        <durationSecond>960,609</durationSecond>
    </durationSeconds>
</compactMatrixResultV2>
```

Case of a forgotten origin or destination specification tool (compactMatrixResultV2/status is ERROR)

```
<compactMatrixResultV2>
    <message>Origins and destinations must be not null</message>
    <status>ERROR</status>
</compactMatrixResultV2>
```

Case of a faulty type (serviceResult/status is ERROR)

```
<serviceResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
</serviceResult>
```

Case of a snap to graph error (compactMatrixResultV2/status is OK) / (cell/status is KO)

```
<compactMatrixResultV2>
    <status>OK</status>
    <origins>
        <origin>-0.36147,49.18392</origin>
        <origin>7.26132,43.70629</origin>
    </origins>
    <destinations>
        <destination>146.08821,48.43253</destination>
        <destination>2.34878,48.86473</destination>
    </destinations>
    <rows>
        <row>
            <cells>
                <cell>
                    <distanceMeters>0.0</distanceMeters>
                    <durationSeconds>0.0</durationSeconds>
                    <status>KO</status>
                </cell>
                <cell>
                    <distanceMeters>234196.77000000002</distanceMeters>
                    <durationSeconds>9451.57</durationSeconds>
                    <status>OK</status>
                </cell>
            </cells>
        </row>
        <row>
            <cells>
                <cell>
                    <distanceMeters>0.0</distanceMeters>
                    <durationSeconds>0.0</durationSeconds>
                    <status>KO</status>
                </cell>
                <cell>
                    <distanceMeters>930934.64</distanceMeters>
                    <durationSeconds>31773.440000000002</durationSeconds>
                <status>OK</status>
             </cell>
          </cells>
       </row>
    </rows>
</compactMatrixResultV2>
```

Case of a problem with the graph: absent file, bad filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in matrix computation
Error in smartrouting
datasource is null</message>
   <status>ERROR</status>
</serviceResult>
```

V1

Parameters / properties

Input

parameter	description	optional	default
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
origins	List of the coordinates for points of origin. Longitude and Latitude coordinates are separated by , characters.	yes	
destinations	List of coordinates for destination points. Longitude and Latitude coordinates are separaged by . characters.	yes	
graphName	Name of the graph to use	yes	
method	shortest route (distance) or fastest route (time)	yes	time
profileId	Vehicle identifier (saved under vehicle profiles) to use	yes	
profileName	Vehicle profile (saved under vehicle profiles) to use	yes	
exclusions	List of restriction rules to use, separated by the , or ; character (Example: "Toll", "Tunnel", "Bridge")	yes	
startDateTime	Departure Date and Time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a departure on 21 Januray 2014, at 9.00am in Paris	yes	
snapMethod	Snap to graph method - standard: to nearest connectable road section - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without restriction rules	yes	standard

Output

Matrix (matrixResultV2)

parameter	type	min/max	description
origins/origin (or origins in JSON / JSON-P)	string (or array of origins/origin in JSON / JSON-P)	0/ unlimited	origin positions.

parameter	type	min/max	description
destinations/destination (or destinations in JSON / JSON-P)	string (or array of destinations/destination in JSON / JSON-P)	0/ unlimited	destination positions.
rows/row (or rows in JSON / JSON-P)	matrixRowV2 (or array of rows/row (matrixRow) in JSON / JSON-P)	0/ unlimited	Matrix line

Matrix line (matrixRowV2)

parameter	type	min/max	description
cells/cell (or cells in JSON / JSON-P)	matrixCellV2 (or array of cells/cell (matrixCellV2) in JSON / JSON-P)	0/ unlimited	Matrix cell

Matrix cell (matrixCellV2)

parameter	type	min/max	description
distanceMeters	double	1/1	Matrix cell distance (in metres)
durationSeconds	double	1/1	Matrix cell duration (in seconds)
status	string	0/1	Matrix cell status: - OK : the itinerary has been found for this cell - KO : no itinerary found for this cell

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/compactMatrixService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:compactMatrixV1>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <srs>epsg:27572</srs>
            <!--Optional:-->
            <origins>
               <!--1 or more repetitions:-->
               <origin>
                  <x>315846.96</x>
                  <y>2254268.3500000001</y>
               </origin>
                <origin>
                           <x>313584.77</x>
                  <y>2251648.9700000002</y>
               </origin>
            </origins>
            <!--Optional:-->
            <destinations>
               <!--1 or more repetitions:-->
               <destination>
                  <x>321442.89</x>
                  <y>2251013.98</y>
               </destination>
                <destination>
                  <x>318982.27</x>
                  <y>2248315.2200000002</y>
               </destination>
            </destinations>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <method>time</method>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <snapMethod></snapMethod>
         </request>
      </sch:compactMatrixV1>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:compactMatrixV1Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <MatrixResult>
            <status>OK</status>
            <origins>
               <origin>315846.96,2254268.35</origin>
               <origin>313584.77,2251648.97</origin>
            </origins>
            <destinations>
               <destination>321442.89,2251013.98</destination>
               <destination>318982.27,2248315.22</destination>
            </destinations>
            <srs>epsg:27572</srs>
            <rows>
               <row>
                  <cells>
                     <cell>
                        <distanceMeters>8109.04</distanceMeters>
                        <durationSeconds>894.48</durationSeconds>
                        <status>OK</status>
                     </cell>
                     <cell>
                        <distanceMeters>11127.53</distanceMeters>
                        <durationSeconds>873.21</durationSeconds>
                        <status>OK</status>
                     </cell>
```

```
                    </cells>
                 </row>
                 <row>
                    <cells>
                       <cell>
                          <distanceMeters>11736.87</distanceMeters>
                          <durationSeconds>1158.04</durationSeconds>
                          <status>OK</status>
                       </cell>
                       <cell>
                          <distanceMeters>7514.82</distanceMeters>
                          <durationSeconds>615.79</durationSeconds>
                          <status>OK</status>
                       </cell>
                    </cells>
                 </row>
              </rows>
           </MatrixResult>
        </ns2:compactMatrixV1Response>
     </soap:Body>
  </soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/compactMatrix/v1.json?
origins=315846.96,2254268.3500000001;313584.77,2251648.9700000002&destinations=321442.89,2251013.98;318982.27,2248315.2200
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/compactMatrix/v1.json?
origins=315846.96,2254268.3500000001;313584.77,2251648.9700000002&destinations=321442.89,2251013.98;318982.27,2248315.2200
```

XML query

```
http://<server>/<webapp>/api/lbs/compactMatrix/v1.xml?
origins=315846.96,2254268.3500000001;313584.77,2251648.9700000002&destinations=321442.89,2251013.98;318982.27,2248315.2200
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message": null,
   "status": "OK",
   "distanceMeters":    [
            [
         8109,
         11249
      ],
            [
         11737,
         7637
      ]
   ],
   "durationSeconds":    [
            [
         657,
         889
      ],
            [
         960,
         609
      ]
   ]
}
```

JSON-P format

```
myCallback(
    {
       "message": null,
       "status": "OK",
       "distanceMeters":    [
                [
             8109,
             11249
          ],
                [
             11737,
             7637
          ]
       ],
       "durationSeconds":    [
                [
             657,
             889
          ],
                [
             960,
             609
          ]
       ]
    }
);
```

XML format

```
<compactMatrixResult>
   <status>OK</status>
   <distanceMeters>
      <distanceMeter>8109,11249</distanceMeter>
      <distanceMeter>11737,7637</distanceMeter>
   </distanceMeters>
   <durationSeconds>
      <durationSecond>657,889</durationSecond>
      <durationSecond>960,609</durationSecond>
   </durationSeconds>
</compactMatrixResult>
```

Possible responses

Case of an itinerary found (matrixResultV2/status is OK)

```
<compactMatrixResult>
   <status>OK</status>
   <distanceMeters>
      <distanceMeter>8109,11249</distanceMeter>
      <distanceMeter>11737,7637</distanceMeter>
   </distanceMeters>
   <durationSeconds>
      <durationSecond>657,889</durationSecond>
      <durationSecond>960,609</durationSecond>
   </durationSeconds>
</compactMatrixResult>
```

Case of forgotten origin or destination specification (compactMatrixResultV1/status is ERROR)

```
<comcompactMatrixResultV1>
   <message>Origins and destinations must be not null</message>
   <status>ERROR</status>
</compactMatrixResultV1>
```

Case of a faulty type (serviceResult/status is ERROR)

```
<serviceResult>
        <message>NumberFormatException: For input string: "AAA"</message>
        <status>ERROR</status>
</serviceResult>
```

Case of a snap-to-graph error (compactMatrixResultV1/status is OK) / (cell/status is KO)

```
<compactMatrixResultV1>
   <status>OK</status>
   <origins>
      <origin>-0.36147,49.18392</origin>
      <origin>7.26132,43.70629</origin>
   </origins>
   <destinations>
      <destination>146.08821,48.43253</destination>
      <destination>2.34878,48.86473</destination>
   </destinations>
   <rows>
      <row>
         <cells>
            <cell>
               <distanceMeters>0.0</distanceMeters>
               <durationSeconds>0.0</durationSeconds>
               <status>KO</status>
            </cell>
            <cell>
               <distanceMeters>234196.77000000002</distanceMeters>
               <durationSeconds>9451.57</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
      <row>
         <cells>
            <cell>
               <distanceMeters>0.0</distanceMeters>
               <durationSeconds>0.0</durationSeconds>
               <status>KO</status>
            </cell>
            <cell>
               <distanceMeters>930934.64</distanceMeters>
               <durationSeconds>31773.440000000002</durationSeconds>
               <status>OK</status>
            </cell>
         </cells>
      </row>
   </rows>
</compactMatrixResultV1>
```

Case of a problem with the graph: absent file, bad filepath, etc… (serviceResult/status is ERROR)

```
<serviceResult>
   <message>ServiceException: Error in matrix computation
Error in smartrouting
datasource is null</message>
   <status>ERROR</status>
</serviceResult>
```

FAQ

1. Can I use aliases instead of .siti file names to call a datasource?

   Yes, cf. details under FAQ for the reverse geocoding Web Service.

2. How can I use route statistics?

   Check that the graph does contain these statistics and use these two parameters: **`computeOptions`** with a value of *trafficPatterns* and **`startDateTime`** to specify the departure date/time.

3. How can I perform a route matrix calculation excluding toll roads?

   If the *Toll* constraint has been included in the graph, place an exclusion in **`exclusions`** :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:matrix>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <srs></srs>
            <!--Optional:-->
            <origins>
               <!--1 or more repetitions:-->
               <origin>
                  <x>-0.361470</x>
                  <y>49.183920</y>
               </origin>
               <origin>
                  <x>7.261320</x>
                  <y>43.706290</y>
               </origin>
            </origins>
            <!--Optional:-->
            <destinations>
               <!--1 or more repetitions:-->
               <destination>
```

```
                    <x>0.088210</x>
                    <y>48.432530</y>
                 </destination>
                 <destination>
                    <x>2.348780</x>
                    <y>48.864730</y>
                 </destination>
              </destinations>
              <!--Optional:-->
              <graphName></graphName>
              <!--Optional:-->
              <method></method>
              <!--Optional:-->
              <profileId></profileId>
              <!--Optional:-->
              <profileName></profileName>
              <!--Optional:-->
              <exclusions>Toll</exclusions>
           </request>
        </sch:matrix>
     </soapenv:Body>
  </soapenv:Envelope>
```

4. What are Speed Patterns? How can I use them?

   In order to propose journey times that are as accurate as possible and reflect changing traffic conditions, graphs supplied by GEOCONCEPT SAS include, from version M18 upwards, 5 speed profiles (Speed Patterns) for cars and lorries, to take into account the different levels of congestion during the course of one day:

   - standard *normal-speed* corresponds to the speed at a time when the roads have an average congestion level (eg 11.00am)
   - night *fast-speed* corresponds to a very fluid traffic situation, often experienced at night-time (3.00am)
   - congested *slow-speed* corresponds to the times when traffic is densest (8.00am)
   - rush hour *very-slow-speed* corresponds to times when traffic is densest in urban and built-up areas, and is slower than the speed above (8.00am)
   - default *default* corresponds to speeds averaged out over an entire day

   To use them, you need to pass to parameter, when calling the web service, the **`computeOptions`** parameter with the *speedPattern* option and specify the value of the Speed Pattern requested, without forgetting a vehicle profile
   Example:

+

```
&computeOptions=speedPattern:fast-speed&profileId=1
```

How to use Heavy Goods Vehicle attributes?

The graph must include the attributes for Heavy Goods Vehicles (as standard in the graphs supplied by GEOCONCEPT SAS from version M18 upwards) and either calculate an itinerary using a vehicle profile using restrictions (cf. the catalogue of vehicles, editable, defined in the SmartRoutingVehicles.xml file, stored in the folder `‘*<GEOCONCEPT_WEB_HOME>”\smartrouting*

\jee\smartrouting\conf\), or overwrite the call to the web service by using the **`computeOptions`** parameter with the options *length*, *width*, *height*, *weight*, *axles* and/or *weightPerAxle*.
Example for a journey with a vehicle 4.5 metres high:

```
&computeOptions=height:450
```

Search Around

(fr) Cette page n’est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

This web service allows you to classify a list of resources positioned around a particular point in terms of distance or time. It takes as input a start point (in this case, the target location) and a list of resources with their geographic locations, plus a priority category of either 1 or 2 (this is an optional parameter). The resources are records saved in a database of the user’s choice, depending on their requirements. It is the allotted task of the client to select these resources in the database. The resources are returned sorted firstly on the attribute priority1, and then by time or distance (according to which has been chosen). The priorities are optional. The default value of the priorities is 0. This depends on the configured graph, the name of which has been specified in the Geoconcept Web administration interface.

Availability

This web service is available at all times with Geoconcept Web and a graph or network.

Version change

Earlier versions of the web service are conserved in Geoconcept Web to ensure compatibility with earlier software developments. We reommend using the most recent version.

Changes in relation to v4

- Addition of "maxCost", "timeOut" and "computeOptions" parameters.
- Addition in the "nodes" snapMethod of snapping to nearest nodes.

Changes in relation to v3

- Addition of the notion of mode, faster, to snap to graph nodes rather than to geographic coordinates. Addition of the following elements: "locationNode", of the "node" parameter in resources, and "nodes" in snapMethods.
- Addition of the "graphName", "profileId", "profileName", "avoidArea" and "configName" parameters.

Changes in relation to v2

- Addition of the "snapMethod" parameter

- Deletion of the "targetX" and "targetY" double type parameters, replaced by the string type parameter "location"
- Deletion of the "searchMethod" parameter, replaced by "method"
- Deletion of the "rejectFlags" parameter, replaced by "exclusions"
- Deletion of the "projection" parameter, replaced by "srs"
- The "distances" parameter has been renamed "distanceMeters" and it has changed type from integer to double
- The "time" parameter has been renamed "durationSeconds" and has changed type from integer to double

V4

Parameters / properties

Input

parameter	description	optional	default
location	Coordinates of the start point (or arrival point) in SOAP, points are stored in the "x" and "y" parameters themselves in the "geographicPoint" parameter in REST, the points are separated by the , character	yes *	
locationNode	Start node (or arrival node). Care: a physical node does not have the same ID in another graph.	yes *	
method	The shortest (distance) or fastest (time) route, or the route *as the crow flies* (flying)	yes	time
reverse	if *true*, the target is considered as an arrival	yes	false
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
exclusions	List of restriction rules to use, separated by the ";" character (Example: Toll; Tunnel; Bridge)	yes	
resources	List of resources, separated by ";" characters. Each resource takes the form "id,x,y,node,priority1,priority2"	yes	
startDateTime	Start date and time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a departure on 21 January 2014, at 9.00am in Paris. Caution: the + character can be misinterpreted by browsers, so it is best to substitute it when encountered with *%2B*.	yes	
snapMethod	Snap to graph method - standard: to the nearest connectable road segment - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without any restriction rules - nodes: snap directly to the nodes provided by the locationNode parameter, or, if no value is assigned, to the node(s) nearest to the location parameter	yes	standard
graphName (depreciated)	Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
profileId (depreciated)	Vehicle identifier (saved in the vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	

parameter	description	optional	default
profileName (depreciated)	Vehicle profile (saved in the vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84: `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))` Care: WKT geometries must be closed.	yes	
configName	Name of the configuration to use (defined in Geoconcept Web - Administration / Tools / Road graphs) This replaces the use of graphName, profileId and profileName	yes	
computeOptions	List of options for the calculation,separated by *;* characters - trafficPatterns: uses routing statistics (you will need to supply a value for the startDateTime parameter and use a graph that includes traffic patterns information) - speedPattern (M18): uses a *speed pattern* as defined in the SmartRoutingVehicles.xml file, located in the `*'<GEOCONCEPT_WEB_HOME>''\smartrouting\jee\smartrouting\conf\* folder. Use as follows: "speedPattern:slow-speed" - length (M18): maximum authorised length in centimeters (you need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "length:950" - width (M18): maximum authorised width in centimeters (You will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "width:255" - height (M18): maximum authorised height in centimeters (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "height:360" - weight (M18): maximum authorised weight in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weight:18000" - axles (M18): maximum authorised number of axles (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "axles:2" - weightPerAxle (M18): maximum authorised weight per axle in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weightPerAxle:9000" - snapSpeed: snap-to-graph speed in kilometers per hour. Use as follows: "snapSpeed:10"	yes	
maxCost	Maximum cost not to exceed in the calculation -1: no maximum cost to take into account 0: take the default value defined in the SmartRouting Server configuration if not: value in metres if method=distance or in seconds if method=time	yes	
timeOut	Time out for the calculation (in milliseconds)	yes	

(*) At least one of the two parameters location, and locationNode must be entered.

(M18) Available from version M18 and later versions of graphs supplied by GEOCONCEPT SAS.

Output

Search Around (searchAroundWebResultsV4)

parameter	type	min/max	description
location	double	0/1	Coordinates of the start point (or arrival point) in SOAP, points are stored in the "x" and "y" parameters themselves in the "geographicPoint" parameter in REST, the points are separated by the , character
method	string	0/1	Specified input method (time, distance or flying), and time by default.
srs	string	0/1	Input projection specified.
exclusions	string	0/ unlimited	List of rules for specified input restrictions
searchAroundResult	searchAroundWebResultV4 (or array in JSON / JSON-P)	0/ unlimited	List of results for each candidate.

Search Around element (searchAroundWebResultV4)

parameter	type	min/max	description
id	string	0/1	Candidate identifier
distanceMeters	double	1/1	Distance in metres
durationSeconds	double	1/1	Time in seconds

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/searchAroundService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:searchAroundV4>
         <!--Optional:-->
         <request>
            <location>
               <x>-1.553927</x>
               <y>47.218580</y>
            </location>
            <!--Optional:-->
            <locationNode></locationNode>
            <!--Optional:-->
            <method>time</method>
            <reverse></reverse>
            <!--Optional:-->
            <srs></srs>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <resources>
               <!--Zero or more repetitions:-->
               <resource>
                  <x>-1.593927</x>
                  <y>47.188580</y>
                  <!--Optional:-->
                  <id>1</id>
                  <!--Optional:-->
                  <node></node>
                  <!--Optional:-->
                  <priority1>1</priority1>
                  <!--Optional:-->
                  <priority2>2</priority2>
               </resource>
               <resource>
                  <x>-1.556927</x>
                  <y>47.219580</y>
                  <!--Optional:-->
                  <id>2</id>
                  <!--Optional:-->
                  <node></node>
                  <!--Optional:-->
                  <priority1>1</priority1>
                  <!--Optional:-->
                  <priority2>2</priority2>
               </resource>
            </resources>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <snapMethod></snapMethod>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <avoidArea></avoidArea>
            <!--Optional:-->
            <configName></configName>
            <!--Optional:-->
            <computeOptions></computeOptions>
            <!--Optional:-->
            <maxCost></maxCost>
            <!--Optional:-->
            <timeOut></timeOut>
         </request>
      </sch:searchAroundV4>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:searchAroundV4Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <SearchAroundResult>
            <status>OK</status>
            <location>-1.553927,47.21858</location>
            <method>time</method>
            <srs/>
```

```
               <exclusions/>
               <searchAroundResult>
                  <id>2</id>
                  <distanceMeters>424.0</distanceMeters>
                  <durationSeconds>100.0</durationSeconds>
               </searchAroundResult>
               <searchAroundResult>
                  <id>1</id>
                  <distanceMeters>6682.0</distanceMeters>
                  <durationSeconds>942.0</durationSeconds>
               </searchAroundResult>
            </SearchAroundResult>
         </ns2:searchAroundV4Response>
      </soap:Body>
   </soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/searchAround/v4.json?
method=time&location=-1.593927,47.218580&resources=1,-1.593927,47.188580,1,2;2,-1.556927,47.219580,1,1
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/searchAround/v4.json?
method=time&location=-1.593927,47.218580&resources=1,-1.593927,47.188580,1,2;2,-1.556927,47.219580,1,1&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/searchAround/v4.xml?
method=time&location=-1.593927,47.218580&resources=1,-1.593927,47.188580,1,2;2,-1.556927,47.219580,1,1
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message": null,
   "status": "OK",
   "location": "-1.593927,47.21858",
   "locationNode": null,
   "method": "time",
   "srs": null,
   "exclusions": [],
   "searchAroundResult":    [
            {
         "id": "2",
         "distanceMeters": 4474,
         "durationSeconds": 796
      },
            {
         "id": "1",
         "distanceMeters": 10453,
         "durationSeconds": 1114
      }
   ]
}
```

JSON-P format

```
myCallback(
    {
       "message": null,
       "status": "OK",
       "location": "-1.593927,47.21858",
       "method": "time",
       "srs": null,
       "exclusions": [],
       "searchAroundResult":    [
                {
             "id": "2",
             "distanceMeters": 4474,
             "durationSeconds": 796
          },
                {
             "id": "1",
             "distanceMeters": 10453,
             "durationSeconds": 1114
          }
       ]
    }
);
```

XML format

```
<searchAroundResponseV4>
   <status>OK</status>
   <location>-1.593927,47.21858</location>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distanceMeters>4474.0</distanceMeters>
      <durationSeconds>796.0</durationSeconds>
   </searchAroundResult>
   <searchAroundResult>
      <id>1</id>
      <distanceMeters>10453.0</distanceMeters>
      <durationSeconds>1114.0</durationSeconds>
   </searchAroundResult>
</searchAroundResponseV4>
```

Possible responses

Case of a proximity search finding(searchAroundResponse/status is OK)

```
<searchAroundResponseV4>
   <status>OK</status>
   <location>-1.593927,47.21858</location>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distanceMeters>4474.0</distanceMeters>
      <durationSeconds>796.0</durationSeconds>
   </searchAroundResult>
   <searchAroundResult>
      <id>1</id>
      <distanceMeters>10453.0</distanceMeters>
      <durationSeconds>1114.0</durationSeconds>
   </searchAroundResult>
</searchAroundResponseV4>
```

Case of a problem with the graph: file missing, faulty filepath, etc… (searchAroundResponse/status is OK)

```
<searchAroundResponseV4>
   <status>OK</status>
   <location>-1.593927,47.21858</location>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
   <searchAroundResult>
      <id>1</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
</searchAroundResponseV4>
```

Case of a snap-to-graph error (searchAroundResponse/status is OK)

```
<searchAroundResponseV4>
   <status>OK</status>
   <location>-1.593927,47.21858</location>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
   <searchAroundResult>
      <id>1</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
</searchAroundResponseV4>
```

Case of an absent location parameter (searchAroundResponse/status is ERROR)

```
<searchAroundResponseV4>
   <message>Location must be not null</message>
   <status>ERROR</status>
</searchAroundResponseV4>
```

Case of an incomplete location parameter, or one with the wrong separator (searchAroundResponse/status is ERROR)

```
<searchAroundResponseV4>
   <message>Location point must have 2 components separated with a ,</message>
   <status>ERROR</status>
</searchAroundResponseV4>
```

Case of the absence of a resource (searchAroundResponse/status is OK)

```
<searchAroundResponseV4>
    <message>resources not defined</message>
    <status>ERROR</status>
</searchAroundResponseV4>
```

Case of a badly formatted resource (searchAroundResponse/status is ERROR)

```
<serviceResult>
   <message>ServiceException: not enough fields in candidate 1.-1.5939</message>
   <status>ERROR</status>
</serviceResult>
```

V3

Parameters / properties

Input

parameter	description	optional	default
location	Coordinates of the start point (or arrival point) in SOAP, points are stored in the "x" and "y" parameters themselves in the "geographicPoint" parameter in REST, the points are separated by the , character	yes *	
locationNode	Start node (or arrival node). Care: a physical node does not have the same ID in another graph.	yes *	
method	The shortest (distance) or fastest (time) route, or the route *as the crow flies* (flying)	yes	time
reverse	if *true*, the target is considered as an arrival	yes	false
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
exclusions	List of restriction rules to use, separated by the ";" character (Example: "Toll"; "Tunnel"; "Bridge")	yes	
resources	List of resources, separated by ";" characters. Each resource takes the form "id,x,y,node,priority1,priority2"	yes	
startDateTime	Start date and time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a departure on 21 January 2014, at 9.00am in Paris. Caution: the + character can be misinterpreted by browsers, so it is best to substitute it when encountered with *%2B*.	yes	
snapMethod	Snap to graph method - standard: to the nearest connectable road segment - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without any restriction rules - nodes: snap directly to the nodes indicated by locationNode and the *node* parameters of resources	yes	standard

parameter	description	optional	default
graphName	Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
profileId	Vehicle identifier (saved in the vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
profileName	Vehicle profile (saved in the vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84: `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))` Care: WKT geometries must be closed.	yes	
configName	Name of the configuration to use (defined in Geoconcept Web - Administration / Tools / Road graphs) This replaces the use of graphName, profileId and profileName	yes	

(*) At least one of the two parameters location, and locationNode must be entered.

Output

Search Around (searchAroundWebResultsV3)

parameter	type	min/max	description
location	double	0/1	Coordinates of the start point (or arrival point) in SOAP, points are stored in the "x" and "y" parameters themselves in the "geographicPoint" parameter in REST, the points are separated by the , character
method	string	0/1	Specified input method (time, distance or flying), and time by default.
srs	string	0/1	Input projection specified.
exclusions	string	0/ unlimited	List of rules for specified input restrictions
searchAroundResult	searchAroundWebResultV3 (or array in JSON / JSON-P)	0/ unlimited	List of results for each candidate.

Search Around element (searchAroundWebResultV3)

parameter	type	min/max	description
id	string	0/1	Candidate identifier
distanceMeters	double	1/1	Distance in metres
durationSeconds	double	1/1	Time in seconds

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/searchAroundService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:searchAroundV3>
         <!--Optional:-->
         <request>
            <location>
               <x>-1.553927</x>
               <y>47.218580</y>
            </location>
            <!--Optional:-->
            <locationNode></locationNode>
            <!--Optional:-->
            <method>time</method>
            <reverse></reverse>
            <!--Optional:-->
            <srs></srs>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <resources>
               <!--Zero or more repetitions:-->
               <resource>
                  <x>-1.593927</x>
                  <y>47.188580</y>
                  <!--Optional:-->
                  <id>1</id>
                  <!--Optional:-->
                  <node></node>
                  <!--Optional:-->
                  <priority1>1</priority1>
                  <!--Optional:-->
                  <priority2>2</priority2>
               </resource>
               <resource>
                  <x>-1.556927</x>
                  <y>47.219580</y>
                  <!--Optional:-->
                  <id>2</id>
                  <!--Optional:-->
                  <node></node>
                  <!--Optional:-->
                  <priority1>1</priority1>
                  <!--Optional:-->
                  <priority2>2</priority2>
               </resource>
            </resources>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <snapMethod></snapMethod>
            <!--Optional:-->
            <graphName></graphName>
            <!--Optional:-->
            <profileId></profileId>
            <!--Optional:-->
            <profileName></profileName>
            <!--Optional:-->
            <avoidArea></avoidArea>
            <!--Optional:-->
            <configName></configName>
         </request>
      </sch:searchAroundV3>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:searchAroundV3Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <SearchAroundResult>
            <status>OK</status>
            <location>-1.553927,47.21858</location>
            <method>time</method>
            <srs/>
            <exclusions/>
            <searchAroundResult>
               <id>2</id>
               <distanceMeters>424.0</distanceMeters>
               <durationSeconds>147.0</durationSeconds>
            </searchAroundResult>
            <searchAroundResult>
               <id>1</id>
               <distanceMeters>6682.0</distanceMeters>
               <durationSeconds>1059.0</durationSeconds>
            </searchAroundResult>
         </SearchAroundResult>
      </ns2:searchAroundV3Response>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/searchAround/v3.json?
method=time&location=-1.593927,47.218580&resources=1,-1.593927,47.188580,1,2;2,-1.556927,47.219580,1,1
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/searchAround/v3.json?
method=time&location=-1.593927,47.218580&resources=1,-1.593927,47.188580,1,2;2,-1.556927,47.219580,1,1&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/searchAround/v3.xml?
method=time&location=-1.593927,47.218580&resources=1,-1.593927,47.188580,1,2;2,-1.556927,47.219580,1,1
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message": null,
   "status": "OK",
   "location": "-1.593927,47.21858",
   "locationNode": null,
   "method": "time",
   "srs": null,
   "exclusions": [],
   "searchAroundResult":    [
            {
         "id": "2",
         "distanceMeters": 4474,
         "durationSeconds": 796
      },
            {
         "id": "1",
         "distanceMeters": 10453,
         "durationSeconds": 1114
      }
   ]
}
```

JSON-P format

```
myCallback(
    {
       "message": null,
       "status": "OK",
       "location": "-1.593927,47.21858",
       "method": "time",
       "srs": null,
       "exclusions": [],
       "searchAroundResult":    [
                {
             "id": "2",
             "distanceMeters": 4474,
             "durationSeconds": 796
          },
                {
             "id": "1",
             "distanceMeters": 10453,
             "durationSeconds": 1114
          }
       ]
    }
);
```

XML format

```
<searchAroundResponseV3>
   <status>OK</status>
   <location>-1.593927,47.21858</location>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distanceMeters>4474.0</distanceMeters>
      <durationSeconds>796.0</durationSeconds>
```

```
    </searchAroundResult>
    <searchAroundResult>
       <id>1</id>
       <distanceMeters>10453.0</distanceMeters>
       <durationSeconds>1114.0</durationSeconds>
    </searchAroundResult>
 </searchAroundResponseV3>
```

Possible responses

Case of a proximity search finding(searchAroundResponse/status is OK)

```
 <searchAroundResponseV3>
    <status>OK</status>
    <location>-1.593927,47.21858</location>
    <method>time</method>
    <searchAroundResult>
       <id>2</id>
       <distanceMeters>4474.0</distanceMeters>
       <durationSeconds>796.0</durationSeconds>
    </searchAroundResult>
    <searchAroundResult>
       <id>1</id>
       <distanceMeters>10453.0</distanceMeters>
       <durationSeconds>1114.0</durationSeconds>
    </searchAroundResult>
 </searchAroundResponseV3>
```

Case of a problem with the graph: file missing, faulty filepath, etc… (searchAroundResponse/status is OK)

```
 <searchAroundResponseV3>
    <status>OK</status>
    <location>-1.593927,47.21858</location>
    <method>time</method>
    <searchAroundResult>
       <id>2</id>
       <distance>-1</distance>
       <time>-1</time>
    </searchAroundResult>
    <searchAroundResult>
       <id>1</id>
       <distance>-1</distance>
       <time>-1</time>
    </searchAroundResult>
 </searchAroundResponseV3>
```

Case of a snap-to-graph error (searchAroundResponse/status is OK)

```
 <searchAroundResponseV3>
    <status>OK</status>
    <location>-1.593927,47.21858</location>
    <method>time</method>
    <searchAroundResult>
       <id>2</id>
       <distance>-1</distance>
       <time>-1</time>
    </searchAroundResult>
    <searchAroundResult>
```

```
        <id>1</id>
        <distance>-1</distance>
        <time>-1</time>
    </searchAroundResult>
</searchAroundResponseV3>
```

Case of an absent location parameter (searchAroundResponse/status is ERROR)

```
<searchAroundResponseV3>
    <message>Location must be not null</message>
    <status>ERROR</status>
</searchAroundResponseV3>
```

Case of an incomplete location parameter, or one with the wrong separator (searchAroundResponse/status is ERROR)

```
<searchAroundResponseV3>
    <message>Location point must have 2 components separated with a ,</message>
    <status>ERROR</status>
</searchAroundResponseV3>
```

Case of the absence of a resource (searchAroundResponse/status is OK)

```
<searchAroundResponseV3>
    <status>OK</status>
</searchAroundResponseV3>
```

Case of a badly formatted resource (searchAroundResponse/status is ERROR)

```
<serviceResult>
    <message>ServiceException: not enough fields in candidate 1.-1.5939</message>
    <status>ERROR</status>
</serviceResult>
```

V2

Parameters / properties

Input

parameter	description	optional	default
location	Coordinates of the start point (or arrival point) in SOAP, points are stored in the "x" and "y" parameters themselves in the "geographicPoint" parameter in REST, the points are separated by the , character	no	
method	The shortest (distance) or fastest (time) route, or the route *as the crow flies* (flying)	yes	time
reverse	if *true*, the target is considered as an arrival	yes	false
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
exclusions	List of restriction rules to use, separated by the ";" character (Example: "Toll"; "Tunnel"; "Bridge")	yes	
resources	List of resources, separated by ";" characters.	yes	

parameter	description	optional	default
	Each resource takes the form "id,x,y,priority1,priority2"		
startDateTime	Start date and time (format ISO8601: local time) example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a start on 21 January 2014, at 9.00am in Paris	yes	
snapMethod	Snap to graph method - standard: to the nearest connectable road segment - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without any restriction rules	yes	standard

Output

Search Around (searchAroundWebResultsV2)

parameter	type	min/max	description
location	double	0/1	Coordinates of the start point (or arrival point) in SOAP, points are stored in the "x" and "y" parameters themselves in the "geographicPoint" parameter in REST, the points are separated by the , character
method	string	0/1	Specified input method (time, distance or flying), and time by default.
srs	string	0/1	Input projection specified.
exclusions	string	0/ unlimited	List of rules for specified input restrictions
searchAroundResult	searchAroundWebResultV2 (or array in JSON / JSON-P)	0/ unlimited	List of results for each candidate.

Search Around element(searchAroundWebResultV2)

parameter	type	min/max	description
id	string	0/1	Candidate identifier
distanceMeters	double	1/1	Distance in metres
durationSeconds	double	1/1	Time in seconds

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/searchAroundService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:searchAroundV2>
         <!--Optional:-->
         <request>
            <location>
               <x>-1.553927</x>
               <y>47.218580</y>
            </location>
            <!--Optional:-->
            <method>time</method>
            <reverse></reverse>
            <!--Optional:-->
            <srs></srs>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <resources>
               <!--Zero or more repetitions:-->
               <resource>
                  <x>-1.593927</x>
                  <y>47.188580</y>
                  <!--Optional:-->
                  <id>1</id>
                  <!--Optional:-->
                  <priority1>1</priority1>
                  <!--Optional:-->
                  <priority2>2</priority2>
               </resource>
               <resource>
               <x>-1.556927</x>
               <y>47.219580</y>
                  <!--Optional:-->
                  <id>2</id>
                  <!--Optional:-->
                  <priority1>1</priority1>
                  <!--Optional:-->
                  <priority2>2</priority2>
               </resource>
            </resources>
            <!--Optional:-->
            <startDateTime></startDateTime>
            <!--Optional:-->
            <snapMethod></snapMethod>
         </request>
      </sch:searchAroundV2>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:searchAroundV2Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <SearchAroundResult>
            <status>OK</status>
            <location>-1.553927,47.21858</location>
            <method>time</method>
            <srs/>
            <exclusions/>
            <searchAroundResult>
               <id>2</id>
               <distanceMeters>424.0</distanceMeters>
               <durationSeconds>147.0</durationSeconds>
            </searchAroundResult>
            <searchAroundResult>
               <id>1</id>
               <distanceMeters>6682.0</distanceMeters>
               <durationSeconds>1059.0</durationSeconds>
            </searchAroundResult>
         </SearchAroundResult>
      </ns2:searchAroundV2Response>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/searchAround/v2.json?
method=time&location=-1.593927,47.218580&resources=1,-1.593927,47.188580,1,2;2,-1.556927,47.219580,1,1
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/searchAround/v2.json?
method=time&location=-1.593927,47.218580&resources=1,-1.593927,47.188580,1,2;2,-1.556927,47.219580,1,1&callback=myCallback
```

XML query

```
http://<server>/<webapp>/api/lbs/searchAround/v2.xml?
method=time&location=-1.593927,47.218580&resources=1,-1.593927,47.188580,1,2;2,-1.556927,47.219580,1,1
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message": null,
   "status": "OK",
   "location": "-1.593927,47.21858",
   "method": "time",
   "srs": null,
   "exclusions": [],
   "searchAroundResult":    [
            {
         "id": "2",
         "distanceMeters": 3376,
         "durationSeconds": 735
      },
            {
         "id": "1",
         "distanceMeters": 10453,
         "durationSeconds": 1114
      }
   ]
}
```

JSON-P format

```
myCallback(
    {
       "message": null,
       "status": "OK",
       "location": "-1.593927,47.21858",
       "method": "time",
       "srs": null,
       "exclusions": [],
       "searchAroundResult":    [
                {
             "id": "2",
             "distanceMeters": 3376,
             "durationSeconds": 735
          },
                {
             "id": "1",
             "distanceMeters": 10453,
             "durationSeconds": 1114
          }
       ]
    }
);
```

XML format

```
<searchAroundResponse>
   <status>OK</status>
   <location>-1.593927,47.21858</location>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distanceMeters>3376.0</distanceMeters>
      <durationSeconds>735.0</durationSeconds>
   </searchAroundResult>
   <searchAroundResult>
      <id>1</id>
      <distanceMeters>10453.0</distanceMeters>
      <durationSeconds>1114.0</durationSeconds>
   </searchAroundResult>
</searchAroundResponse>
```

Possible responses

Case of a proximity search finding(searchAroundResponse/status is OK)

```
<searchAroundResponse>
   <status>OK</status>
   <location>-1.593927,47.21858</location>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distanceMeters>3376.0</distanceMeters>
      <durationSeconds>735.0</durationSeconds>
   </searchAroundResult>
   <searchAroundResult>
      <id>1</id>
      <distanceMeters>10453.0</distanceMeters>
      <durationSeconds>1114.0</durationSeconds>
   </searchAroundResult>
</searchAroundResponse>
```

Case of a problem with the graph: file missing, faulty filepath, etc… (searchAroundResponse/status is OK)

```
<searchAroundResponse>
   <status>OK</status>
   <location>-1.593927,47.21858</location>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
   <searchAroundResult>
      <id>1</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
</searchAroundResponse>
```

Case of a snap-to-graph error (searchAroundResponse/status is OK)

```
<searchAroundResponse>
   <status>OK</status>
   <location>-1.593927,47.21858</location>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
   <searchAroundResult>
      <id>1</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
</searchAroundResponse>
```

Case of an absent location parameter (searchAroundResponse/status is ERROR)

```
<searchAroundResponse>
   <message>Location must be not null</message>
   <status>ERROR</status>
</searchAroundResponse>
```

Case of an incomplete location parameter, or one with the wrong separator (searchAroundResponse/status is ERROR)

```
<searchAroundResponse>
   <message>Location point must have 2 components separated with a ,</message>
   <status>ERROR</status>
</searchAroundResponse>
```

Case of the absence of a resource (searchAroundResponse/status is OK)

```
<searchAroundResponse>
```

```
   <status>OK</status>
</searchAroundResponse>
```

Case of a badly formatted resource (searchAroundResponse/status is ERROR)

```
<serviceResult>
   <message>ServiceException: not enough fields in candidate 1.-1.5939</message>
   <status>ERROR</status>
</serviceResult>
```

V1

Parameters / properties

Input

parameter	description	optional	default
targetX	X coordinates or longitude of the target location	yes	
targetY	Y coordinates or latitude of the target location	yes	
searchMethod	Deprecated, replaced by method	yes	time
method	The shortest (distance) or fastest (time) route	yes	time
reverse	if *true*, the target is considered as a finish point	yes	false
projection	Deprecated, replaced by srs	yes	
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
rejectFlags	Deprecated, replaced by exclusions	yes	
exclusions	List of restriction rules to use, separated by the ";" character (Example: "Toll"; "Tunnel"; "Bridge")	yes	
resources	List of resources, separated by ";" characters. Each resource takes the form "id,x,y,priority1,priority2"	yes	

Output

Search Around (searchAroundWebResults)

parameter	type	min/max	description
targetX	double	1/1	X coordinates or longitude of the target location
targetY	double	1/1	Y coordinates or latitude of the target location
method	string	0/1	Specified input method
projection	string	0/1	Deprecated, replaced by srs
srs	string	0/1	Specified input projection
exclusions	string	0/ unlimited	List of restriction rules specified as input
searchAroundResult	array	0/ unlimited	List of responses

Search Around element(searchAroundWebResult)

parameter	type	min/max	description
id	string	0/1	Candidate identifier
distance	int	1/1	Distance in metres
time	int	1/1	Time in seconds

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/searchAroundService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header/>
   <soapenv:Body>
      <sch:searchAround>
         <!--Optional:-->
         <request>
            <!--Optional:-->
            <target>
               <x>-1.553927</x>
               <y>47.218580</y>
            </target>
            <!--Optional:-->
            <searchMethod></searchMethod>
            <!--Optional:-->
            <method>time</method>
            <reverse></reverse>
            <!--Optional:-->
            <srs></srs>
            <!--Optional:-->
            <exclusions>
               <!--Zero or more repetitions:-->
               <exclusion></exclusion>
            </exclusions>
            <!--Optional:-->
            <rejectFlags>
               <!--Zero or more repetitions:-->
               <rejectFlag></rejectFlag>
            </rejectFlags>
            <!--Optional:-->
            <resources>
               <!--Zero or more repetitions:-->
               <resource>
               <x>-1.593927</x>
               <y>47.188580</y>
                  <!--Optional:-->
                  <id>1</id>
                  <!--Optional:-->
                  <priority1>1</priority1>
                  <!--Optional:-->
                  <priority2>2</priority2>
               </resource>
               <resource>
               <x>-1.556927</x>
               <y>47.219580</y>
                  <!--Optional:-->
                  <id>2</id>
```

```
                    <!--Optional:-->
                    <priority1>1</priority1>
                    <!--Optional:-->
                    <priority2>2</priority2>
                 </resource>
              </resources>
           </request>
        </sch:searchAround>
     </soapenv:Body>
  </soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:searchAroundResponse xmlns:ns2="http://geoconcept.com/gc/schemas">
         <SearchAroundResult>
            <status>OK</status>
            <targetX>-1.553927</targetX>
            <targetY>47.21858</targetY>
            <method>time</method>
            <projection/>
            <srs/>
            <exclusions/>
            <searchAroundResult>
               <id>2</id>
               <distance>485</distance>
               <time>87</time>
            </searchAroundResult>
            <searchAroundResult>
               <id>1</id>
               <distance>6788</distance>
               <time>526</time>
            </searchAroundResult>
         </SearchAroundResult>
      </ns2:searchAroundResponse>
   </soap:Body>
</soap:Envelope>
```

REST

Query

JSON query

```
http://<server>/<webapp>/api/lbs/searchAround.json?
method=time&targetX=-1.593927&targetY=47.218580&resources=1,-1.593927,47.188580,1,2;2,-1.556927,47.219580,1,1
```

JSON-P query

```
http://<server>/<webapp>/api/lbs/searchAround.json?
method=time&targetX=-1.593927&targetY=47.218580&resources=1,-1.593927,47.188580,1,2;2,-1.556927,47.219580,1,1&callback=myCallb
```

XML query

```
http://<server>/<webapp>/api/lbs/searchAround.xml?
method=time&targetX=-1.593927&targetY=47.218580&resources=1,-1.593927,47.188580,1,2;2,-1.556927,47.219580,1,1
```

Response

The response is always in UTF-8 format.

JSON format

```
{
   "message": null,
   "status": "OK",
   "targetX": -1.593927,
   "targetY": 47.21858,
   "method": "time",
   "projection": null,
   "srs": null,
   "exclusions": [],
   "searchAroundResult":    [
            {
         "id": "2",
         "distance": 3880,
         "time": 777
      },
            {
         "id": "1",
         "distance": 10448,
         "time": 1178
      }
   ]
}
```

JSON-P format

```
myCallback(
        {
           "message": null,
           "status": "OK",
           "targetX": -1.593927,
           "targetY": 47.21858,
           "method": "time",
           "projection": null,
           "srs": null,
           "exclusions": [],
           "searchAroundResult":    [
                                {
                         "id": "2",
                         "distance": 3880,
                         "time": 777
                  },
                                {
                         "id": "1",
                         "distance": 10448,
                         "time": 1178
                  }
           ]
        }
);
```

XML format

```
<searchAroundResponse>
   <status>OK</status>
   <targetX>-1.593927</targetX>
   <targetY>47.21858</targetY>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distance>3880</distance>
      <time>777</time>
   </searchAroundResult>
   <searchAroundResult>
      <id>1</id>
      <distance>10448</distance>
      <time>1178</time>
   </searchAroundResult>
</searchAroundResponse>
```

Possible responses

Case of a proximity search finding(searchAroundResponse/status is OK)

```
<searchAroundResponse>
   <status>OK</status>
   <targetX>-1.593927</targetX>
   <targetY>47.21858</targetY>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distance>3880</distance>
      <time>777</time>
   </searchAroundResult>
   <searchAroundResult>
      <id>1</id>
      <distance>10448</distance>
      <time>1178</time>
   </searchAroundResult>
</searchAroundResponse>
```

Case of a problem with the graph: file missing, faulty filepath, etc… (searchAroundResponse/status is OK)

```
<searchAroundResponse>
   <status>OK</status>
   <targetX>-1.593927</targetX>
   <targetY>47.21858</targetY>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
   <searchAroundResult>
      <id>1</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
</searchAroundResponse>
```

Case of a snap-to-graph error (searchAroundResponse/status is OK)

```
<searchAroundResponse>
```

```
   <status>OK</status>
   <targetX>-51.593927</targetX>
   <targetY>47.21858</targetY>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
   <searchAroundResult>
      <id>1</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
</searchAroundResponse>
```

Case of unspecified target point (searchAroundResponse/status is OK)

```
<searchAroundResponse>
   <status>OK</status>
   <targetX>0.0</targetX>
   <targetY>0.0</targetY>
   <method>time</method>
   <searchAroundResult>
      <id>2</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
   <searchAroundResult>
      <id>1</id>
      <distance>-1</distance>
      <time>-1</time>
   </searchAroundResult>
</searchAroundResponse>
```

Case of the absence of a resource (searchAroundResponse/status is OK)

```
<searchAroundResponse>
   <status>OK</status>
   <targetX>0.0</targetX>
   <targetY>0.0</targetY>
</searchAroundResponse>
```

Case of a badly formatted resource (searchAroundResponse/status is ERROR)

```
<serviceResult>
   <message>ServiceException: not enough fields in candidate 2,-1.556927,</message>
   <status>ERROR</status>
</serviceResult>
```

FAQ

1. Is it possible to give priority to either journey time or distance?

   Yes, by changing the method: distance, time or flying = as the crow flies (at a speed of 30 km/h).

2. Can I use aliases instead of .siti file names to call a datasource?

Yes, refer to details in the FAQ of the reverse geocoding Web Service.

3. How can I use route statistics?

   Check that the graph does contain these statistics and use these two parameters: `computeOptions` with a value of *trafficPatterns* and `startDateTime` to specify the departure date/time.

4. How is the classification of returned addresses structured, in relation to distance, time and priorities generally?

   The classification creates a hierarchy of priority 1 in increasing order, then priority 2 in increasing order, then by time or distance or distance as the crow flies in increasing order.

5. What format should the priority take and is it possible to define a classification order? (increasing/ decreasing)

   The priority is an integer, and currently the result is always in increasing order. To obtain the reverse order, you should put n-priority in the attribute.

6. If just one address in the list has a priority of 0, is the priority taken into account for any or all of the addresses in the list?

   Currently, a priority of 0 is not treated in any special way. This means you need to set all priorities to 0 if you want to ignore the criterion.

7. How can I perform a route calculation excluding toll roads?

   If the *Toll* constraint has been included in the graph, place an exclusion in `exclusions` :

```
<sch:searchAroundV2>
 <!--Optional:-->
 <request>
    <location>
       <x>-1.553927</x>
       <y>47.218580</y>
    </location>
    <!--Optional:-->
    <method>time</method>
    <reverse></reverse>
    <!--Optional:-->
    <srs></srs>
    <!--Optional:-->
    <exclusions>
       <!--Zero or more repetitions:-->
       <exclusion>Toll</exclusion>
    </exclusions>
    <!--Optional:-->
    <resources>
       <!--Zero or more repetitions:-->
       <resource>
          <x>-1.593927</x>
          <y>47.188580</y>
          <!--Optional:-->
          <id>1</id>
          <!--Optional:-->
          <priority1>1</priority1>
          <!--Optional:-->
          <priority2>2</priority2>
       </resource>
```

```
      </resources>
      <!--Optional:-->
      <startDateTime></startDateTime>
   </request>
  </sch:searchAroundV2>
```

8. What are Speed Patterns? How can I use them?

   In order to propose journey times that are as accurate as possible and reflect changing traffic conditions, graphs supplied by GEOCONCEPT SAS include, from version M18 upwards, 5 speed profiles (Speed Patterns) for cars and lorries, to take into account the different levels of congestion during the course of one day:

   - standard *normal-speed* corresponds to the speed at a time when the roads have an average congestion level (eg 11.00am)
   - night *fast-speed* corresponds to a very fluid traffic situation, often experienced at night-time (3.00am)
   - congested *slow-speed* corresponds to the times when traffic is densest (8.00am)
   - rush hour *very-slow-speed* corresponds to times when traffic is densest in urban and built-up areas, and is slower than the speed above (8.00am)
   - default *default* corresponds to speeds averaged out over an entire day

   To use them, you need to pass to parameter, when calling the web service, the `computeOptions` parameter with the *speedPattern* option and specify the value of the Speed Pattern requested, without forgetting a vehicle profile
   Example:

+

```
&computeOptions=speedPattern:fast-speed&profileId=1
```

How to use Heavy Goods Vehicle attributes?

The graph must include the attributes for Heavy Goods Vehicles (as standard in the graphs supplied by GEOCONCEPT SAS from version M18 upwards) and either calculate an itinerary using a vehicle profile using restrictions (cf. the catalogue of vehicles, editable, defined in the SmartRoutingVehicles.xml file, stored in the folder `‘*<GEOCONCEPT_WEB_HOME>”\smartrouting \jee\smartrouting\conf\* ), or overwrite  the call to the web service by using the `computeOptions` parameter with the options *length*, *width*, *height*, *weight*, *axles* and/or *weightPerAxle*.
Example for a journey with a vehicle 4.5 metres high:

```
&computeOptions=height:450
```

Search ALong

(fr) Cette page n’est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

The Search Along web service allows you to identify the best candidates from which to select the next nearest step in a pre-existing itinerary.

The algorithm explores all possible solutions before returning the best candidate selected as a function of best score for the requested criteria.

Availability

This web service is available at all times with Geoconcept Web and a road graph or network.

Parameters / properties

Input

parameter	description	optional	default
routes array of routes/ route (inputRoute)	Table of journey input data. Pre-existing itineraries in which candidate steps must be inserted.	yes *	
routeNodes array of routeNodes/routeNode (inputRouteNode)	Table of journey input data based on graph nodes (for the fastest calculation). Caution: a physical node does not have the same ID in another graph.	yes *	
resources	Table of candidates	yes	
method	Shortest (distance) or fastest (time) route	yes	time
exclusions	List of restriction rules to use, separated by the ";" character (Example: Toll, Tunnel, Bridge)	yes	
snapMethod	Snap-to-graph method - standard: to the nearest connectable road section - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only - unrestricted: without any restriction rules - nodes: Snap directly to nodes provided by the locationNode parameter or, if no value has been assigned, to the node nearest to the location parameter	yes	standard
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84: `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))` NOTE: WKT geometries must be closed.	yes	
startDateTime	Departure Date and Time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a departure date on 21 Januray 2014, at 9.00am in Paris. Caution: the + character may be misinterpreted by some browsers, and in this instance, you will need to replace it with *%2B*.	yes	
graphName (depreciated)	Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
profileId (depreciated)	Vehicle identifier (saved under vehicle profiles)	yes	

parameter	description	optional	default
	This parameter is omitted if the configName parameter is used.		
profileName (depreciated)	Vehicle profile (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
configName	Name of the configuration to use (defined in Geoconcept Web - Administration / Tools / Road graphs) This replaces the use of graphName, profileId and profileName	yes	
computeOptions	List of options for the calculation,separated by *;* characters - trafficPatterns: uses routing statistics (you will need to supply a value for the startDateTime parameter and use a graph that includes traffic patterns information) - speedPattern (M18): uses a *speed pattern* as defined in the SmartRoutingVehicles.xml file, located in the *`‘<GEOCONCEPT_WEB_HOME>’’\smartrouting\jee\smartrouting\conf\* folder. Use as follows: "speedPattern:slow-speed" - length (M18): maximum authorised length in centimeters (you need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "length:950" - width (M18): maximum authorised width in centimeters (You will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "width:255" - height (M18): maximum authorised height in centimeters (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "height:360" - weight (M18): maximum authorised weight in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weight:18000" - axles (M18): maximum authorised number of axles (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "axles:2" - weightPerAxle (M18): maximum authorised weight per axle in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weightPerAxle:9000" - snapSpeed: snap-to-graph speed in kilometers per hour. Use as follows: "snapSpeed:10"	yes	
maxDetourDurationSeconds	Filter: maximum detour authorised (in seconds)	yes	
maxDetourDistanceMeters	Filter: maximum detour authorised (in metres)	yes	
maxDetourOriginDurationSeconds	Filter: maximum authorised duration between the journey start point and the candidate (in seconds)	yes	
maxDetourOriginDistanceMeters	Filter: maximum authorised detour between the candidate and the journey arrival point (in metres)	yes	
maxDetourDestinationDurationSeconds	Filter: maximum authorised duration between the candidate and the journey arrival point (in seconds)	yes	
maxDetourDestinationDistanceMeters	Filter: maximum authorised detour from the arrival point (in metres)	yes	
timeOut	Time out for the calculation (in milliseconds)	yes	

(*) At least one of the two parameters routes and routeNodes must be assigned a value.

(M18) Available from version M18 and later versions of graphs supplied by GEOCONCEPT SAS.

Journeys in input (inputRoute)

parameter	description	optional	default
id	Journey identifier	No	
departurePoint (geographicPoint)	Coordinates of the journey departure point	No	
arrivalPoint (geographicPoint)	Coordinates of the journey arrival point	No	

Coordinates (geographicPoint)

parameter	description	optional	default
x	First coordinate or longitude	No	
y	Second coordonnée or latitude	No	

Journeys in input based on the graph nodes (inputRouteNode)

parameter	description	optional	default
id	Journey identifier	No	
departureNode (geographicPoint)	Journey departure node	No	
arrivalNode (geographicPoint)	Journey arrival node	No	

Candidates (searchAlongResource)

parameter	description	optional	default
id	Candidate identifier	No	
x	First coordinate or longitude	No	
y	Second coordonnée or latitude	No	
node	Candidate node	Yes	
priority1	Priority 1	Yes	0
priority2	Priority 2	Yes	0

Output

parameter	type	min/max	description
routes	array of routes/route (searchAlongRouteResult)	0/ unlimited	List of calculated journeys.

List of calculated journeys (searchAlongRouteResult)

parameter	type	min/max	description
id	string	0/1	Journey identifier
directDistanceMeters	double	1/1	Total distance without detour (in metres)
directDurationSeconds	double	1/1	Total duration without detour (in seconds)

parameter	type	min/max	description
resources	(searchAlongResourceResult)	0/unlimited	List of candidates

Candidates (searchAlongResourceResult)

parameter	type	min/max	description
id	string	0/1	Candidate identifier
totalDistanceMeters	double	1/1	Total distance with detour (in metres)
totalDurationSeconds	double	1/1	Total duration with detour (in seconds)
detourDistanceMeters	double	1/1	Detour distance (in metres)
detourDurationSeconds	double	1/1	Detour duration (in seconds)
detourOriginDistanceMeters	double	1/1	Detour distance from the journey departure point to the candidate (in metres)
detourOriginDurationSeconds	double	1/1	Detour duration from the journey departure point to the candidate (in seconds)
detourDestinationDistanceMeters	double	1/1	Detour distance from the candidate to the arrival point (in metres)
detourDestinationDurationSeconds	double	1/1	Detour duration from the candidate to the arrival point (in seconds)

SOAP

WSDL

http://*<server>*/*<webapp>*/api/ws/searchAlongService?wsdl

Query

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sch="http://
geoconcept.com/gc/schemas">
   <soapenv:Header />
   <soapenv:Body>
      <sch:searchAlongV1>
         <!--Optional:-->
         <request>
            <routes>
               <route>
                  <id>dep</id>
                  <departurePoint>
                     <x>-1.553927</x>
                     <y>47.21858</y>
                  </departurePoint>
                  <arrivalPoint>
                     <x>-1.593927</x>
                     <y>47.18858</y>
                  </arrivalPoint>
               </route>
            </routes>
            <resources>
               <resource>
                  <id>res1</id>
                  <node />
                  <priority1>1</priority1>
                  <priority2>2</priority2>
                  <x>-1.511092</x>
                  <y>47.208355</y>
               </resource>
               <resource>
                  <id>res2</id>
                  <node />
                  <priority1>1</priority1>
                  <priority2>1</priority2>
                  <x>-1.549524</x>
                  <y>47.195484</y>
               </resource>
            </resources>
            <srs>epsg:4326</srs>
            <maxDetourDestinationDistanceMeters>-1</maxDetourDestinationDistanceMeters>
            <maxDetourDestinationDurationSeconds>-1</maxDetourDestinationDurationSeconds>
            <maxDetourDistanceMeters>-1</maxDetourDistanceMeters>
            <maxDetourDurationSeconds>-1</maxDetourDurationSeconds>
            <maxDetourOriginDistanceMeters>-1</maxDetourOriginDistanceMeters>
            <maxDetourOriginDurationSeconds>-1</maxDetourOriginDurationSeconds>
            <method>time</method>
                        <timeOut></timeOut>
         </request>
      </sch:searchAlongV1>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:searchAlongV1Response xmlns:ns2="http://geoconcept.com/gc/schemas">
         <SearchAlongResult>
            <status>OK</status>
            <routes>
               <route>
                  <id>dep</id>
                  <directDistanceMeters>6850.71</directDistanceMeters>
                  <directDurationSeconds>973.14</directDurationSeconds>
                  <resources>
                     <id>res2</id>
                     <totalDistanceMeters>7808.36</totalDistanceMeters>
                     <totalDurationSeconds>1197.28</totalDurationSeconds>
                     <detourDistanceMeters>957.65</detourDistanceMeters>
                     <detourDurationSeconds>224.14</detourDurationSeconds>
                     <detourOriginDistanceMeters>3684.98</detourOriginDistanceMeters>
                     <detourOriginDurationSeconds>752.34</detourOriginDurationSeconds>
                     <detourDestinationDistanceMeters>4123.38</detourDestinationDistanceMeters>
                     <detourDestinationDurationSeconds>444.94</detourDestinationDurationSeconds>
                  </resources>
                  <resources>
                     <id>res1</id>
                     <totalDistanceMeters>13714.6</totalDistanceMeters>
                     <totalDurationSeconds>2261.52</totalDurationSeconds>
                     <detourDistanceMeters>6863.89</detourDistanceMeters>
                     <detourDurationSeconds>1288.38</detourDurationSeconds>
                     <detourOriginDistanceMeters>5731.27</detourOriginDistanceMeters>
                     <detourOriginDurationSeconds>1156.94</detourOriginDurationSeconds>
                     <detourDestinationDistanceMeters>7983.33</detourDestinationDistanceMeters>
                     <detourDestinationDurationSeconds>1104.58</detourDestinationDurationSeconds>
                  </resources>
```

```
                </route>
             </routes>
             <computationTime>118.72</computationTime>
          </SearchAlongResult>
       </ns2:searchAlongV1Response>
    </soap:Body>
</soap:Envelope>
```

REST (POST JSON)

Query

Query

```
http://<server>/<webapp>/api/lbs/searchAlong.json
```

Data (JSON)

```
{
   "routes":[
      {
         "id":"dep",
         "departurePoint":{
            "x":-1.553927,
            "y":47.218580
         },
         "arrivalPoint":{
            "x":-1.593927,
            "y":47.188580
         }
      }
   ],
   "resources":[
      {
         "id":"res1",
         "node":"",
         "priority1":1,
         "priority2":2,
         "x":-1.511092,
         "y":47.208354
      }
   ],
   "resources":[
      {
         "id":"res2",
         "node":"",
         "priority1":1,
         "priority2":1,
         "x":-1.549524,
         "y":47.195483
      }
   ],
   "method":"time",
   "srs":"epsg:4326",
   "maxDetourDurationSeconds":-1,
   "maxDetourDistanceMeters":-1,
   "maxDetourOriginDurationSeconds":-1,
   "maxDetourOriginDistanceMeters":-1,
   "maxDetourDestinationDurationSeconds":-1,
   "maxDetourDestinationDistanceMeters":-1
}
```

Response

The response is always in UTF-8 format.

JSON format

```
{
    "message": null,
    "status": "OK",
    "routes": [
        {
            "id": "dep",
            "directDistanceMeters": 6850.71,
            "directDurationSeconds": 973.14,
            "resources": [
                {
                    "id": "res2",
                    "totalDistanceMeters": 7808.36,
                    "totalDurationSeconds": 1197.28,
                    "detourDistanceMeters": 957.65,
                    "detourDurationSeconds": 224.14,
                    "detourOriginDistanceMeters": 3684.98,
                    "detourOriginDurationSeconds": 752.34,
                    "detourDestinationDistanceMeters": 4123.38,
                    "detourDestinationDurationSeconds": 444.94
                }
            ]
        }
    ]
}
```

REST (POST XML)

Query

Query

```
http://<server>/<webapp>/api/lbs/searchAlong.xml
```

Data (XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<searchAlongRequest>
   <routes>
      <route>
         <id>dep</id>
         <departurePoint>
            <x>-1.553927</x>
            <y>47.21858</y>
         </departurePoint>
         <arrivalPoint>
            <x>-1.593927</x>
            <y>47.18858</y>
         </arrivalPoint>
      </route>
   </routes>
   <resources>
      <resource>
         <id>res1</id>
         <node />
         <priority1>1</priority1>
         <priority2>2</priority2>
         <x>-1.511092</x>
         <y>47.208355</y>
      </resource>
      <resource>
         <id>res2</id>
         <node />
         <priority1>1</priority1>
         <priority2>1</priority2>
         <x>-1.549524</x>
         <y>47.195484</y>
      </resource>
   </resources>
   <srs>epsg:4326</srs>
   <maxDetourDestinationDistanceMeters>-1</maxDetourDestinationDistanceMeters>
   <maxDetourDestinationDurationSeconds>-1</maxDetourDestinationDurationSeconds>
   <maxDetourDistanceMeters>-1</maxDetourDistanceMeters>
   <maxDetourDurationSeconds>-1</maxDetourDurationSeconds>
   <maxDetourOriginDistanceMeters>-1</maxDetourOriginDistanceMeters>
   <maxDetourOriginDurationSeconds>-1</maxDetourOriginDurationSeconds>
   <method>time</method>
</searchAlongRequest>
```

Response

The response is always in UTF-8 format.

XML format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<searchAlongResult>
    <status>OK</status>
    <routes>
        <route>
            <id>dep</id>
            <directDistanceMeters>6850.71</directDistanceMeters>
            <directDurationSeconds>973.14</directDurationSeconds>
            <resources>
                <id>res2</id>
                <totalDistanceMeters>7808.36</totalDistanceMeters>
                <totalDurationSeconds>1197.28</totalDurationSeconds>
                <detourDistanceMeters>957.65</detourDistanceMeters>
                <detourDurationSeconds>224.14</detourDurationSeconds>
                <detourOriginDistanceMeters>3684.98</detourOriginDistanceMeters>
                <detourOriginDurationSeconds>752.34</detourOriginDurationSeconds>
                <detourDestinationDistanceMeters>4123.38</detourDestinationDistanceMeters>
                <detourDestinationDurationSeconds>444.94</detourDestinationDurationSeconds>
            </resources>
            <resources>
                <id>res1</id>
                <totalDistanceMeters>13714.6</totalDistanceMeters>
                <totalDurationSeconds>2261.52</totalDurationSeconds>
                <detourDistanceMeters>6863.89</detourDistanceMeters>
                <detourDurationSeconds>1288.38</detourDurationSeconds>
                <detourOriginDistanceMeters>5731.27</detourOriginDistanceMeters>
                <detourOriginDurationSeconds>1156.94</detourOriginDurationSeconds>
                <detourDestinationDistanceMeters>7983.33</detourDestinationDistanceMeters>
                <detourDestinationDurationSeconds>1104.58</detourDestinationDurationSeconds>
            </resources>
        </route>
    </routes>
</searchAlongResult>
```

Possible responses

Case of a search around found (searchAlongResult/status is OK)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<searchAlongResult>
    <status>OK</status>
    <routes>
        <route>
            <id>dep</id>
            <directDistanceMeters>6850.71</directDistanceMeters>
            <directDurationSeconds>973.14</directDurationSeconds>
            <resources>
                <id>res2</id>
                <totalDistanceMeters>7808.36</totalDistanceMeters>
                <totalDurationSeconds>1197.28</totalDurationSeconds>
                <detourDistanceMeters>957.65</detourDistanceMeters>
                <detourDurationSeconds>224.14</detourDurationSeconds>
                <detourOriginDistanceMeters>3684.98</detourOriginDistanceMeters>
                <detourOriginDurationSeconds>752.34</detourOriginDurationSeconds>
                <detourDestinationDistanceMeters>4123.38</detourDestinationDistanceMeters>
                <detourDestinationDurationSeconds>444.94</detourDestinationDurationSeconds>
            </resources>
            <resources>
                <id>res1</id>
                <totalDistanceMeters>13714.6</totalDistanceMeters>
                <totalDurationSeconds>2261.52</totalDurationSeconds>
                <detourDistanceMeters>6863.89</detourDistanceMeters>
                <detourDurationSeconds>1288.38</detourDurationSeconds>
                <detourOriginDistanceMeters>5731.27</detourOriginDistanceMeters>
                <detourOriginDurationSeconds>1156.94</detourOriginDurationSeconds>
                <detourDestinationDistanceMeters>7983.33</detourDestinationDistanceMeters>
                <detourDestinationDurationSeconds>1104.58</detourDestinationDurationSeconds>
            </resources>
        </route>
    </routes>
</searchAlongResult>
```

Case of an incorrect SRS

```
{"message":"NullPointerException: null","status":"ERROR"}
```

Case of an absent resource (searchAroundResponse/status is OK)

```
{"message":"Input resources list must not be null and empty!","status":"ERROR","routes":[]}
```

FAQ

1. Is it possible to give priority to either journey time or distance?

   Yes, by changing the method `method` distance or time

2. Can I use aliases instead of .siti file names to call a datasource?

   Yes, refer to details in the FAQ of the reverse geocoding Web Service.

3. How can I use route statistics?

   Check that the graph does contain these statistics and use these two parameters: `computeOptions` with a value of *trafficPatterns* and `startDateTime` to specify the departure date/time.

4. How is the classification of returned addresses structured, in relation to distance, time and priorities generally?

   The classification creates a hierarchy of priority 1 in increasing order, then priority 2 in increasing order, then by time or distance or distance as the crow flies in increasing order.

5. What format should the priority take and is it possible to define a classification order? (increasing/ decreasing)

   The priority is an integer, and currently the result is always in increasing order. To obtain the reverse order, you should put n-priority in the attribute.

6. If just one address in the list has a priority of 0, is the priority taken into account for any or all of the addresses in the list?

   Currently, a priority of 0 is not treated in any special way. This means you need to set all priorities to 0 if you want to ignore the criterion.

7. How can I perform a route calculation excluding toll roads?

   If the *Toll* constraint has been included in the graph, place an exclusion in `exclusions`

8. What are Speed Patterns? How can I use them?

   In order to propose journey times that are as accurate as possible and reflect changing traffic conditions, graphs supplied by GEOCONCEPT SAS include, from version M18 upwards, 5 speed profiles (Speed Patterns) for cars and lorries, to take into account the different levels of congestion during the course of one day:

   - standard *normal-speed* corresponds to the speed at a time when the roads have an average congestion level (eg 11.00am)
   - night *fast-speed* corresponds to a very fluid traffic situation, often experienced at night-time (3.00am)
   - congested *slow-speed* corresponds to the times when traffic is densest (8.00am)
   - rush hour *very-slow-speed* corresponds to times when traffic is densest in urban and built-up areas, and is slower than the speed above (8.00am)
   - default *default* corresponds to speeds averaged out over an entire day

To use them, you need to pass to parameter, when calling the web service, the `computeOptions` parameter with the *speedPattern* option and specify the value of the Speed Pattern requested, without forgetting a vehicle profile

Example:

+

```
&computeOptions=speedPattern:fast-speed&profileId=1
```

How to use Heavy Goods Vehicle attributes?

> The graph must include the attributes for Heavy Goods Vehicles (as standard in the graphs supplied by GEOCONCEPT SAS from version M18 upwards) and either calculate an itinerary using a vehicle profile using restrictions (cf. the catalogue of vehicles, editable, defined in the SmartRoutingVehicles.xml file, stored in the folder `*‘<GEOCONCEPT_WEB_HOME>”\smartrouting \jee\smartrouting\conf\* ), or overwrite the call to the web service by using the **`computeOptions`** parameter with the options *length*, *width*, *height*, *weight*, *axles* and/or *weightPerAxle*.
> Example for a journey with a vehicle 4.5 metres high:
>
> ```
> &computeOptions=height:450
> ```

Pickup and Delivery

(fr) Cette page n’est plus maintenue depuis la version 2022 de Geoconcept Web. Pour accéder à la documentation à jour de ce web service, veuillez suivre ce lien [https://mygeoconcept.com/doc/geoapi/docs/en/geoptimization-api-book/geoptimization-api-intro.html].

Basic principles

The pickup and delivery service can be used to list the best solutions for pickup/collection and drop-off/delivery, whether for the transport of people or of goods.

The web service takes two types of data input:

- Existing journeys
- Pickup/collection points and drop-off/delivery points

The algorithm explores the solutions to find a pair that includes one pick-up point and one drop-off point in the existing journeys. It returns, for each journey, the best detours possible among the candidates (pick-up and drop-off points).

In the case where only one candidate (pickup or drop-off) is available, the web service tries to insert one step in existing journeys instead of two.

Availability

This web service is an option in Geoconcept Web : please contact us to find out how you can purchase the service.

It requires access to an accelerated graph or network.

Parameters / properties

Input

parameter	type	optional	description
routes	array of routes/route (pickupDeliveryRouteInput)	No	Table of journeys as input. Pre-existing journeys in which candidates must be inserted.
pickupPoints	array of pickupPoints/pickupPoint (wayPoint)	Yes	Table of candidate pick-up points to insert in journeys.
deliveryPoints	array of deliveryPoints/deliveryPoint (wayPoint)	Yes	Table of possible drop-off candidates to insert in journeys.
constraints	(pickupDeliveryConstraints)	Yes	Definition of constraints to filter the journeys to use.
sortOptions	(pickupDeliverySortingOptions)	Yes	Definition of the sort on the results.
srs	projection (EPSG code such as epsg:4326 or wgs84)	yes	
graphName (depreciated)	Name of the graph to use This parameter is omitted if the configName parameter is used.	yes	
profileId (depreciated)	Vehicle identifier (saved under vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
profileName (depreciated)	Vehicle profile (saved in the vehicle profiles) This parameter is omitted if the configName parameter is used.	yes	
configName	Name of the configuration to use (defined in Geoconcept Web - Administration / Tools / Road graphs) This replaces the use of graphName, profileId and profileName	yes	
snapMethod	Snap to graph method - standard: to the nearest connectable road section - extended: via restricted road sections (pedestrian routes…) - nearest: to the nearest road section only	yes	standard

parameter	type	optional	description
	- unrestricted: without any restriction rules - nodes: Snap directly to nodes supplied by the originNode, destinationNode and waypointNodes parameters, or, if these parameters have not been set, to the nearest nodes sourced by the origin, destination and waypoint parameters		
exclusions	List of restriction rules to use, separated by the *;* character (Example: Toll, Tunnel, Bridge)	yes	
startDateTime	Start date and time (format ISO8601: local time) Example: 2014-01-21T09:00:00.000+01:00 (or 2014-01-21T09:00:00.000%2B01:00) for a start date of 21 January 2014, at 9.00am in Paris. Caution: the + character can be misinterpreted by browsers, so in this case, it should be replaced by *%2B*.	yes	
avoidArea	Forbidden transit zone in WKT format (POLYGON or MULTIPOLYGON) in the requested projection (srs parameter) Example in wgs84: `POLYGON ((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689))` - `MULTIPOLYGON (((-1.556892 47.21689, -1.556892 47.216904, [...] -1.556892 47.21689)), ((-1.558639 47.218144, -1.558639 47.218193, [...] -1.558639 47.218144)))` Care: WKT geometries must be closed.	yes	
size	(pickupDeliverySizes)	No	Number of journeys and candidates to return.
dataVersionHash	string	Yes	Graph identifier (non-utilised)
computeOptions	List of options for the calculation,separated by *;* characters - trafficPatterns: uses routing statistics (you will need to supply a value for the startDateTime parameter and use a graph that includes traffic patterns information) - speedPattern (M18): uses a *speed pattern* as defined in the SmartRoutingVehicles.xml file, located in the `*‘<GEOCONCEPT_WEB_HOME>’’\smartrouting\jee\smartrouting\conf\* folder. Use as follows: "speedPattern:slow-speed" - length (M18): maximum authorised length in centimeters (you need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "length:950" - width (M18): maximum authorised width in centimeters (You will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "width:255" - height (M18): maximum authorised height in centimeters (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "height:360" - weight (M18): maximum authorised weight in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weight:18000" - axles (M18): maximum authorised number of axles (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "axles:2"	yes	

parameter	type	optional	description
	- weightPerAxle (M18): maximum authorised weight per axle in kilograms (you will need to use a graph that includes Heavy Goods Vehicle attributes). Use as follows: "weightPerAxle:9000" - snapSpeed: snap-to-graph speed in kilometers per hour. Use as follows: "snapSpeed:10"		

(pickupDeliveryRouteInput) journeys as input

parameter	type	optional	description
id	string	No	Journey identifier
departure	(location)	No	Start point coordinates
arrival	(location)	No	Finish point coordinates
distance	long	Yes	Journey distance (in metres)
duration	long	Yes	Journey duration (in seconds)
waypoints	Array of wayPoint	Yes	List of journey steps

(wayPoint) steps

parameter	type	optional	description
id	string	No	Step identifier
location	(location)	No	Step coordinates
stopoverPosition	long	Yes	Position of the step in the journey, if the step is a stop. If no values are assigned, this step is just a waypoint.

(location) coordinates

parameter	type	optional	description
lat	double	No	Latitude of the point (in wgs84)
long	double	No	Longitude of the point (in wgs84)
nodeId	string	Yes	Graph node. Caution: a physical node does not have the same ID in another graph.

(pickupDeliveryConstraints) constraints

parameter	type	optional	description
maxDetourDuration	long	Yes	Maximum duration of detour (in seconds). Only resulting journeys for which the duration added is less than this limit will be returned by the web service. The detour duration is defined as: (duration with detour) - (duration without detour)
minSharedDistance	long	Yes	Minimum value of common

parameter	type	optional	description
			distance ratio. Only those resulting journeys for which the common distance ratio exceeds this limit will be returned by the web service. The common distance ratio is defined as: (common distance) / (total distance with detour)

(pickupDeliverySortingOptions) sort

parameter	type	optional	description
routesAndCandidatesSortCriterium	(sortCriteriaEnum)	Yes	Sort options: - MIN_DETOUR_DURATION = sort (ascending) results according to detour duration - MIN_DETOUR_DISTANCE = sort (ascending) results according to detour distance - MAX_SHARED_DISTANCE = sort

parameter	type	optional	description
			(descending) results according to common distance ratio

(pickupDeliverySizes) number of returns

parameter	type	optional	description
routes	long	No	Maximum number of journeys to return.
candidates	long	No	Maximum number of candidates to return for each journey.

Output

parameter	type	description
results	array of results/result (pickupDeliveryRouteResult)	List of journeys calculated.

(pickupDeliveryRouteResult) list of journeys calculated

parameter	type	description
routeId	string	List of journeys calculated.
candidates	array of candidates/ candidate (pickupDeliveryRouteCandidate)	List of possible detours for the journey.

(pickupDeliveryRouteCandidate) list of detours possible for the journey

parameter	type	description
pickup	(meetingPointCandidate)	Pick-up candidate.
delivery	(meetingPointCandidate)	Drop-off candidate.
sharedRouteDuration	long	Common journey duration (in seconds).
sharedRouteDistance	long	Common journey distance (in metres).
totalDurationWithDetour	long	Total journey duration including the detour (in seconds).
totalDistanceWithDetour	long	Total journey distance including the detour (in metres).
detourDuration	long	Detour duration = (duration with detour) - (original journey duration)
detourDistance	long	Detour distance = (distance with detour) - (original journey distance)
sharedDistance	double	Common distance = ratio between (common distance) and (Total journey distance including the detour)

(meetingPointCandidate) candidates

parameter	type	description
meetingPointId	string	Identifier for the pickup/drop-off location
duration	long	For a pickup link: duration from the journey departure point to the pickup point (in seconds) For a drop-off point: duration from the journey drop-off point to the arrival point (in seconds)
distance	long	For a pickup link: distance from the journey departure point to the pickup point (in metres) For a drop-off point: the distance from the journey drop-off point to the arrival point (in metres)
meetingPointSubPath	int	Optimum segment position (in the journey) for the pickup/drop-off point.

REST (POST)

Query

Query

```
http://<server>/<webapp>/api/lbs/pickupDelivery.json
```

Data (JSON)

```
{
  "routes" : [ {
    "id" : "trip_1",
    "departure" : {
      "lat" : "47.333990",
      "lon" : "-1.805604"
    },
    "arrival" : {
      "lat" : "47.479740",
      "lon" : "-1.095251"
    },
    "waypoints" : [ {
      "id" : "waypoint_1",
      "location" : {
        "lat" : "47.157530",
        "lon" : "-1.421958"
      }
    }]
  }],
  "pickupPoints" : [ {
    "id" : "depMeet_1",
    "location" : {
      "lat" : "47.228659",
      "lon" : "-1.600995"
    }
  }],
  "deliveryPoints" : [ {
    "id" : "arrMeet_1",
    "location" : {
      "lat" : "47.293823",
      "lon" : "-1.480789"
    }
  }],
  "constraints" : {
    "maxDetourDuration" : 3600,
    "minSharedDistance" : 0.1
  },
  "sortOptions" : {
    "routesAndCandidatesSortCriterium" : "MIN_DETOUR_DURATION"
  },
  "size" : {
    "routes" : 1,
    "candidates" : 1
  }
}
```

Query

Response

The response is always in UTF-8 format.

JSON format

```
  {
      "message": null,
      "status": "OK",
      "results": [
          {
              "routeId": "trip_1",
              "candidates": [
                  {
                      "pickup": {
                          "meetingPointId": "depMeet_1",
                          "duration": 1349,
                          "distance": 23426,
                          "meetingPointSubPath": 0
                      },
                      "delivery": {
                          "meetingPointId": "arrMeet_1",
                          "duration": 2177,
                          "distance": 43159,
                          "meetingPointSubPath": 0
                      },
                      "sharedRouteDuration": 1333,
                      "sharedRouteDistance": 18767,
                      "totalDurationWithDetour": 4860,
                      "totalDistanceWithDetour": 85352,
                      "detourDuration": -577,
                      "detourDistance": -19590,
                      "sharedDistance": 0.21
                  }
              ]
          }
      ]
  }
```

FAQ

1. How can I use route statistics?

   Check that the graph does contain these statistics and use these two parameters: `computeOptions` with a value of *trafficPatterns* and `startDateTime` to specify the departure date/time.

2. How can I perform a route calculation excluding toll roads?

   If the *Toll* constraint has been included in the graph, place an exclusion in `exclusions`

3. What are Speed Patterns? How can I use them?

   In order to propose journey times that are as accurate as possible and reflect changing traffic conditions, graphs supplied by GEOCONCEPT SAS include, from version M18 upwards, 5 speed profiles (Speed Patterns) for cars and lorries, to take into account the different levels of congestion during the course of one day:

   - standard *normal-speed* corresponds to the speed at a time when the roads have an average congestion level (eg 11.00am)
   - night *fast-speed* corresponds to a very fluid traffic situation, often experienced at night-time (3.00am)
   - congested *slow-speed* corresponds to the times when traffic is densest (8.00am)
   - rush hour *very-slow-speed* corresponds to times when traffic is densest in urban and built-up areas, and is slower than the speed above (8.00am)
   - default *default* corresponds to speeds averaged out over an entire day

To use them, you need to pass to parameter, when calling the web service, the `computeOptions` parameter with the *speedPattern* option and specify the value of the Speed Pattern requested, without forgetting a vehicle profile
Example:

+

```
&computeOptions=speedPattern:fast-speed&profileId=1
```

How to use Heavy Goods Vehicle attributes?

The graph must include the attributes for Heavy Goods Vehicles (as standard in the graphs supplied by GEOCONCEPT SAS from version M18 upwards) and either calculate an itinerary using a vehicle profile using restrictions (cf. the catalogue of vehicles, editable, defined in the SmartRoutingVehicles.xml file, stored in the folder `*‘<GEOCONCEPT_WEB_HOME>”\smartrouting \jee\smartrouting\conf\* ), or overwrite  the call to the web service by using the `computeOptions` parameter with the options *length*, *width*, *height*, *weight*, *axles* and/or *weightPerAxle*.
Example for a journey with a vehicle 4.5 metres high:

```
&computeOptions=height:450
```

Optimisation (complete version)

Basic principles

This web service allows you to define optimum routes while minimising operating costs by taking into account the following: road mapping data and road characteristics data in relation to the profiles of the vehicles used - «Business» constraints that relate to (A) the «Clients» to whom deliveries will be made, who will be visited or at whose premises technical interventions are performed, and (B) the «Resources» who will be responsible for implementing the above tasks.

The optimisation service allows us to determine the best possible distribution of clients in relation to resources. It is based on:

- the geographic positions of the full set of clients and resources: a distance matrix is calculated beforehand by the route calculation functions and takes into account the profile of the vehicles used, heavy goods vehicle attributes, …
- customer constraints: time windows, quantities to deliver, special conditions or requirements, …
- parameters for resources: working hours, cost of the resource, …

The optimisation is performed by searching for the solution with the lowest overall cost on the basis of costs of resources as supplied by the user, or using the default costs supplied by the optimisation module.

Availability

This web service is an option in Geoconcept Web: please contact us to find out how to purchase the full documentation.

Pop-up/iframe pages

Introduction

Geoconcept Call Center Architecture

Geoconcept Call Center is a Cllient light application enabling a geographic search for resources. It is a J2EE application, using a databasen and designed to utilise Geoconcept Web web services.

Operating principles

The web services published via the Geoconcept Web solution make light work of integrating geographic search functionalities within existing applications.

Taking as an example an incident management application requiring a search function for locating the nearest emergency and rescue services via the available Web services. In this application, the emergency services concerned are stored in a traditional database (Oracle, SQL Server…).

Using infoboxes or pop-ups, and how to integrate them within applications

Integration depends on Web screens of the *pop-up* type called via the main application. At certain moments during the user's browsing activities, windows issued by Geoconcept Web make geographic data available to the user. The user's selection is returned by a POST form to the master application.

Priority is given to data exchanges via XML for data items containing more than 100 characters.

Geocoding

The application displays an address recognition tool.

For this, as a function of an address in the form (street number and street name, town, post code), a list of the nearest addresses found is suggested (this might be just one, if there are no errors present) along with their geographic locations.

Availability

Available at all times with Geoconcept Web.

The pop-up

The pop-up URL is: http://<serveur>/callcenter/Ext/geocode.do

The pop-up is called by the test page present at the address:

http://<serveur>/callcenter/Ext/geocodetest.do

The layer displayed in the map is configured in the popup.geographics.map.layer server parameter

XML form and structure

The variables below must be sent to the pop-up via the POST method;

urlPost: URL receiving the geocoding results

userData: no matter what the data item consists of, it will be submitted without any modification to the results reception form. This data enables the master application to register a context, that could be useful for the return.

xmlData: the data to geocode in xml. The format is that of a SOAP geocoding query.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
       <gc:GeocodeRequest xmlns:gc="http://geoconcept.com/gc/schemas">
          <gc:Address>
             <gc:CountryCode>fr</gc:CountryCode>
             <gc:City>Paris</gc:City>
             <gc:PostalCode>75013</gc:PostalCode>
             <gc:AddressLine>25 rue de Toldiac</gc:AddressLine>
          </gc:Address>
       </gc:GeocodeRequest>
```

Retrieving the geocoding results

The choice of a user is returned via a POST query containing the following parameters:

userData: the user data passed previously

address: the address recognised in practise (note: does not include the number if the number was not present in the geocoding database)

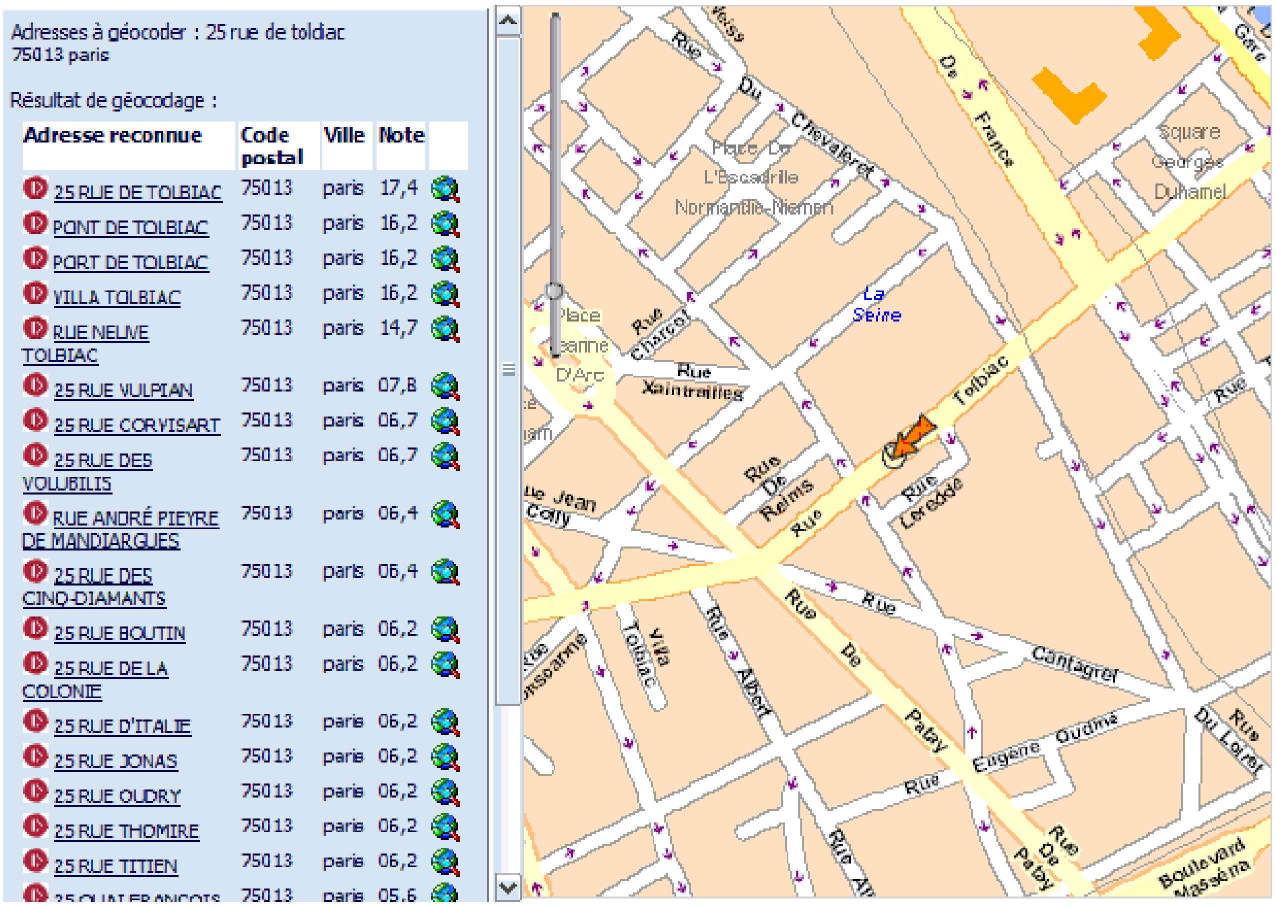
city: the town recognised.

postalCode: the recognised Post code

x: the X position (in the projection configured in the parameter server, or in longitude if a WGS84 projection has been requested)

y: Y position (idem)

Result


Adresses à géocoder : 25 rue de tolbiac 75013 paris

Résultat de géocodage :

Adresse reconnue	Code postal	Ville	Note
25 RUE DE TOLBIAC	75013	paris	17,4
PONT DE TOLBIAC	75013	paris	16,2
PORT DE TOLBIAC	75013	paris	16,2
VILLA TOLBIAC	75013	paris	16,2
RUE NELVIE TOLBIAC	75013	paris	14,7
25 RUE VULPIAN	75013	paris	07,8
25 RUE CORVISART	75013	paris	06,7
25 RUE DES VOLUBILIS	75013	paris	06,7
RUE ANDRÉ PIEYRE DE MANDIARGUES	75013	paris	06,4
25 RUE DES CINQ-DIAMANTS	75013	paris	06,4
25 RUE BOUTIN	75013	paris	06,2
25 RUE DE LA COLONIE	75013	paris	06,2
25 RUE D'ITALIE	75013	paris	06,2
25 RUE JONAS	75013	paris	06,2
25 RUE OUDRY	75013	paris	06,2
25 RUE THOMIRE	75013	paris	06,2
25 RUE TITIEN	75013	paris	06,2
25 QUAI FRANCOIS	75013	paris	05,6

Search for a resource

Following a click on the «search» button, the master application displays a pop-up window that calls a search page on the Geoconcept Web server. This page makes a first request to select a certain number of candidates fulfilling the criteria indicated in the database of service providers. It can, notably, search for resources in a square around the point of research, while integrating business criteria. It is worth noting that it can use stored procedures if required.

Availability

This section is under development.

The pop-up

The pop-up URL is :http://<serveur>/callcenter/Ext/search.do

The pop-up is called by the test page present at the following address:

http://<serveur>/callcenter/Ext/searchtest.do

The layer displayed in the map is configured in the popup.geographics.map.layer server parameter

Parameters

- `urlPost` : the URL receiving the Gocoding results
- `userData` : any data item, it will be submitted, without modification, to the form assigned to receive the geocoding results. This data allows the master application to set up a context which can be useful for the return.
- `xmlData` : the data to geocode in xml. The format is that of a SearchAround SOAP query.

If the projection is not configured in gc:Options, the projection used will be the one configured in the popup.ws.defaultSrs server parameter.

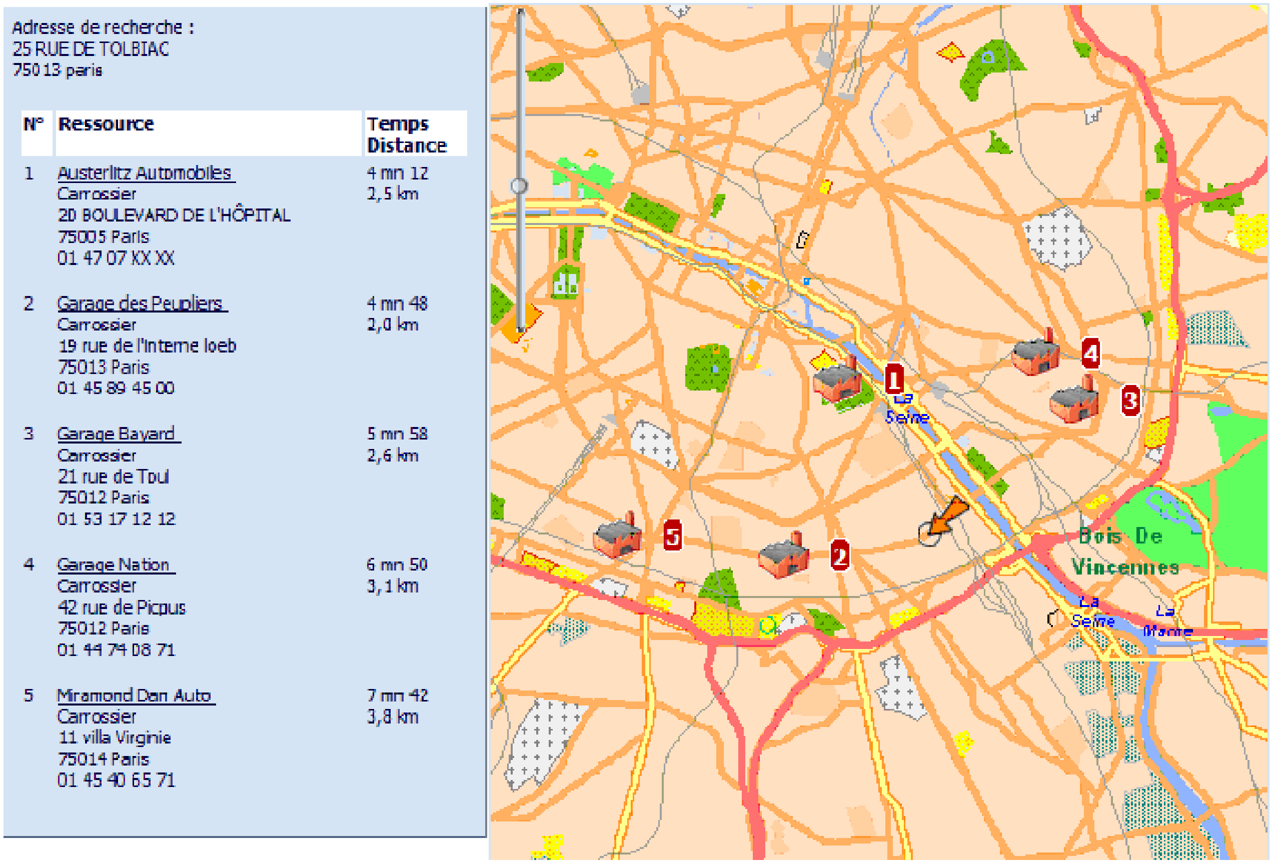
Nevertheless, one can add the following information, used only for the display, in the parameters describing the target, or a resource being searched for:

- `Name` : name of the resource
- `Html` : an HTML code to display under the resource name
- `sheetHtml` : an html code to display on the infobox describing the resource (appears on the map when you click on the resource)
- `ImageUrl` : url of the resource’s image

Example of a query

```
<gc:SearchAroundRequest xmlns:gc="http://geoconcept.com/gc/schemas">
          <gc:Options>
          </gc:Options>
          <gc:Target>
             <gc:X>602725</gc:X>
             <gc:Y>2425604</gc:Y>
             <gc:Id>1</gc:Id>
          </gc:Target>
          <gc:Resource>
             <gc:X>599266</gc:X>
             <gc:Y>2425096</gc:Y>
             <gc:Id>2</gc:Id>
<gc:Name>car1</gc:Name>
<gc:Address>12 rue de tolbaic</gc:Address>

<gc:Html><![CDATA[<b>ici</b>]]></gc:Html>
          </gc:Resource>
          <gc:Resource>
             <gc:X>599366</gc:X>
             <gc:Y>2425196</gc:Y>
             <gc:Id>2</gc:Id>
<gc:Name>nom1</gc:Name>
<gc:Html><![CDATA[<b>decription html</b>]]></gc:Html>
          </gc:Resource>
</gc:SearchAroundRequest>
```

Pop-up display



Adresse de recherche :
25 RUE DE TOLBIAC
75013 paris

N°	Ressource	Temps Distance
1	Austerlitz Automobiles Carrossier 2D BOULEVARD DE L'HÔPITAL 75005 Paris 01 47 07 XX XX	4 mn 12 2,5 km
2	Garage des Peupliers Carrossier 19 rue de l'Interne loeb 75013 Paris 01 45 89 45 00	4 mn 48 2,0 km
3	Garage Bayard Carrossier 21 rue de Toul 75012 Paris 01 53 17 12 12	5 mn 58 2,6 km
4	Garage Nation Carrossier 42 rue de Picpus 75012 Paris 01 44 74 08 71	6 mn 50 3,1 km
5	Miramond Dan Auto Carrossier 11 villa Virginie 75014 Paris 01 45 40 65 71	7 mn 42 3,8 km

Retrieving the result

The choice of a user is returned via a POST query containing the following parameters:

- `userData` : the user data passed previously
- `resourceId` : ID of the chosen resource
- `resourceName` : name of the chosen resource
- `distance` : distance, in meters, to the chosen resource
- `time` : access time in seconds

Zone definition

Integration principle

The integration depends on Web screens of the *pop-up* type called from the main application. At certain moments, windows issued from the GeoConcept LBS Platform server present geographic data to the user. The choice of the user is returned via a POST form to the master application.

XML data echange is given priority for data exchanges consisting of more than 100 characters.

The pop-up

The URL for the pop-up is: http://<serveur>/callcenter/Ext/zonedefine.do

The pop-up is called by the test page present at the address:

http://<serveur>/callcenter/Ext/zonedefinetest.do

The layer displayed in the map is configured in the popup.geographics.map.layer server parameter.

If the name of the entities to select is in a unicode field, set the popup.geographics.entityNameUnicode server parameter to true (otherwise it will be false).

Parameters

- `urlPost` : the URL receiving the Gocoding results
- `userData` : any data item, it will be submitted, without modification, to the form assigned to receive the geocoding results. This data allows the master application to set up a context which can be useful for the return.
- `xmlData` : data for the xml structure zone, with «DefineZoneRequest» as root… this xml element may contain the following information:
- `CurrentZone` : list of geographically referenced elements, defined by the type, subtype and code attributes.
- `Entity` : geographic element defined by Id
- `DisplayElements` : list of elements to display (there could be several lists))

Note: you could send the geographic coordinates in MAP projection (the projection utilised is that configured in the popup.ws.defaultSrs server parameter) or as WGS84 coordinates (EPSG:4326 code).

Example of a query

```
<gc:DefineZoneRequest xmlns:gc="http://geoconcept.com/gc/schemas">
  <!-- The Map must have a tab  type.subtype where type/subtype is selectable. code is a field of  type/
subtype -->
  <gc:CurrentZone type="Administrative unit" subtype="Order8" code="Government code">
    <!-- select Bagneux -->
    <gc:Entity id="92007"/>
    <!-- select Cachan -->
    <gc:Entity id="94016"/>
    <!-- select Châtillon -->
    <gc:Entity id="92020"/>
  </gc:CurrentZone>
  <gc:DisplayElements img="http://en.geoconcept.com/sites/all/themes/geoconcept/images/flags/fr.png"
 projection="EPSG:4326">
    <gc:Symbol x="2.31035" y="48.80165" />
    <gc:Symbol x="2.31035" y="48.89165" />
  </gc:DisplayElements>
  <!-- An image must be called Pin_user_gcblue  -->
  <gc:DisplayElements img="Pin_user_gcblue">
    <gc:Symbol x="601725" y="2424604" />
  </gc:DisplayElements>
</gc:DefineZoneRequest >
```

Pop-up display

The page displays a map with the selection tools. By default, the selection tool is the tool for adjusting the selection.

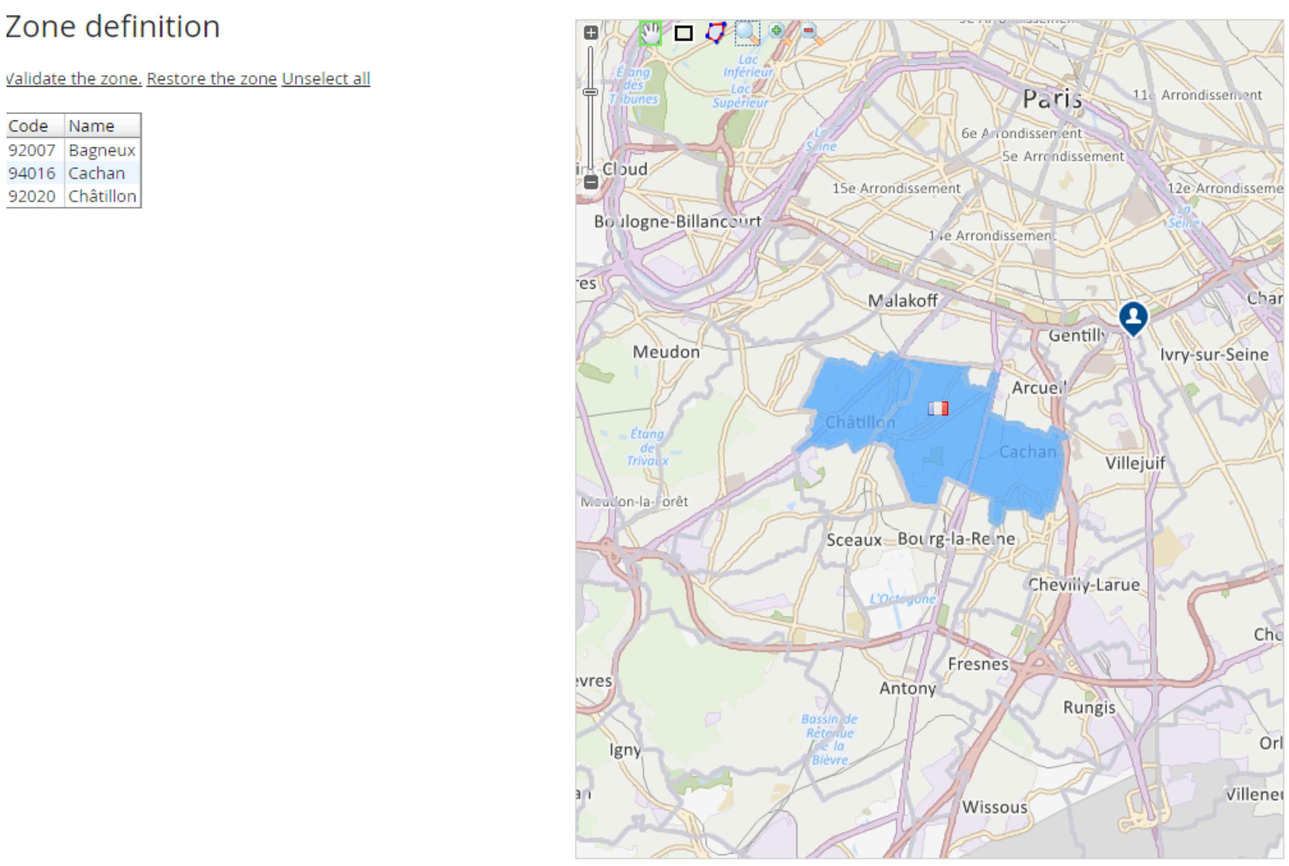
The entities passed to parameter are coloured.

The symbols passed in the list are displayed with an image, the name of which is indicated.

- pan tool: enables movement in the map
- draw a zone tool: this tool draws a polygon, point by point
- the entities that are integrally or partially included in the design are then selected, and the drawing is deleted.
- adjust the selection tool: click on an entity to select/deselect it, depending on its current status.

a button allows you to terminate the entry operation.

Example of an interface



Zone definition

Validate the zone. Restore the zone Unselect all

Code	Name
92007	Bagneux
94016	Cachan
92020	Châtillon

Retrieving the result

The user’s choice is returned by a POST query containing the following parameters:

```
<gc:DefineZoneResult xmlns:gc="http://geoconcept.com/gc/schemas">
```

```
    <gc:Zone type="Administrative unit" subtype="Order8" code="Government code">
       <gc:Entity id="92007"/>
       <gc:Entity id="94016"/>
       <gc:Entity id="92020"/>
    </gc:Zone>
  </gc:DefineZoneResult>
```

Examples

Example of how web services can be integrated: incident management application

This example shows you how to use the different components of Geoconcept Web, and to link one component in with another to construct a complete application.

We will take as an example an accident management application requiring search for the nearest rescue service. In this application, the service operators feature in a traditional kind of database (Oracle, SQL Server, …).

Geocoding of service providers

Add to the table of service providers two numeric columns with X and Y (or latitude and longitude) as column headers.

For a full scale and rapid geocoding operation (several thousand lines) use the Universal Geocoder application in desktop mode, or in batch mode.

For a less ambitious Geocoding operation, one can also use the geocoding web service described in the documentation. This web service takes an address as input, and returns the best address or addresses found, with their respective X and Y coordinates.

Indexing longitude and latitude fields

Making modifications to the application

Adding a Geocoding functionality to handle incidents

After typing in the address of an accident via a form, a geocoding step is added (étape_géocoder_) that calls the geocoding web service. It is also possible to display a map of the location at which the accident happened, with the Javascript component, using the GCIS component. Following geocoding, a position for the accident (X,Y) can be obtained. Thanks to the Javascript functionalities, it is possible to centre the map displayed on the X, Y position returned.

Search for candidate rescue services

In the first instance, a preliminary selection needs to be made of the nearest rescue providers occurring within the limits of a Bounding Box that has to be defined.

First, construct an approximate first query in the table of providers to search for providers in a square X kilometers around the site of the accident. Then calculate (approximate formula):

```
longitude0 : longitude sinistre
latitude0 : latitude sinistre
```

```
l : distance de recherche (en km)
R = 6371 (en km)
delta_latitude = (180 / PI) * l / R
delta_longitude = (180 / PI) * l / R / cos(latitude0)
```

which can be achieved with a simple SQL query:

```
SELECT * FROM prestataire
WHERE longitude > longitude0 - delta_longitude
AND longitude < longitude0 + delta_longitude
AND latitude > latitude0 - delta_latitude
AND latitude < latitude0 + delta_latitude
```

This example of a query defines a certain number of candidate providers among the nearest. If the number of candidates is too high (more than 100), one can restart the operation by dividing the distance in two. If the number of candidates is too low, one could also restart the operation by doubling the search area, until the correct number of candidates is found (a minimum of 10, for example). The next step will be to classify these potential candidates as a function of the method chosen, in order to highlight the nearest.

Calculating distance, time and sort operations on candidate entities

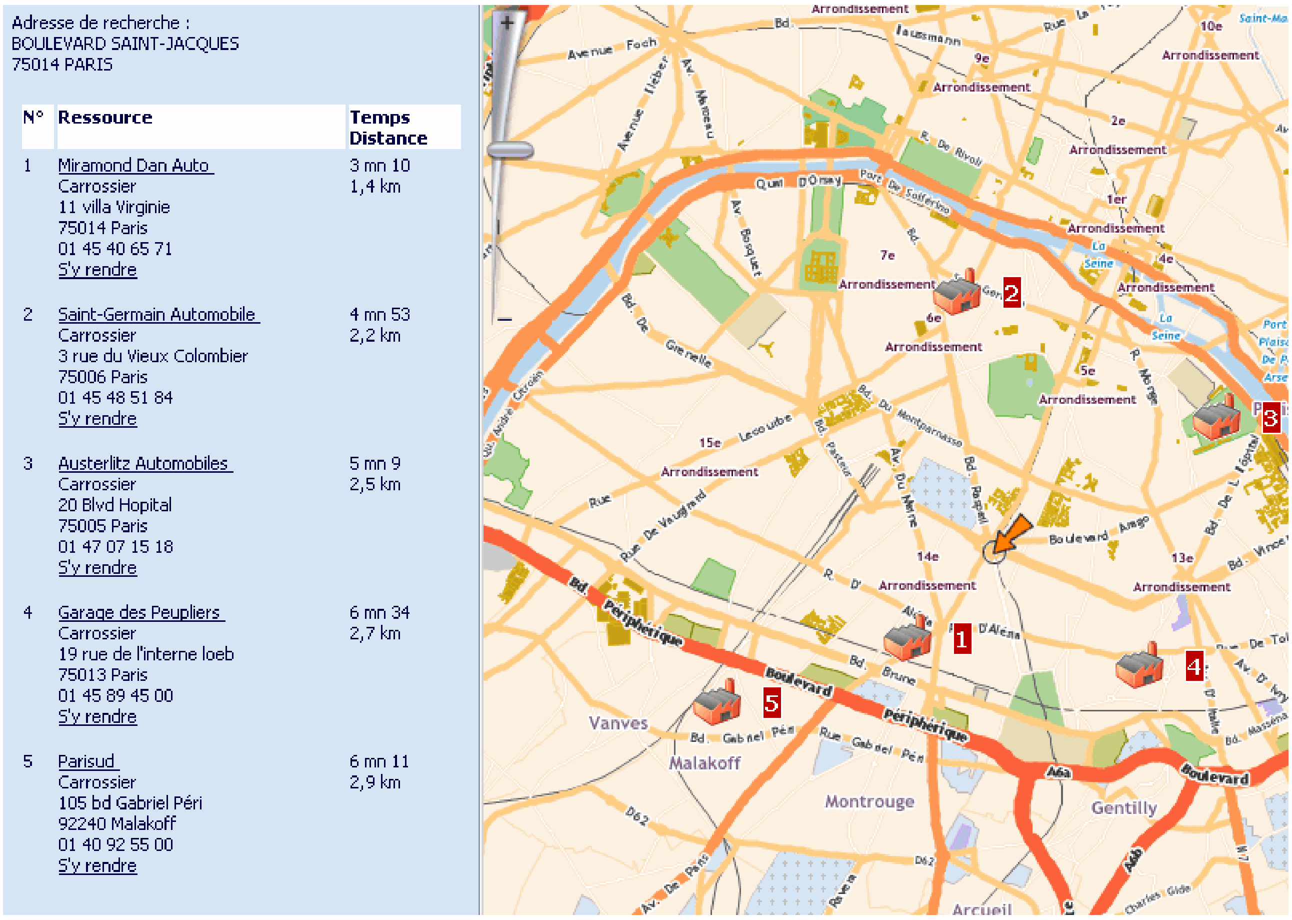
We use the “SearchAround” Web Service described in the documentation to classify candidates as a function of the chosen method.

The user only needs to present the result of classified providers to the user for them to then choose the most suitable provider (the nearest in distance or the nearest in terms of travel time).

Display of candidates on a map

We would use the API Javascript component to include the map in the existing web application and display the providers on a street map background.

Utilisation of API Javascript


Adresse de recherche :
BOULEVARD SAINT-JACQUES
75014 PARIS

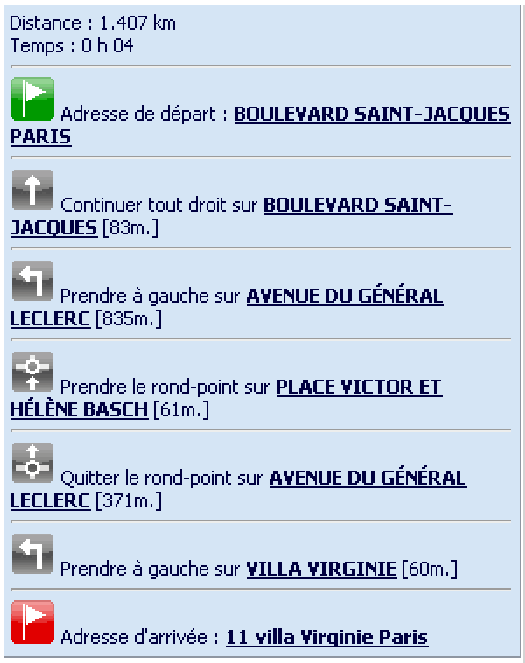
N°	Ressource	Temps Distance
1	Miramond Dan Auto Carrossier 11 villa Virginie 75014 Paris 01 45 40 65 71 S'y rendre	3 mn 10 1,4 km
2	Saint-Germain Automobile Carrossier 3 rue du Vieux Colombier 75006 Paris 01 45 48 51 84 S'y rendre	4 mn 53 2,2 km
3	Austerlitz Automobiles Carrossier 20 Blvd Hopital 75005 Paris 01 47 07 15 18 S'y rendre	5 mn 9 2,5 km
4	Garage des Peupliers Carrossier 19 rue de l'interne loeb 75013 Paris 01 45 89 45 00 S'y rendre	6 mn 34 2,7 km
5	Parisud Carrossier 105 bd Gabriel Péri 92240 Malakoff 01 40 92 55 00 S'y rendre	6 mn 11 2,9 km

Route calculation

The route calculation web service returns a route sheet in xml format between a start address (supplied by the user via the form filled in) and a service provider chosen by the user.

It is then possible to use this route sheet to track an itinerary between the start and arrival points.

Route calculation



Distance : 1.407 km
Temps : 0 h 04

Adresse de départ : **BOULEVARD SAINT-JACQUES PARIS**

Continuer tout droit sur **BOULEVARD SAINT-JACQUES** [83m.]

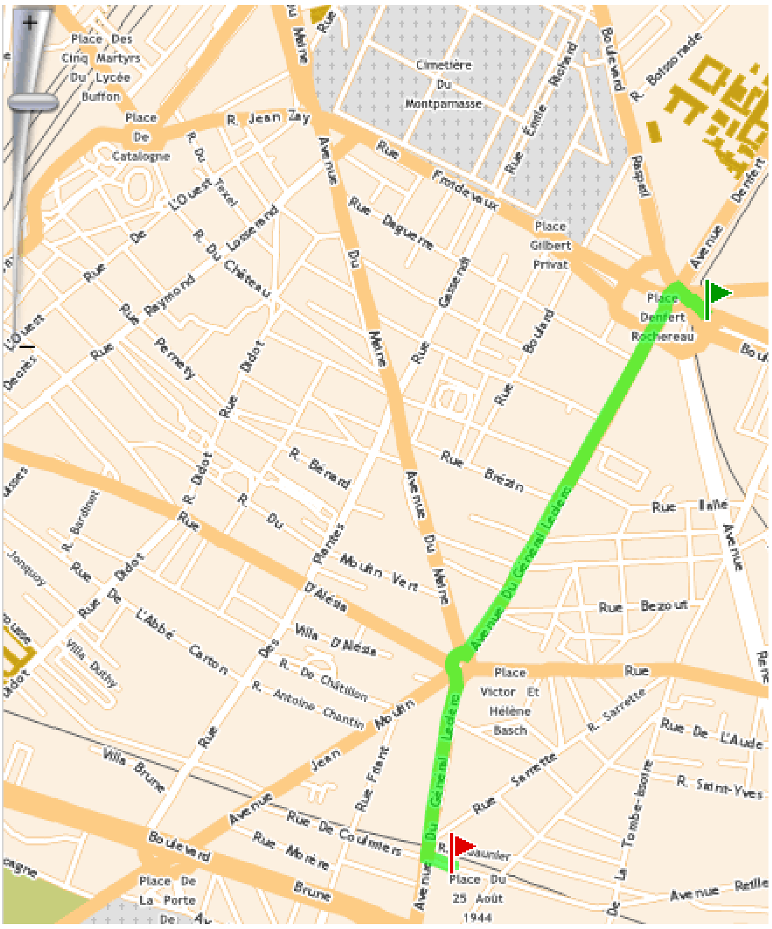
Prendre à gauche sur **AVENUE DU GÉNÉRAL LECLERC** [835m.]

Prendre le rond-point sur **PLACE VICTOR ET HÉLÈNE BASCH** [61m.]

Quitter le rond-point sur **AVENUE DU GÉNÉRAL LECLERC** [371m.]

Prendre à gauche sur **VILLA VIRGINIE** [60m.]

Adresse d'arrivée : **11 villa Virginie Paris**



Example of scripts in PHP 5

Geocoding

```
<h1>Geocoding service</h1>

<?php
// Display all errors:
error_reporting(E_ALL);
ini_set('display_errors', '1');

///////////////////////////////////////////////////////////////////////
///////////////////////////  CREATE REQUEST ///////////////////////////
///////////////////////////////////////////////////////////////////////

class AddressType{

   function __construct($countryCode,$city,$postalCode,$addressLine,$srs="epsg:4326",$maxResponses=2)  {
     $this->countryCode=$countryCode;
     $this->city=$city;
     $this->postalCode=$postalCode;
     $this->addressLine=$addressLine;
     $this->srs=$srs;
     $this->maxResponses=$maxResponses;
   }
};


$addressType=new AddressType("fr","Paris","75013","25 rue de Toldiac","wgs84");
$request=array(
  "geocode" => $addressType
);

////////////////////////////////////////////////////////////////////////
/////////////////////////////  CALL SOAP  /////////////////////////////
////////////////////////////////////////////////////////////////////////

try{
  $clientOptions=array(
    'trace'=>true,
    'exceptions'=>true,
    'encoding'=>'utf-8'
  );

  $client = new SoapClient('http://gcweb.geoconcept.com/gws/api/ws/geocodeService?wsdl', $clientOptions);


  echo "<h2>Request</h2>";
  echo "<pre>";
  var_dump($request);
  echo "</pre>";

  $response = $client->__soapCall("geocode",$request);


  echo "<h2>Response (initialAddress)</h2>";
  echo "<pre>";
  var_dump($response->GeocodeResult->initialAddress);
  echo "</pre>";

  echo "<h2>Response (geocodedAddress)</h2>";
  echo "<pre>";
  var_dump($response->GeocodeResult->geocodedAddress);
  echo "</pre>";
}
catch (SoapFault $e) {
  echo $e;
}
?>
```

Route calculation

```
<h1>Route service</h1>

<?php
// Display all errors:
error_reporting(E_ALL);
ini_set('display_errors', '1');

////////////////////////////////////////////////////////////////////////
///////////////////////////  CREATE REQUEST ///////////////////////////
////////////////////////////////////////////////////////////////////////

class RoutePointType {

  function __construct($x,$y)  {
    $this->x=$x;
    $this->y=$y;
  }
}

$step1=new RoutePointType(0.691012,47.384813);
$step2=new RoutePointType(0.693012,47.385813);

$route=array(
  "request" => array(
     "origin" => $step1,
     "destination" => $step2,
     "srs" => "epsg:4326",
     "method" => "time",
     "format" => "STANDARD",
     "rejectFlags" => array("Toll"),
     "tolerance" => array()
    )

);

$request=array(
   "route" => $route
);

///////////////////////////////////////////////////////////////////////
/////////////////////////////  CALL SOAP  /////////////////////////////
///////////////////////////////////////////////////////////////////////

try{
  $clientOptions=array(
    'trace'=>true,
    'exceptions'=>true,
    'encoding'=>'utf-8'
  );

  $client = new SoapClient('http://pouget/geoconcept-web/api/ws/routeService?wsdl', $clientOptions);


  echo "<h2>Request</h2>";
  echo "<pre>";
  var_dump($request);
  echo "</pre>";

  $response = $client->__soapCall("route",$request);

  echo "<h2>Response</h2>";
  echo "<pre>";
  var_dump($response);
  echo "</pre>";
}
catch (SoapFault $e) {
  echo $e;
}
?>
```

Search Around (proximity search)

```
<h1>SearchAround service</h1>

<?php
// Display all errors:
error_reporting(E_ALL);
```

```
ini_set('display_errors', '1');

////////////////////////////////////////////////////////////////////////
///////////////////////////  CREATE REQUEST ///////////////////////////
////////////////////////////////////////////////////////////////////////

class SearchAroundTargetType {

  function __construct($x,$y)  {
    $this->x=$x;
    $this->y=$y;
  }
}

class SearchAroundPointType {

  function __construct($x,$y,$id="",$priority1="",$priority2="")  {
    $this->x=$x;
    $this->y=$y;
    $this->id=$id;
    $this->priority1=$priority1;
    $this->priority2=$priority2;
  }
}

$target=new SearchAroundTargetType(-1.593927,47.218580);
$resource1=new SearchAroundPointType(-1.593927,47.188580,2,1,1);
$resource2=new SearchAroundPointType(-1.556927,47.188580,3,1,2);
$resource3=new SearchAroundPointType(-1.557927,47.189580,4,1,1);

$searchAround=array(
       "request" => array (
                "target" => $target,
                "method" => "time",
                "reverse"   => "",
                "srs" => "epsg:4326",
                 "exclusions" => array(),
                "resources" => array($resource1,$resource2,$resource3),
       )

);

$request = array("searchAround" => $searchAround);


////////////////////////////////////////////////////////////////////////
/////////////////////////////  CALL SOAP  /////////////////////////////
////////////////////////////////////////////////////////////////////////

try{
  $clientOptions=array(
    'trace'=>true,
    'exceptions'=>true,
    'encoding'=>'utf-8'
  );

  $client = new SoapClient('http://pouget/geoconcept-web/api/ws/searchAroundService?wsdl', $clientOptions);

  echo "<h2>Request</h2>";
  echo "<pre>";
  var_dump($request);
  echo "</pre>";
```

```
    $response = $client->__soapCall("searchAround",$request);

    echo "<h2>Response</h2>";
    echo "<pre>";
    var_dump($response);
    echo "</pre>";
  }
  catch (SoapFault $e) {
    echo $e;
  }
  ?>
```

UGC Server

Installing UGC Server

This document describes the installation procedure for the Universal Geocoder Server application. Selecting which components to install is a vital stage in the installation process:

- UGC JEE,
- UGC Command Line,
- UGC .NET. (not available with Geoconcept Web)

These components are quite independent of one another, and fulfil different purposes. For example, the JEE UGC component is needed by the “Geocoder” widget.

This document also describes how to set up the UGC resource adaptor as a function of the target application server.

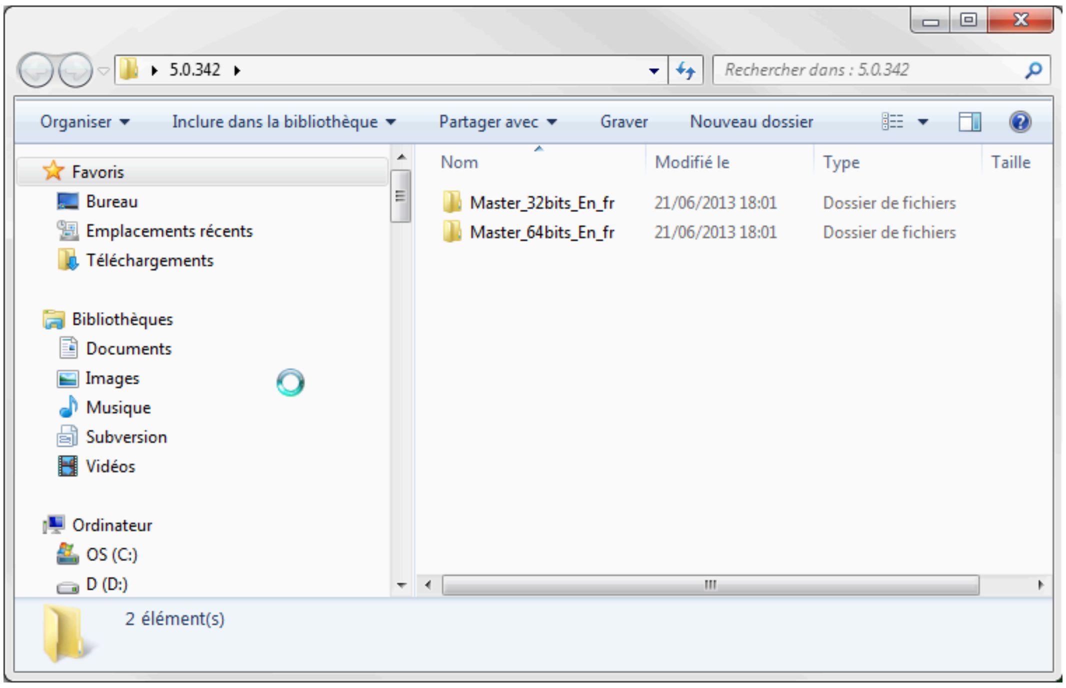
Finally, the installation of the files needed for the autocomplete functionality is described in this document. For further information about using the autocomplete web service, please refer to the section API and Web Services section of Geoconcept Web.

Installing the UGC Server application

The following procedure does not concern Geoconcept Web since its installer takes charge of installing and configuring the JEE UC component.

Having unzipped the downloaded folder, you will find two directories: one for the 32-bit installer and one for the 64-bit installer. Choose the one that is best for your particular architecture.

Choice of the 32-bit or 64-bit master



Nom	Modifié le	Type	Taille
Master_32bits_En_fr	21/06/2013 18:01	Dossier de fichiers	
Master_64bits_En_fr	21/06/2013 18:01	Dossier de fichiers	

When using UGC JEE, the 32-bit or 64-bit version should be selected to suit your Java and Apache Tomcat architecture :

- Choose the 32-bit master if your JRE is a 32-bit version (Apache Tomcat is then in 32-bits).
- Choose the 64-bit master if your JRE is a 64-bit version (Apache Tomcat is then in 64-bits).

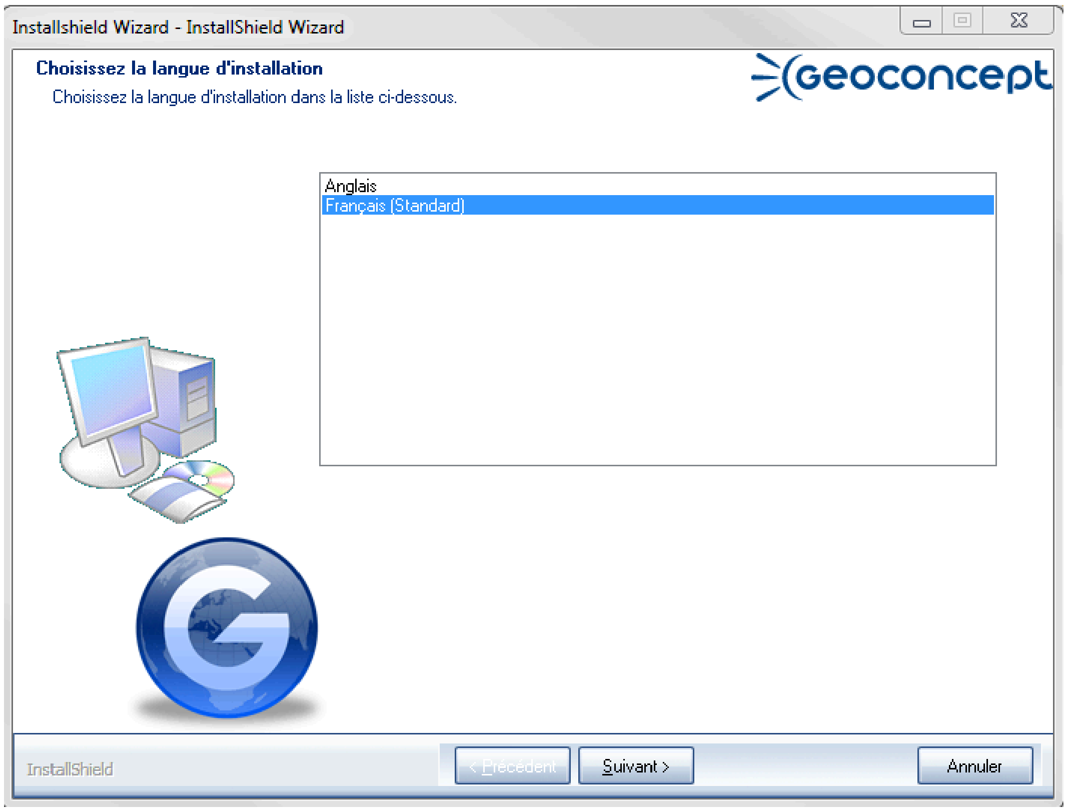
The importance of this choice is crucial with regard to the DLL utilised to make the UGC Server function.

A 32-bit JEE UGC deployed in a 64-bit environment will not function, and vice versa.

We recommend installing the 64-bit version.

Run the setup.exe installer and then follow the different steps as guided by the installation assistant.

Choice of a language version

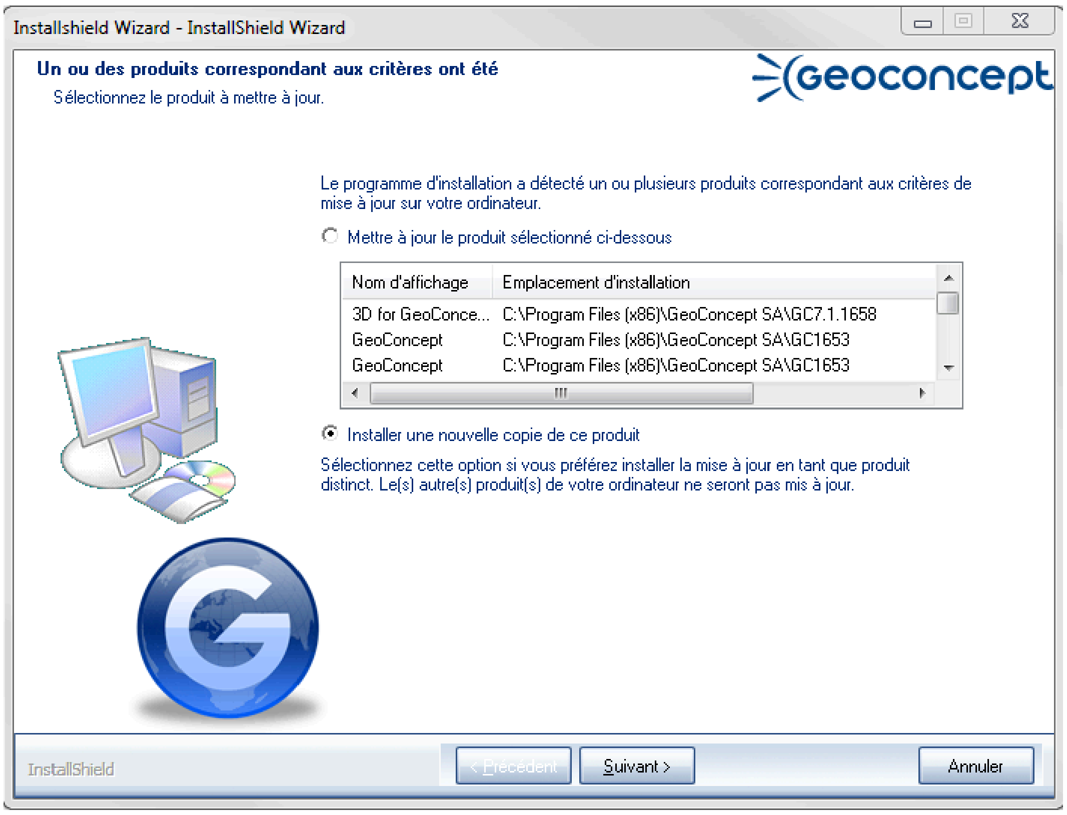

Installshield Wizard - InstallShield Wizard

Choisissez la langue d'installation

Choisissez la langue d'installation dans la liste ci-dessous.

Anglais
Français (Standard)

Installing a new copy



Installshield Wizard - InstallShield Wizard

Un ou des produits correspondant aux critères ont été

Sélectionnez le produit à mettre à jour.

Le programme d'installation a détecté un ou plusieurs produits correspondant aux critères de mise à jour sur votre ordinateur.

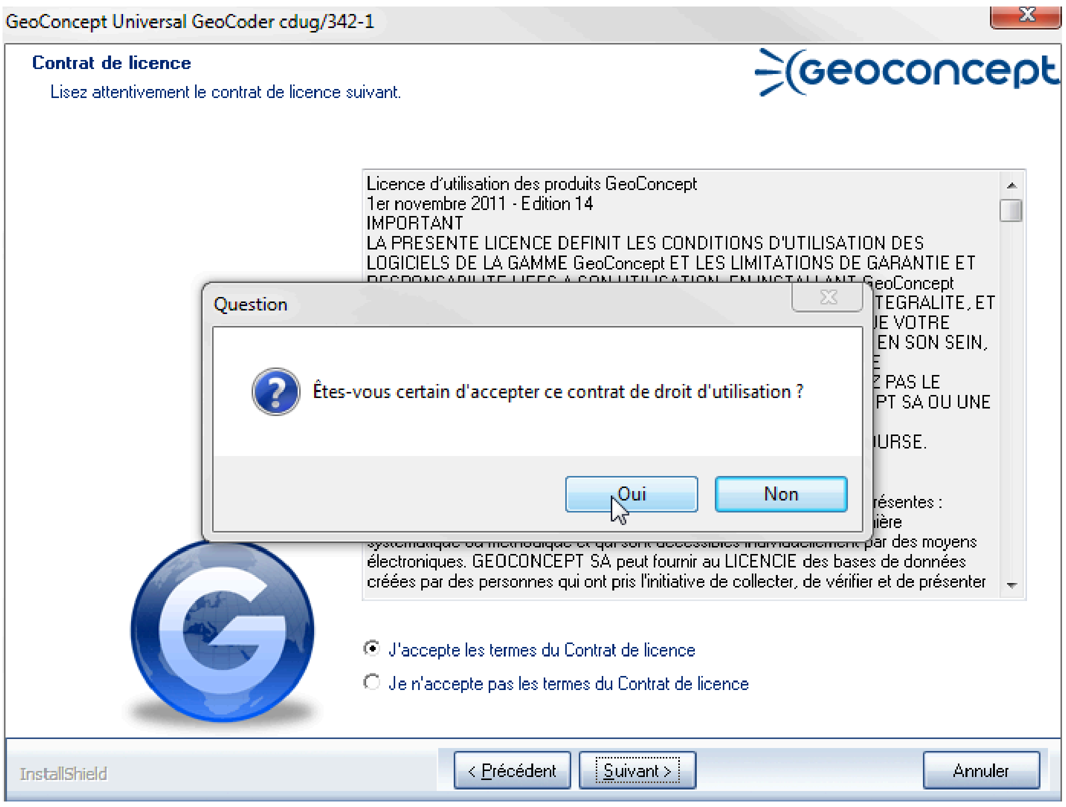
( ) Mettre à jour le produit sélectionné ci-dessous

Nom d'affichage	Emplacement d'installation
3D for GeoConce...	C:\Program Files (x86)\GeoConcept SA\GC7.1.1658
GeoConcept	C:\Program Files (x86)\GeoConcept SA\GC1653
GeoConcept	C:\Program Files (x86)\GeoConcept SA\GC1653

[✓ Installer une nouvelle copie de ce produit]

Sélectionnez cette option si vous préférez installer la mise à jour en tant que produit distinct. Le(s) autre(s) produit(s) de votre ordinateur ne seront pas mis à jour.

Accepting the terms and conditions of use



GeoConcept Universal GeoCoder cdug/342-1

Contrat de licence

Lisez attentivement le contrat de licence suivant.

Licence d'utilisation des produits GeoConcept
1er novembre 2011 - Edition 14
IMPORTANT
LA PRESENTE LICENCE DEFINIT LES CONDITIONS D'UTILISATION DES LOGICIELS DE LA GAMME GeoConcept ET LES LIMITATIONS DE GARANTIE ET
...
électroniques. GEOCONCEPT SA peut fournir au LICENCIE des bases de données créées par des personnes qui ont pris l'initiative de collecter, de vérifier et de présenter

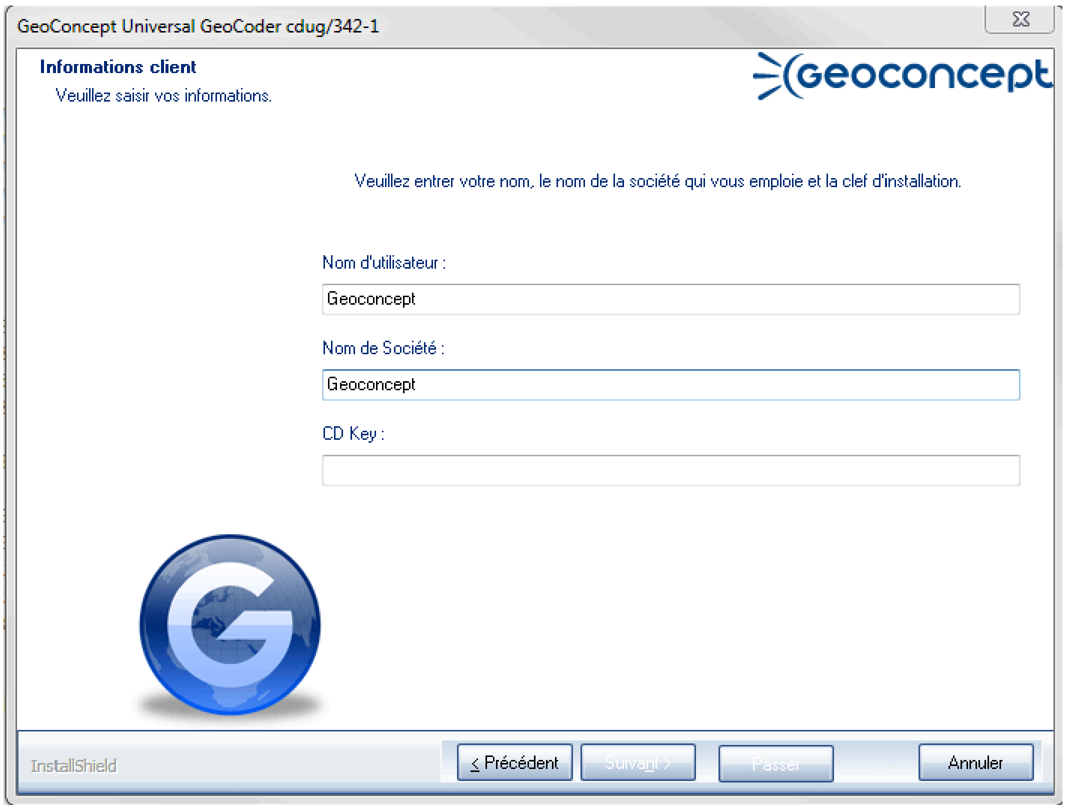
Question

Êtes-vous certain d'accepter ce contrat de droit d'utilisation ?

[✓ J'accepte les termes du Contrat de licence]

( ) Je n'accepte pas les termes du Contrat de licence

Entering the key


GeoConcept Universal GeoCoder cdug/342-1

Informations client

Veuillez saisir vos informations.

Veuillez entrer votre nom, le nom de la société qui vous emploie et la clef d'installation.

Nom d'utilisateur : Geoconcept

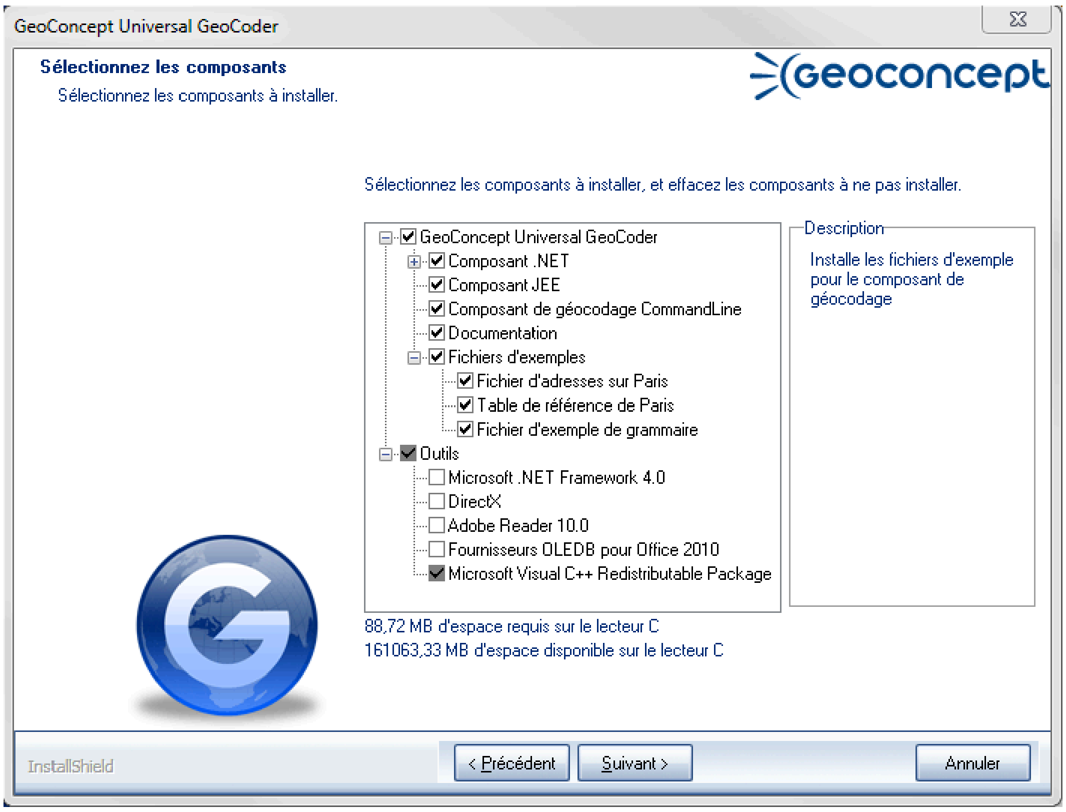
Nom de Société : Geoconcept

CD Key :

You must choose the components needed for your installation depending on how you intend to use the configuration:

- .NET component,
- JEE Component,
- CommandLine component,

Choice of components


GeoConcept Universal GeoCoder

Sélectionnez les composants

Sélectionnez les composants à installer.

Sélectionnez les composants à installer, et effacez les composants à ne pas installer.

- [✓] GeoConcept Universal GeoCoder
  - [✓] Composant .NET
  - [✓] Composant JEE
  - [✓] Composant de géocodage CommandLine
  - [✓] Documentation
  - [✓] Fichiers d'exemples
    - [✓] Fichier d'adresses sur Paris
    - [✓] Table de référence de Paris
    - [✓] Fichier d'exemple de grammaire
- [✓] Outils
  - [ ] Microsoft .NET Framework 4.0
  - [ ] DirectX
  - [ ] Adobe Reader 10.0
  - [ ] Fournisseurs OLEDB pour Office 2010
  - [✓] Microsoft Visual C++ Redistributable Package

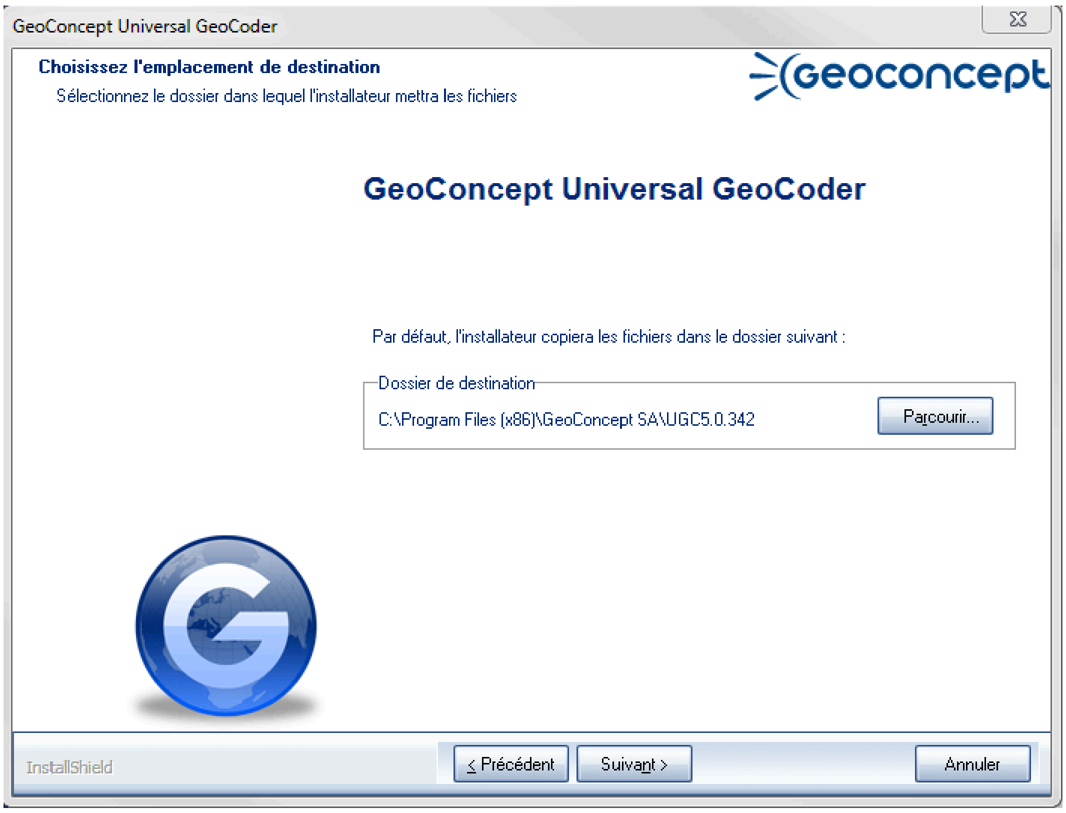
Description

Installe les fichiers d'exemple pour le composant de géocodage

88,72 MB d'espace requis sur le lecteur C

161063,33 MB d'espace disponible sur le lecteur C

Choosing an installation directory


GeoConcept Universal GeoCoder

Choisissez l'emplacement de destination

Sélectionnez le dossier dans lequel l'installateur mettra les fichiers

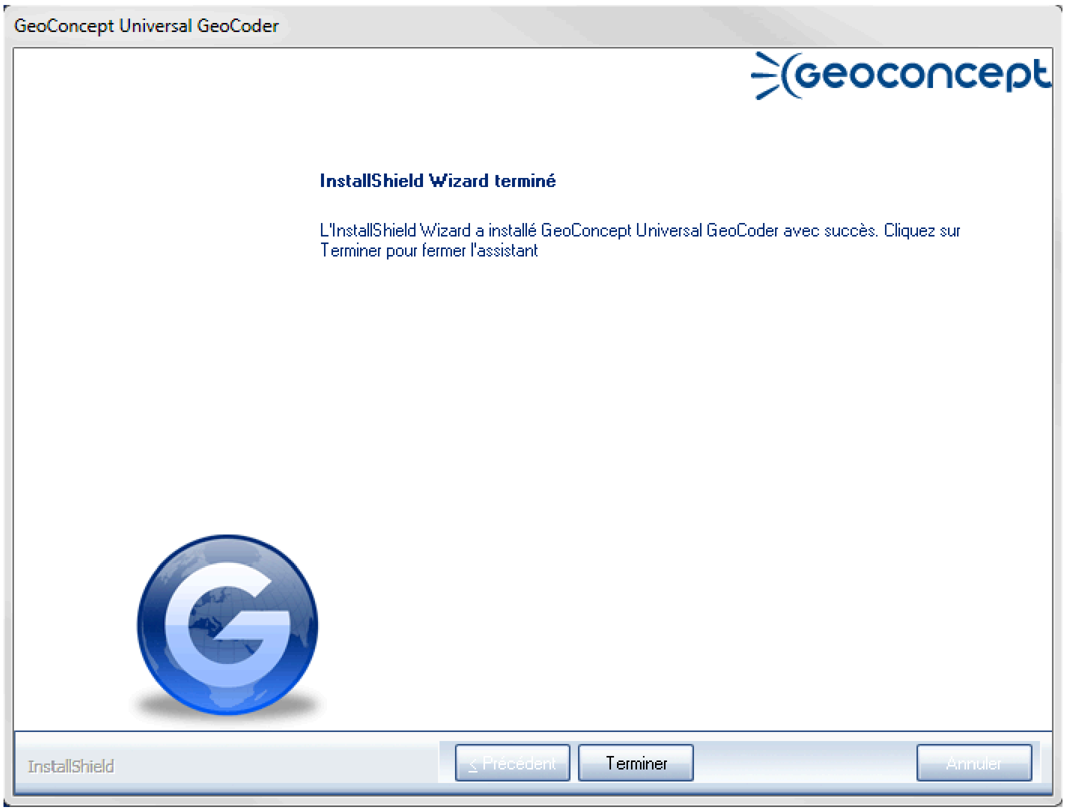
GeoConcept Universal GeoCoder

Par défaut, l'installateur copiera les fichiers dans le dossier suivant :

Dossier de destination

C:\Program Files (x86)\GeoConcept SA\UGC5.0.342

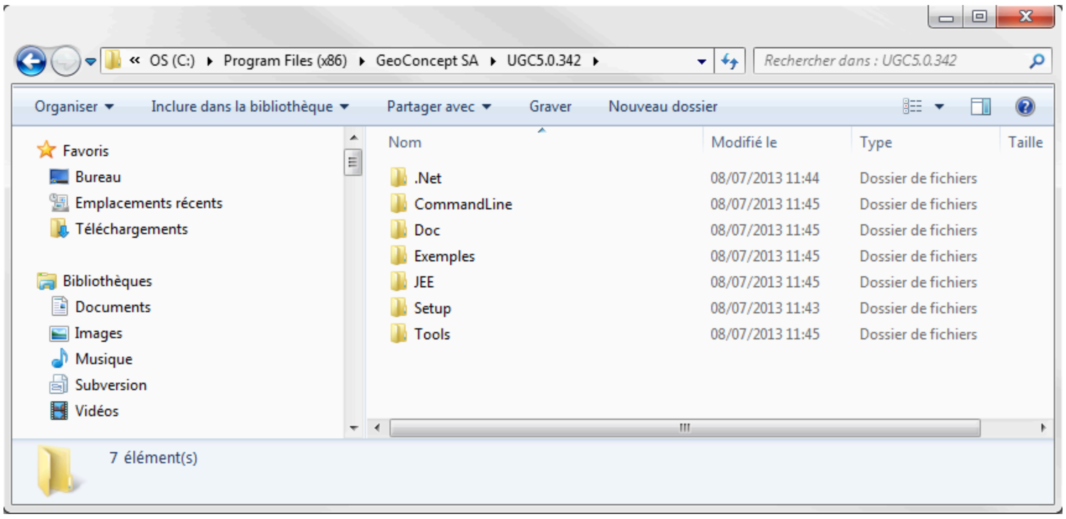
End of the installation


GeoConcept Universal GeoCoder

InstallShield Wizard terminé

L'InstallShield Wizard a installé GeoConcept Universal GeoCoder avec succès. Cliquez sur Terminer pour fermer l'assistant

Directories deployed on the server


OS (C:) ▸ Program Files (x86) ▸ GeoConcept SA ▸ UGC5.0.342 ▸

Nom	Modifié le	Type	Taille
.Net	08/07/2013 11:44	Dossier de fichiers	
CommandLine	08/07/2013 11:45	Dossier de fichiers	
Doc	08/07/2013 11:45	Dossier de fichiers	
Exemples	08/07/2013 11:45	Dossier de fichiers	
JEE	08/07/2013 11:45	Dossier de fichiers	
Setup	08/07/2013 11:43	Dossier de fichiers	
Tools	08/07/2013 11:45	Dossier de fichiers	

7 élément(s)

The .ugc.xxi referencial geocoding should be stored in the “<UGC_HOME>”\JEE\ugc\reftables directory (this is the default directory) or in a dedicated directory of the D:\TableRef type. This dedicated directory must be configured in the `server.xml` file (please refer to the section entitled ‘`Configuration for JEE UGC ’’ in the Geoconcept Web manual).

Deployment and configuration of the application server

This chapter describes the setting up of the UGC resource adaptor. The deployment mode depends on the target application Server.

Configuring Apache Tomcat

Using the Apache Tomcat server in the framework of an installation of Geoconcept Web is described in the Geoconcept Web manual.

The remainder of this section describes the configuration in a framework other than that of Geoconcept Web. Certain elements may be similar to a utilisation in the framework of Geoconcept Web.

The directory in which Tomcat is installed is referred to as %tomcat%.

Common Lib

In %tomcat%/common/lib, copy the files stored into %liverable%/tomcat/lib:

- ugc.jar: main library for the ugc-jee resource adaptor,
- jdom.jar: manipulation of XML documents,
- activation.jar: as this is a necessary item, it should be copied if not present,
- javax-resource.jar: required by the resource adaptor.

Configuration of server.xml

At the level of the main Tomcat configuration file, you need to declare the primary service provider by associating it with a logical name (JNDI): in tomcat%/conf/server.xml, edit the tag GlobalNamingResources and add:

- for Tomcat 7 and 8:

```
<Resource
        name="geoconcept/ugc/default"
        type="com.geoconcept.ugc.connect.tomcat.ConnectionFactory"
        scope="Shareable"
        description="UGC connection factory - local dll"
        auth="Container"
        RootDirectory="%geoconcept%\ugc\"
        factory="com.geoconcept.ugc.connect.tomcat.ConnectionFactory"
        ConnectionMode="LocalDll"
/>
```

For all versions of Tomcat, the RootDirectory parameter must correspond to the root of the ugc hierarchical tree external to the application server (containing the conf, native and reftables directories).

You can add the parameter `RefTablesDirectory` that allows definition of a directory in which you can store the reference tables. It will take the format: `RefTablesDirectory="D:\UGC\TablesRef"` .

In the Server section, you must add:

```
<Listener className="com.geoconcept.ugc.connect.tomcat.LifeCycleListener" resource="geoconcept/ugc/
default" />
```

Addition of the `LifeCycleListener` parameter is not essential under Windows, but it is, nevertheless, advisable. Conversely, under Linux it is essential (as otherwise, the Tomcat host will not terminate correctly).

In the scenario where the logical name used is not that used by default (geoconcept/ugc/defaut) adapt the configuration.

Creating resource / webapps links in Tomcat

For Tomcat, for webapps that are using a resource, it will be necessary to declare this at the level of the webapp context descriptor. For Tomcat 7 and 8, the webapp configuration files can be externalised from conf/server.xml and be placed under conf/catalina/localhost (by default).

So, for each webapp using the primary provider (referenced via JNDI), such as ugc-admin or ugc-samples, a series of context files are provided (for Tomcat 7 and 8), and for this reason, for each webapp that must be deployed, it will be necessary to:

- copy the .war into /webapps
- copy the context description .xml file into /conf/catalina/localhost

In the same way, if you create your own webapp that uses the primary provider, you must declare this link:

- by a separate file for Tomcat 7 and 87

for Tomcat 7 and 8:

- Copy the ugc-admin.xml file that is under %deliverable%\tomcat\conf into the %tomcat%/conf/catalina/localhost directory;
- Copy the axis.xml file that is in%liverable%\tomcat\webapps into the %tomcat%/conf/catalina/localhost directory (this is useful exclusively for using JEE Universal Geocoder in remote access mode).

To define other context files (file.xml) for other web applications, it will be necessary to add the following lines:

```
<ResourceLink
        global="geoconcept/ugc/default"
        name="geoconcept/ugc/default"
```

```
          type="com.geoconcept.ugc.connect.tomcat.ConnectionFactory"
/>
```

Axis

This part of the installation is only useful when using Universal geocoder JEE in remote access mode.

Copy the axis directory that is stored under install\tomcat\webapps into the %tomcat%\webapps directory.

The documentation for installing axis is accessible at : http://ws.apache.org/axis/java/install.html#webapp. To test axis, call http://localhost:8080/axis and then validate.

HappyAxis is an auto-test. Verify that it is not flagging up any problem with the libraries.

Any missing optional libraries will not be a problem (mail, attachments, xml security, etc).

AddressFinderService Installation

In %tomcat%/webapps/axis/WEB-INF/lib, copy the file called ugc-axis.jar that is in install\tomcat\axis-std.

Deployment of the AddressFinder service:

1. Run Tomcat,
2. Open a command line window in the \install\tomcat\axis-std directory and then execute deploy. (If the server port is not 8080, modify the file called deploy.bat. Do the same thing for the host if this is not localhost).

The Axis libraries must be in the CLASSPATH.

If the service is correctly deployed, the server responds:

```
...
Processing file AddressFinderServiceAxis.wsdd
<Admin>Done processing</Admin>
```

If the service is not correctly deployed, an error message displays.

Configuring JBoss

There are two deployment modes:

- EITHER the resource adaptor is deployed in standalone mode (on its own) and in this case, the service is accessible to any other module deployed on the same Jboss instance,
- OR the resource adaptor is deployed in an enterprise application that contains the publishing modules and in this case the service is not accessible to any other module deployed on the same Jboss instance (this is equivalent to a full pre-deployment).

Standalone adaptor

- Copy the ugc.rar file that is in%livrable%\jboss\standalone\ra\deploy into the %jboss%/deploy directory,

- Copy the ugc-ds.xml file that is in %deliverable%\jboss\standalone\ra\deploy into the %jboss%/deploy directory; then edit it in order to modify the RootDirectory value,
- Copy the ugc-common.jar file that is in %deliverable%\jboss\standalone\ra\lib into the %jboss%/lib directory.

In the form of the enterprise application archive (EAR)

- Copy the ugc-all.ear file that is in %livrable%\jboss\standalone\ra\deploy into the %jboss%/deploy directory,
- Copy the file called ugc-ds.xml stored in %livrable%\jboss\standalone\ra\deploy into the %jboss%/ deploy directory; then edit it in order to modify the RootDirectory value.

Configuring Jonas

The resource adaptor (ugc.rar) is deployed in standalone mode (on its own), the service is accessible to any other module deployed on the same Jboss instance whether it is a module supplied as ugc-admin or ugc-ws or a specific module written by the user).

The deployment and configuration procedure follows Jonas recommendations: the module configuration has to be modified and then deployed via one of the methods proposed by Jonas.

Deliverable

If the deliverable is in the form of an archive (a .zip file or .tar.gz) it needs to be extracted into a work directory (for example, via the “tar xvfz ugc-jee-5.0.112.tar.gz” under linux), if in expanded form, recopy it into a work directory.

Modify the ugc.rar configuration to adapt it to the local installation

The UGC resource adaptor configuration must be modified (ugc.rar located in jonas/standalone/ra/ deploy) in order to indicate the local installation filepath.

To do this, the Jonas procedure is as follows: the jonas-ra.xml file must be extracted from the archive (.rar) and then modified, and then the archive updated with the modified file. The Jonas RAConfig tool serves to generate the jonas-ra.xml from the standard ra.xml file, but here this is of no interest since the jonas-ra.xml file is supplied. So the jar tool from java jdk will suffice to extract and update the archive. If JAVA_HOME/bin is not in the PATH, it should be added temporarily (or, alternatively, invoke the jar tool by including this directory, e.g. /etc/java/jdk1.5.0_22/bin/jar)

1. Extraction of the jonas-ra.xml file: jar xvf ugc.rar META-INF/jonas-ra.xml. This will allow creation of a META-INF sub-directory and the jonas-ra.xml file inside,
2. Editing the jonas-ra.xml file: modify the extracted jonas-ra.xml file in order to indicate in the RootDirectory property value the directory in which UGC has been installed. If you wish to use a reference tables directory that is different to the default directory (%ugc%/reftables) indicate this in the RefTablesDirectory optional property. Otherwise, leave the option property as it is.
3. Update the archive (.rar): jar uvf ugc.rar META-INF/jonas-ra.xml

The updated archive is ready to be deployed on a Jonas instance.

Deployment on Jonas 4.x

The Jonas resource service must be configured and run at start-up (which is the case in the default configuration). That is to say that in the jonas.properties file, the resource value must be present in the list of values for the jonas.services property.

There are several ways of deploying a RAR file on a Jonas instance:

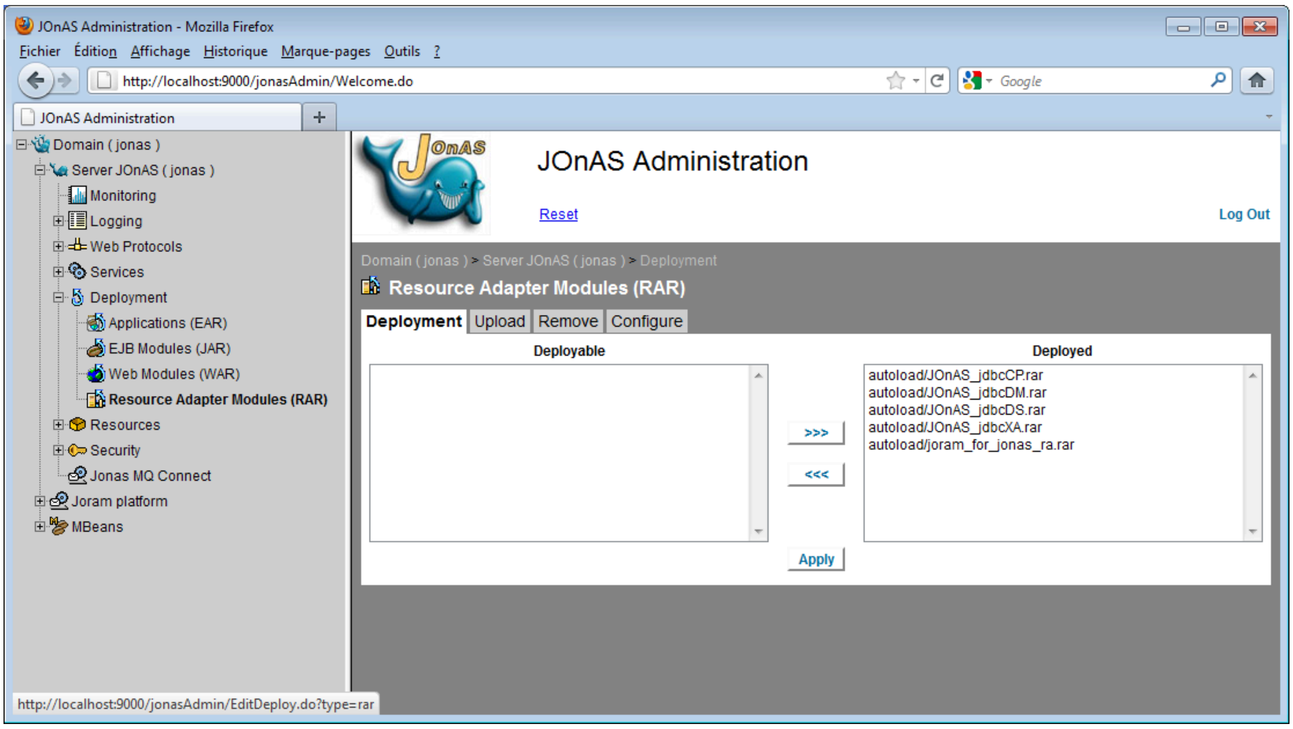
- modify the values of jonas.service.resource.resources property in the jonas.properties file: to add to it ugc (the suffix .rar is not mandatory). Example: jonas.service.resource.resources ugc,
- place the file in the autoload directory of connectors, by default this being $JONAS_BASE/rars/autoload,
- on the command line via the jonas admin -a ugc.rar command,
- using the web administration console (jonasAdmin) accessible by default via http://localhost:9000/jonasAdmin,

For further details, consult http://jonas.ow2.org/JOnAS_4_7/doc/PG_Connector.html#PG_Connector-Use

Deployment via the rars/autoload directory and deployment via jonasAdmin are described below.

- Deployment via the rars/autoload directory: copy the ugc.rar file prepared in $JONAS_BASE/rars/autoload and restart Jonas.
- Deployment via jonasAdmin
  - Select Deployment Deployment > Resource Adaptor Modules (RAR) in the hierarchical tree,

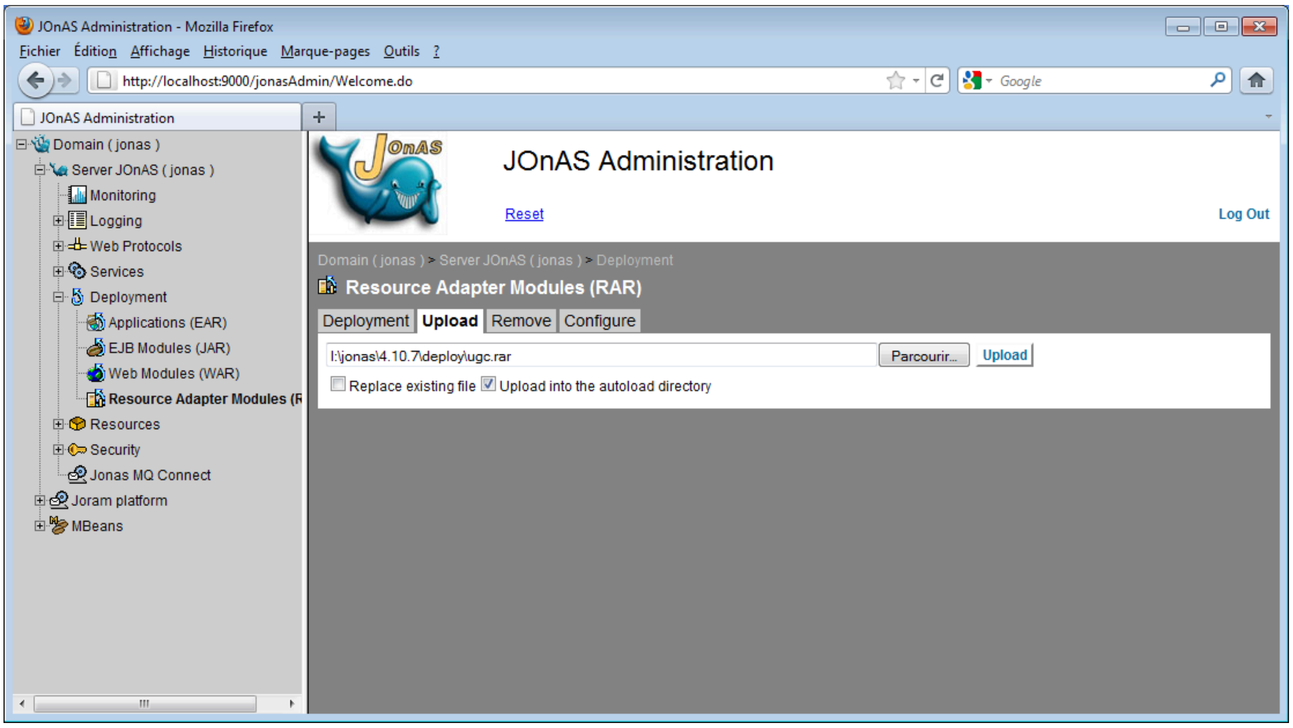
Jonas Deployment



autoload/JOnAS_jdbcCP.rar
autoload/JOnAS_jdbcDM.rar
autoload/JOnAS_jdbcDS.rar
autoload/JOnAS_jdbcXA.rar
autoload/joram_for_jonas_ra.rar

- Select the Upload tab,

Upload tab

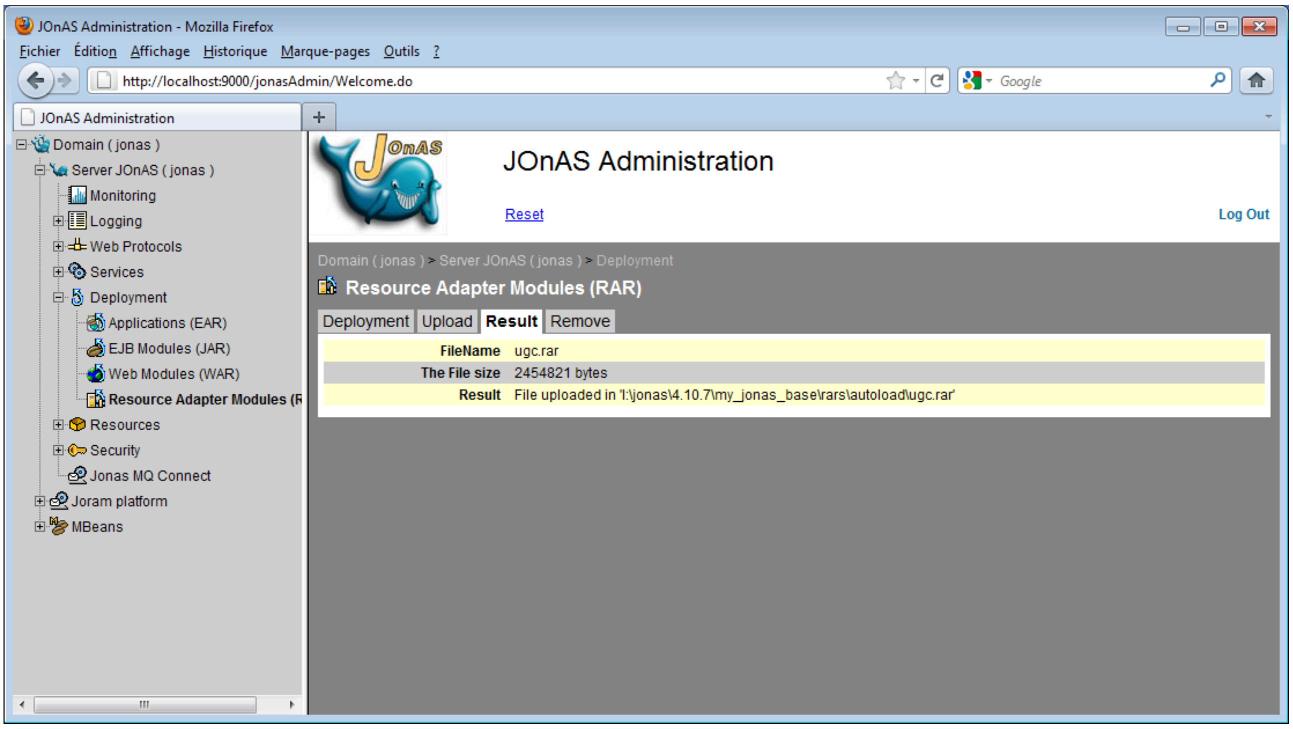


I:\jonas\4.10.7\deploy\ugc.rar

[ ] Replace existing file [✓] Upload into the autoload directory

• Upload the file from the location where it has been prepared on the disk. Check upload into the autoload directory, if not, the deployment will only take place for the current session of the Jonas instance. Then validate via the Upload button.

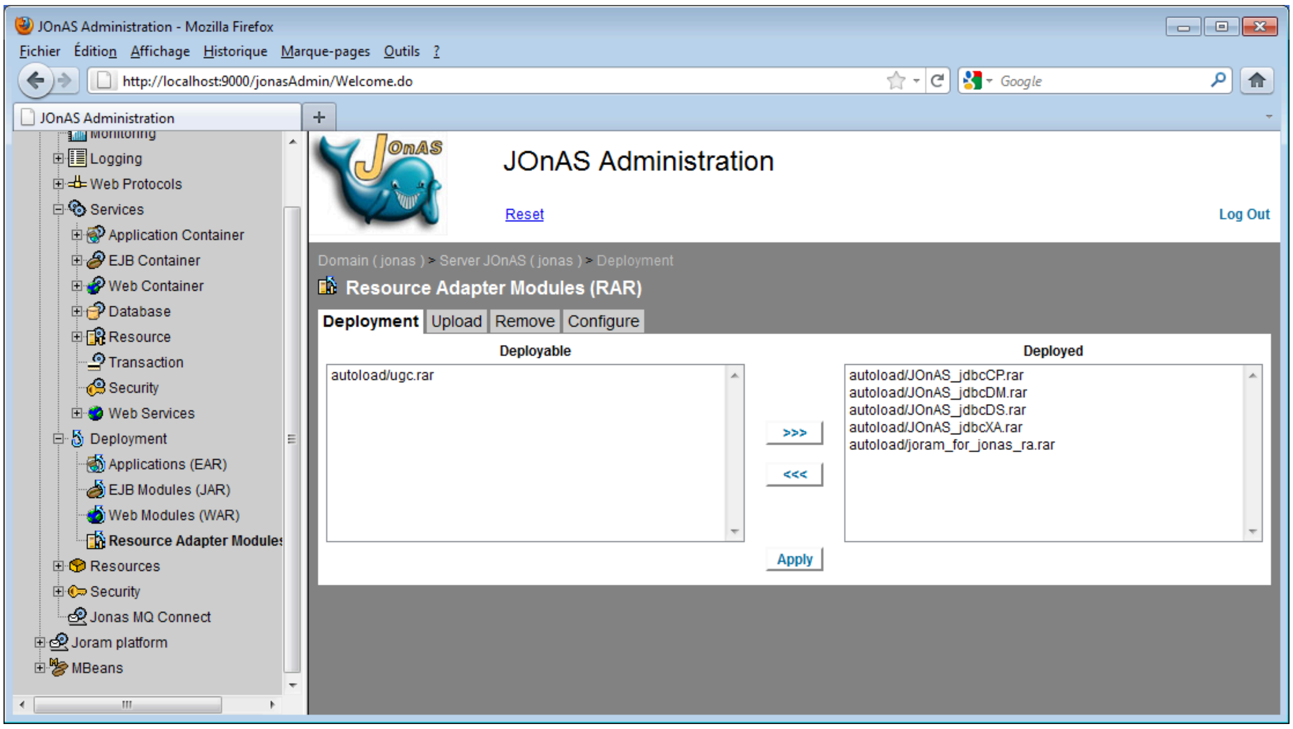
Result tab



FileName	ugc.rar
The File size	2454821 bytes
Result	File uploaded in 'I:\jonas\4.10.7\my_jonas_base\rars\autoload\ugc.rar'

This will have the effect of copying ugc.rar into the JONAS_BASE/rars/autoload directory.

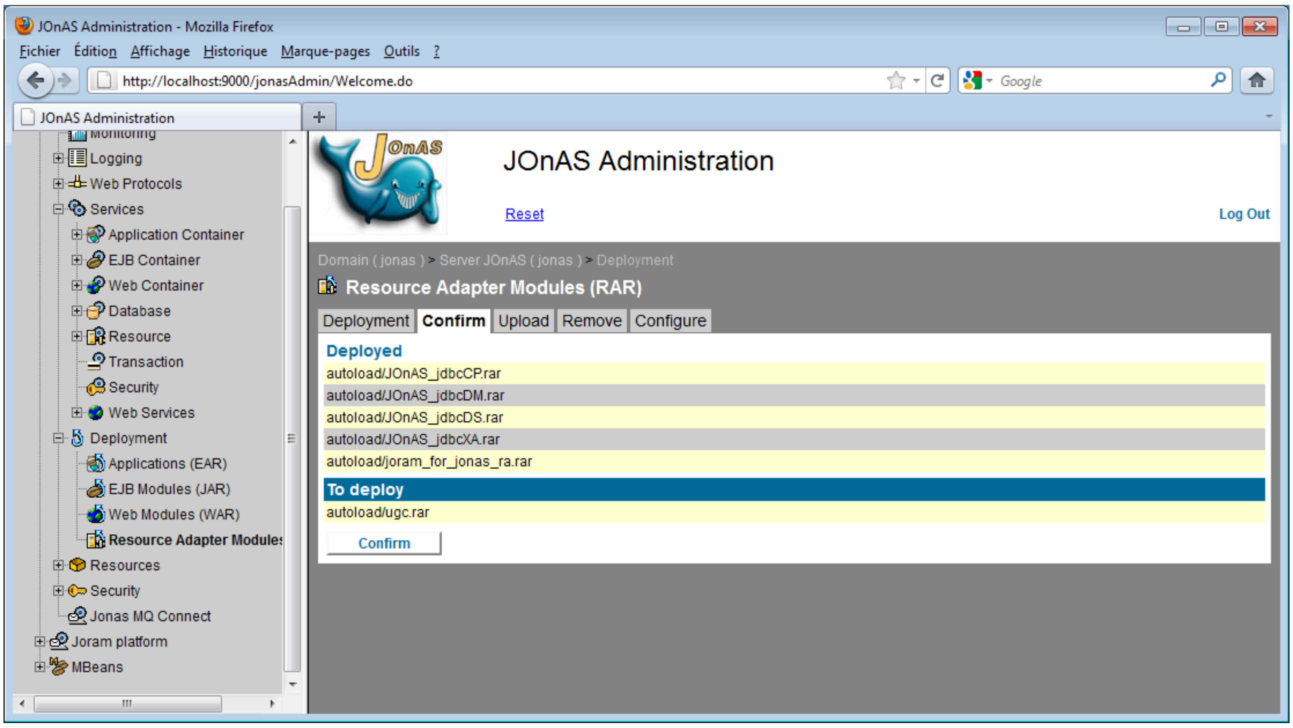
Deployment tab



Deployable	Deployed
autoload/ugc.rar	autoload/JOnAS_jdbcCP.rar
	autoload/JOnAS_jdbcDM.rar
	autoload/JOnAS_jdbcDS.rar
	autoload/JOnAS_jdbcXA.rar
	autoload/joram_for_jonas_ra.rar

The module is now “deployable”.

Confirm tab

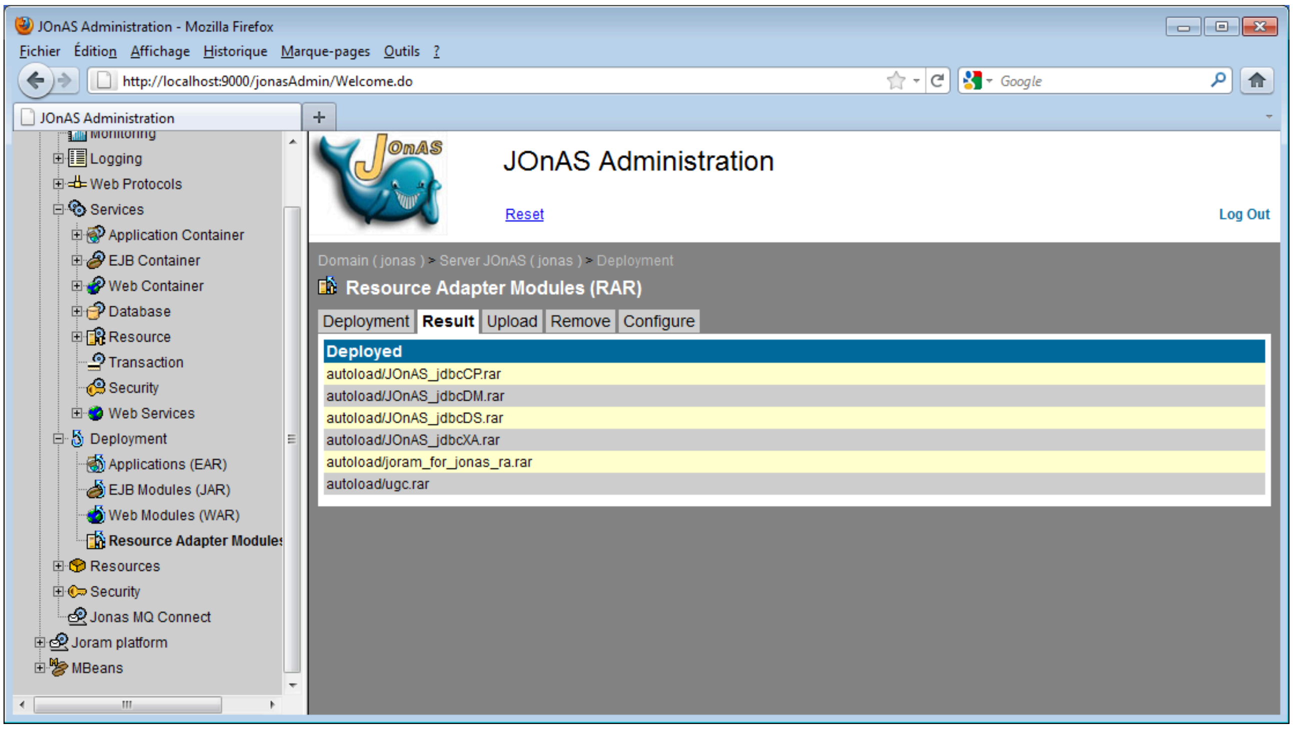

JOnAS Administration - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils ?
http://localhost:9000/jonasAdmin/Welcome.do
JOnAS Administration
Reset
Log Out
Domain (jonas) > Server JOnAS (jonas) > Deployment
Resource Adapter Modules (RAR)
Deployment
Confirm
Upload
Remove
Configure
Confirm

Deployed
autoload/JOnAS_jdbcCP.rar
autoload/JOnAS_jdbcDM.rar
autoload/JOnAS_jdbcDS.rar
autoload/JOnAS_jdbcXA.rar
autoload/joram_for_jonas_ra.rar

To deploy
autoload/ugc.rar

- D) Add ugc.rar to the modules already deployed and confirm.

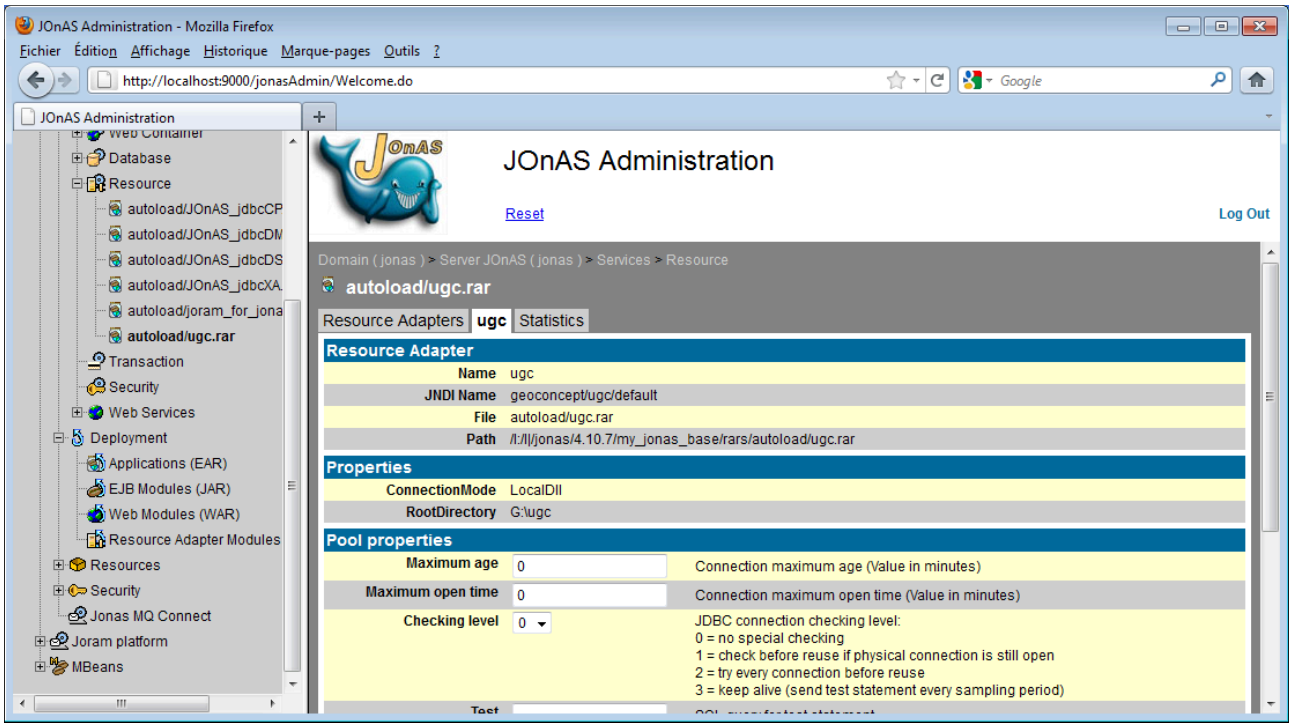
Result tab



Deployed
autoload/JOnAS_jdbcCP.rar
autoload/JOnAS_jdbcDM.rar
autoload/JOnAS_jdbcDS.rar
autoload/JOnAS_jdbcXA.rar
autoload/joram_for_jonas_ra.rar
autoload/ugc.rar

The module is now deployed. At the level of the hierarchical tree on the server (on the right) it appears in Services/Resource:

UGC tab



Resource Adapter	
Name	ugc
JNDI Name	geoconcept/ugc/default
File	autoload/ugc.rar
Path	/l:/l/jonas/4.10.7/my_jonas_base/rars/autoload/ugc.rar

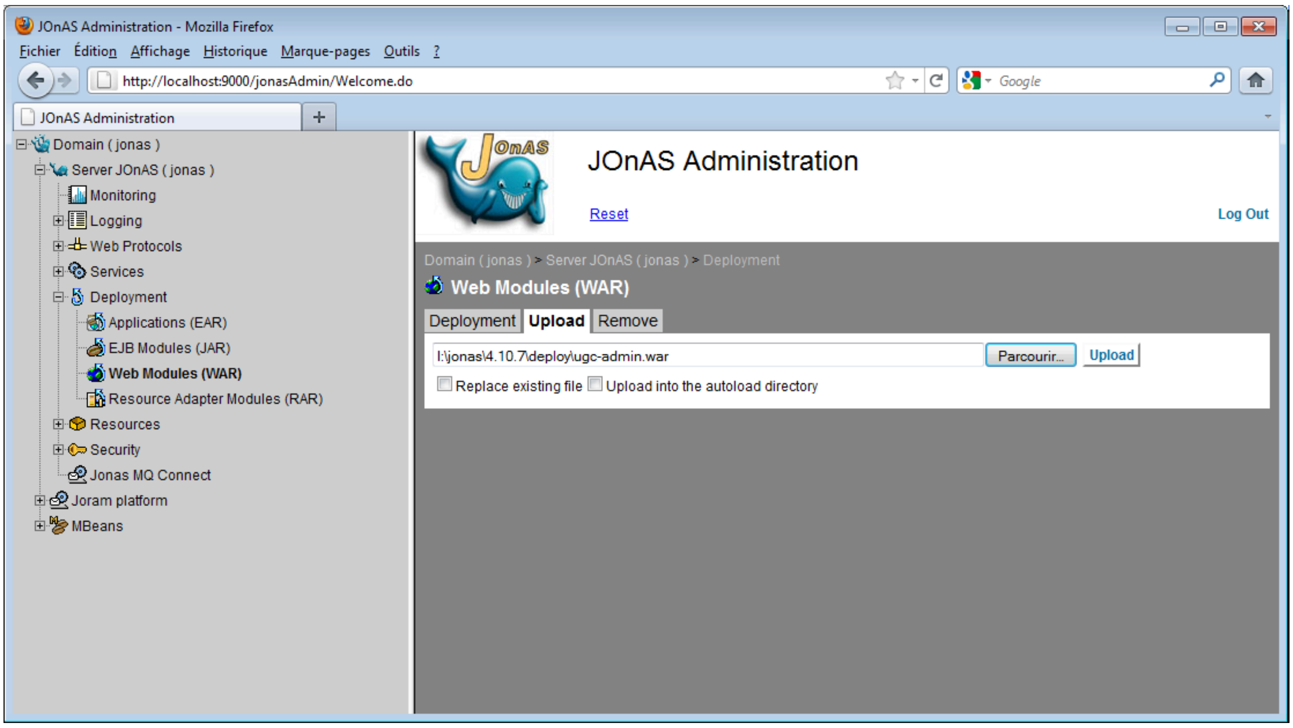
Properties	
ConnectionMode	LocalDll
RootDirectory	G:\ugc

Pool properties		
Maximum age	0	Connection maximum age (Value in minutes)
Maximum open time	0	Connection maximum open time (Value in minutes)
Checking level	0	JDBC connection checking level: 0 = no special checking 1 = check before reuse if physical connection is still open 2 = try every connection before reuse 3 = keep alive (send test statement every sampling period)

Deployment of ugc-admin (optional):

ugc-admin is an administration webapp that tests (among other things) for the correct functioning of ugc-jee. Its deployment is optional. The principle for deployment is the same as for the (ugc.rar) resource adaptor, even though in this instance it concerns just the Web Modules (WAR) section, and this module does not require configuration.

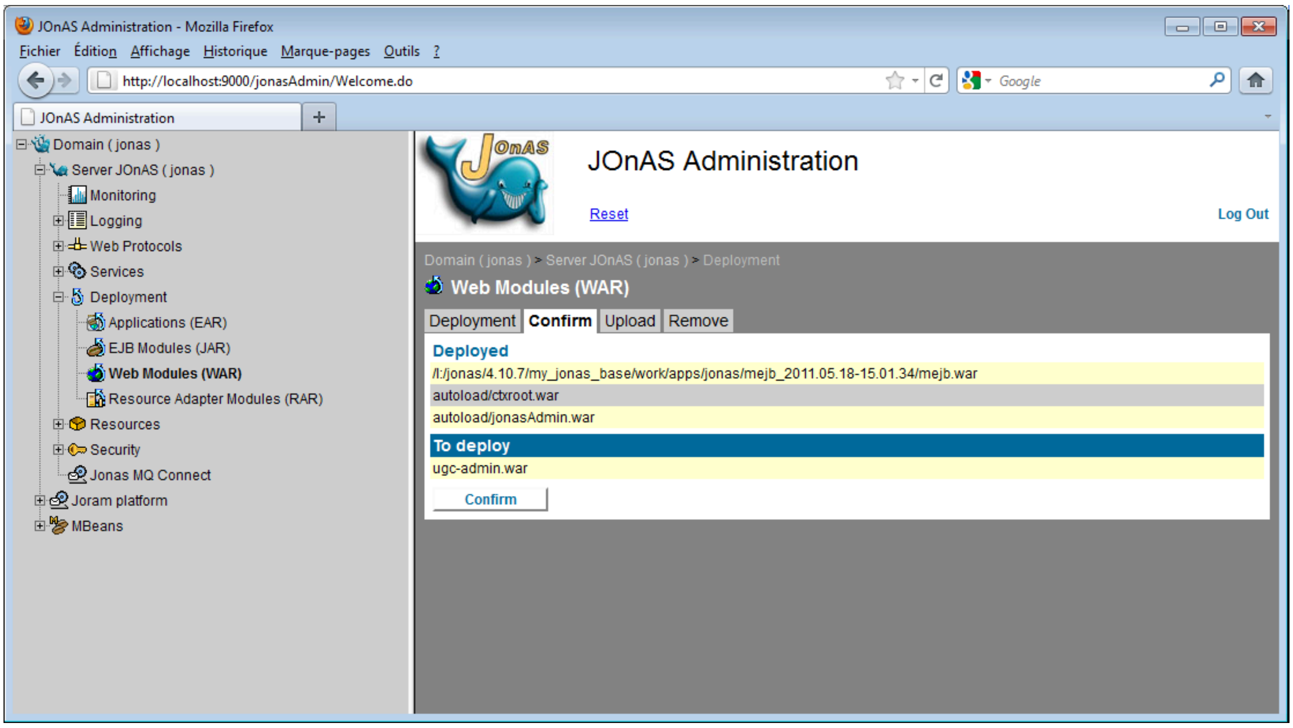
Uploading the ugc-admin.war file



I:\jonas\4.10.7\deploy\ugc-admin.war

Replace existing file
Upload into the autoload directory

Then deployment:



Deployment of the ugc-admin.war file

JOnAS Administration - Mozilla Firefox

http://localhost:9000/jonasAdmin/Welcome.do

Domain (jonas) > Server JOnAS (jonas) > Deployment

Web Modules (WAR)

Deployment | Confirm | Upload | Remove

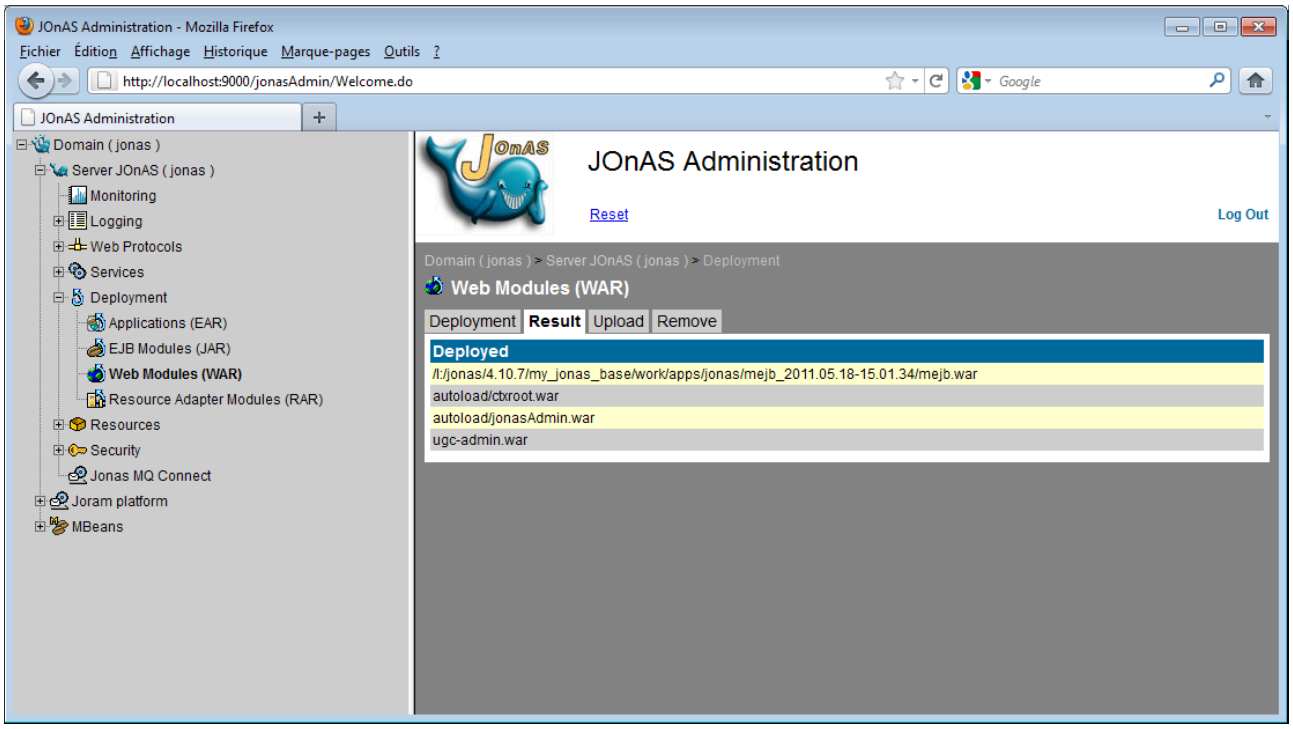
Deployed

/l:/jonas/4.10.7/my_jonas_base/work/apps/jonas/mejb_2011.05.18-15.01.34/mejb.war
autoload/ctxroot.war
autoload/jonasAdmin.war

To deploy

ugc-admin.war

Confirm



Deployment of the ugc-admin.war file

JOnAS Administration - Mozilla Firefox

http://localhost:9000/jonasAdmin/Welcome.do

Domain (jonas) > Server JOnAS (jonas) > Deployment

Web Modules (WAR)

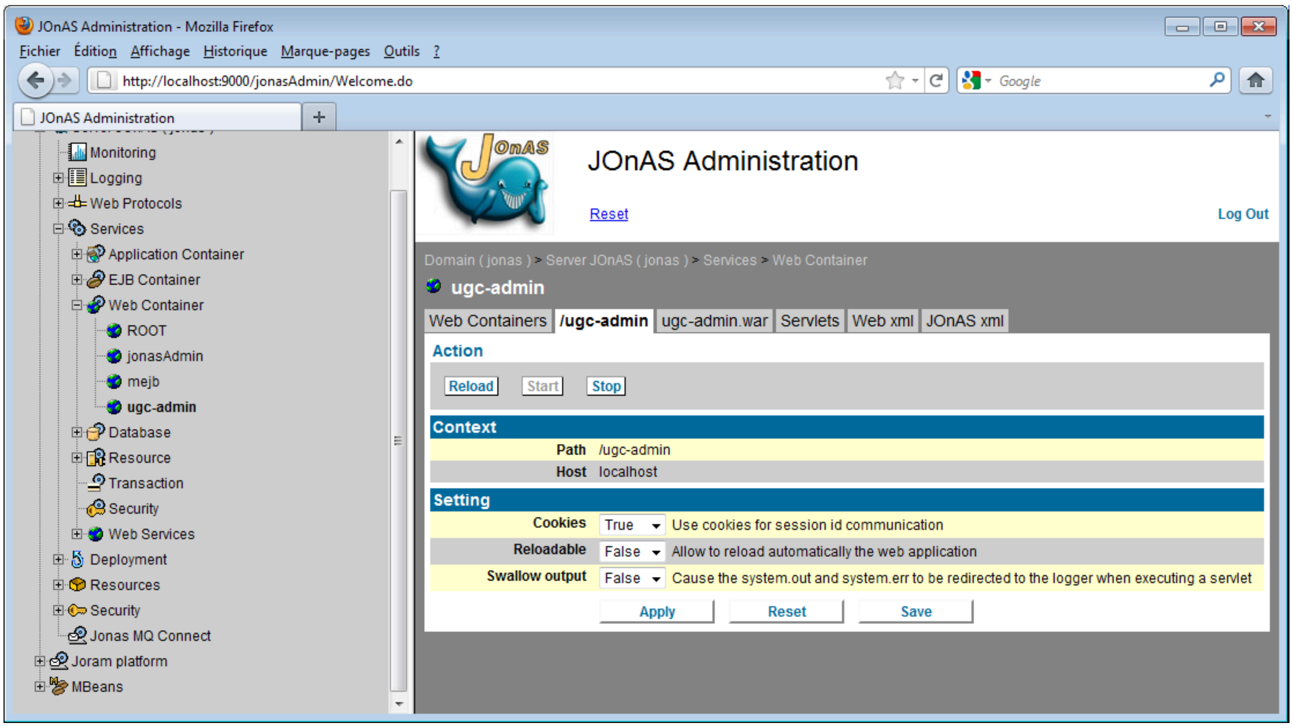
Deployment | Result | Upload | Remove

Deployed

/l:/jonas/4.10.7/my_jonas_base/work/apps/jonas/mejb_2011.05.18-15.01.34/mejb.war
autoload/ctxroot.war
autoload/jonasAdmin.war
ugc-admin.war

The webapp is deployed.

Deployment of the ugc-admin.war file



JOnAS Administration

Domain (jonas) > Server JOnAS (jonas) > Services > Web Container

ugc-admin

Action

Context

Path	/ugc-admin
Host	localhost

Setting

Cookies	True	Use cookies for session id communication
Reloadable	False	Allow to reload automatically the web application
Swallow output	False	Cause the system.out and system.err to be redirected to the logger when executing a servlet

Setting up a web service client provider on jonas 4.x

Jonas 4.x not delivering any saaj 1.3 implementation, it is necessary to place the saaj-impl-1.3.2.jar library (that can be taken in redist/providers-libs/ws-saas-client-saaj13/libs) or a more recent version in JONAS_BASE/libs/apps.

Example of configuring a web service client provider of the saas type based on saaj (extracted from a jonas-ra.xml file):

```
<jonas-config-property>
        <jonas-config-property-name>ConnectionMode</jonas-config-property-name>
        <jonas-config-property-value>WssClient</jonas-config-property-value>
</jonas-config-property>
<jonas-config-property>
        <jonas-config-property-name>ServerUri</jonas-config-property-name>
        <jonas-config-property-value>http://owss.geoconcept.com:2010/service/soap11</jonas-config-property-
value>
</jonas-config-property>
<jonas-config-property>
        <jonas-config-property-name>Key</jonas-config-property-name>
        <jonas-config-property-value>123456</jonas-config-property-value>
</jonas-config-property>
```

Deployment on Jonas 5.x

The deployment is identical to Jonas 4.x, except for the fact that the $JONAS_BASE/deploy directory replaces the directories that are differentiated $JONAS_BASE/rars/autoload, $JONAS_BASE/webapps/autoload) for the deployment.

Deploying and configuring BES

The deployment environment described is Borland Enterprise server, AppServer Edition version 5.2.1.

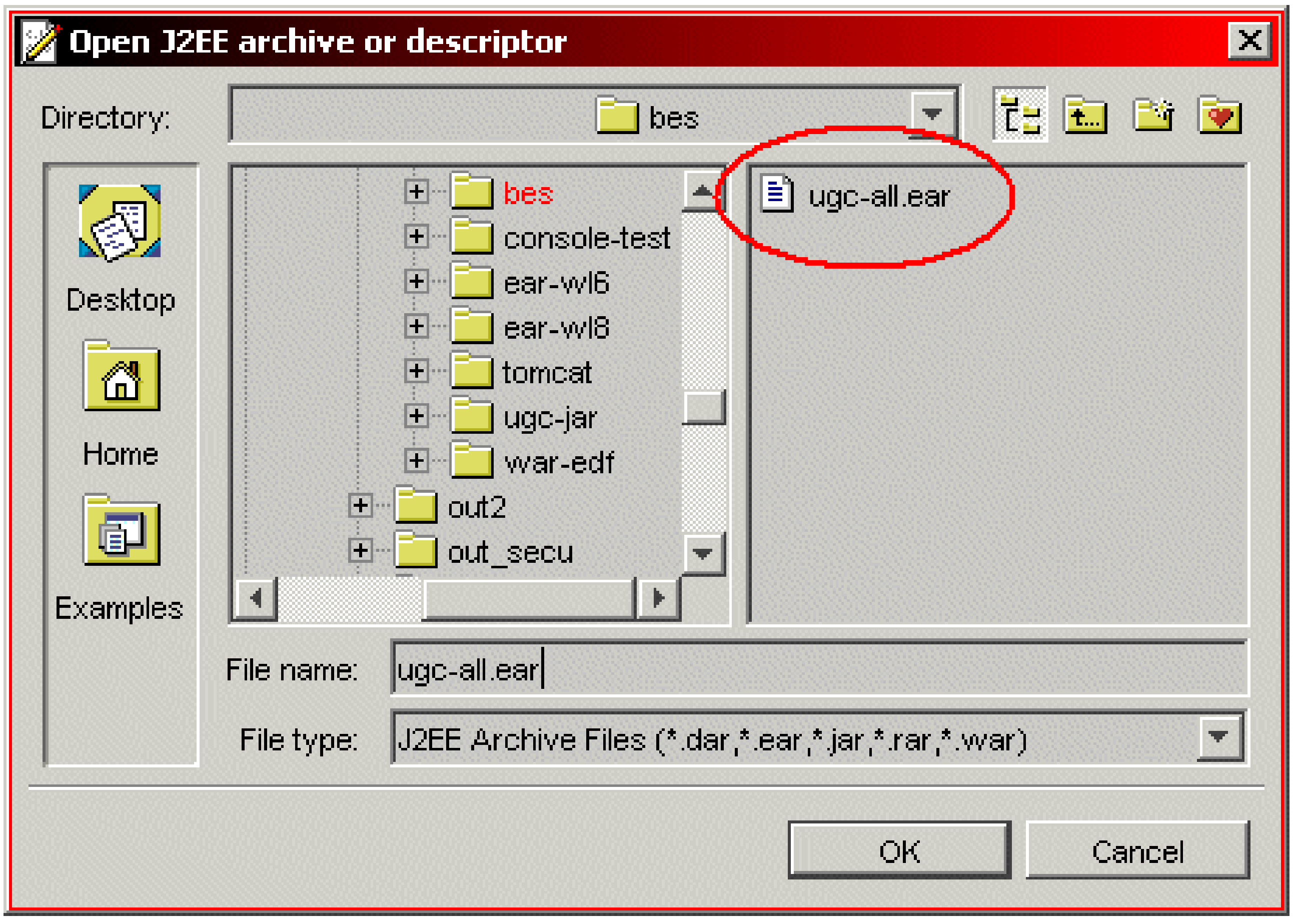
The resource adaptor (ugc.rar) and a series of publishing modules (ugc-admin.rar) are deployed in the form of an enterprise application (.ear archive): ugc-all.ear.

Configuring ugc-all.ear: before deploying ugc-all.ear, it will be necessary to configure it to update the local installation filepaths.

To do this, the simplest solution is to use Borland Deployment Descriptor Editor (DDEditor).

1. Execute ddeditor and open ugc-all.ear

ugc-all.ear


Open J2EE archive or descriptor

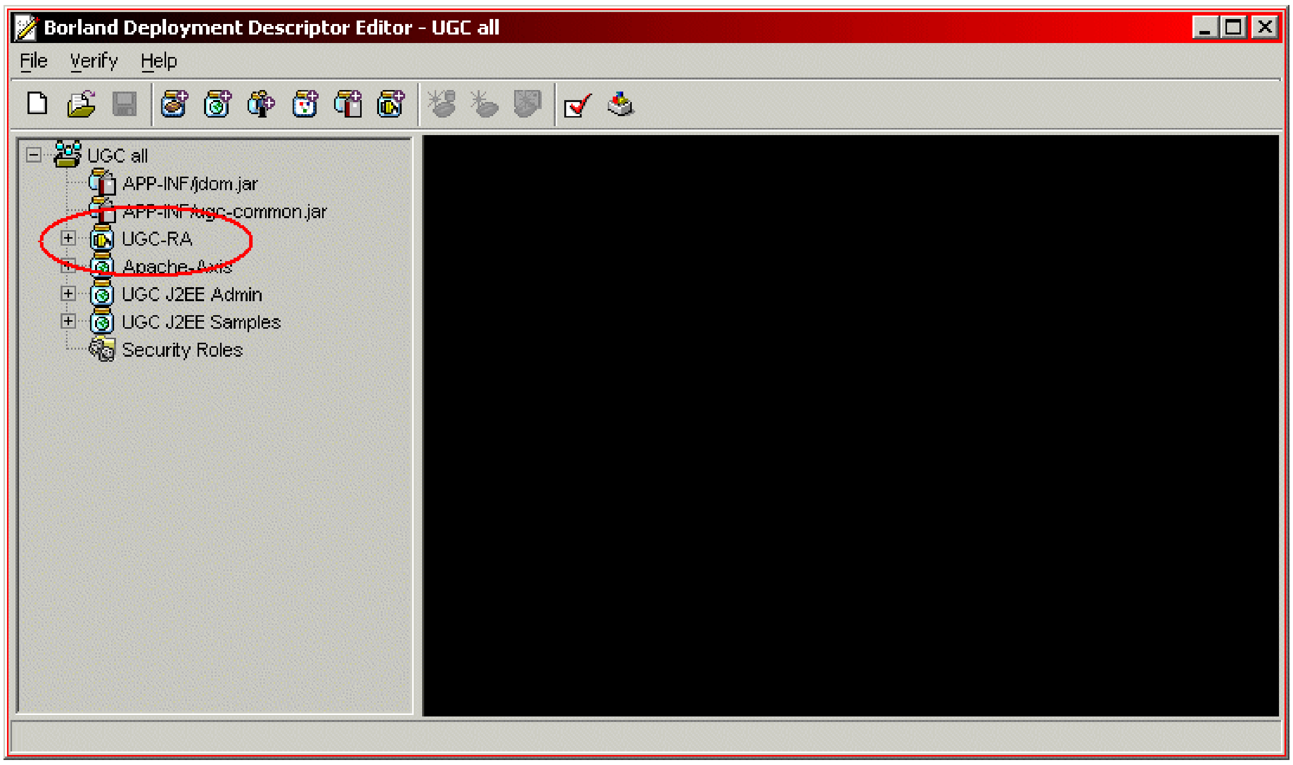
Directory: bes

File name: ugc-all.ear

File type: J2EE Archive Files (*.dar,*.ear,*.jar,*.rar,*.war)

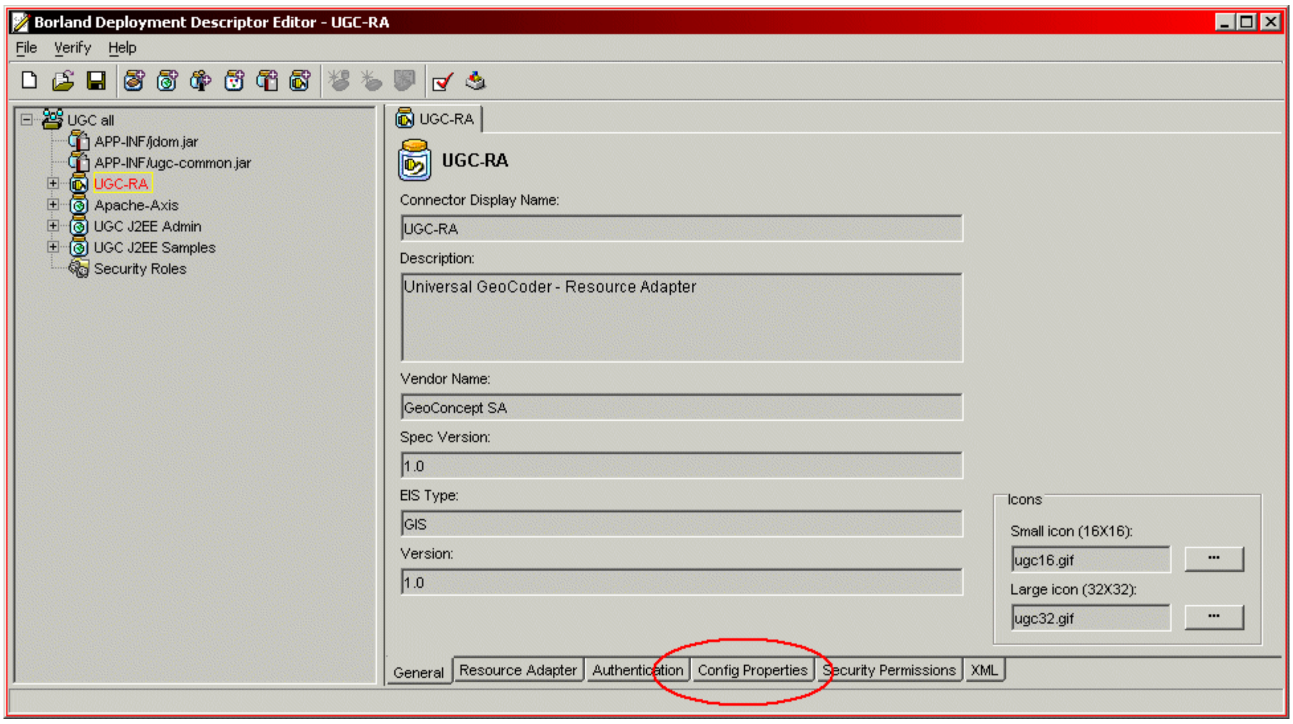
1. Select the resource adaptor module (rar) in the enterprise application archive (ear)

Resource adaptor module



1. Select the config properties tag in the resource adaptor module

Config properties tab



UGC-RA

Connector Display Name: UGC-RA

Description: Universal GeoCoder - Resource Adapter

Vendor Name: GeoConcept SA

Spec Version: 1.0

EIS Type: GIS

Version: 1.0

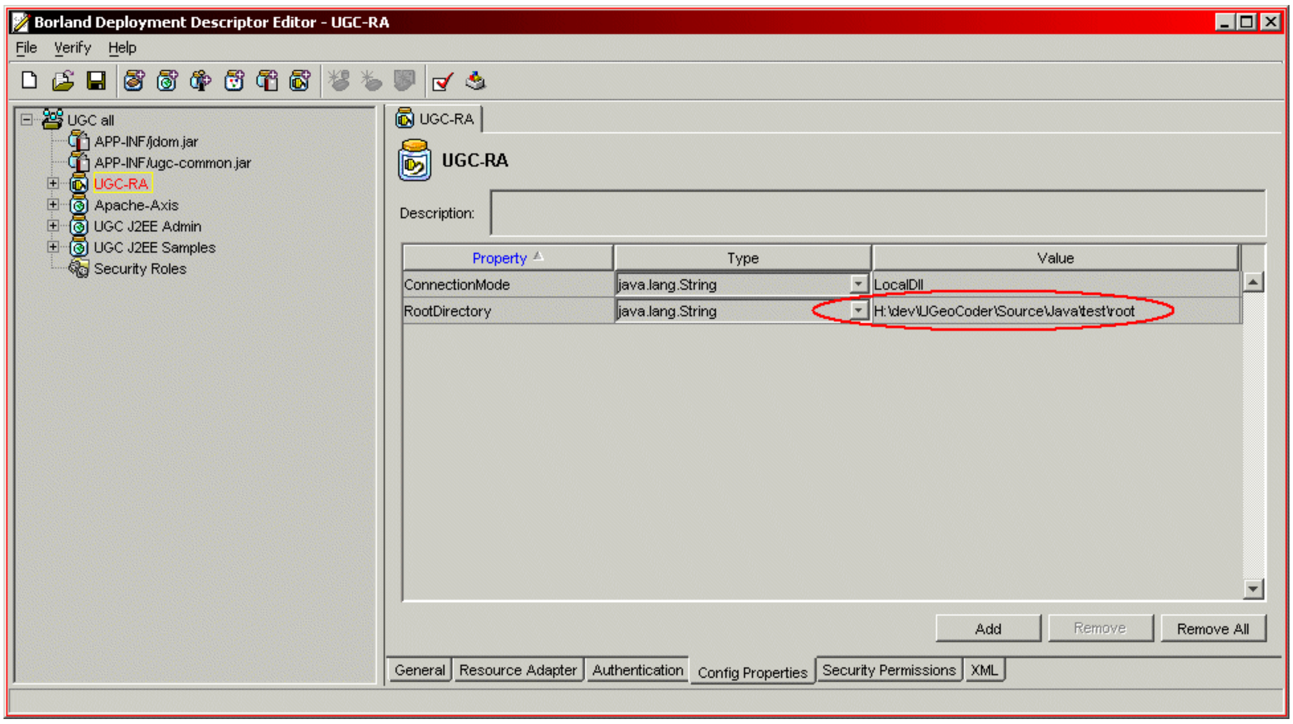
Icons

Small icon (16X16): ugc16.gif

Large icon (32X32): ugc32.gif

1. Indicate the installation directory for external files to the application server

Config properties tab



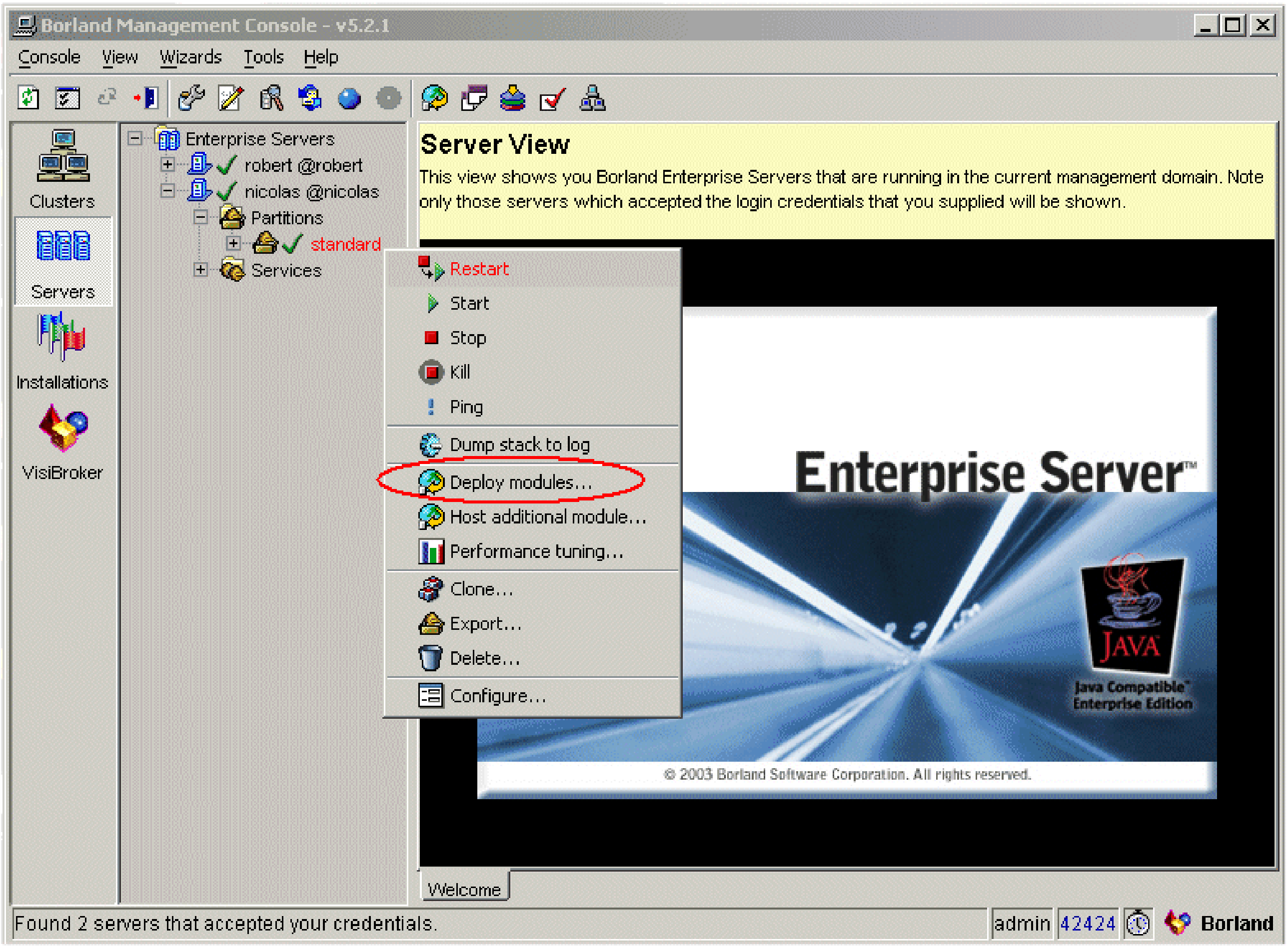
Property	Type	Value
ConnectionMode	java.lang.String	LocalDll
RootDirectory	java.lang.String	H:\dev\UGeoCoder\Source\Java\test\root

1. Save any changes applied

Deployment of the enterprise application (ugc-all.ear) in Borland Management Console

Select the partition on which you want to deploy:

Selecting the partition



Server View

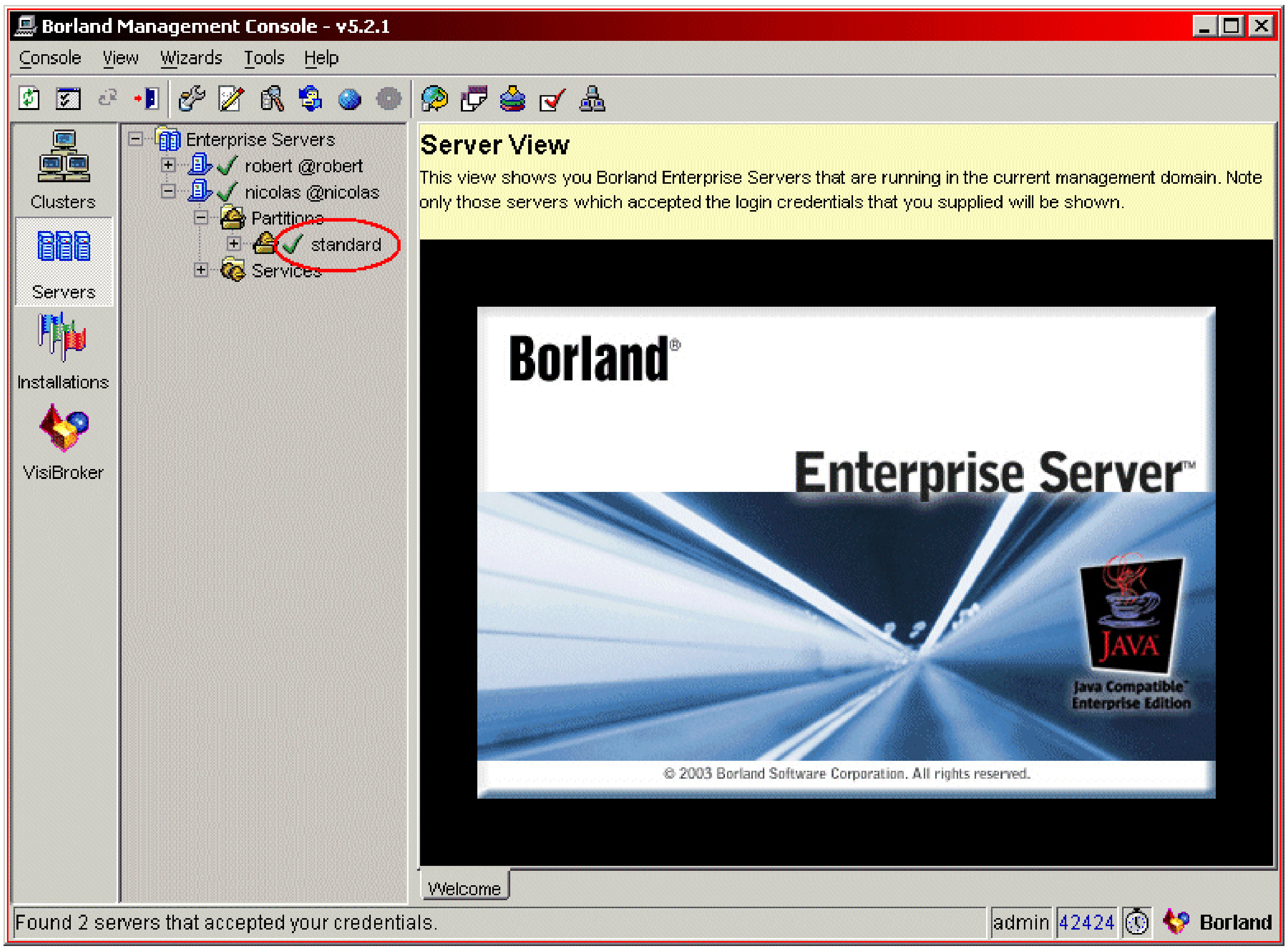
This view shows you Borland Enterprise Servers that are running in the current management domain. Note only those servers which accepted the login credentials that you supplied will be shown.

Found 2 servers that accepted your credentials.

VisiConnect must be enabled in the PartitionServices of the target partition.

Apply a right-click to the partition and select '*Deploy modules …*'

Deploy modules



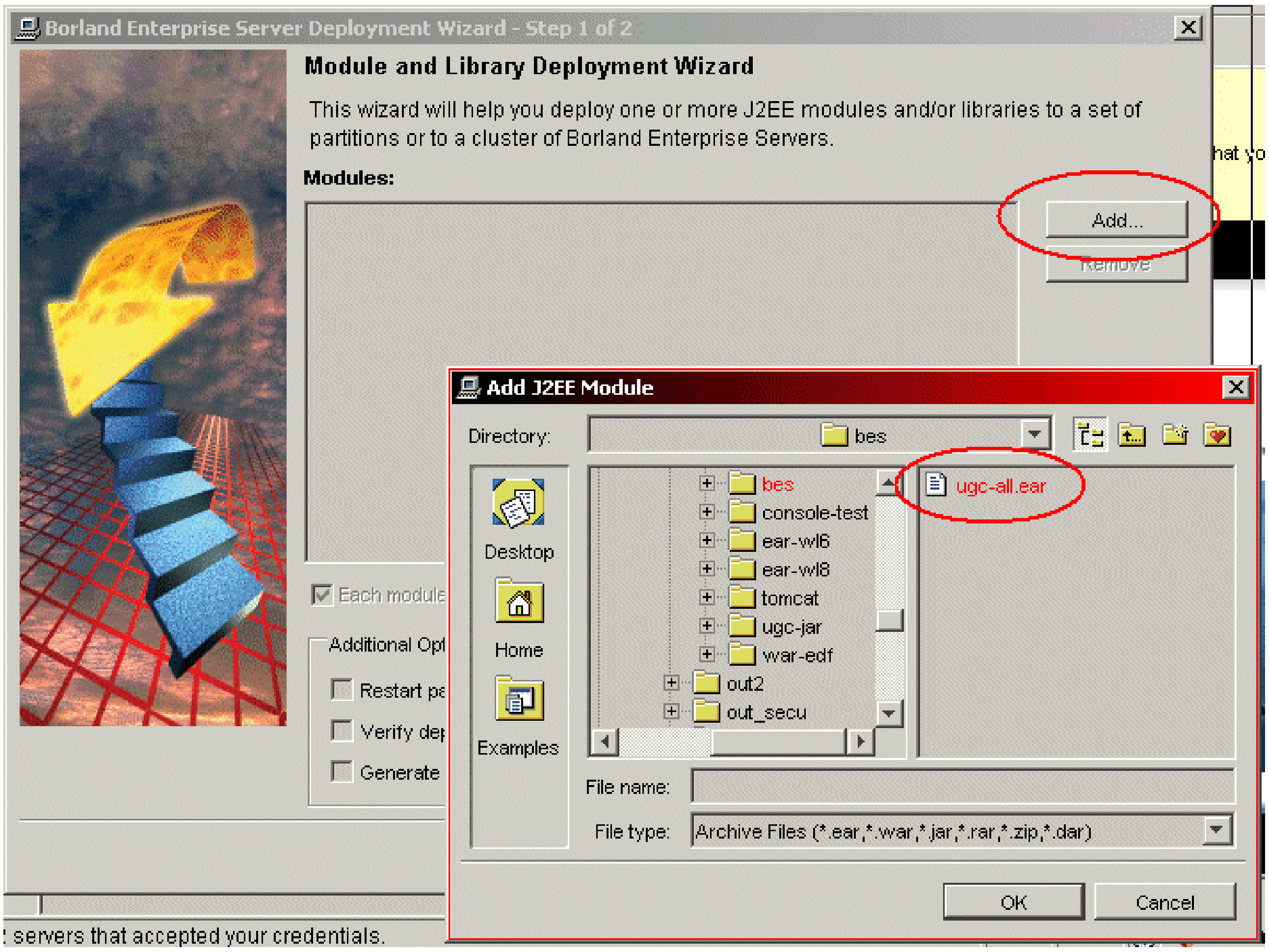
Server View

This view shows you Borland Enterprise Servers that are running in the current management domain. Note only those servers which accepted the login credentials that you supplied will be shown.

Found 2 servers that accepted your credentials.

Select the ugc-all.ear file that you have modified in the previous step and validate:

Validate the file



Borland Enterprise Server Deployment Wizard - Step 1 of 2

Module and Library Deployment Wizard

This wizard will help you deploy one or more J2EE modules and/or libraries to a set of partitions or to a cluster of Borland Enterprise Servers.

Modules:

Each module

Additional Opt

Restart pa

Verify dep

Generate

servers that accepted your credentials.

Add J2EE Module

Directory: bes

bes
console-test
ear-wl6
ear-wl8
tomcat
ugc-jar
war-edf
out2
out_secu

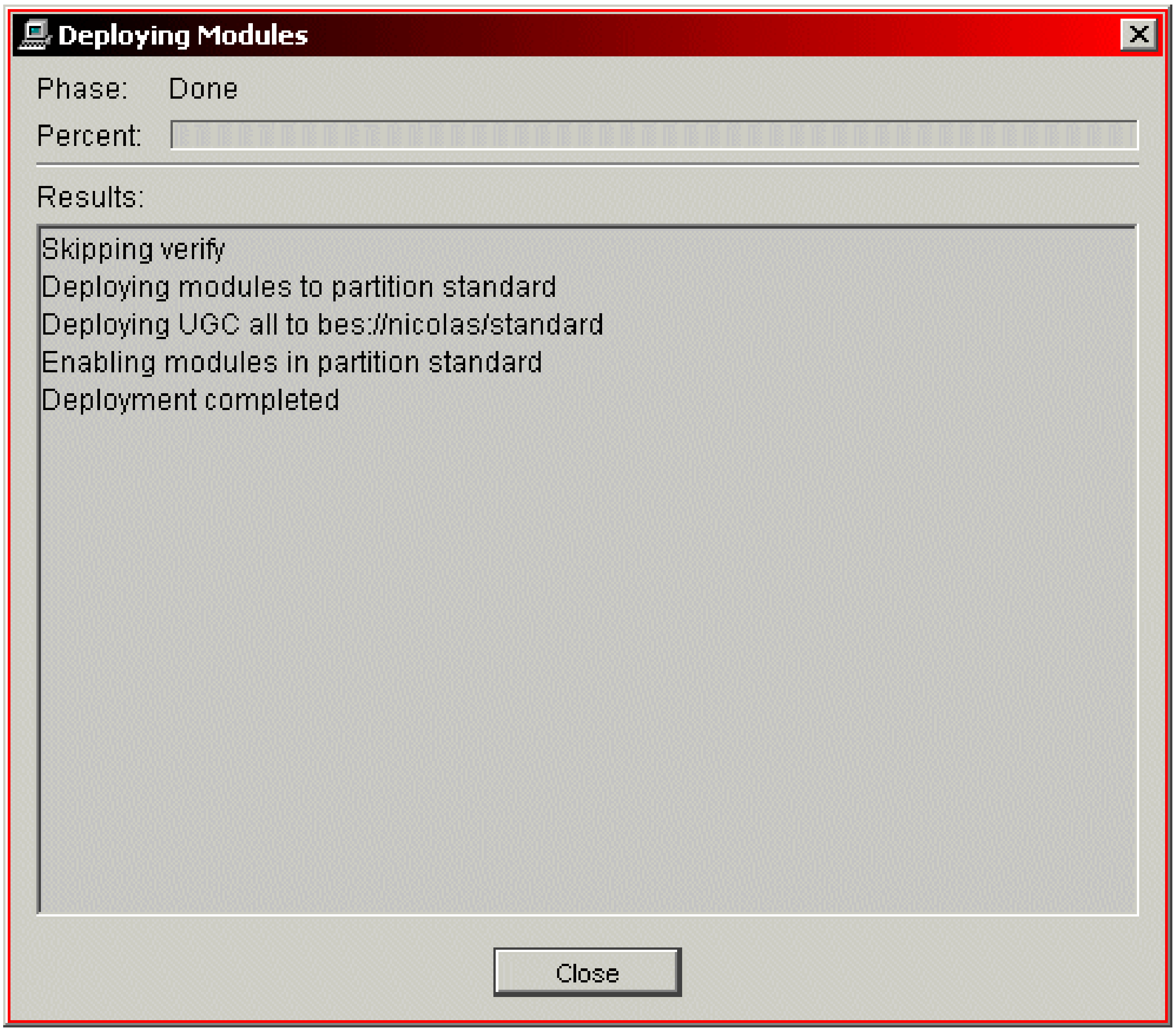
ugc-all.ear

File name:

File type: Archive Files (*.ear,*.war,*.jar,*.rar,*.zip,*.dar)

Select *finish* and the deployment terminates:

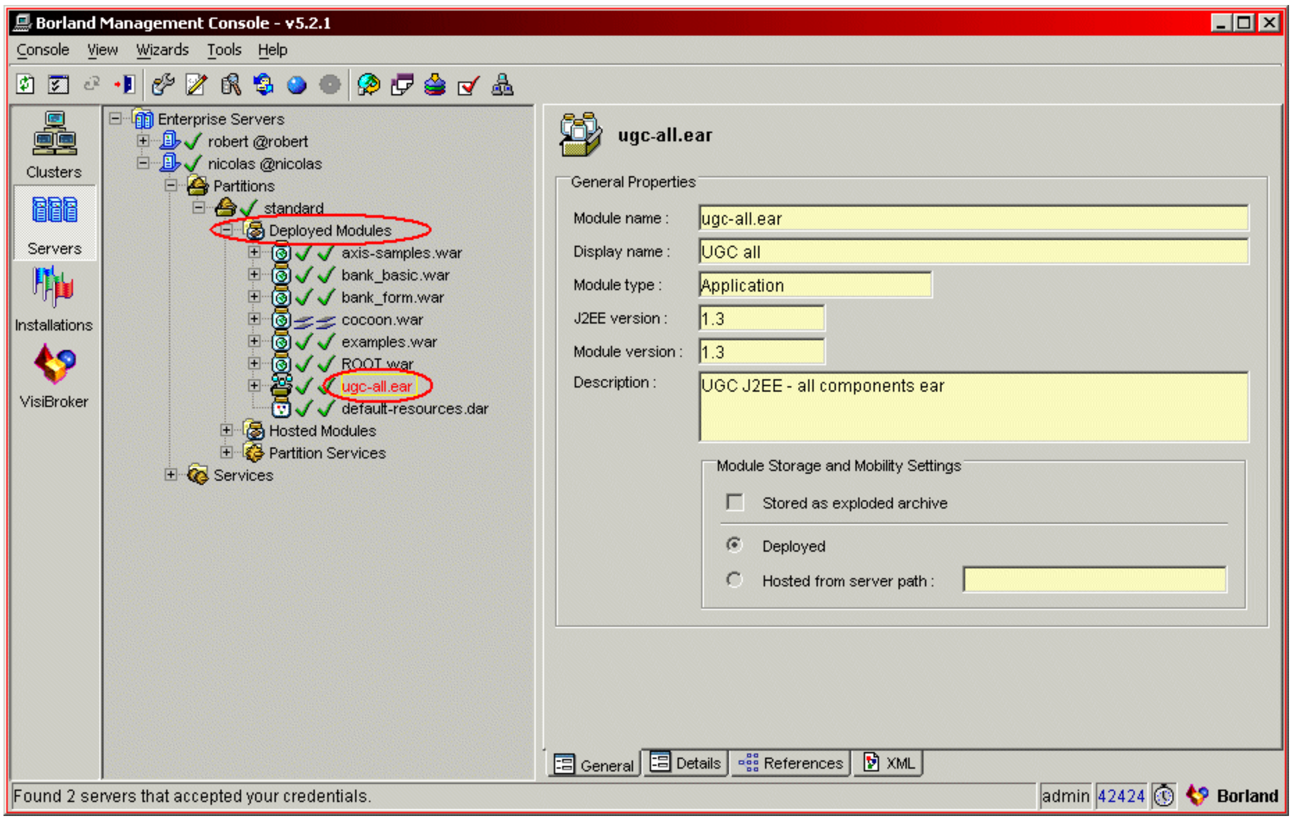
Deployment



```
Skipping verify
Deploying modules to partition standard
Deploying UGC all to bes://nicolas/standard
Enabling modules in partition standard
Deployment completed
```

The ugc-all application appears now in the list of modules deployed for the selected partition:

List of modules deployed



ugc-all.ear

General Properties

Module name :	ugc-all.ear
Display name :	UGC all
Module type :	Application
J2EE version :	1.3
Module version :	1.3
Description :	UGC J2EE - all components ear

Module Storage and Mobility Settings

☐ Stored as exploded archive

[✓ Deployed]

Hosted from server path :

Found 2 servers that accepted your credentials.

Deployment and configuration of WebSphere

The deployment environment described is IBM WebSphere Application Server version 8.5.

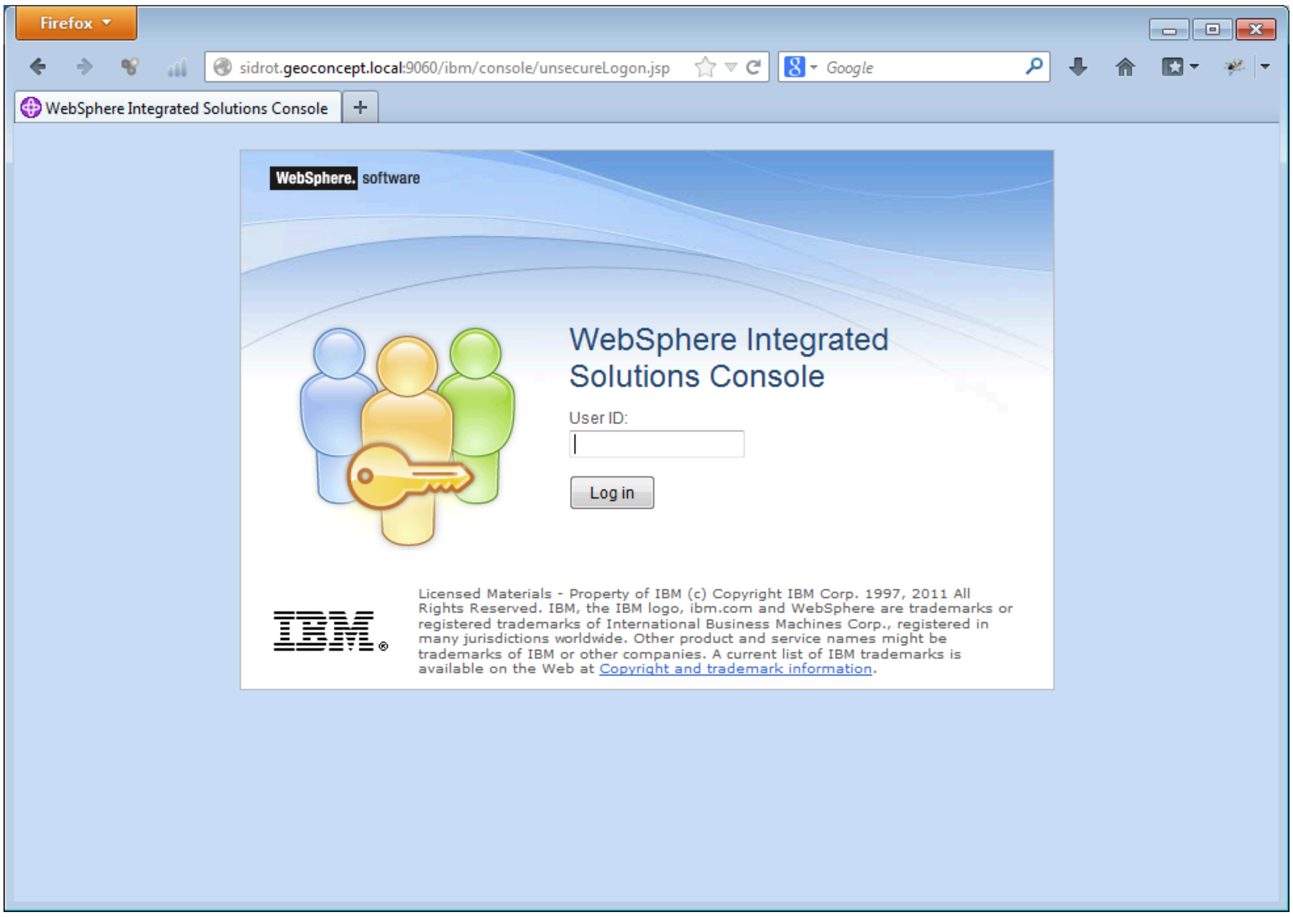
The resource adaptor (ugc.rar) and a series of optional publishing modules (ugc-admin.war, ugc-samples.war, ugc-axis-fusion.war, etc) are deployed in the form of separate (standalone) modules.

ugc-admin is an optional module but it allows, among other things, validation of the correct deployment of the resource adaptor.

There are many ways to execute deployments in WebSphere Application Server (via scripting, or programming, via files of properties, via wsadmin, etc), here the deployment process described will be that of WebSphere integrated Solutions Console (also called the administrative console).

1. Deployment of the resource adaptor (ugc.rar): log in to WebSphere Integrated Solutions Console

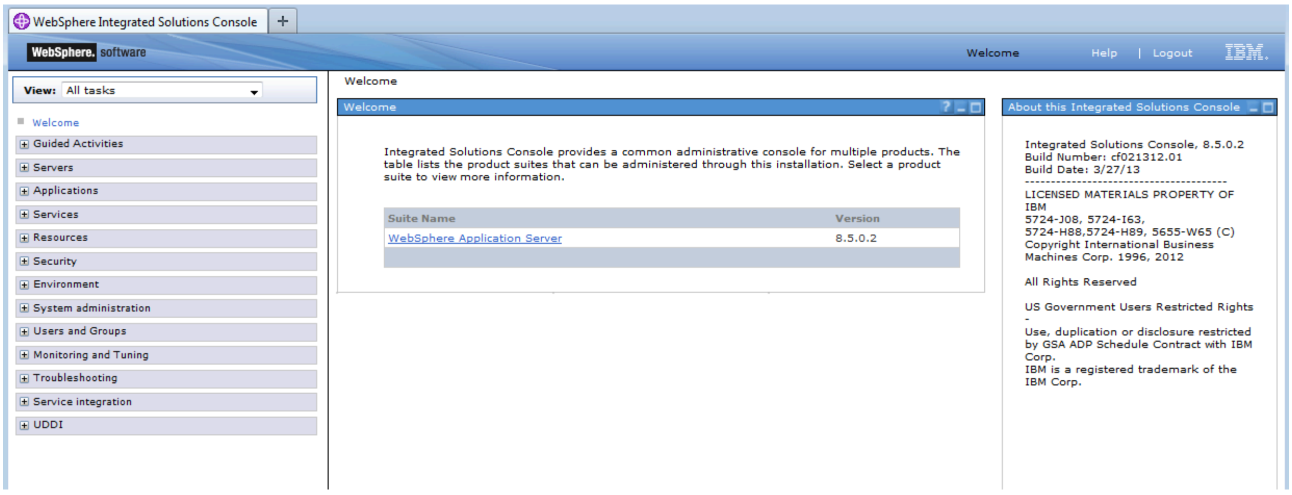
Deployment of the resource adaptor


Firefox
sidrot.geoconcept.local:9060/ibm/console/unsecureLogon.jsp
Google
WebSphere Integrated Solutions Console
WebSphere. software
User ID:
Log in

WebSphere Integrated Solutions Console

Licensed Materials - Property of IBM (c) Copyright IBM Corp. 1997, 2011 All Rights Reserved. IBM, the IBM logo, ibm.com and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information.

Deployment of the resource adaptor



Integrated Solutions Console provides a common administrative console for multiple products. The table lists the product suites that can be administered through this installation. Select a product suite to view more information.

Suite Name	Version
WebSphere Application Server	8.5.0.2

Integrated Solutions Console, 8.5.0.2
Build Number: cf021312.01
Build Date: 3/27/13

LICENSED MATERIALS PROPERTY OF IBM
5724-J08, 5724-I63, 5724-H88,5724-H89, 5655-W65 (C) Copyright International Business Machines Corp. 1996, 2012

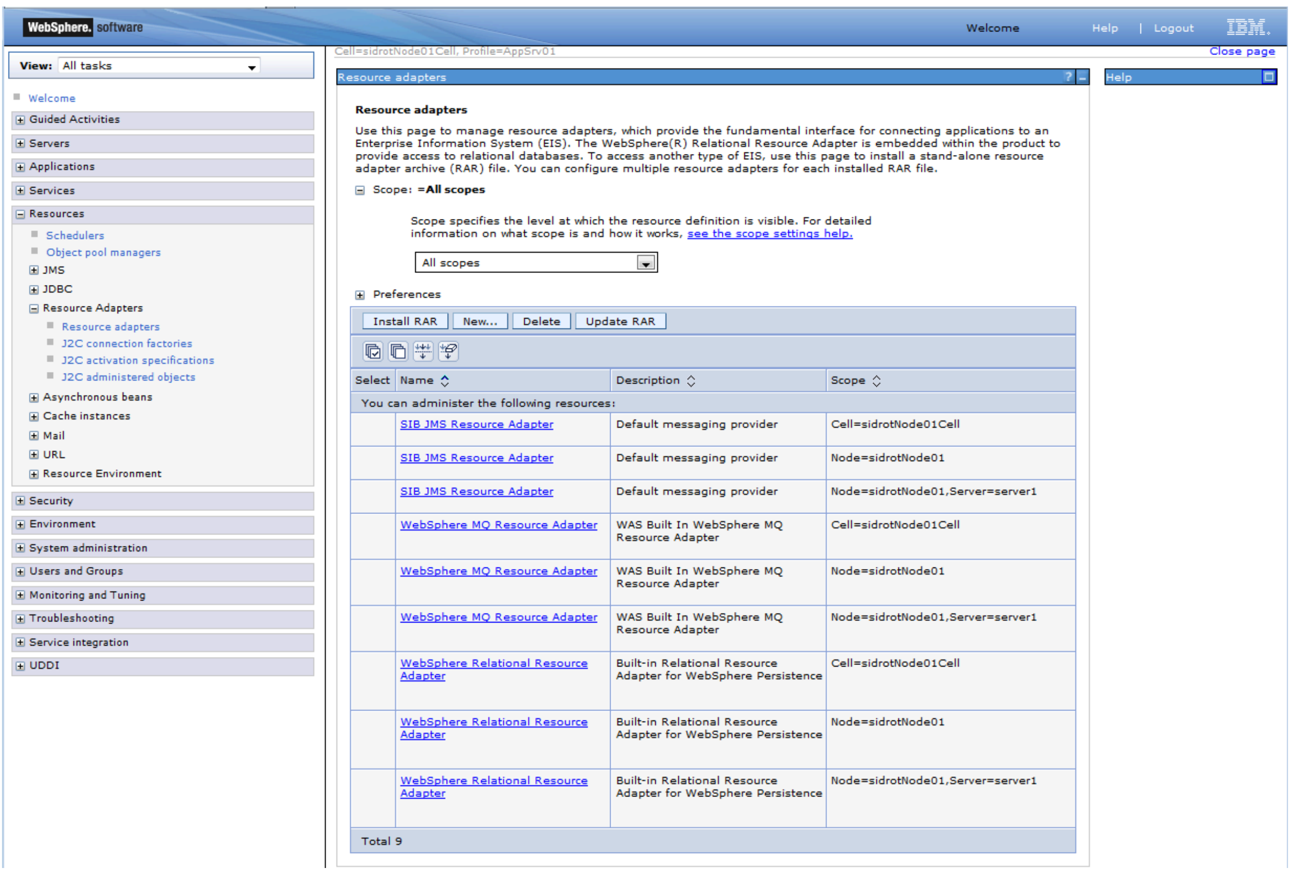
All Rights Reserved

US Government Users Restricted Rights -

Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
IBM is a registered trademark of the IBM Corp.

Go into Resources > Resource Adaptors > Resource adaptors (left-hand menu).

Deployment of the resource adaptor



Resource adapters

Use this page to manage resource adapters, which provide the fundamental interface for connecting applications to an Enterprise Information System (EIS). The WebSphere(R) Relational Resource Adapter is embedded within the product to provide access to relational databases. To access another type of EIS, use this page to install a stand-alone resource adapter archive (RAR) file. You can configure multiple resource adapters for each installed RAR file.

Scope: =**All scopes**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

All scopes

Preferences

You can administer the following resources:

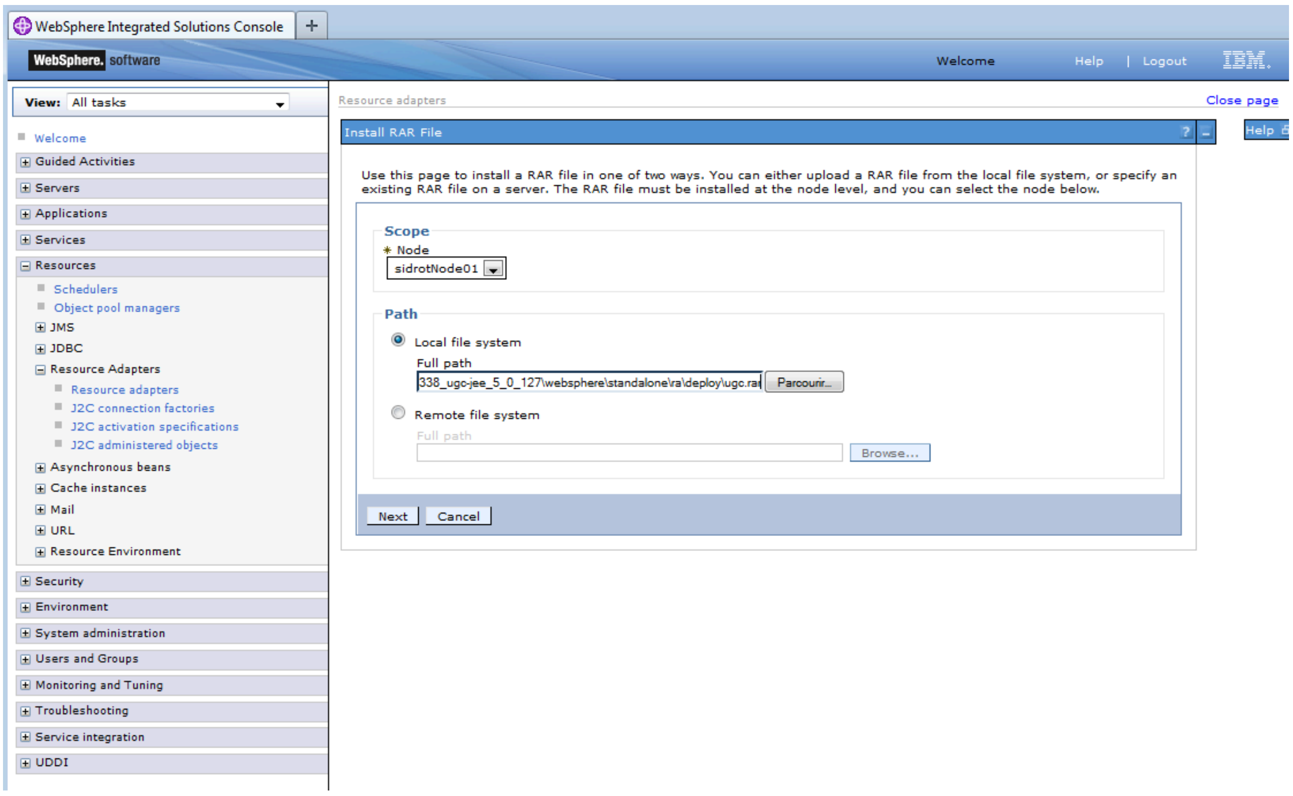
Select	Name	Description	Scope
	SIB JMS Resource Adapter	Default messaging provider	Cell=sidrotNode01Cell
	SIB JMS Resource Adapter	Default messaging provider	Node=sidrotNode01
	SIB JMS Resource Adapter	Default messaging provider	Node=sidrotNode01,Server=server1
	WebSphere MQ Resource Adapter	WAS Built In WebSphere MQ Resource Adapter	Cell=sidrotNode01Cell
	WebSphere MQ Resource Adapter	WAS Built In WebSphere MQ Resource Adapter	Node=sidrotNode01
	WebSphere MQ Resource Adapter	WAS Built In WebSphere MQ Resource Adapter	Node=sidrotNode01,Server=server1
	WebSphere Relational Resource Adapter	Built-in Relational Resource Adapter for WebSphere Persistence	Cell=sidrotNode01Cell
	WebSphere Relational Resource Adapter	Built-in Relational Resource Adapter for WebSphere Persistence	Node=sidrotNode01
	WebSphere Relational Resource Adapter	Built-in Relational Resource Adapter for WebSphere Persistence	Node=sidrotNode01,Server=server1

Total 9

In our case, Preferences / "show built-in resources" is checked (it is for this reason that one sees built-in adaptors).

Click on "Install RAR" and select the "ugc.rar" file located in the sub-directory websphere/standalone/ra/deploy of the deliverable

RAR installation



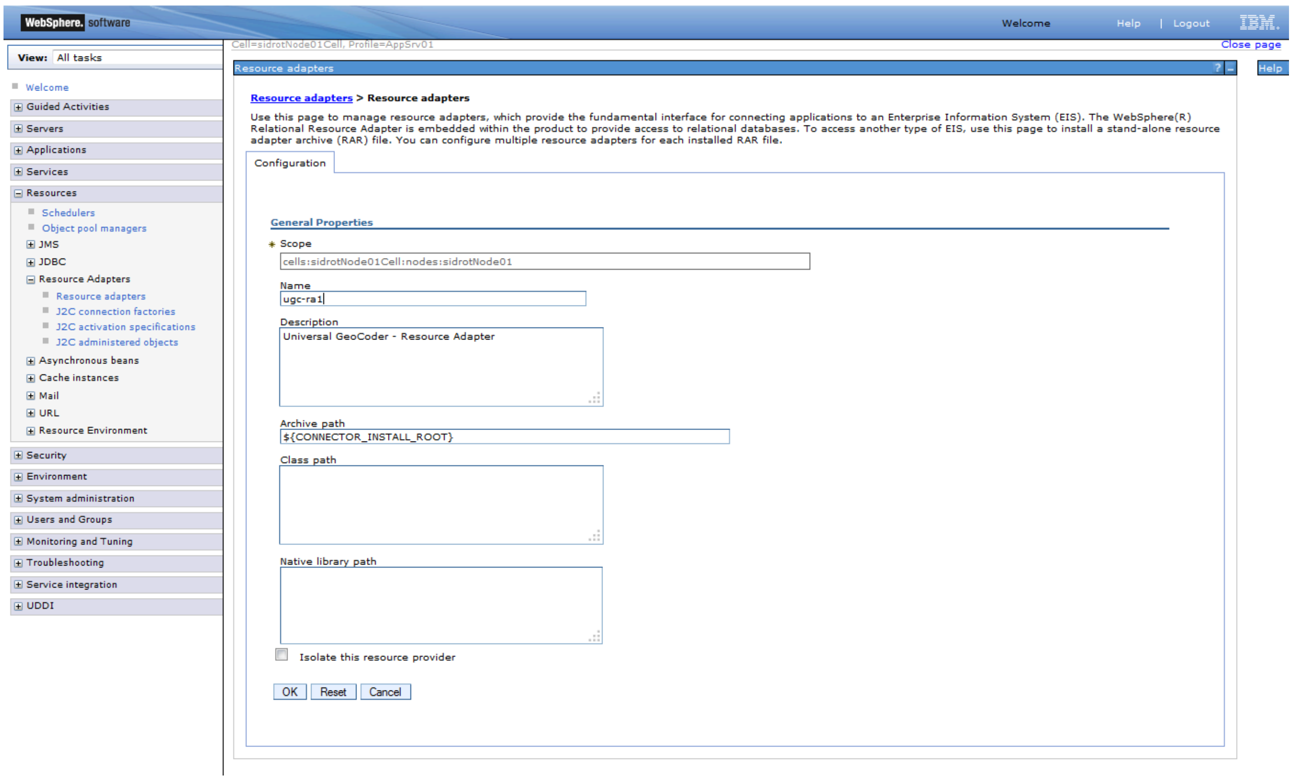
Use this page to install a RAR file in one of two ways. You can either upload a RAR file from the local file system, or specify an existing RAR file on a server. The RAR file must be installed at the node level, and you can select the node below.

Scope
* Node
sidrotNode01

Path
[✓ Local file system]
Full path
338_ugc-jee_5_0_127\websphere\standalone\ra\deploy\ugc.rar
Remote file system
Full path

Then '*next*' ; in the '*name*' field, replace geoconcept/ugc/default with ugc-ra1 and then validate.

RAR installation



Cell=sidrotNode01Cell, Profile=AppSrv01

Resource adapters > Resource adapters

Use this page to manage resource adapters, which provide the fundamental interface for connecting applications to an Enterprise Information System (EIS). The WebSphere(R) Relational Resource Adapter is embedded within the product to provide access to relational databases. To access another type of EIS, use this page to install a stand-alone resource adapter archive (RAR) file. You can configure multiple resource adapters for each installed RAR file.

General Properties

* Scope
cells:sidrotNode01Cell:nodes:sidrotNode01

Name
ugc-ra1

Description
Universal GeoCoder - Resource Adapter

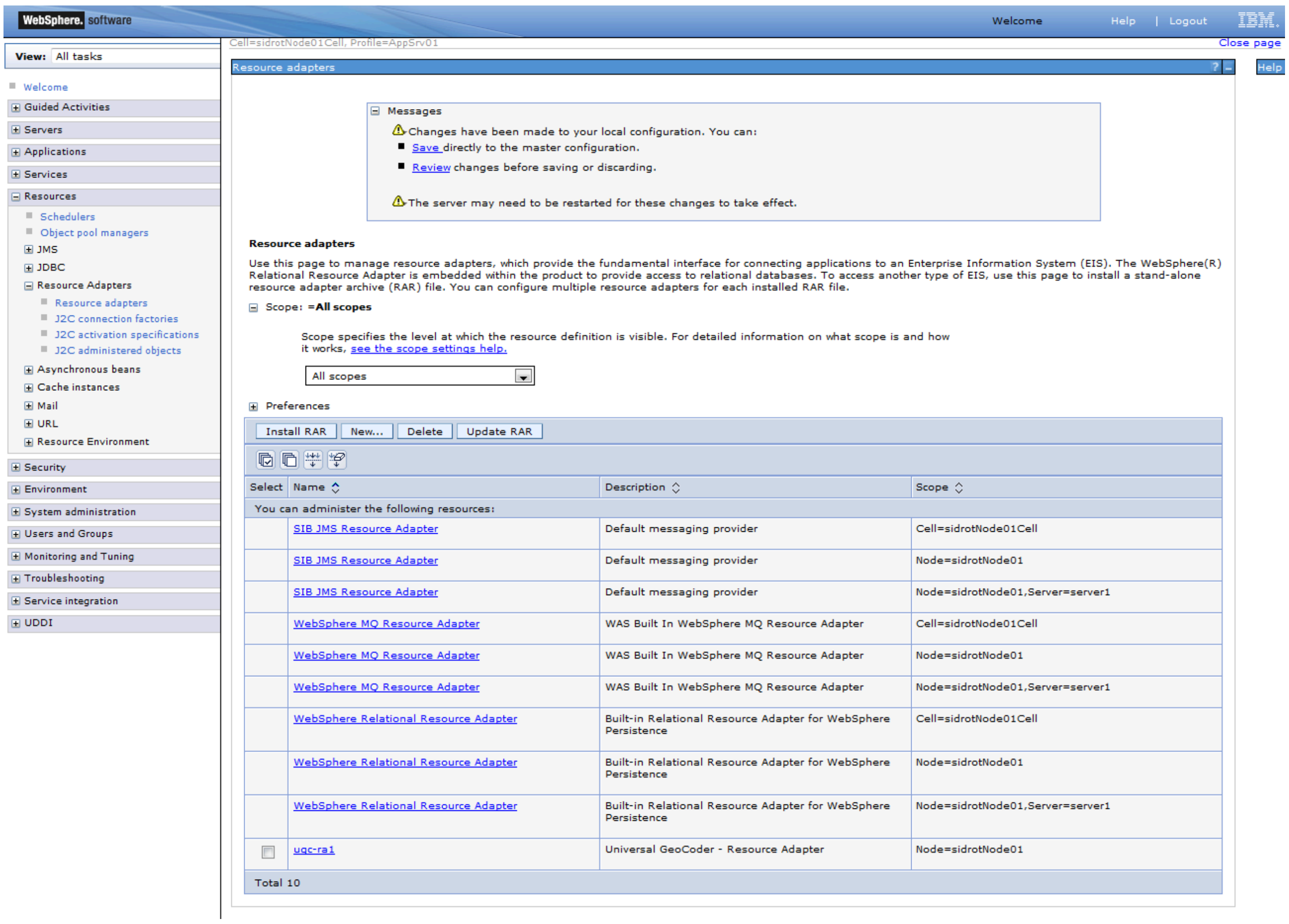
Archive path
${CONNECTOR_INSTALL_ROOT}

Class path

Native library path

Isolate this resource provider

RAR installation



Cell=sidrotNode01Cell, Profile=AppSrv01

Resource adapters

Messages

⚠ Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

Resource adapters

Use this page to manage resource adapters, which provide the fundamental interface for connecting applications to an Enterprise Information System (EIS). The WebSphere(R) Relational Resource Adapter is embedded within the product to provide access to relational databases. To access another type of EIS, use this page to install a stand-alone resource adapter archive (RAR) file. You can configure multiple resource adapters for each installed RAR file.

Scope: =**All scopes**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

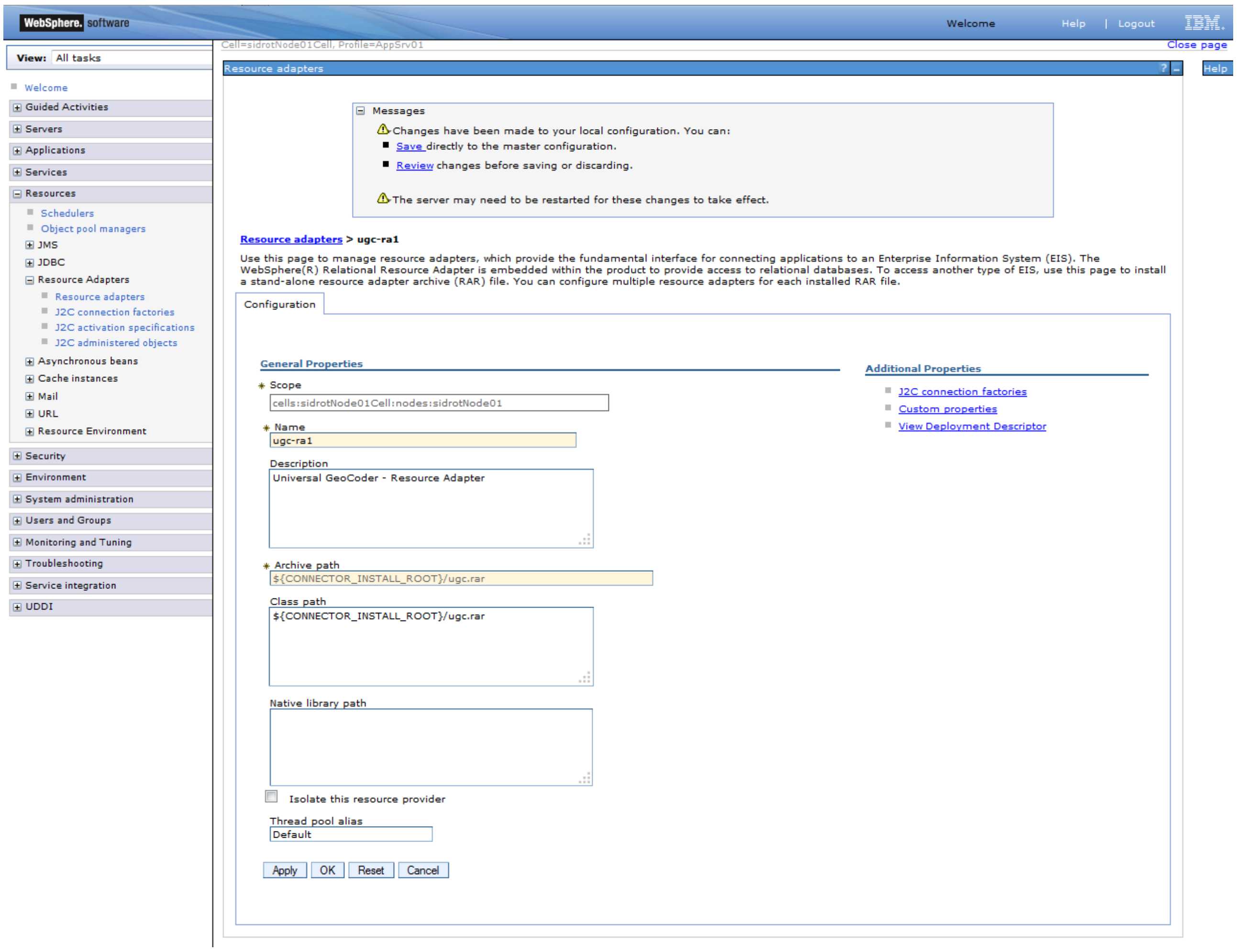
All scopes

Preferences

Select	Name	Description	Scope
You can administer the following resources:			
	SIB JMS Resource Adapter	Default messaging provider	Cell=sidrotNode01Cell
	SIB JMS Resource Adapter	Default messaging provider	Node=sidrotNode01
	SIB JMS Resource Adapter	Default messaging provider	Node=sidrotNode01,Server=server1
	WebSphere MQ Resource Adapter	WAS Built In WebSphere MQ Resource Adapter	Cell=sidrotNode01Cell
	WebSphere MQ Resource Adapter	WAS Built In WebSphere MQ Resource Adapter	Node=sidrotNode01
	WebSphere MQ Resource Adapter	WAS Built In WebSphere MQ Resource Adapter	Node=sidrotNode01,Server=server1
	WebSphere Relational Resource Adapter	Built-in Relational Resource Adapter for WebSphere Persistence	Cell=sidrotNode01Cell
	WebSphere Relational Resource Adapter	Built-in Relational Resource Adapter for WebSphere Persistence	Node=sidrotNode01
	WebSphere Relational Resource Adapter	Built-in Relational Resource Adapter for WebSphere Persistence	Node=sidrotNode01,Server=server1
☐	ugc-ra1	Universal GeoCoder - Resource Adapter	Node=sidrotNode01
Total 10			

ugc-ral appears from then on in the list of resource adaptors deployed, click on it in the list

List of resource adaptors deployed



Cell=sidrotNode01Cell, Profile=AppSrv01

Resource adapters

Messages

⚠ Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

Resource adapters > **ugc-ra1**

Use this page to manage resource adapters, which provide the fundamental interface for connecting applications to an Enterprise Information System (EIS). The WebSphere(R) Relational Resource Adapter is embedded within the product to provide access to relational databases. To access another type of EIS, use this page to install a stand-alone resource adapter archive (RAR) file. You can configure multiple resource adapters for each installed RAR file.

Configuration

General Properties

* Scope: cells:sidrotNode01Cell:nodes:sidrotNode01

* Name: ugc-ra1

Description: Universal GeoCoder - Resource Adapter

* Archive path: ${CONNECTOR_INSTALL_ROOT}/ugc.rar

Class path: ${CONNECTOR_INSTALL_ROOT}/ugc.rar

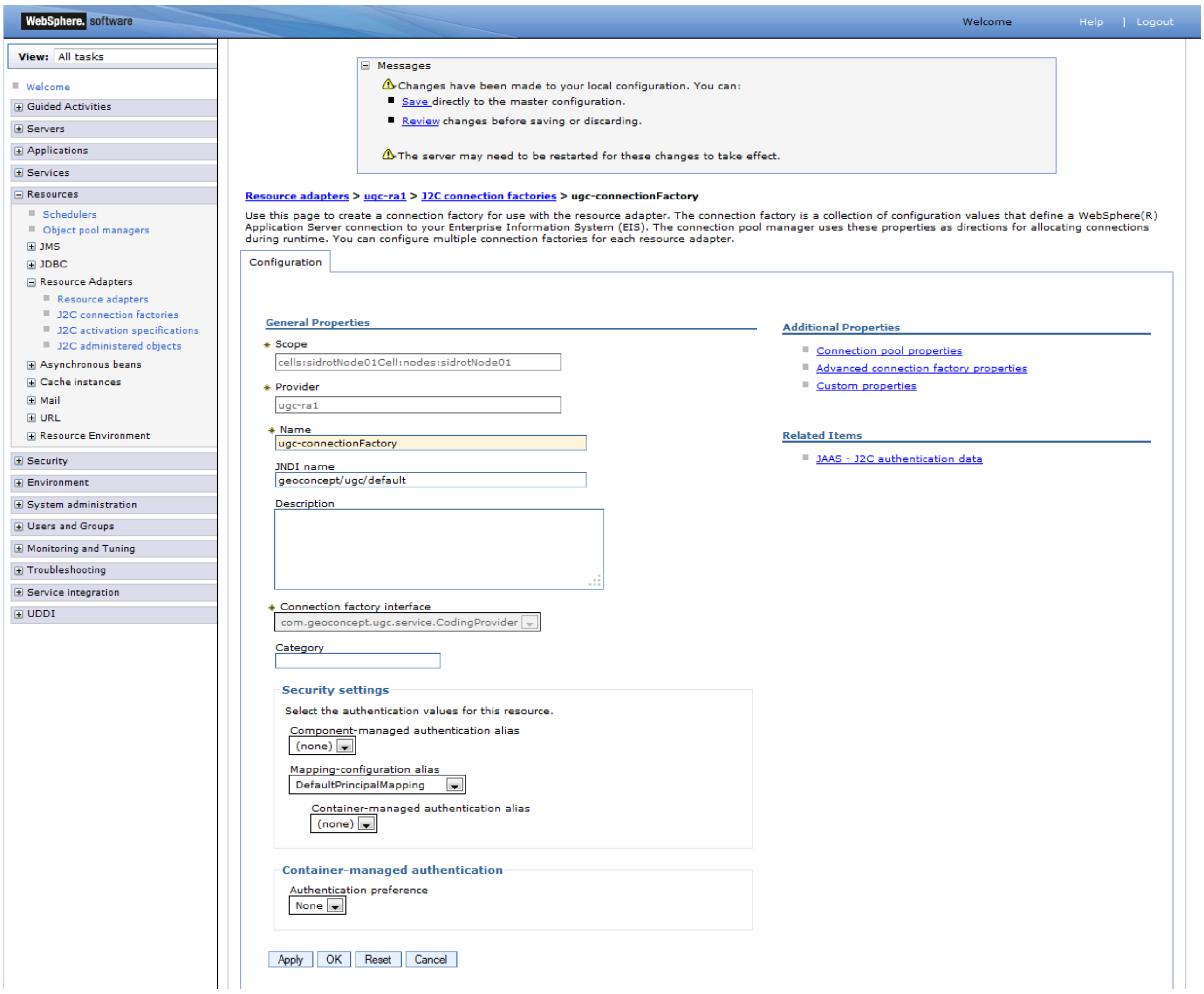
Native library path:

☐ Isolate this resource provider

Thread pool alias: Default

In the column headed Additional properties (on the right) click on "J2C factories connection"

Additional properties column


Messages
Changes have been made to your local configuration. You can:
- Save directly to the master configuration.
- Review changes before saving or discarding.

The server may need to be restarted for these changes to take effect.

Resource adapters > ugc-ra1 > J2C connection factories > ugc-connectionFactory

Use this page to create a connection factory for use with the resource adapter. The connection factory is a collection of configuration values that define a WebSphere(R) Application Server connection to your Enterprise Information System (EIS). The connection pool manager uses these properties as directions for allocating connections during runtime. You can configure multiple connection factories for each resource adapter.

Configuration

General Properties
* Scope: cells:sidrotNode01Cell:nodes:sidrotNode01
* Provider: ugc-ra1
* Name: ugc-connectionFactory
JNDI name: geoconcept/ugc/default
Description:
* Connection factory interface: com.geoconcept.ugc.service.CodingProvider
Category:

Security settings
Select the authentication values for this resource.
Component-managed authentication alias: (none)
Mapping-configuration alias: DefaultPrincipalMapping
Container-managed authentication alias: (none)

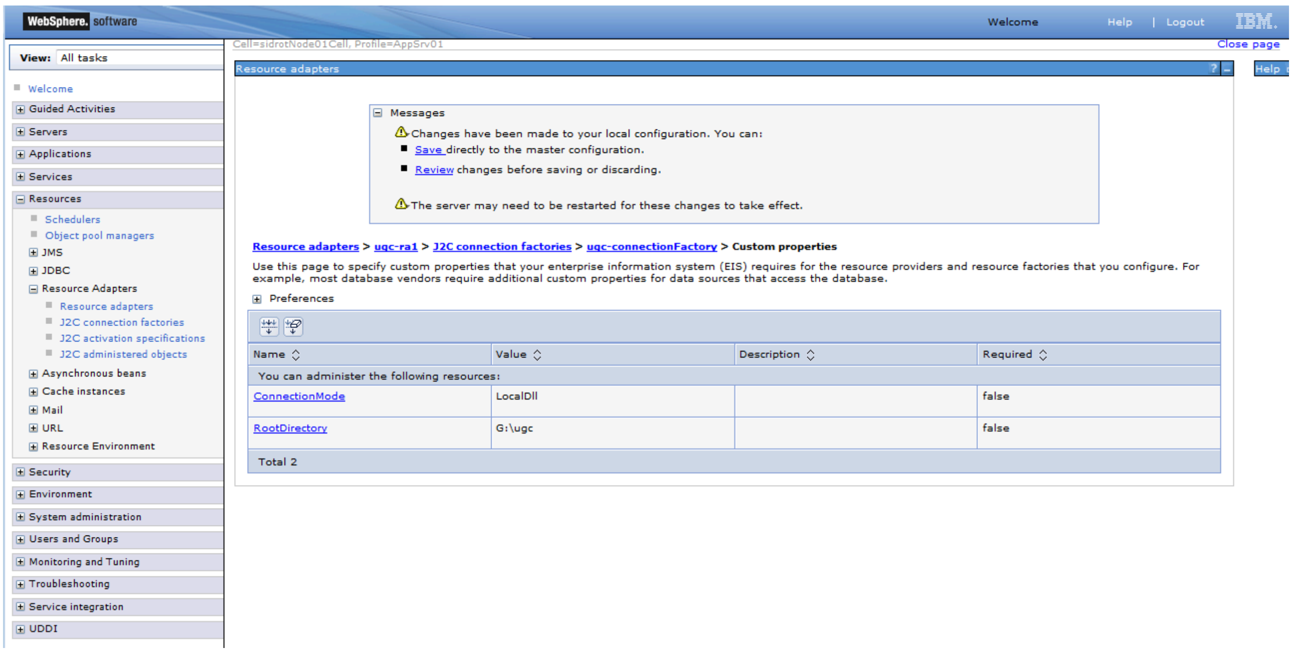
Container-managed authentication
Authentication preference: None

Additional Properties
- Connection pool properties
- Advanced connection factory properties
- Custom properties

Related Items
- JAAS - J2C authentication data

In the Name file, enter ugc-connectionFactory, and in the JNDI name field, enter geoconcept/ugc/default, then “Apply”. In the Additional properties column (on the right), click on Custom properties.

Custom properties column



Cell=sidrotNode01Cell, Profile=AppSrv01

Resource adapters

Messages

⚠ Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

Resource adapters > ugc-ra1 > J2C connection factories > ugc-connectionFactory > Custom properties

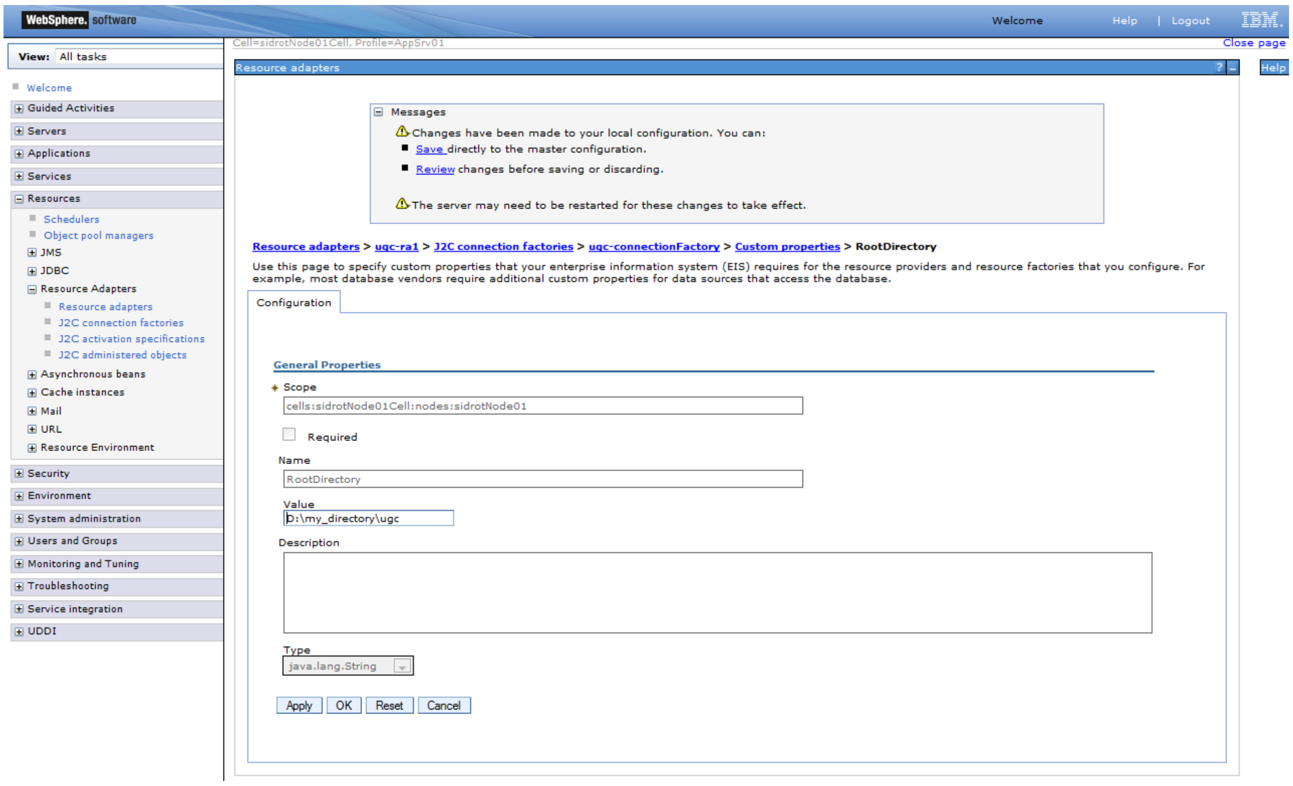
Use this page to specify custom properties that your enterprise information system (EIS) requires for the resource providers and resource factories that you configure. For example, most database vendors require additional custom properties for data sources that access the database.

Preferences

Name	Value	Description	Required
You can administer the following resources:			
ConnectionMode	LocalDll		false
RootDirectory	G:\ugc		false
Total 2			

Edit the value of RootDirectory to give it the directory in which the external ugc files are, and then validate.

RootDirectory value



Cell=sidrotNode01Cell, Profile=AppSrv01

Resource adapters

Messages

⚠ Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

⚠ The server may need to be restarted for these changes to take effect.

Resource adapters > ugc-ra1 > J2C connection factories > ugc-connectionFactory > Custom properties > RootDirectory

Use this page to specify custom properties that your enterprise information system (EIS) requires for the resource providers and resource factories that you configure. For example, most database vendors require additional custom properties for data sources that access the database.

Configuration

General Properties

* Scope: cells:sidrotNode01Cell:nodes:sidrotNode01

Required: ☐

Name: RootDirectory

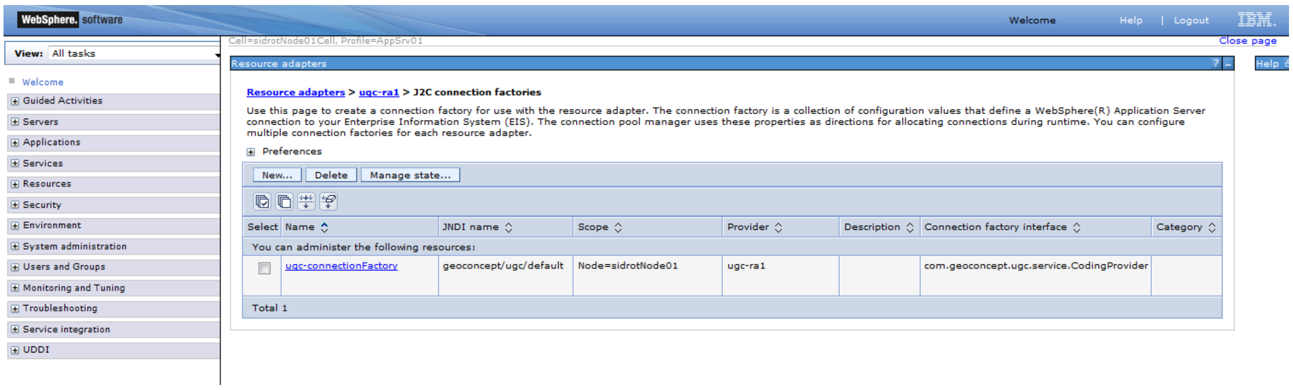
Value: D:\my_directory\ugc

Description:

Type: java.lang.String

Save (Save directly to the master configuration)

Saving or backing up



Cell=sidrotNode01Cell, Profile=AppSrv01

Resource adapters > ugc-ra1 > J2C connection factories

Use this page to create a connection factory for use with the resource adapter. The connection factory is a collection of configuration values that define a WebSphere(R) Application Server connection to your Enterprise Information System (EIS). The connection pool manager uses these properties as directions for allocating connections during runtime. You can configure multiple connection factories for each resource adapter.

Select	Name	JNDI name	Scope	Provider	Description	Connection factory interface	Category
You can administer the following resources:							
☐	ugc-connectionFactory	geoconcept/ugc/default	Node=sidrotNode01	ugc-ra1		com.geoconcept.ugc.service.CodingProvider	
Total 1							

The factory connection is now deployed.

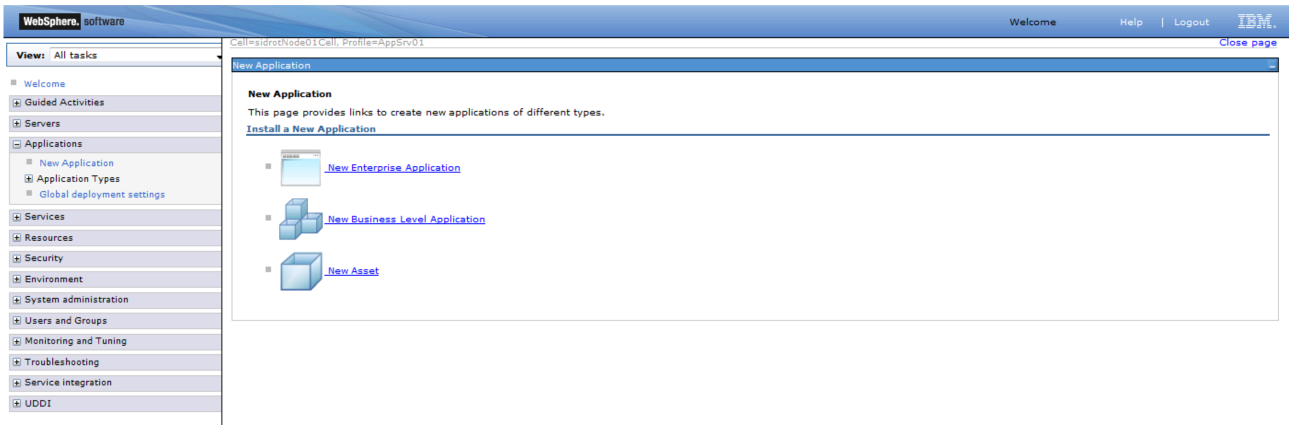
1. Deployment of an optional module: we are now going to deploy an optional module supplied as standard, in this case the webapp ugc-admin (a war) that will consume the services of the resource adaptor.

The principle is the same as for any module (supplied or developed specifically) consuming the services of the resource adaptor.

In addition ugc-admin enables validation of the deployment of the resource adaptor.

In the menu on the left-hand side, go into Applications > New Application

New application



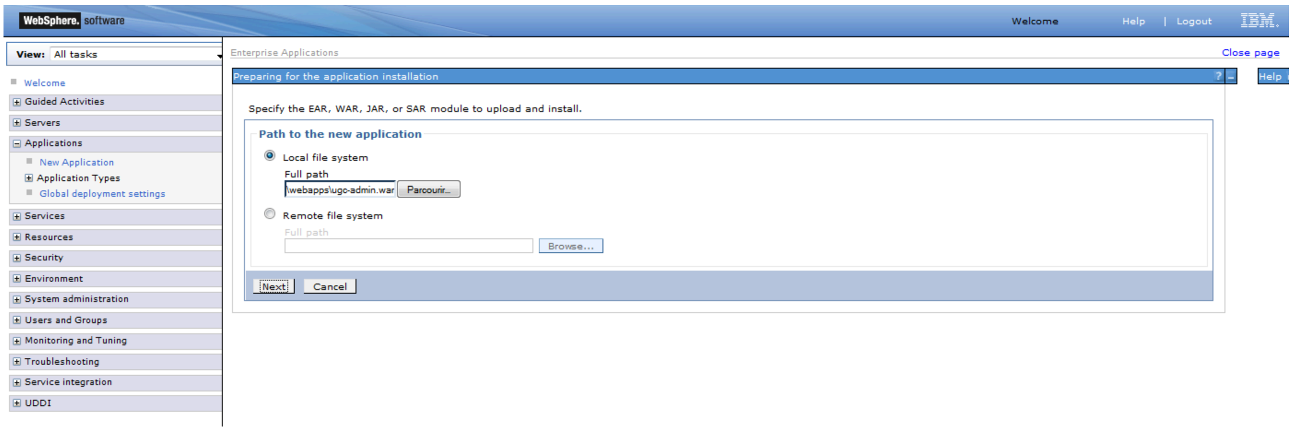
Cell=sidrotNode01Cell, Profile=AppSrv01

New Application

This page provides links to create new applications of different types.

Choose New Enterprise Application.

New Enterprise Application



Preparing for the application installation

Specify the EAR, WAR, JAR, or SAR module to upload and install.

Path to the new application

(●) Local file system

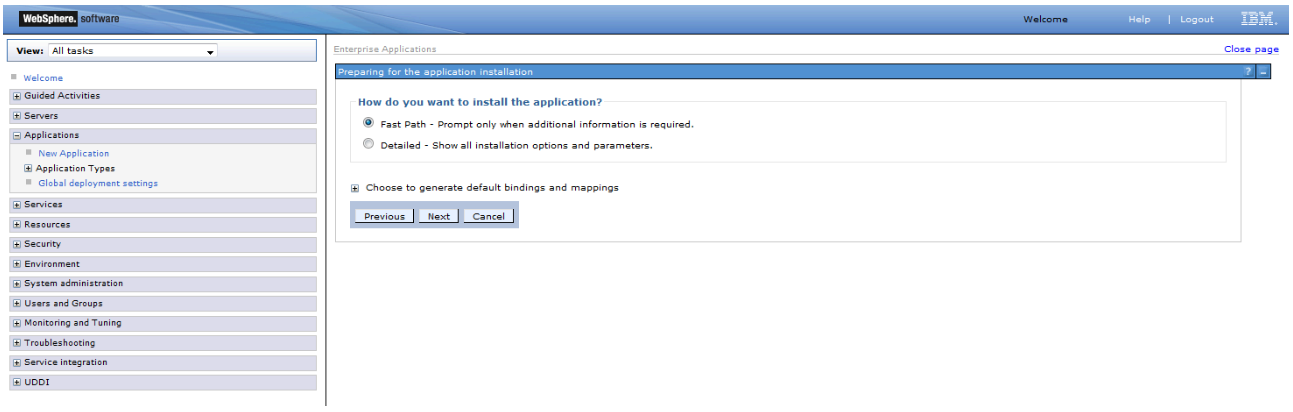
Full path: \webapps\ugc-admin.war

( ) Remote file system

Full path:

Click on browse and select the ugc-admin.war file located in the websphere/standalone/webapps sub-directory of the deliverable and then click on Next.

The ugc-admin.war file



Preparing for the application installation

How do you want to install the application?

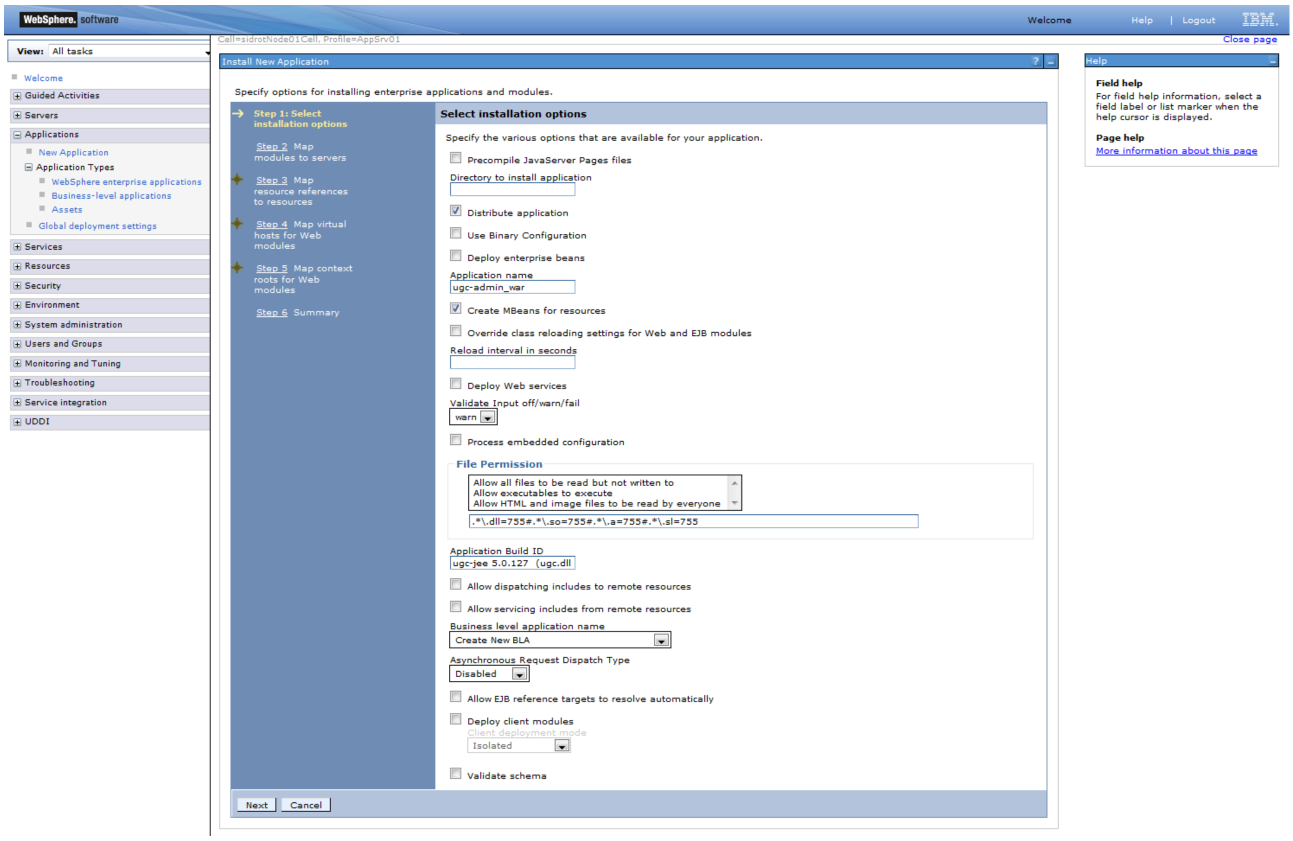
(●) Fast Path - Prompt only when additional information is required.

( ) Detailed - Show all installation options and parameters.

⊞ Choose to generate default bindings and mappings

Leave fastpath by default and click on Next.

The fastpath parameter



Cell=sidrotNode01Cell, Profile=AppSrv01

Install New Application

Specify options for installing enterprise applications and modules.

Select installation options

Specify the various options that are available for your application.

- [ ] Precompile JavaServer Pages files
- Directory to install application:
- [✓] Distribute application
- [ ] Use Binary Configuration
- [ ] Deploy enterprise beans
- Application name: ugc-admin_war
- [✓] Create MBeans for resources
- [ ] Override class reloading settings for Web and EJB modules
- Reload interval in seconds:
- [ ] Deploy Web services
- Validate Input off/warn/fail: warn
- [ ] Process embedded configuration

File Permission

Allow all files to be read but not written to
Allow executables to execute
Allow HTML and image files to be read by everyone

.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755

- Application Build ID: ugc-jee 5.0.127 (ugc.dll
- [ ] Allow dispatching includes to remote resources
- [ ] Allow servicing includes from remote resources
- Business level application name: Create New BLA
- Asynchronous Request Dispatch Type: Disabled
- [ ] Allow EJB reference targets to resolve automatically
- [ ] Deploy client modules
- Client deployment mode: Isolated
- [ ] Validate schema

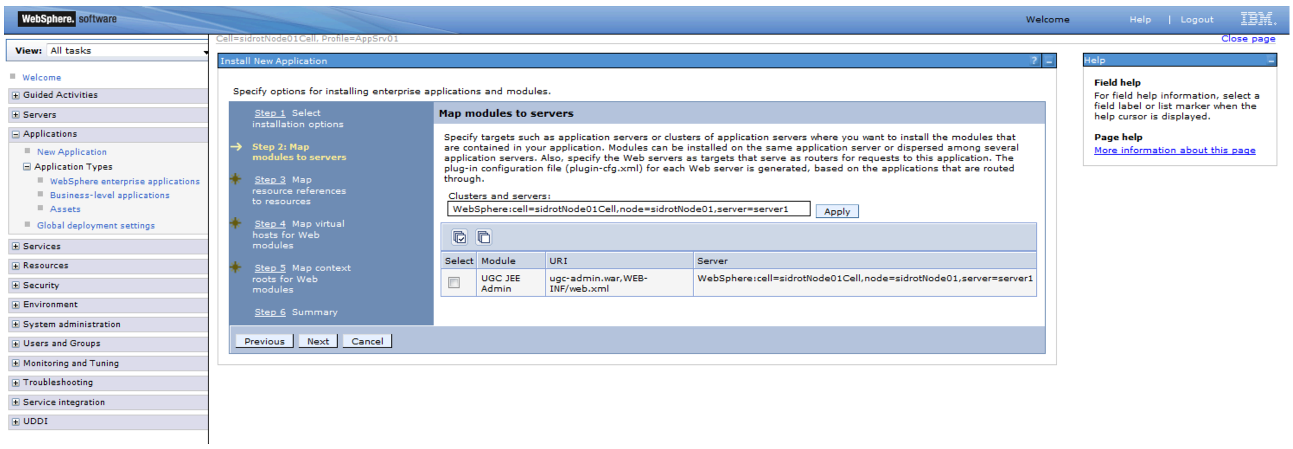
Help

Field help
For field help information, select a field label or list marker when the help cursor is displayed.

Page help
More information about this page

Step 1: leave everything on Default, then click on Next.

Step 1



Cell=sidrotNode01Cell, Profile=AppSrv01

Install New Application

Specify options for installing enterprise applications and modules.

Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed through.

Clusters and servers:
WebSphere:cell=sidrotNode01Cell,node=sidrotNode01,server=server1

Select	Module	URI	Server
☐	UGC JEE Admin	ugc-admin.war,WEB-INF/web.xml	WebSphere:cell=sidrotNode01Cell,node=sidrotNode01,server=server1

Help

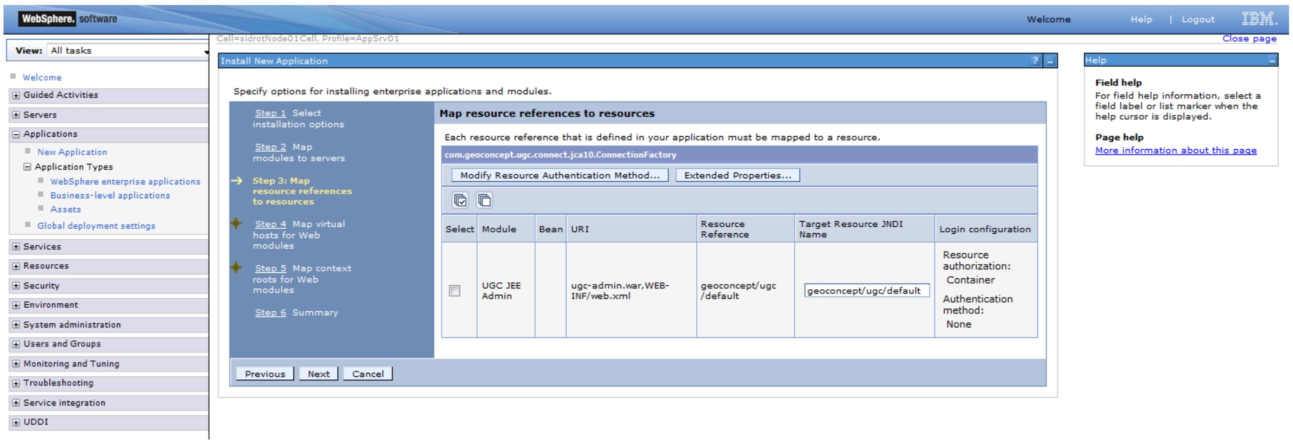
Field help
For field help information, select a field label or list marker when the help cursor is displayed.

Page help
More information about this page

Step 2: leave everything on Default, then click on Next.

Step 3: enter geoconcept/ugc/default in Target Resource JNDI Name.

Step 3



Cell=sidrotNode01Cell, Profile=AppSrv01

Specify options for installing enterprise applications and modules.

Map resource references to resources

Each resource reference that is defined in your application must be mapped to a resource.

com.geoconcept.ugc.connect.jca10.ConnectionFactory

Select	Module	Bean	URI	Resource Reference	Target Resource JNDI Name	Login configuration
☐	UGC JEE Admin		ugc-admin.war,WEB-INF/web.xml	geoconcept/ugc/default	geoconcept/ugc/default	Resource authorization: Container Authentication method: None

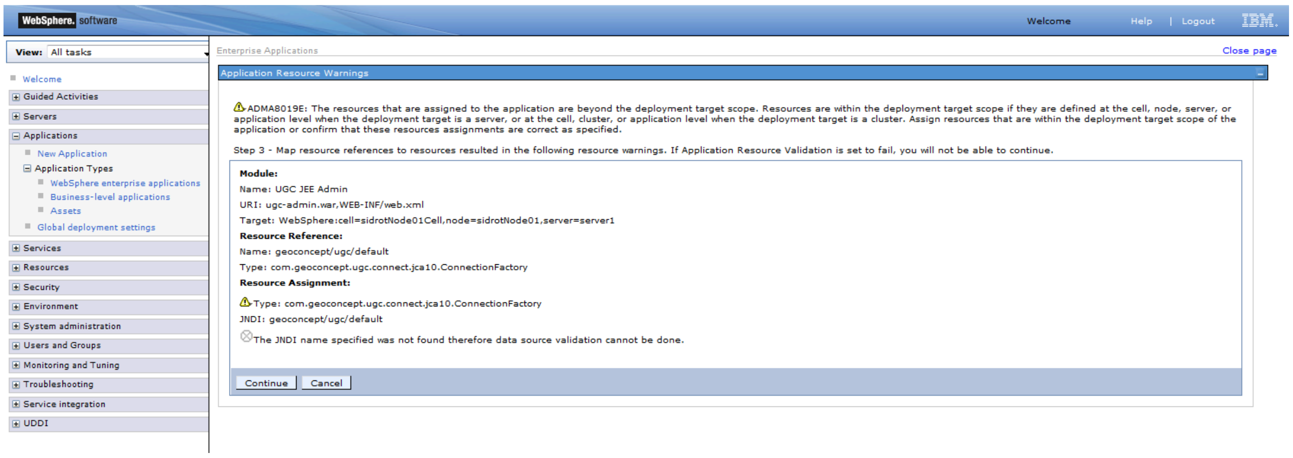
Help

Field help
For field help information, select a field label or list marker when the help cursor is displayed.

Page help
More information about this page

A warning is indicated, then click on '*continue*'.

Step 3



Enterprise Applications

Application Resource Warnings

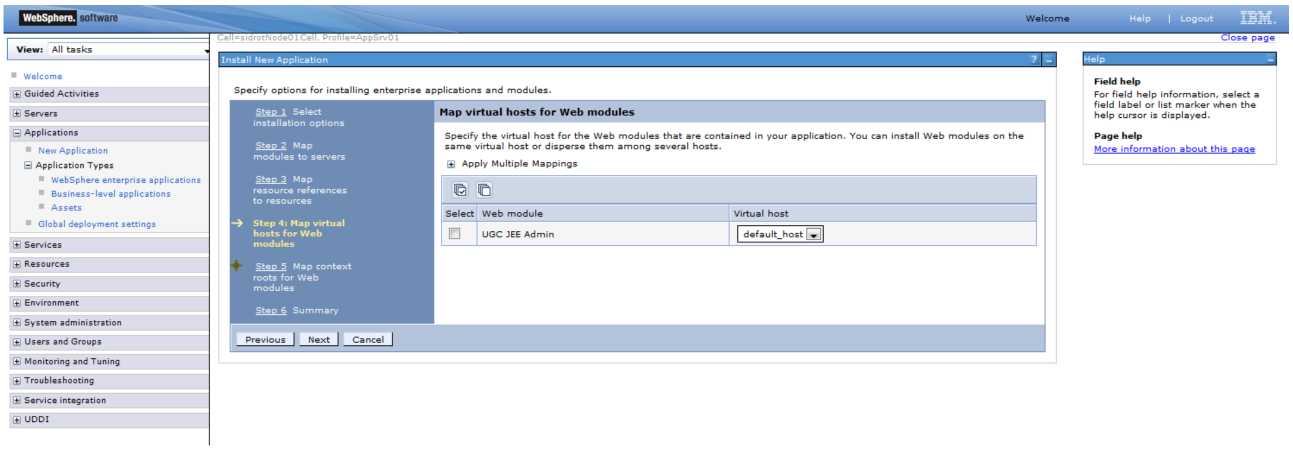
ADMA8019E: The resources that are assigned to the application are beyond the deployment target scope. Resources are within the deployment target scope if they are defined at the cell, node, server, or application level when the deployment target is a server, or at the cell, cluster, or application level when the deployment target is a cluster. Assign resources that are within the deployment target scope of the application or confirm that these resources assignments are correct as specified.

Step 3 - Map resource references to resources resulted in the following resource warnings. If Application Resource Validation is set to fail, you will not be able to continue.

Module:
Name: UGC JEE Admin
URI: ugc-admin.war,WEB-INF/web.xml
Target: WebSphere:cell=sidrotNode01Cell,node=sidrotNode01,server=server1
Resource Reference:
Name: geoconcept/ugc/default
Type: com.geoconcept.ugc.connect.jca10.ConnectionFactory
Resource Assignment:
Type: com.geoconcept.ugc.connect.jca10.ConnectionFactory
JNDI: geoconcept/ugc/default
The JNDI name specified was not found therefore data source validation cannot be done.

Step 4: leave everything on Default, and then click on Next.

Step 4



Cell=sidrotNode01Cell, Profile=AppSrv01

Specify options for installing enterprise applications and modules.

Map virtual hosts for Web modules

Specify the virtual host for the Web modules that are contained in your application. You can install Web modules on the same virtual host or disperse them among several hosts.

Select	Web module	Virtual host
☐	UGC JEE Admin	default_host

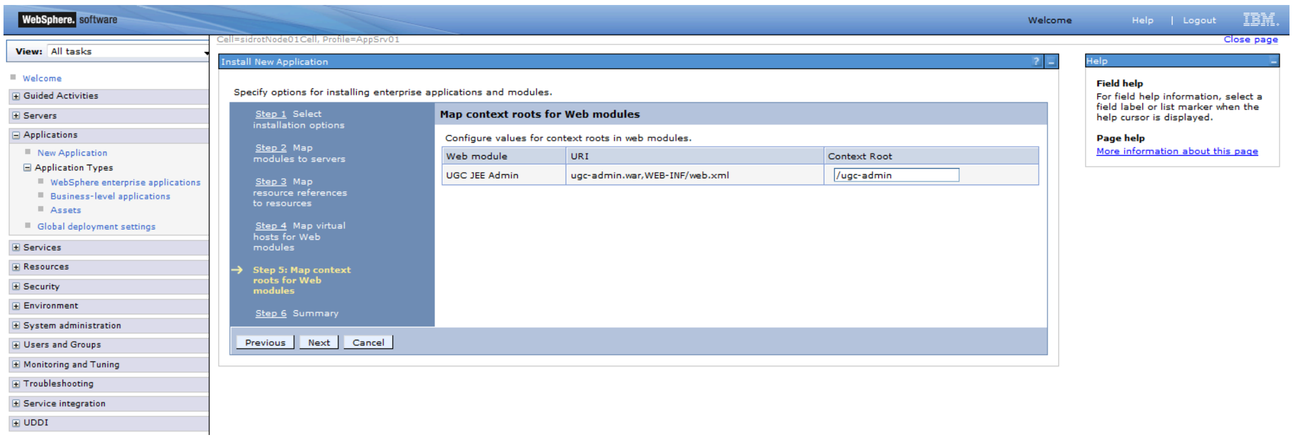
Help

Field help
For field help information, select a field label or list marker when the help cursor is displayed.

Page help
More information about this page

Step 5: enter the root / ugc-admin context (it proposes / by default)

Step 5



Cell=sidrotNode01Cell, Profile=AppSrv01

Install New Application

Specify options for installing enterprise applications and modules.

Map context roots for Web modules

Configure values for context roots in web modules.

Web module	URI	Context Root
UGC JEE Admin	ugc-admin.war,WEB-INF/web.xml	/ugc-admin

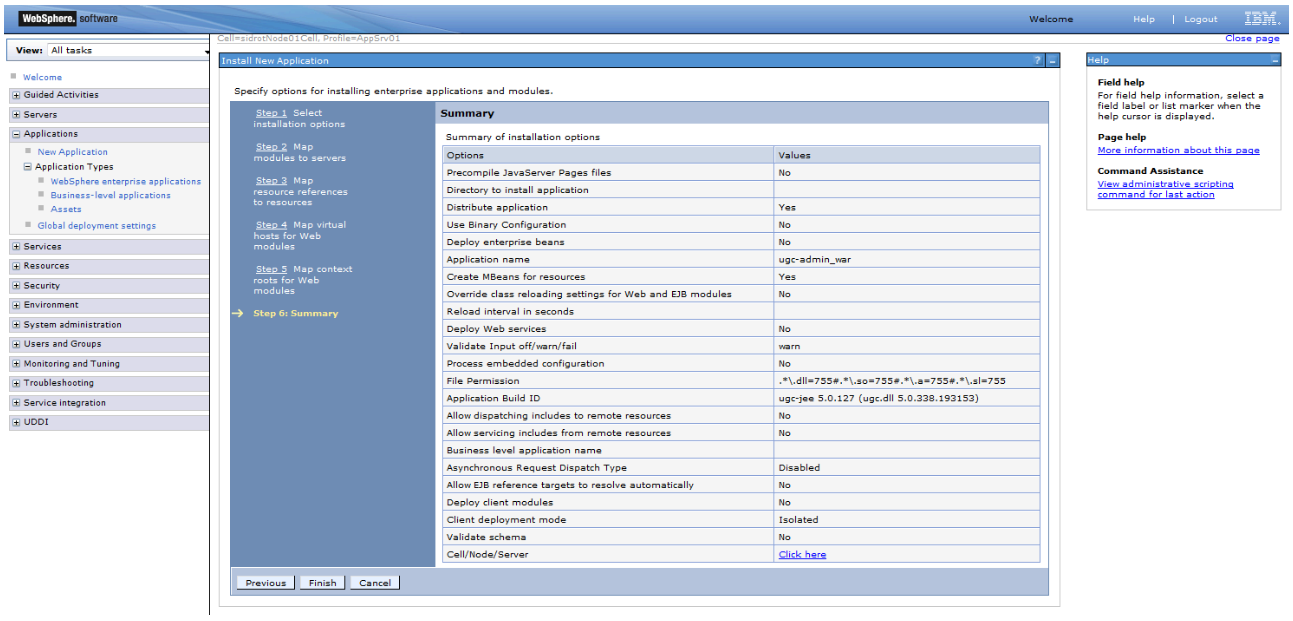
Help

Field help
For field help information, select a field label or list marker when the help cursor is displayed.

Page help
More information about this page

Step 6: summary (validate with Finish)

Step 6



Cell=sidrotNode01Cell, Profile=AppSrv01

Install New Application

Specify options for installing enterprise applications and modules.

Summary

Summary of installation options

Options	Values
Precompile JavaServer Pages files	No
Directory to install application	
Distribute application	Yes
Use Binary Configuration	No
Deploy enterprise beans	No
Application name	ugc-admin_war
Create MBeans for resources	Yes
Override class reloading settings for Web and EJB modules	No
Reload interval in seconds	
Deploy Web services	No
Validate Input off/warn/fail	warn
Process embedded configuration	No
File Permission	.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755
Application Build ID	ugc-jee 5.0.127 (ugc.dll 5.0.338.193153)
Allow dispatching includes to remote resources	No
Allow servicing includes from remote resources	No
Business level application name	
Asynchronous Request Dispatch Type	Disabled
Allow EJB reference targets to resolve automatically	No
Deploy client modules	No
Client deployment mode	Isolated
Validate schema	No
Cell/Node/Server	Click here

Help

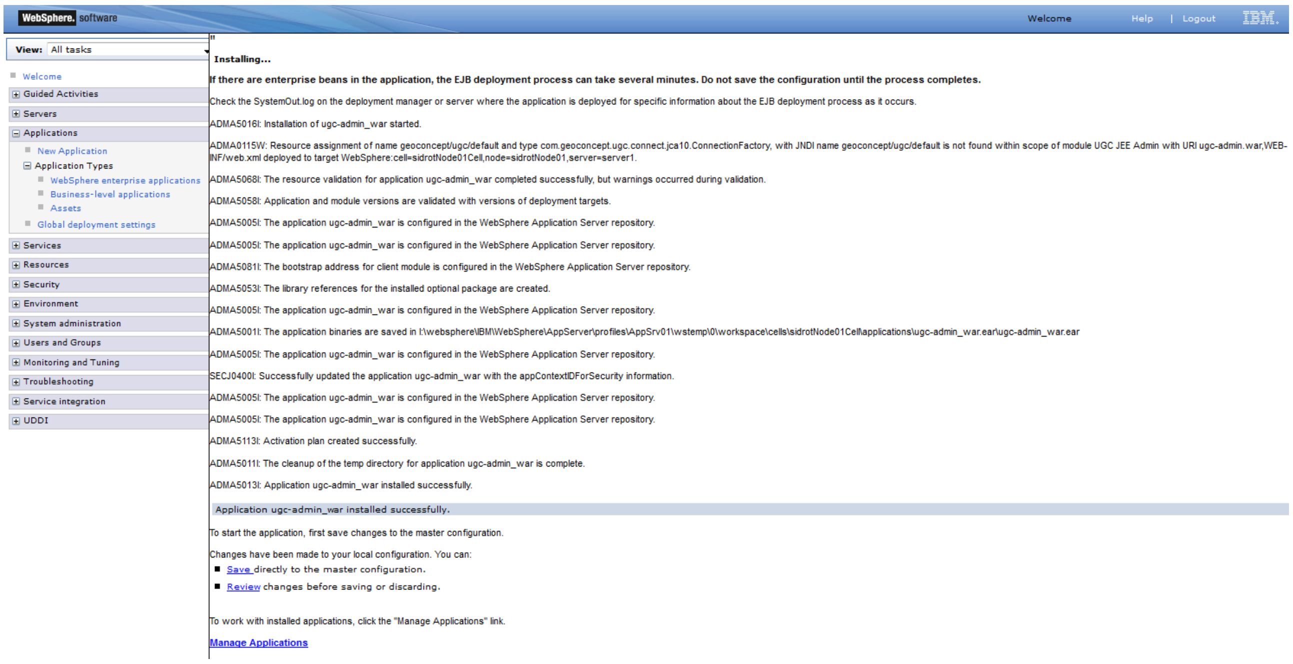
Field help
For field help information, select a field label or list marker when the help cursor is displayed.

Page help
More information about this page

Command Assistance
View administrative scripting command for last action

The console presents the status of the deployment and indicates Application ugc-admin_war installed successfully

Deployment status



Installing...

If there are enterprise beans in the application, the EJB deployment process can take several minutes. Do not save the configuration until the process completes.

```
Check the SystemOut.log on the deployment manager or server where the application is deployed for specific information about the EJB deployment process as it occurs.

ADMA5016I: Installation of ugc-admin_war started.

ADMA0115W: Resource assignment of name geoconcept/ugc/default and type com.geoconcept.ugc.connect.jca10.ConnectionFactory, with JNDI name geoconcept/ugc/default is not found within scope of module UGC JEE Admin with URI ugc-admin.war,WEB-
INF/web.xml deployed to target WebSphere:cell=sidrotNode01Cell,node=sidrotNode01,server=server1.

ADMA5068I: The resource validation for application ugc-admin_war completed successfully, but warnings occurred during validation.

ADMA5058I: Application and module versions are validated with versions of deployment targets.

ADMA5005I: The application ugc-admin_war is configured in the WebSphere Application Server repository.

ADMA5005I: The application ugc-admin_war is configured in the WebSphere Application Server repository.

ADMA5081I: The bootstrap address for client module is configured in the WebSphere Application Server repository.

ADMA5053I: The library references for the installed optional package are created.

ADMA5005I: The application ugc-admin_war is configured in the WebSphere Application Server repository.

ADMA5001I: The application binaries are saved in I:\websphere\IBM\WebSphere\AppServer\profiles\AppSrv01\wstemp\0\workspace\cells\sidrotNode01Cell\applications\ugc-admin_war.ear\ugc-admin_war.ear

ADMA5005I: The application ugc-admin_war is configured in the WebSphere Application Server repository.

SECJ0400I: Successfully updated the application ugc-admin_war with the appContextIDForSecurity information.

ADMA5005I: The application ugc-admin_war is configured in the WebSphere Application Server repository.

ADMA5005I: The application ugc-admin_war is configured in the WebSphere Application Server repository.

ADMA5113I: Activation plan created successfully.

ADMA5011I: The cleanup of the temp directory for application ugc-admin_war is complete.

ADMA5013I: Application ugc-admin_war installed successfully.
```

Application ugc-admin_war installed successfully.

To start the application, first save changes to the master configuration.

Changes have been made to your local configuration. You can:

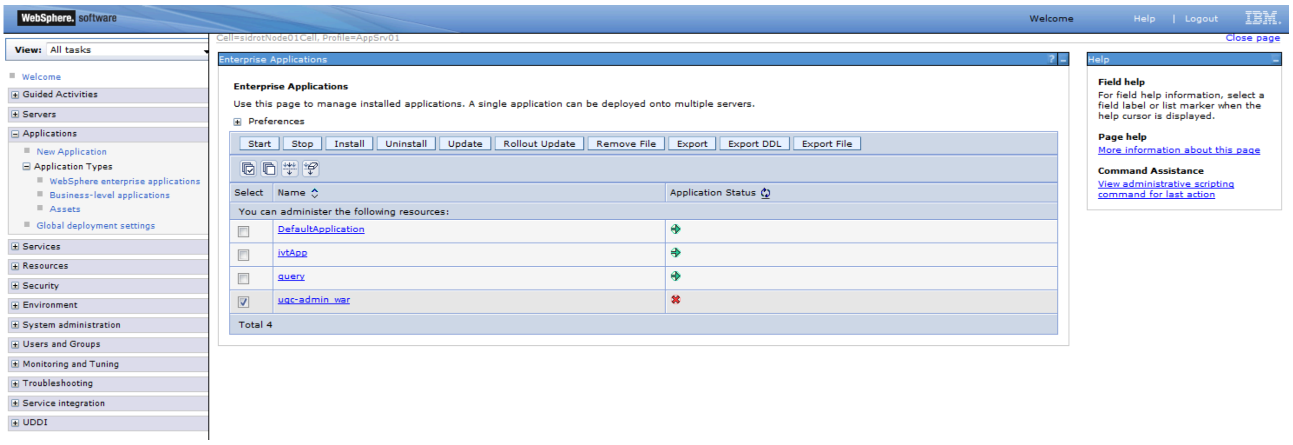
- Save directly to the master configuration.
- Review changes before saving or discarding.

To work with installed applications, click the "Manage Applications" link.

Manage Applications

Start-up the webapp (check ugc-admin.war and click on Start)

Deployment status



Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

Select	Name	Application Status
You can administer the following resources:		
☐	DefaultApplication	➜
☐	ivtApp	➜
☐	query	➜
☑	ugc-admin_war	✖
Total 4		

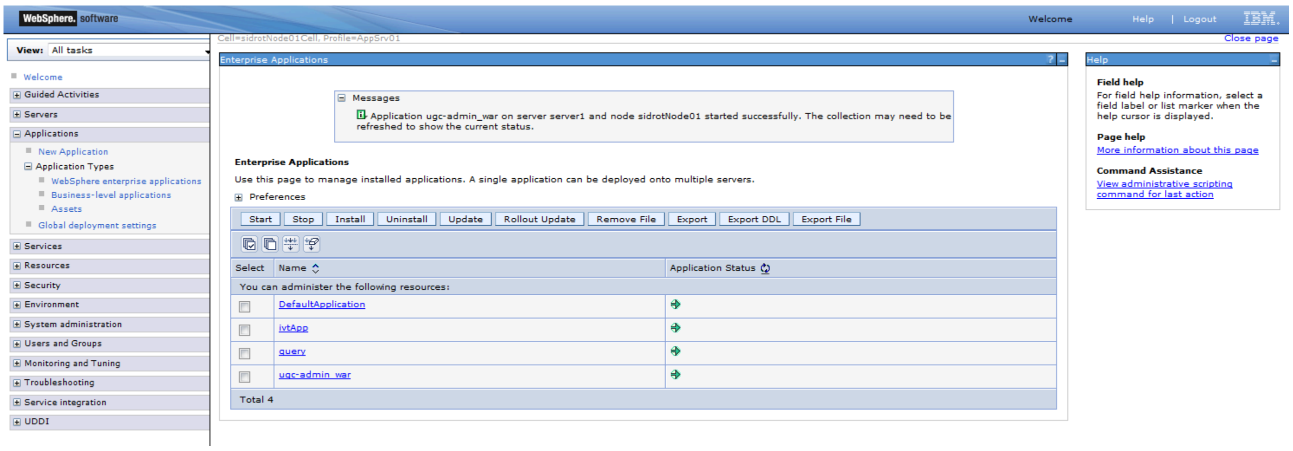
Field help
For field help information, select a field label or list marker when the help cursor is displayed.

Page help
More information about this page

Command Assistance
View administrative scripting command for last action

The console indicates "ugc-admin_war on server server1 and node sidrotNode01 started successfully. The collection may need to be refreshed to show the current status."

Console



Cell=sidrotNode01Cell, Profile=AppSrv01

Enterprise Applications

Messages

Application ugc-admin_war on server server1 and node sidrotNode01 started successfully. The collection may need to be refreshed to show the current status.

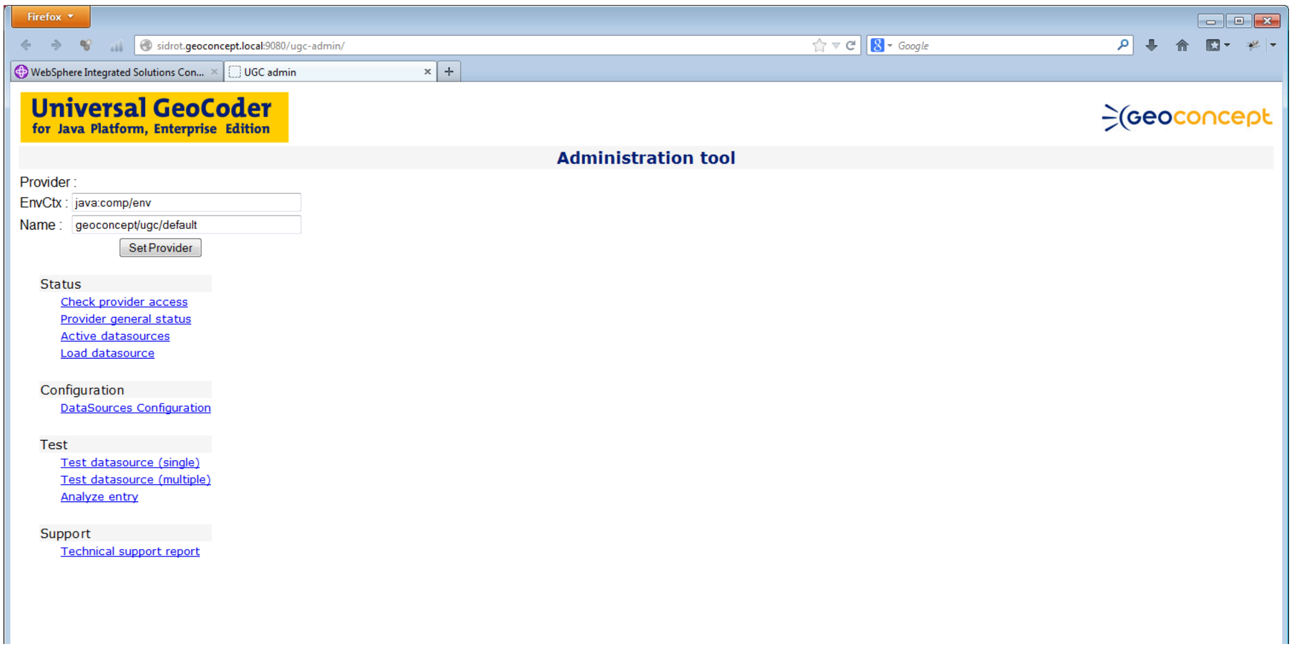
Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

Select	Name	Application Status
You can administer the following resources:		
☐	DefaultApplication	➔
☐	ivtApp	➔
☐	query	➔
☐	ugc-admin_war	➔
Total 4		

1. Verification of the deployment with ugc-admin

Access to the ugc-admin webapp previously deployed (by default, port 9080):



sidrot.geoconcept.local:9080/ugc-admin/

Administration tool

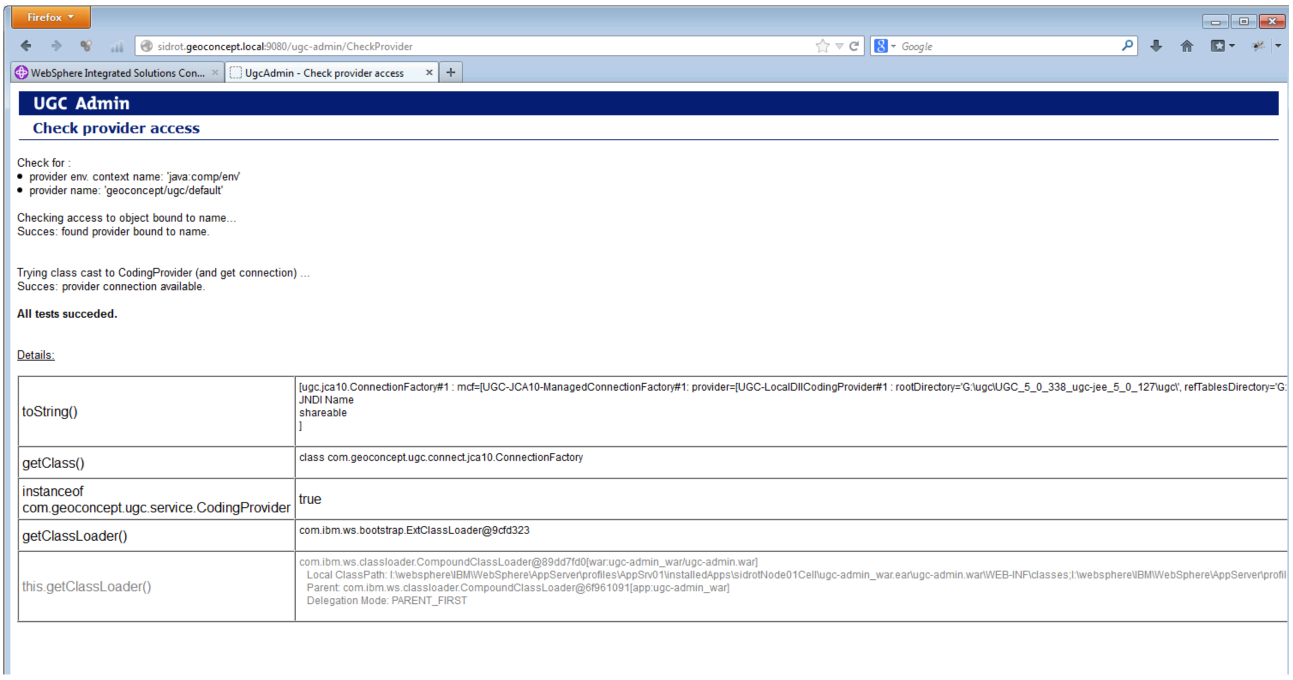
Provider :

EnvCtx : java:comp/env

Name : geoconcept/ugc/default

Click on Check provider access

Check provider access



sidrot.geoconcept.local:9080/ugc-admin/CheckProvider

UGC Admin

Check provider access

Check for :
- provider env. context name: 'java:comp/env'
- provider name: 'geoconcept/ugc/default'

Checking access to object bound to name...
Succes: found provider bound to name.

Trying class cast to CodingProvider (and get connection) ...
Succes: provider connection available.

All tests succeded.

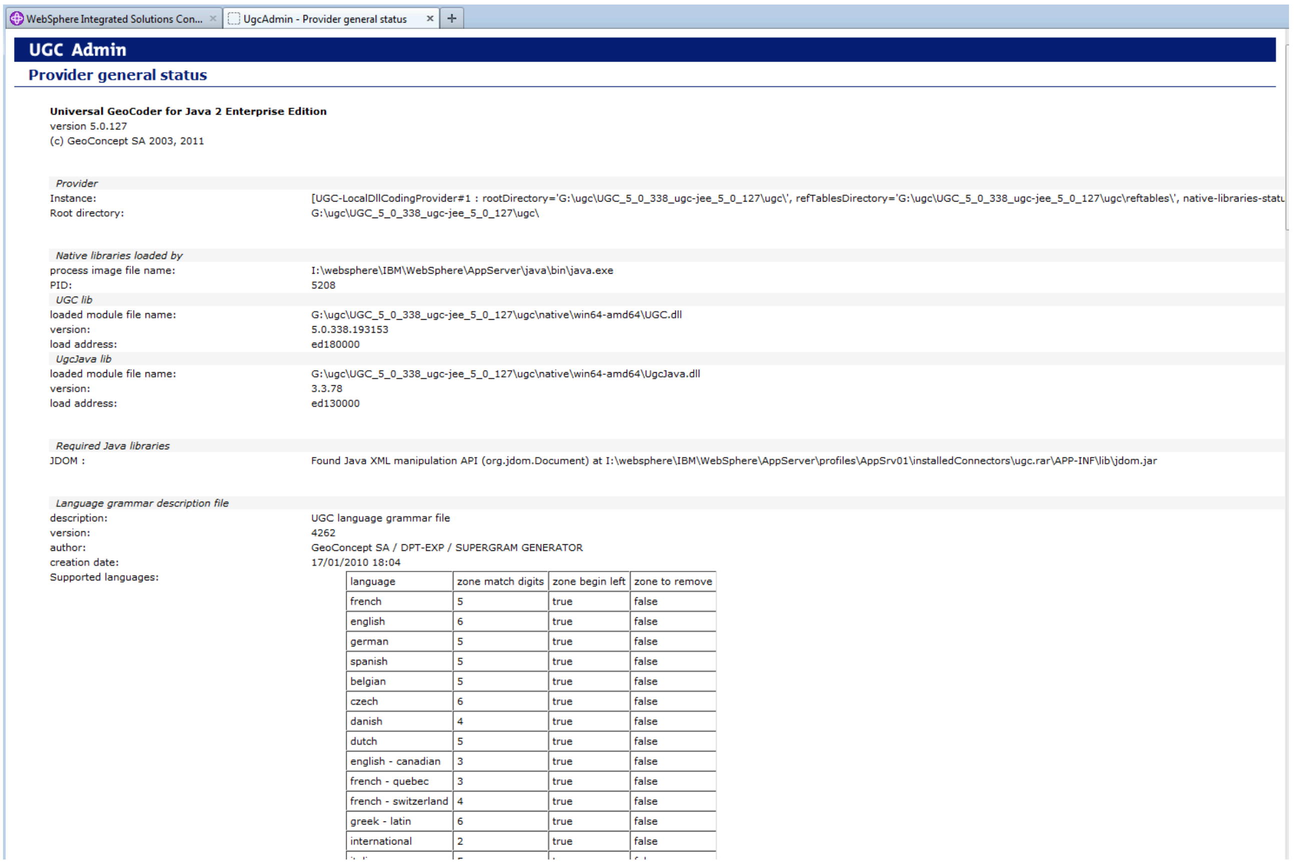
Details:

toString()	[ugc.jca10.ConnectionFactory#1 : mcf=[UGC-JCA10-ManagedConnectionFactory#1: provider=[UGC-LocalDllCodingProvider#1 : rootDirectory='G:\ugc\UGC_5_0_338_ugc-jee_5_0_127\ugc\', refTablesDirectory='G: JNDI Name shareable]
getClass()	class com.geoconcept.ugc.connect.jca10.ConnectionFactory
instanceof com.geoconcept.ugc.service.CodingProvider	true
getClassLoader()	com.ibm.ws.bootstrap.ExtClassLoader@9cfd323
this.getClassLoader()	com.ibm.ws.classloader.CompoundClassLoader@89dd7fd0[war:ugc-admin_war/ugc-admin.war] Local ClassPath: I:\websphere\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\sidrotNode01Cell\ugc-admin_war.ear\ugc-admin.war\WEB-INF\classes;I:\websphere\IBM\WebSphere\AppServer\profil Parent: com.ibm.ws.classloader.CompoundClassLoader@6f961091[app:ugc-admin_war] Delegation Mode: PARENT_FIRST

The page must indicate All tests succeeded.

Return to the home page of ugc-admin and click on Provider general status

General Provider Status



UGC Admin

Provider general status

Universal GeoCoder for Java 2 Enterprise Edition
version 5.0.127
(c) GeoConcept SA 2003, 2011

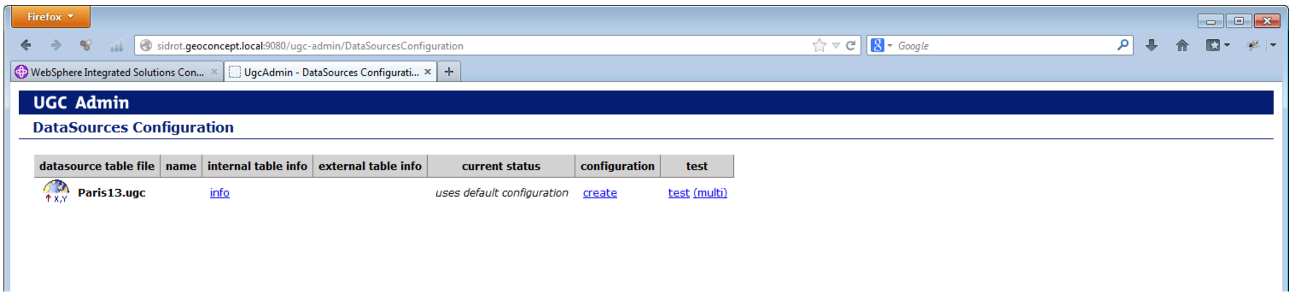
Provider	
Instance:	[UGC-LocalDllCodingProvider#1 : rootDirectory='G:\ugc\UGC_5_0_338_ugc-jee_5_0_127\ugc\', refTablesDirectory='G:\ugc\UGC_5_0_338_ugc-jee_5_0_127\ugc\reftables\', native-libraries-statu
Root directory:	G:\ugc\UGC_5_0_338_ugc-jee_5_0_127\ugc\
Native libraries loaded by	
process image file name:	I:\websphere\IBM\WebSphere\AppServer\java\bin\java.exe
PID:	5208
UGC lib	
loaded module file name:	G:\ugc\UGC_5_0_338_ugc-jee_5_0_127\ugc\native\win64-amd64\UGC.dll
version:	5.0.338.193153
load address:	ed180000
UgcJava lib	
loaded module file name:	G:\ugc\UGC_5_0_338_ugc-jee_5_0_127\ugc\native\win64-amd64\UgcJava.dll
version:	3.3.78
load address:	ed130000
Required Java libraries	
JDOM :	Found Java XML manipulation API (org.jdom.Document) at I:\websphere\IBM\WebSphere\AppServer\profiles\AppSrv01\installedConnectors\ugc.rar\APP-INF\lib\jdom.jar
Language grammar description file	
description:	UGC language grammar file
version:	4262
author:	GeoConcept SA / DPT-EXP / SUPERGRAM GENERATOR
creation date:	17/01/2010 18:04
Supported languages:	

language	zone match digits	zone begin left	zone to remove
french	5	true	false
english	6	true	false
german	5	true	false
spanish	5	true	false
belgian	5	true	false
czech	6	true	false
danish	4	true	false
dutch	5	true	false
english - canadian	3	true	false
french - quebec	3	true	false
french - switzerland	4	true	false
greek - latin	6	true	false
international	2	true	false

The page shows versioning and runtime information for the provider instance and its environment

Return to the ugc-admin home page and click on DataSources Configuration

Datasources configuration



sidrot.geoconcept.local:9080/ugc-admin/DataSourcesConfiguration

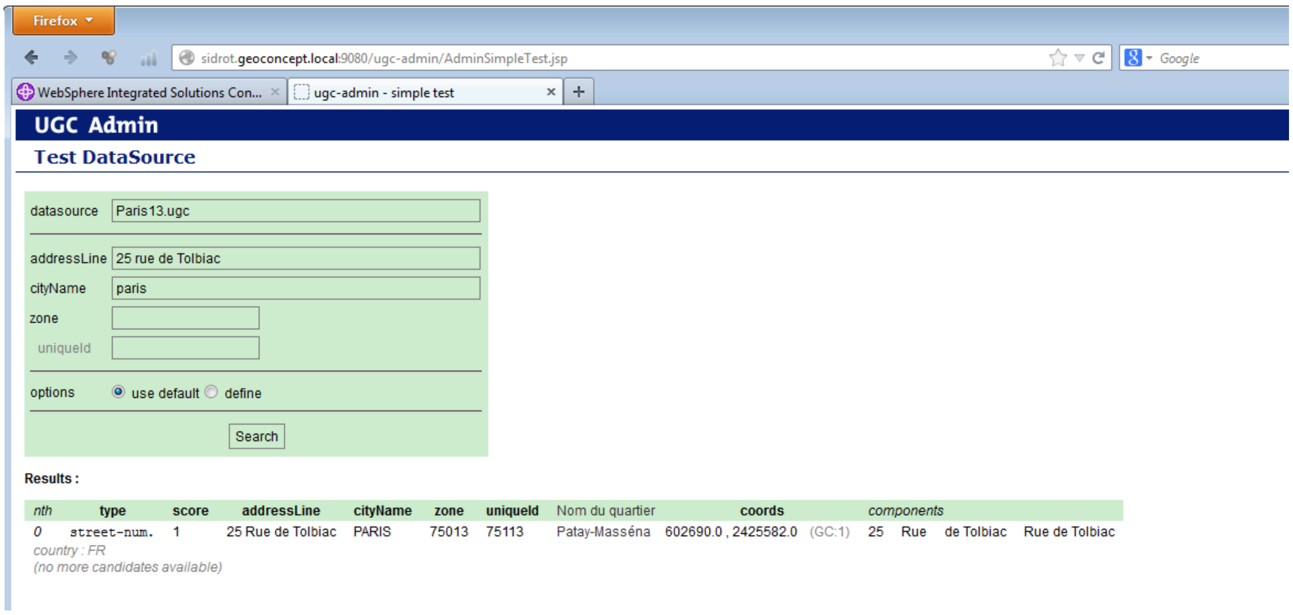
UGC Admin

DataSources Configuration

datasource table file	name	internal table info	external table info	current status	configuration	test
Paris13.ugc		info		*uses default configuration*	create	test (multi)

The page lists all available reference tables. Click on test for one of the reference tables listed. Click on Search

UGC test



sidrot.geoconcept.local:9080/ugc-admin/AdminSimpleTest.jsp

UGC Admin

Test DataSource

datasource	Paris13.ugc
addressLine	25 rue de Tolbiac
cityName	paris
zone	
uniqueId	
options	[✓ use default] define

Results :

nth	type	score	addressLine	cityName	zone	uniqueId	Nom du quartier	coords		components			
0	street-num.	1	25 Rue de Tolbiac	PARIS	75013	75113	Patay-Masséna	602690.0 , 2425582.0	(GC:1)	25	Rue	de Tolbiac	Rue de Tolbiac

country : FR
(no more candidates available)

The page should display a correct result, and this validates the deployment

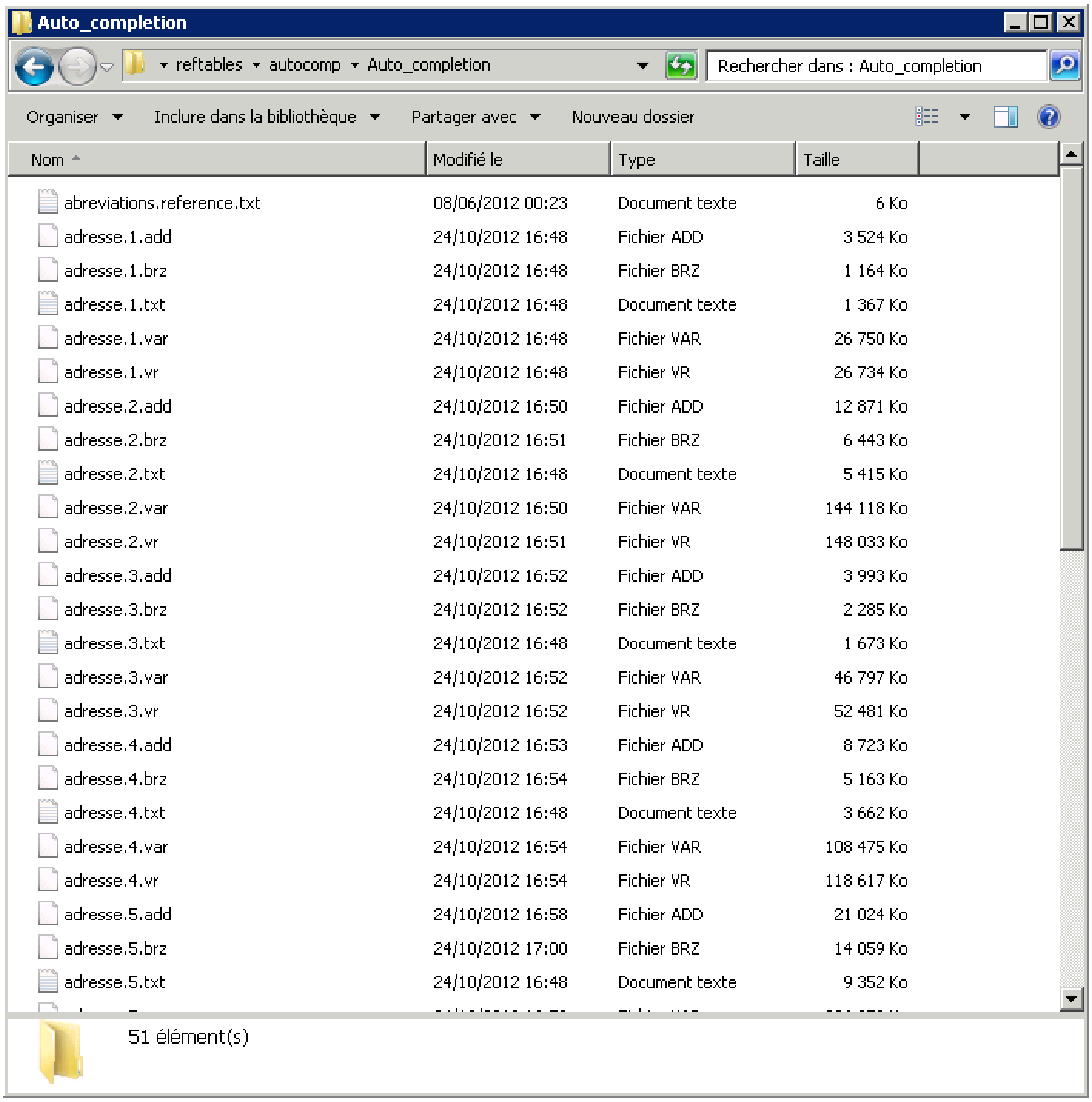
Files for the autocomplete feature

If the auto-complete web service supplied by Geoconcept Web is used, it will be necessary to dispose of and install the relevant reference files (simply copy the files into a directory).

Follow the steps described below to ensure correct functioning of the autocomplete web service:

- In the `reftables` directory present in UGC, create an autocomp directory,
- Copy the directory supplied by Geoconcept into this directory : the directory supplied by Geoconcept contains the reference files. You should have the following architecture :

Preview of the directory containing the reference files for the autocomplete feature


Auto_completion
reftables ▸ autocomp ▸ Auto_completion
Rechercher dans : Auto_completion
Organiser
Inclure dans la bibliothèque
Partager avec
Nouveau dossier

Nom	Modifié le	Type	Taille
abreviations.reference.txt	08/06/2012 00:23	Document texte	6 Ko
adresse.1.add	24/10/2012 16:48	Fichier ADD	3 524 Ko
adresse.1.brz	24/10/2012 16:48	Fichier BRZ	1 164 Ko
adresse.1.txt	24/10/2012 16:48	Document texte	1 367 Ko
adresse.1.var	24/10/2012 16:48	Fichier VAR	26 750 Ko
adresse.1.vr	24/10/2012 16:48	Fichier VR	26 734 Ko
adresse.2.add	24/10/2012 16:50	Fichier ADD	12 871 Ko
adresse.2.brz	24/10/2012 16:51	Fichier BRZ	6 443 Ko
adresse.2.txt	24/10/2012 16:48	Document texte	5 415 Ko
adresse.2.var	24/10/2012 16:50	Fichier VAR	144 118 Ko
adresse.2.vr	24/10/2012 16:51	Fichier VR	148 033 Ko
adresse.3.add	24/10/2012 16:52	Fichier ADD	3 993 Ko
adresse.3.brz	24/10/2012 16:52	Fichier BRZ	2 285 Ko
adresse.3.txt	24/10/2012 16:48	Document texte	1 673 Ko
adresse.3.var	24/10/2012 16:52	Fichier VAR	46 797 Ko
adresse.3.vr	24/10/2012 16:52	Fichier VR	52 481 Ko
adresse.4.add	24/10/2012 16:53	Fichier ADD	8 723 Ko
adresse.4.brz	24/10/2012 16:54	Fichier BRZ	5 163 Ko
adresse.4.txt	24/10/2012 16:48	Document texte	3 662 Ko
adresse.4.var	24/10/2012 16:54	Fichier VAR	108 475 Ko
adresse.4.vr	24/10/2012 16:54	Fichier VR	118 617 Ko
adresse.5.add	24/10/2012 16:58	Fichier ADD	21 024 Ko
adresse.5.brz	24/10/2012 17:00	Fichier BRZ	14 059 Ko
adresse.5.txt	24/10/2012 16:48	Document texte	9 352 Ko

51 élément(s)

The name of the directory in which the reference files are found is important. This name will be supplied in the parameters to be saved in the Geoconcept Webadministration (cf. LBS Platform Manual). Please refer to this manual for details of the parameters to set to ensure the autocomplete web service will operate.

We strongly recommend that the UGC logs remain disabled during utilisation of the autocomplete web service. In effect, these logs are alimented each time a user query is run, and will become huge in size within minutes of a production type utilisation. In addition, performances will be much reduced when logs are activated.

Universal Geocoder Server Reference Guide

Geocoding is a procedure applied to a number of address records or site locations to generate, for each location, a pair of X and Y coordinates. For more information on geocoder terminology (types of geocoding, tolerances…), refer to the Universal Geocoder user documentation.

This document describes the Geoconcept Universal Geocoder component.

The description of the component breaks down into three separate sections:

- UGC JEE: this is a component designed to handle the integration of the Universal Geocoder (UGC) geocoding engine in the Java Enterprise Edition platform and its subsets, like the Tomcat servlet engine. By extension (deployment of an optional module) the product can be used with the aim of deploying a geocoding web service.
- UGC Command Line
- UGC .NET

This documentation is not the documentation of Geoconcept Universal Geocoder standalone.

JEE Universal Geocoder

This component is designed to handle the integration of the Universal Geocoder (UGC) geocoding engine in the Java Enterprise Edition platform and its subsets, such as the Tomcat servlet engine. By extension (deployment of an optional module) the product can be used with the aim of deploying a geocoding web service.

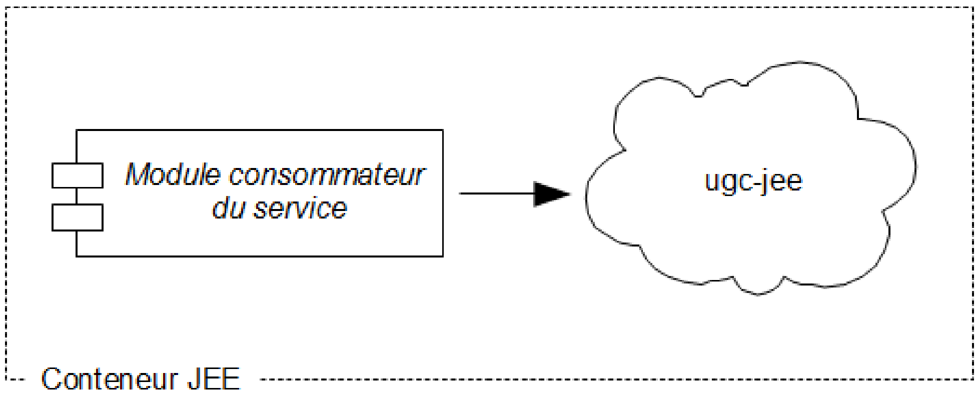
From a functional point of view, geocoding is a procedure applied to a series of address records to obtain their geographic coordinates. For more information about the terminology of the geocoder (types of geocoding, tolerances…) refer to the Universal Geocoder user manual. This manual aims to describe in detail how to deploy Universal geocoder specifically for Java Enterprise Edition (ugc-jee).

Basic principles

JEE Integration

ugc-jee is a JEE platform integration component providing a geocoding service to JEE modules deployed on an application server (in the widest sense, Tomcat type of servlet engine included).

JEE Integration



The consumer module can be of any type (webapp, ejb, etc). Similarly, inside the JEE module the consumer can be of any type (servlet, hsp, pojo, etc).

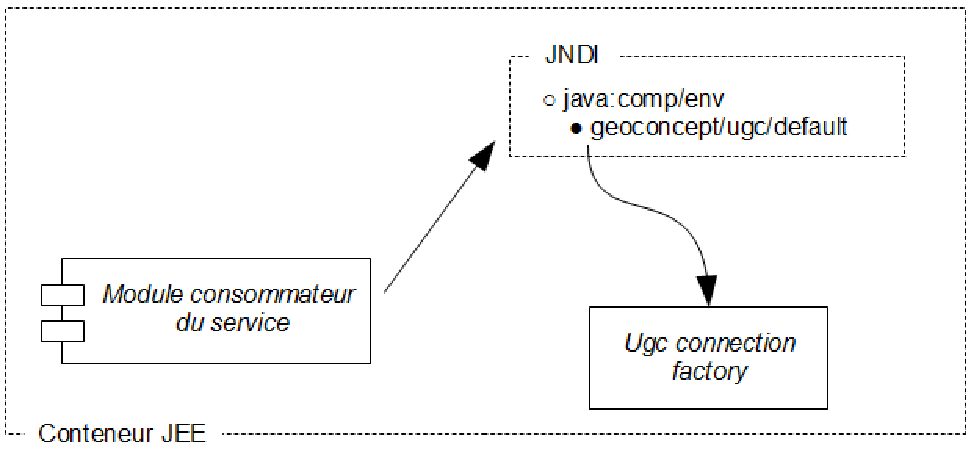
Access mode

The application using the geocoder references the provider via a logical name in the JNDI (Java Naming and Directory interface) directory of the application server. The method to use for setting up the provider will be described later on.

Traditionally, the logical name used is the “java:comp/env” context name “geoconcept/ugc/default”, but it is also possible to use another name, another context, or to set up several providers of different types.

In the interests of simplicity, we will start by describing the most usual utilisation scenario: a single primary provider with naming by default.

Access mode

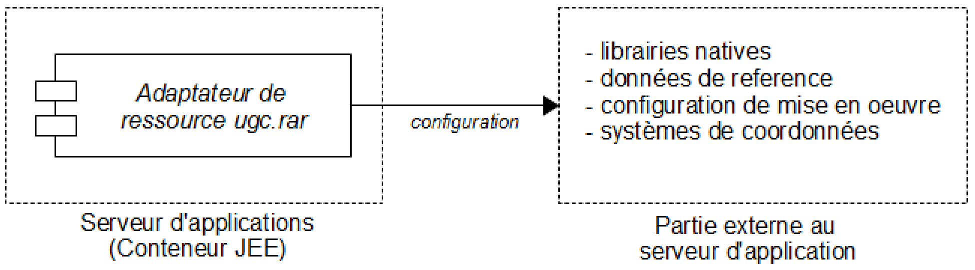


Component structure

Adapting the JEE resource and external components

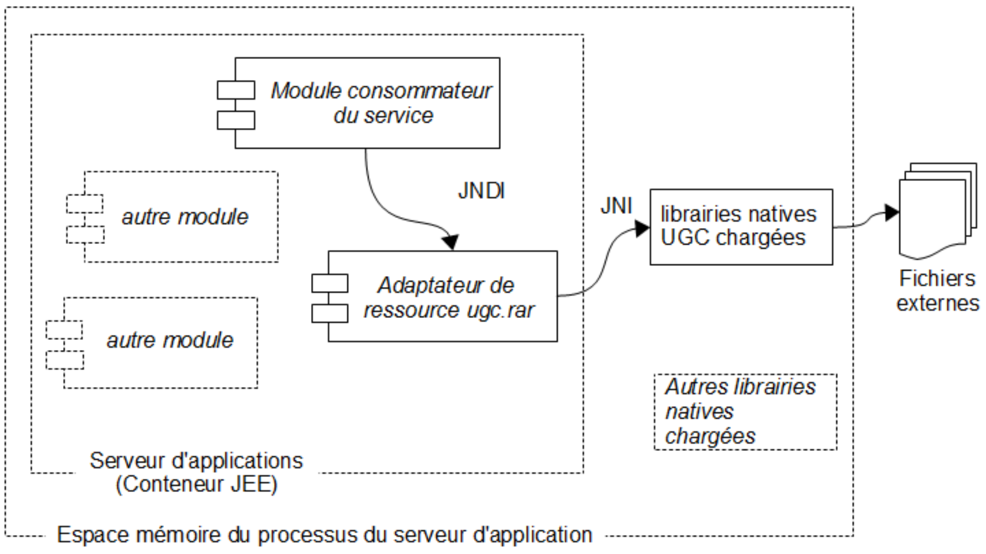
ugc-jee breaks down into several parts. For questions of performances and re-utilisation, the engine (also called the kernel) is written in C++ and is therefore published in the form of native libraries. The integration part (resource adapator) handles the link with the engine by handling its deployment via loading its native libraries. In terms of deployment of files, this then concerns two distinct trees:

Component structure


Adaptateur de ressource ugc.rar
configuration
- librairies natives
- données de reference
- configuration de mise en oeuvre
- systèmes de coordonnées
Serveur d'applications (Conteneur JEE)
Partie externe au serveur d'application

In terms of execution, the native libraries will be loaded into memory in the process of the application server instance (or, where appropriate, its partition, depending on the model) at start-up. The resource adaptor part features a java interface and handles the instanced engine in memory directly via JNI.

Component structure


Module consommateur du service
JNDI
autre module
autre module
Adaptateur de ressource ugc.rar
JNI
librairies natives UGC chargées
Fichiers externes
Autres librairies natives chargées
Serveur d'applications (Conteneur JEE)
Espace mémoire du processus du serveur d'application

Another type of provider

The resource adaptor described previously corresponds to the LocalDll type, and this consists of the primary provider, and namely the one that is linked to the geocoding enfine and in practise, handles processing.

In some situations (for various reasons: separated frontal, sharing, architecture, insulation, etc) it is desirable that the module consuming the service is not deployed in the same instance of the application server as the primary ugc provider (for example, each being one being on a different server machine - remoting).

In this case there are several possibilities:

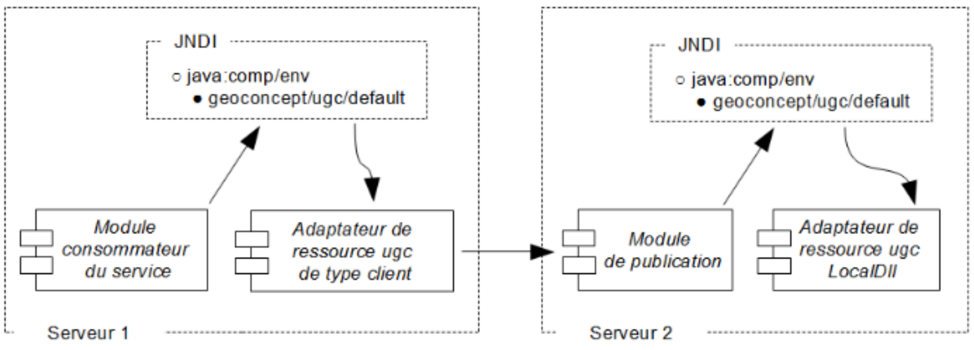
- either, the creation of a dedicated publication module, that will have access to the uge provider (in this case, the protocol used and the procedures are specific to the module created).

- or the consumer is implemented like a client from a module supplied by ugc-jee like, for example, a web service of the ugc-ws module.
- or a module provided by ugc-jee is deployed on the primary provider's machine and a client resource adaptor for this module is set up on the consumer machine.

This last scenario is transparent for the consumer application: whether the provider is local or remote, the accecss code remains the same, it is simply a question of deployment and configuration (it's a bit like the Multiple business delegate pattern).

The transport protocol used depends on the publication module / client pair chosen: this could be web service (http/soap) or rmi.

Another type of provider



In every scenario, in the framework of the deployment of ugc-jee in installed mode (that is, non SAAS mode) there is a provider instance of the LocalDll type. In other respects, and to the extent that this provider is local (no transfer, no serialisation, etc) it is the solution offering highest performances (latency, bandwidth).

Add-on modules

A certain number of add-on modules are supplied with ugc-jee. None are essential to the correct operation of the resource adaptor, and their deployment is optional.

ugc-admin

ugc-admin is a webapp that checks the product has been correctly installed, handles the configuration and tests the application to ensure it is functioning correctly.

ugc-ws

ugc-ws is a webapp that publishes a geocoding web service. It is based on spring-ws (it publishes in document / literal encoding format).

ugc-axis-fusion

ugc-axis-fusion is a webapp that publishes a predeployed geocoding web service on Axis 1.4 (it publishes in rps soapenc encoding format).

ugc-remote

ugc-remote is a webapp for publishing the geocoding service via rmi. It is notably used for remoting in conjunction with the RmiClient type provider.

Administration / Configuration

Configuring the application deployment

A repository made up of a series of files with .ugc.xxi file extensions, constitutes a datasource (datasource).

The service.xml file located in %ugc%/conf defines the general configuration of the geocoding service provider. This contains notably a default configuration for the datasources (default-datasource). You can define a specific configuration to be assigned to a datasource.

To define a specific fonfiguration for a datasource, it is advisable to use the ugc-admin sebapp (see the section on DataSourcesConfiguration, and then configuration > create). It is also possible to edit the service.xml file directly.

If no specific configuration has been defined, then the default configuration is applied on deployment of this datasource.

In other respects, it is possible to define certain parameters during the call (via findAddressOptions). In this case the concrete value that a parameter takes (like, for example discrepancy) may originate (in order):

- from the call if it is assigned in findAddressOptions (this assignment is optional)
- from the value indicated at the level of the specific datasource configuration (if a specific datasource configuration has been defined)
- from the value indicated at the level of the default datasource configuration (default-datasource).

Detailed configuration of a datasource

It is possible to define the configuration of the deployment of a referential in a very detailed way.

This definition may correspond to particular needs, and it is not essential since the default configuration will serve perfectly well in the majority of cases.

Some parameters can be defined at the time of the call, and others are set at the creation of the datasource instance and are then subsequently not modifiable.

The parameters you may want to define at the moment of the call will be described in the FindAddressOptions section.

Datasource identification

This section enables identification of a datasource in the administration.

File: geocoding referential utilised. This is contained in %geoconcept%/ugc/reftables or in a sub-directory.

Name: an alias for the name of the datasource (optional)

Publication information

This section allows you to add information about the datasource.

Title:: a title for the datasource.

Abstract: contains a short description of the datasource.

Online resource: HTTP link giving more information on the datasource.

Reference table settings

Version: version of the referential.

Zone meaning: meaning of the "zone" attribute for the address. Example: Post code.

UniqueID meaning: meaning of the uniqueId search identifier.

secondaryZone meaning: meaning of the secondayZone attribute for the address. For example, IRIS code.

StreetSectionId meaning: meaning of the streetSectionId attribute for the address. For example, PID navteq.

Coordinate system: identifier for the reference coordinates system for the publication (Cf. OpenGIS standard).

Country: the country concerned.

Bounds: rectangular footprint for the data in the referential

Run time settings

Cache: activation or not of the cache. This cache enables towns in the referential to be kept in memory.

Max Cache Mem Size is only used if the cache is active. The size of the memory reserved for the cache in Kbytes.

Min processors: the initial number of geocoding instances.

Max processors: the maximum number of geocoding instances.

Finder general settings

City scoring method: calculation method to use for the town score:

- Standard: rapid, but less precise. Recommended for a batch type of utilisation,

- Levenshtein: less rapid, but more precise. Recommended for a utilisation of the type where the address is input in a line.

Street scoring method: calculation method to use for the score of the street:

- Standard: rapid, but less precise. Utilisation is not recommended.
- Levenshtein : less rapid, but more precise. recommended for a utilisation of the type where the addresses are entered as a line, or in a batch.

Min streets: minimum number of streets for a town to be considered as covered. Used for the tolerance at town level (FindAddressResults.TOLERATE_TYPE_CITY).

Search strategy:

For more detail about search strategies, refer to the Universal Geocoder user guide.

Finder request defaults settings

Max candidates: maximum number of results returned by a geocoding operation. See FindAddressOptions.candidateCount.

Score threshold: the score above which the geocoding results are retained. See FindAddressOptions. ScoreThreshold.

Score threshold: threshold for the suggestion of "street" type candidates. See FindAddressOptions. ScoreThreshold.

Find type: type of geocoding desired. See FindAddressOptions.geocodingType.

Tolerate geocoding type: geocoding tolerance required. See FindAddressOptions.tolerateGeocodingType.

Max meter error: maximum positioning error (in metres) for a geocoding at street level to be considered as a geocoding on street number. See FindAddressOptions.maxMeterError. Discrepency: lateral offset. See FindAddressOptions.discrepency.

Discrepancy along street: longitudinal offset. See FindAddressOptions.discrepancy.

Favor city match element: gear the search for the town with a descriptive element for the town. See FindAddressOptions.favorCityMatchElement.

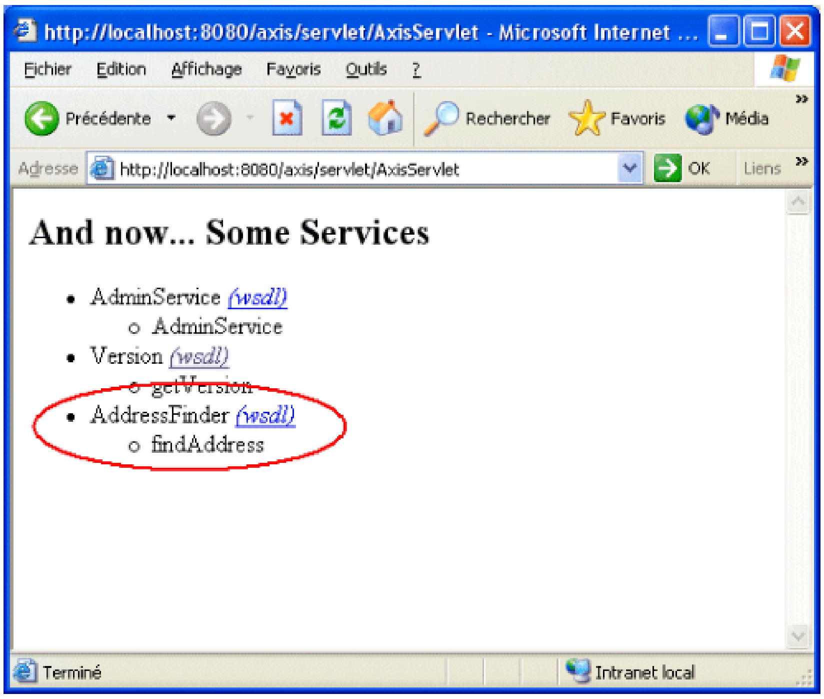
Zone match digits: take into account the n first characters of the area attribute for the address. See FindAddressOptions.zoneMatchDigits.

Tests / Troubleshooting for AXIS 1.4

Check that the resource adaptor has loaded correctly when Tomcat is started. You can use the webapp ugc-admin to validate the installation.

When using a Web Service, check that the AddressFinder service is present in axis:

Tests / Troubleshooting for AXIS 1.4


http://localhost:8080/axis/servlet/AxisServlet - Microsoft Internet ...
Fichier Edition Affichage Favoris Outils ?
Précédente
Rechercher
Favoris
Média
Adresse http://localhost:8080/axis/servlet/AxisServlet
OK
Liens
Terminé
Intranet local

And now... Some Services

- AdminService (wsdl)
  - AdminService
- Version (wsdl)
  - getVersion
- AddressFinder (wsdl)
  - findAddress

Java interface

When the provider is used directly inside the java platform (via single java or «POJO» objects), the applet takes the following form:

- restoration a connection to the provider via JNDI
- calling the findAddress function

The findAddress function has the following simplified form:

```
FindAddressResults findAddress(Address, FindAddressOptions);
```

This means in effect that the call takes as input an address and some options, and returns as output a certain number of candidates.

The precise breakdown is given below.

Package com.geoconcept.ugc.service

Class CodingProvider

getConnection method:

This method allows you to open a connection to the geocoding provider. A connection must be subsequently closed using its Close method.

Prototpye:

getConnection() connection throws ResourceException;

Value returned:

Connection to the geocoding service provider.

Class Connection

findAddress method:

This method enables an address to be geocoded with geocoding options if required.

Prototype

FindAddressResults findAddress(Address address, FindAddressOptions options)

throws InvalidDataSourceException, InternalErrorException;

Parameter Description

address Address to geocode

options Geocoding options

Value returned

Geocoding result list.

Package com.geoconcept.common.geo

Class Location

This class contains the coordinates found during a geocoding operation.

Members

Type	Name	Description
double	x	X Coordinates
double	y	Y coordinates
String	coordinateSystem	Coordinates system identifier. "SRS" Identifier ("Spatial Reference System") for the Web Map Service. See the OpenGIS standard.

Package com.geoconcept.ugc

Class Address

This class contains the description of an address.

Members

Type	Name	Description
String	addressLine	Concatenation of the number, of the street type and the name of the street. For example, 25 rue de Tolbiac.

Type	Name	Description
String	zone	Town code (for example, the post code 75013)
String	cityName	Name of the town (for example, Paris)
String	uniqueID	Unique code (town, zone) to search for (for example, 75113000)
String	secondaryZone	Secondary code for the address (for example, the INSEE code, the IRIS code…)
String	StreetSectionID	Road section identifier (not yet utilised)

Class FindAddressOptions

This class contains the configuration for a geocoding operation.

All elements, except for dataSourceName are optional. If they are not assigned, they take their default value, that is optimum in the majority of cases.

Type	Name	Description
String	dataSourceName	Name of the data source defined in the administrator to use for the geocoding options
short	candidateCount	Maximum number of results to return when geocoding
String	findType	The type of geocoding required. There are three types of geocoding: on towns (FindAddressResults.FIND_TYPE_CITY), on streets (FindAddressResults.FIND_TYPE_STREET) , on street numbers(FindAddressResults.FIND_TYPE_STREET_NUMBER)
String	tolerateFindType	This property allows you to set and obtain the cumulated value of geocodings tolerated. There are three tolerance levels: - on the town (FindAddressResults.TOLERATE_TYPE_CITY) in the case where the street could not be found and the town is not covered (number of streets for this town is insignificant in the reference table), - on the street(FindAddressResults.TOLERATE_TYPE_STREET) in the case where the number requested could not be found, - on the estimated number(FindAddressResults.TOLERATE_TYPE_STREET_ENHANCE), if the number could not be found and the positioning error does not exceed the value of "maxMeterError". The tolerances are "flags" that can be cumulated. For example, if you tolerate the three categories, you can set a tolerance number of 7, while a tolerance uniquely at town level corresponds to a tolerance with a value of 1. The following table summarises the possibilities that exist:
long	maxMeterError	Maximum positioning error (in metres) to consider a geocoding at street level as a geocoding on street number. The type of geocoding will therefore depend on the length of a street.
double	discrepancy	Orthogonal offset or stagger (in metres) to apply to the street found. This allows you to avoid positioning the geocoded address in the middle of the street.
double	discrepancyAlongStreet	Longitudinal stagger (in metres) to apply. This allows you to avoid positioning the geocoded address on a crossroads.
long	favorCityMatchElement	This allows you to improve the search for a town by specifying the element of the town (the name of the town, the town code, or the unique code for a town for which one can be certain of the nature of the data. The assignment of this value will therefore depend on the address to geocode. Three values are possible: the name of the town (FindAddressResults.FAVOR_CITY_NAME), the code for the town (FindAddressResults.FAVOR_ZONE), the unique code for the town (FindAddressResults.FAVOR_UNIQUE_ID).

Type	Name	Description
long	zoneMatchDigits	This allows you to set the number of characters to use for the matching of the town code stored in the reference table and that passed to parameter for geocoding. A value can only be specified if the favorCityMatchElement member is different from (FindAddressResults.FAVOR_ZONE), and you need to use the whole of the post code.
int	scoreThreshold	Minimal score for propositions from the geocoder to select
int	scoreThreshold ToTolerateStreet GeocodingType	Define a minimum score to ensure candidates of the "street" type are returned. If no street attains this threshold, then the town will be returned as a candidate.
String	coordinateSystem	Define the Reference system for the coordinates to return

Class FindAddressResults

Class that contains the results of a geocoding operation, classified by score.

Members

Type	Name	Description
FindAddressResult[]	results	Table of candidates found
int	matchType	Type of geocoding applied

Constants

Type	Name	Value	Description
int	FIND_MATCH_TYPE_CITY	1	Type of geocoding requested: the address must be geocoded on the town
int	FIND_MATCH_TYPE_STREET	2	Type of geocoding requested: the address must be geocoded on the street
int	FIND_MATCH_TYPE_STREET_NUMBER	3	Type of geocoding requested: the address must be geocoded on the street number
int	FOUND_MATCH_TYPE_CITY	1	Type of result geocoding: the address has been geocoded on the town
int	FOUND_MATCH_TYPE_STREET	2	Type of geocoding result: the address has been geocoded on the street
int	FOUND_MATCH_TYPE_STREET_ENHANCED	3	Tye of geocoding result; the address has been geocoded on the approximate street number
int	FOUND_MATCH_TYPE_STREET_NUMBER	4	Type of geocoding result: the address has been geocoded on the exact street number
int	TOLERATE_TYPE_CITY	1	Tolerance of the geocoding on the town
int	TOLERATE_TYPE_STREET	2	Tolerance of the geocoding on the street
int	TOLERATE_TYPE_STREET_ENHANCED	4	Tolerance of the geocoding on the estimated street number
int	FAVOR_CITY_NAME	1	Steers the search for the town to be geocoded by providing the name of the town

Type	Name	Value	Description
int	FAVOR_ZONE	2	Steers the search for the town to be geocoded by providing the code for the town
int	FAVOR_UNIQUE_ID	3	Steers the search for the town to be geocoded with the unique code of the town

Class FindAddressResult

Class that contains the result of a geocoding operation.

Members

Type	Name	Description
Address	address	Address found
Location	location	Coordinates found
double	score	Resemblance score attributed between the address to be geocoded and the address found. Varies between 0 (no resemblance) and 1 (the exact address)
int	type	Type of geocoding found. the possible values are: - FindAddressResults.FOUND_MATCH_TYPE_CITY - FindAddressResults.FOUND_MATCH_TYPE_STREET - FindAddressResults.FOUND_MATCH_TYPE_STREET_ENHANCED - FindAddressResults.FOUND_MATCH_TYPE_STREET_NUMBER

Utilisation

Guiding principles

A geocoding operation consists of the following procedures:

- Connect to the geocoding service provider,
- Construction of the address to geocode,
- Construction of geocoding options,
- Execution of the geocoding,
- Exploitation of the result,
- De-connection from the geocoding service provider.

Example

Connection to the geocoding service provider

```
import com.geoconcept.ugc.service.CodingProvider;
import com.geoconcept.ugc.service.Connection;

/*
*Open a connection on the geocoding server
*/
public Connection getConnection() throws Exception
{
        Connection connection = null;
        try
        {
                // get context
                Context initCtx = new InitialContext();
                Context envCtx = (Context) initCtx.lookup("java:comp/env");

                // retrieves the geocoding server form the logical name
                CodingProvider codingProvider = (CodingProvider) envCtx.lookup("geoconcept/ugc/default");
                connection = codingProvider.getConnection();
        }
        catch (Exception e) { e.printStackTrace(); }
        return connection;
}
```

Construction of the address to geocode

```
import com.geoconcept.ugc.Address;

/*
*Construct an adress to geocode
*@param  addressLine address number + address  way type + address  way name. Sample : "25 rue de Tolbiac"
*@param   zone adress sone. Sample : "75013"
*@param  cityName city name. Sample : "Paris"
*@param  uniqueId city unique identifier. Sample : "75113000"
*
*/
public Address getAddressToGeocode(String  addressLine, String  zone,String cityName,String uniqueId)
throws Exception
{
        Address address = new Address();
        address.addressLine = addressLine;
        address.zone = zone;
        address.cityName = cityName;
        address.uniqueId = uniqueId;
        return  address;
}
```

Construction of the geocoding options

```
import com.geoconcept.ugc.FindAddressOptions;

/*
*Retrieves geocoding options from the data source defined in administration
*@param  datasource Name of a data source
*/
public FindAddressOptions  getOptions(String datasource) throws Exception
{
        FindAddressOptions options = new com.geoconcept.ugc.FindAddressOptions("myDataSource");
     return  options;
}
```

Execution of the geocoding

```
import com.geoconcept.ugc.service.Connection;
import com.geoconcept.ugc.Address;
import com.geoconcept.ugc.FindAddressOptions;
import com.geoconcept.ugc.FindAddressResults;

/*
*Launch geocode process and retrieves results
```

```
*@param  connection Connection to the geocode server
*@param  address Address to geocode
*@param   options Geocoding options
*/
public FindAddressResults  findGeocode(Connection connection, Address address , FindAddressOptions options)
 throws Exception
{
        FindAddressResults findAddressResults  = connection.findAddress(address, options);
        return  findAddressResults;
}
```

Exploitation of the result

```
import com.geoconcept.ugc.FindAddressResults;
import com.geoconcept.ugc.FindAddressResult;
/*
*Browse geocoding results and display result
*@param  findAddressResults  Results of geocode process
*/
public void displayResult(FindAddressResults findAddressResults) throws Exception
{
        // if at least one found result
        if (findAddressResults.results.length > 0)
        {
                // Display colunm name
                system.out.println("Found results : ");
                system.out.println( "N°"+ "\t"
                                + "Geocoding Type"+ "\t"
                                + "Score"+ "\t"
                                + "Address line"+ "\t"
                                + "Zone"+ "\t"
                                + "City Name"+ "\t"
                                + "City Unique Identifier"+ "\t"
                                + "Adress Secondary Zone"+ "\t"
                                + "Coordinates (Coordinates System)");

                // For each found results
                for (int i = 0; i < findAddressResults.results.length; i++)
                {
                // get next found result
                       FindAddressResult findAddressResult = found.results[i];

                String  coordinateSystem = null;
                if (findAddressResult.location.coordinateSystem != null)
                {
                        coordinateSystem = findAddressResult.location.coordinateSystem;
                }
                else
                        coordinateSystem = "(Unknown)";

                        // Display result
                system.out.println( i + "\t"
                                + findAddressResult.address.type + "\t"
                                + findAddressResult.address.score + "\t"
                                + findAddressResult.address.addressLine + "\t"
                                + findAddressResult.address.zone + "\t"
                                + findAddressResult.address.cityName + "\t"
                                + findAddressResult.address.uniqueId + "\t"
                                + findAddressResult.address.secondaryZone + "\t"
                                + findAddressResult.location.x + ","+ findAddressResult.location.y
                                + "( + coordinateSystem + ")");
                }
        }
        else
        {
                system.out.println("No found result.");
        }
}
```

De-connection from the geocoding service provider

```
import com.geoconcept.ugc.FindAddressResult;
import com.geoconcept.ugc.service.Connection;
/*
*Disconnection of the geocode server
*@param  connection  Connection to the geocode server
*/
public void closeConnection(Connection connection) throws Exception
{
         connection.close();
}
```

Full example

```
import com.geoconcept.ugc.service.CodingProvider;
import com.geoconcept.ugc.service.Connection;
import com.geoconcept.ugc.Address;
import com.geoconcept.ugc.FindAddressOptions;
import com.geoconcept.ugc.FindAddressResults;
public void geocodingSample()
{
        try
        {
                // Open connection
                Connection connection = getConnection();
                // Construct the address to geocode
                Address address = getAddressToGeocode("25 rue de Tolbiac","75013","Paris","");
                // retrieves geocode options
                FindAddressOptions options = getOptions("myDataSource");
                // launch geocode process
                FindAddressResults findAddressResults = findGeocode(connection, address , options);
                // print geocoding result.
                displayResult(findAddressResults);
                // disconnection
                closeConnection(connection);
        }
        catch(Exception e)
        {
                // geocoding problem
        }
}
```

SmartRouting Server

Installing the SmartRouting Server component

The Geoconcept Web installer offers to install two versions of SmartRouting Server:

- JEE SmartRouting,
- SmartRouting Command Line.

These versions are independent of one another, and serve different purposes. For example, the JEE SmartRouting component is needed for the “Route Calculation” widget.

Deployment of the native libraries

For Windows

The libraries are loaded explicitly from the sub-directory corresponding to the individual environment (32/64 bits) of the external directory configured (RootDirectory) for the local provider (ConnectionMode="LocalDll").

Deployment under Linux

Only libraries linked dynamically are currently available.

Installation for utilisation of dynamically linked libraries

Loading libraries of dependencies linked dynamically requires that the system variable LD_LIBRARY_PATH contains the libraries that will be used. The way to assign the environment variable LD_LIBRARY_PATH depends on the shell, the start-up mode for the application server, etc. In the same way, the value assigned will vary depending on the host OS, the deployment directory, the loading mode (described below).

For example, for a manual start-up of tomcat via sh or compatible (bash, etc) the user enters export LD_LIBRARY_PATH=/home/smartrouting/native/linux32-x86/debian/3.1/dynamic or export LD_LIBRARY_PATH=/deploy/smartrouting/native/linux64-amd64/red-hat/5.6/dynamic then startup.sh of the tomcat bin directory (the variable is assigned for the session).

It is also possible to modify the catalina.sh file in order to define the LD_LIBRARY_PATH variable for all the tomcat start-ups, or again to modify the /etc/rc.d/.. files for an automated execution.

The service.xml file located in the conf directory of the hierarchical tree external to the application server (%smartrouting%) defines the loading mode for native libraries (under Linux exclusively since under Windows the mode is still the same) in the native-libraries section.

```
<?xml version="1.0" encoding="UTF-8"?>
<service>
  [...]
  <native-libraries>
    <log-enabled>false</log-enabled>
    <always-detect>true</always-detect>
    <detect-method>auto</detect-method>
```

```
        <last-detect />
        <!--force loading libraries from given directory under ugc/native eg 'linux64-amd64/my-dir' : -->
        <dir></dir>
        <link-mode>dynamic</link-mode>
    </native-libraries>
        […]
```

The directory from which the resource adaptor (of the ConnectionMode="LocalDll" type) will load the libraries is defined either automatically (by detection) or it can be configured manually. It is the value of the dir element that will determine this: if it is defined and is not empty, then it is the value of the element that will be used as a sub-directory of the native directory of the hierarchical tree external to the application.

If dir is set to a value of «mydir/test» and the RootDirectory of the resource adaptor is set to a value of «/home/smartrouting» then the directory used for the loading event will be /home/smartrouting/native/mydir/test.

In the event that dir is defined (loading is explicitly forced), the parameters that configure the automated loading by detection will have no effect: link-mode, detect-mode, always-detect.

Where loading is automated loading by detection (dir is undefined or empty), the sub-directory of the native directory of the hierarchical tree that is external to the application server that will be used will depend on:

- the linux (32/64) environment,
- the linux distribution (debian, red hat),
- the version of the linux distribution,
- the link mode for libraries (static/dynamic) - currently only dynamic mode is available.

Whatever the mode used to define the directory to be used (deliberately forced via native-libraries/dir or automatically detected), in dynamic link mode (only), it will be essential that the environment variable LD_LIBRARY_PATH contains this directory.

Deployment and configuration of the application server

Please refer to the section '*Configuration for SmartRouting JEE*' in the Geoconcept Web manual.

SmartRouting Reference Guide

SmartRouting is a powerful route calculation and proximity search engine. Route calculations are based on a file describing the road network, referred to as a *graph*. This file is created from the Geoconcept GIS and a route navigation database, such as that published by Navteq. A series of speed profiles can be defined (pedestrian, heavy goods vehicle, etc) and various traffic circulation rules taken into account (no-entry, one-way street, no U-Turns, etc). SmartRouting Server can perform various operations:

- calculation of an itinerary between two points,
- calculation of an itinerary between one point and several points,
- creation of a route sheet,

- search for entities in proximity to a reference point (Search Around).

SmartRouting can be used to obtain a complete itinerary along with the description of the route (including the path the route follows between two intersections), the duration of the journey between each segment and the distance.

This documentation describes this component, and is in two distinct parts:

- SmartRouting JEE : this is a component designed for the integration of the SmartRouting route calculation engine within the Java Enterprise Edition (JEE) platform and its sub-assemblies such as the Tomcat servlet engine.
- SmartRouting Command line : this is a component designed to calculate itineraries from the command line.

General principles of the SmartRouting route calculation engine

The Graph: any calculation function requiring a graph trajectory requires the utilisation of a graph. The graph constitutes a logical view of the network. It contains and describes all information related to the connectivities between road sections, to the costs and constraints. It is constructed and organised in such a way that the calculations are applied efficiently. The file is in SITI format.

This SITI format corresponds to the second generation of graphs developed for Geoconcept (from version 6.5 of Geoconcept onwards). It potentially integrates numerous additional characteristics (multiple speed profiles, numerous management rules…).

Please refer to the Geoconcept Solution Reference Guide for details of the rules that apply when creating a graph.

Snapping to the graph

When making calculations, the start points, stops and finish points are linked to the network constituted by the graph, even if the objects are distant from it (points situated at a certain distance from the network): this is the snap-to-graph function. This operation aims to search around these points for the closest road sections. Then, these road sections are used to determine the start points of possible solutions.

The itinerary as calculated then integrates a distance covered between these points and the graph, called the snap-to-graph distance. To optimise performances, there is an option that allows a maximum snap-to-graph distance to be set in metres (*max graph snap distance*). This rules out any need to browse the whole of the network to find the nearest road sections.

In addition, the snap-to-graph speed, the speed of traversing the distance separating the start, stop or finish points can also be customised (*graph snap speed*).

The Metagraph

A metagraph consists of the possibility of integrating within the creation of the graph a simplified graph, or a graph that is only a subset of its original content. The utilisation of the metagraph aims, essentially, to accelerate the traversing of the graph, when performing route calculations over long distances.

If an itinerary must be calculated between two distant towns several hundred kilometres apart, it is preferable to give precedence to a graph trajectory at the level of the main routes (tending towards logic level 1, for example, that is considered as the highest level of thoroughfare). Around the start points, and any stop or finish points, the logic levels that are not present in the metagraph will still be employed, but for inter-town links the logic levels used in the metagraph take precedence.

The metagraph is a file accompanying the SITI graph. It is made up of the association of a file in MG format and an MGLIST file. The MG file is loaded into memory.

The reference-level

This is the logic level on which the route calculation will principally be performed. Two scenarios are possible:

- using a metagraph: this is the “normal” scenario, the metagraph being specifically constructed to accelerate the calculation of the itineraries over long distances (only the larger roads will be selected on longer journeys). It is necessary to assign values to the logic level defined in the metagraph in order to optimise calculation times,
- when not using a metagraph: the route calculation will be optimised by interrogating as a priority the road sections at the level specified directly in the graph.

Around way points (for example, start, stop and finish points) all routes will be taken into account, whatever logic level has been selected.

A logic level that is set too high can adversely affect the quality of the route calculation since numerous routes will not be taken into account.

Input and output coordinates

Both input and output coordinates for the SmartRouting Server can be specified as required.

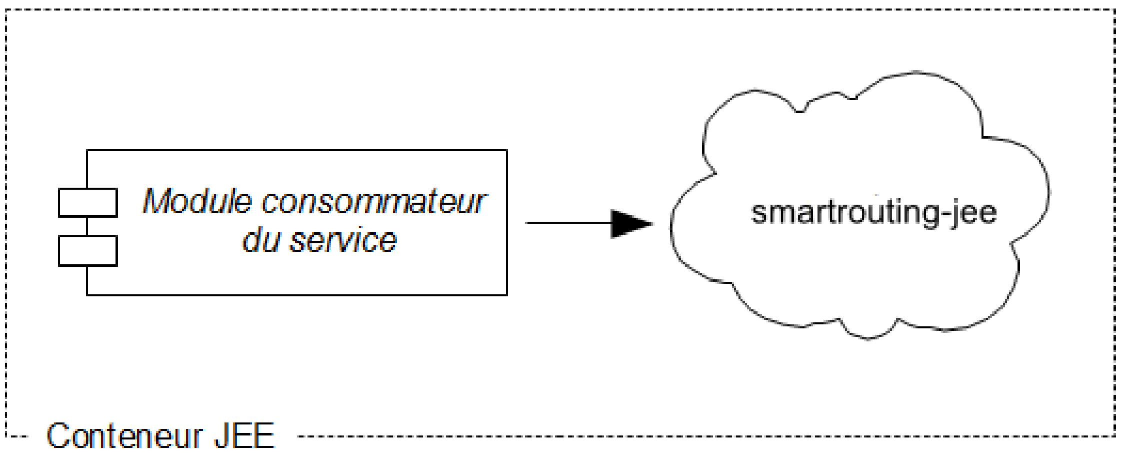
JEE SmartRouting

Basic principles

JEE Integration

JEE SmartRouting is a component for integrating within a JEE platform, and offers a route calculation service to JEE modules deployed on an applications server (in the widest sense, including a Tomcat type of servlet engine).

JEE Integration



The consumer module can be of any type (webapp, ejb, etc). Even inside the JEE module, the consumer can be of any type (servlet, jsp, pojo, etc).

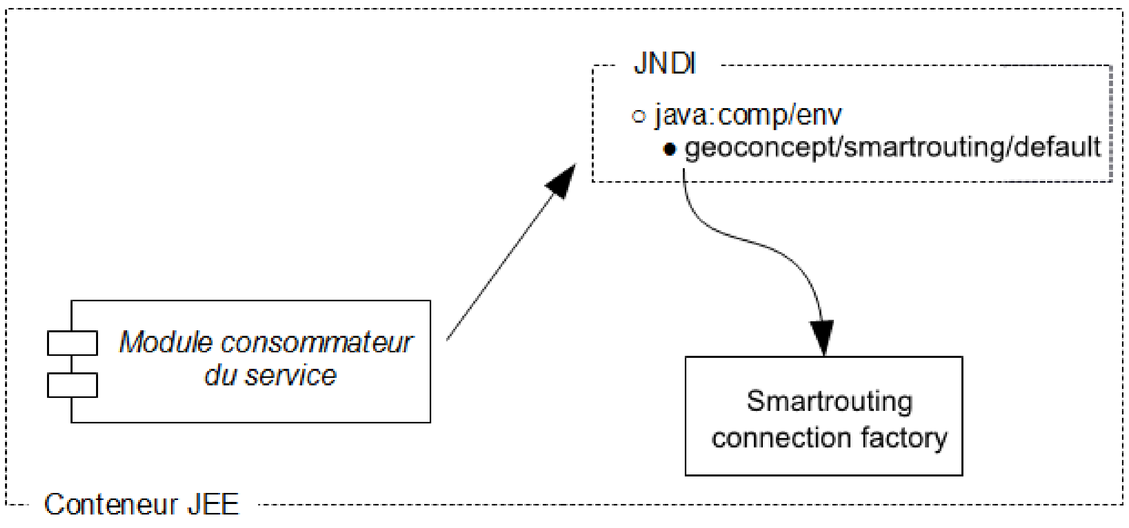
Method of access

The application using the route calculation facility references the provider via a logical name from the JNDI directory (Java Naming and Directory interface) for the application server. The method for setting up a provider will be described later on.

Traditionally, the logical name used is “geoconcept/smartrouting/default” from the “java:comp/env” context, but it is possible to use another name, another context, or to set up several providers of different types.

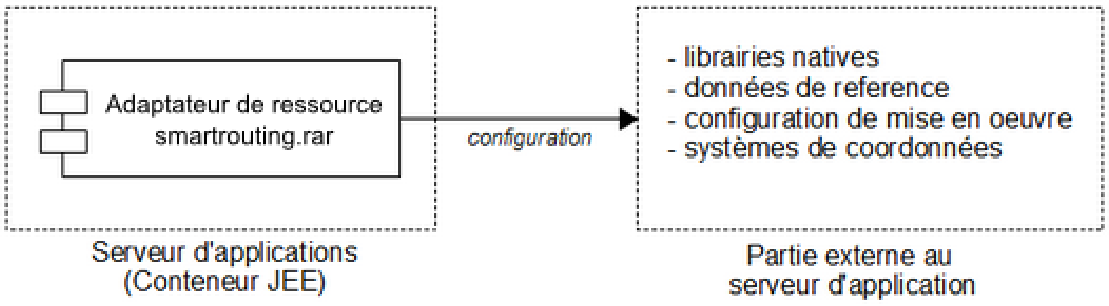
In the interests of simplicity, the case of the most common utilisation (a single local primary provider with a default naming system) will be described first.

Method of access


JNDI
java:comp/env
geoconcept/smartrouting/default
Module consommateur du service
Smartrouting connection factory
Conteneur JEE

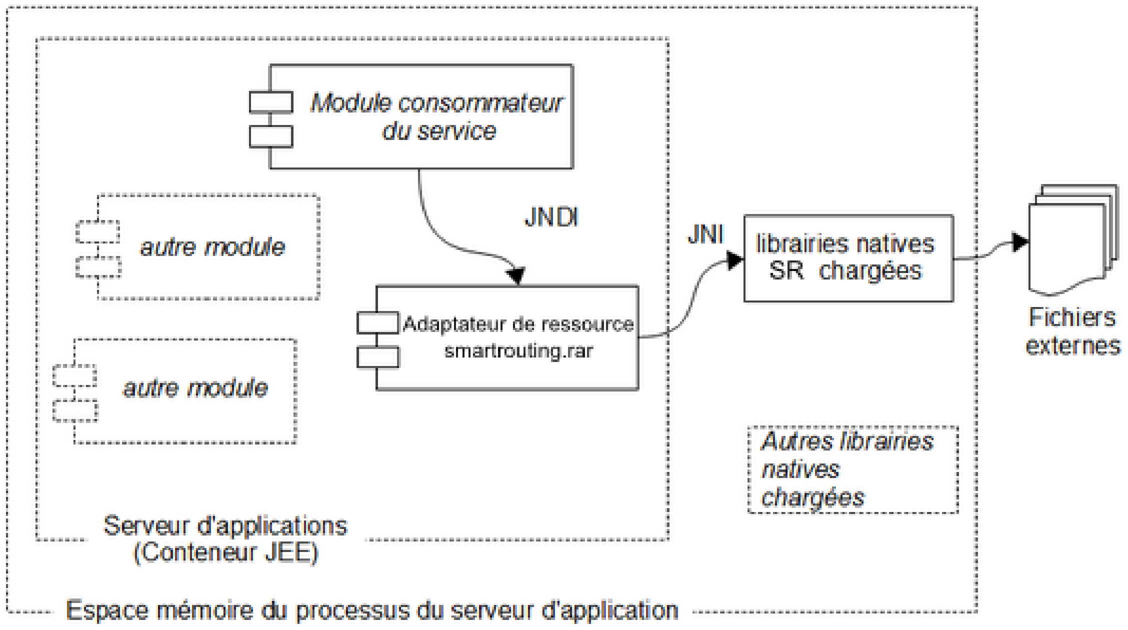
Compoonent structure: Adaptation of the JEE resource and external elements smartrouting-jee breaks down into several parts. For performances and re-utilisation considerations, the engine (also called the kernel) is written in C++ and is therefore distributed in the form of native libraries. The integration part (resource adaptor) constructs the line with the engine while handling its deployment through the loading of its native libraries. In terms of deployment of files, this therefore concerns two distinct trees:

Structure of the component



In terms of execution, the native libraries will be loaded into memory during the processing of the application server instance (or into its partition, if necessary, depending on the model) at start-up. The adaptor part of the resource features a java interface and handles the engine instanced in memory directly via JNI.

Structure of the component



Add-on modules

There is no add-on essential to the currect operation of the resource adaptor, so deployment of add-ons is optional.

smartrouting-admin

smartrouting-admin is a webapp for checking the product is correctly installed, and testing the configuration to ensure all is working smoothly.

Administration / Configuration

Configuring the application deployment

A published graph (.siti file) constitutes a data source (datasource).

The service.xml file located in %smartrouting%/conf defines the general configuration of the route calculation service provider. This contains notably a default configuration for the datasources (*default#datasource*). You can define a configuration that is specific to a particular datasource. To know what the parameters should be, it is best to consult the hater on options.

If no specific configuration has been defines, the default configuration (*default-datasource*) will apply to the deployment of this datasource.

In other respects, it will be possible to define some parameters during the call (via CalculateRouteOptions). This means the real value that a parameter takes can be sourced by (in order):

- from the call if it is assigned in CalculateRouteOptions (this assignment is optional)
- from the value indicated at the level of the particular datasource configuration (if a particular datasource configuration has been defined)
- from the value indicated at the level of the default datasource configuration `default-datasource`).

Detailed configuration of a datasource

The configuration of a deployment of a graph can be defined in a very precise way.

This definition can fulfil particular requirements, and is not essential since the default configuration will be perfectly adequate in the majority of cases.

Some parameters can be defined at the time of the call, while others are fixed on creation of the instance of the datasource, and are not modifiable subsequently.

The parameters that one can define also at the time of the all will be described in the CalculateRouteOptions section.

Datasource identification

This section enables identification of a datasource in the administration.

File: graph.

Name: alias of the datasource name (optional).

Options available in JEE SmartRouting

The options available are those described under the SmartRouting Reference Guide.

These are accessible visually via the smartrouting-admin.war test application, that presents an input for each option in turn so you can observe the behaviour in the result.

Input and output coordinates

If the input coordinates are not in the same coordinates system as the graph, you will have to specify them via the `input coordinate system` parameter (in the form epsg:4326, for example).

The output coordinates will be in the same coordinates system as the graph, except where the `output coordinate system` parameter is filled with the EPSG code (in the form epsg:27572 for example).

Output results

Over and above the component segments of the route calculated, it is possible to receive as a result the attributes of the said segments, thanks to the fields and language fields.

- fields: allows you to select the fields you want to return (by default, all the fields are returned if the option is activated. The name of the fields must be written in English),
- language: allows you to return these fields in the desired language.

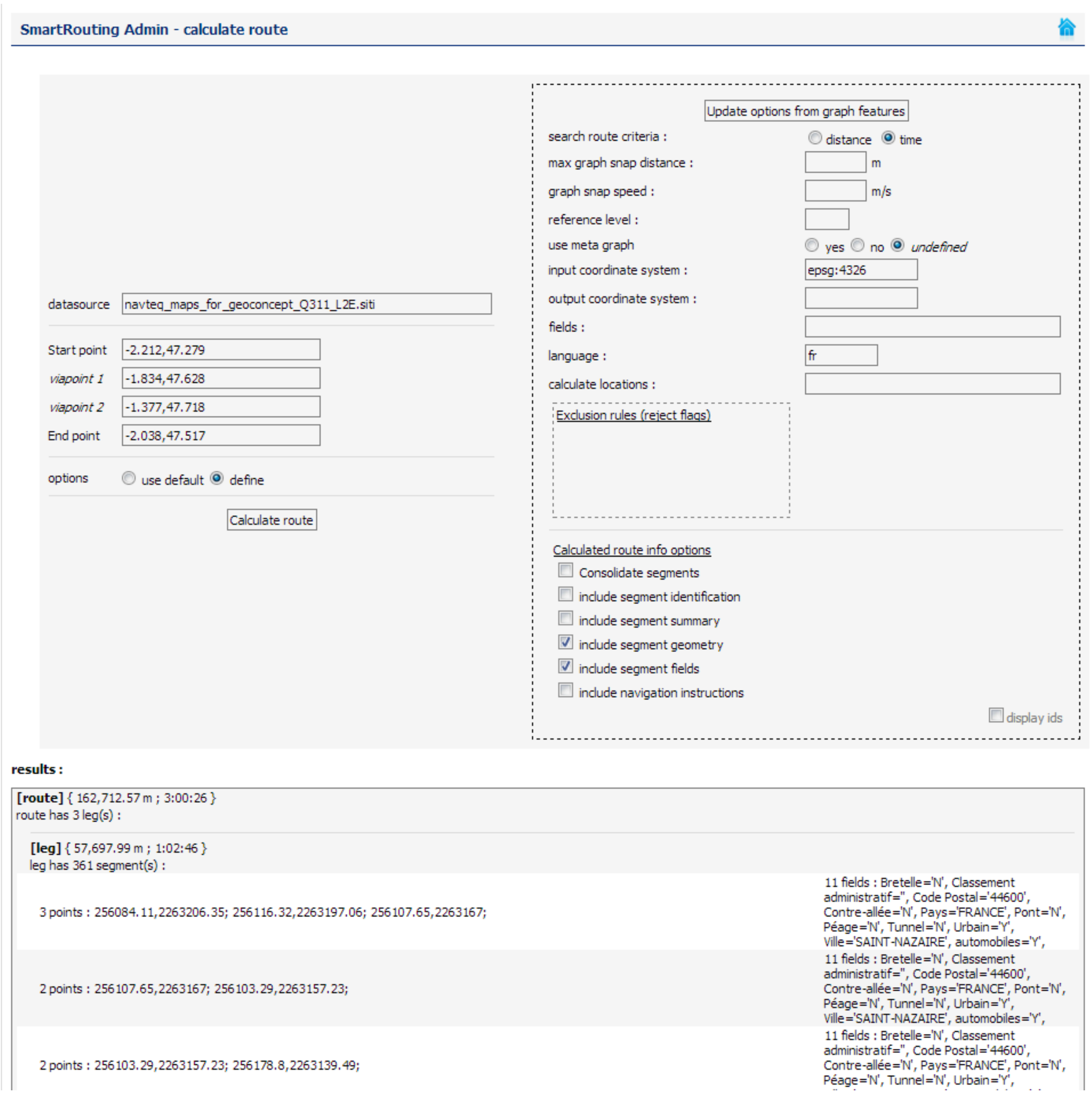
Calculate locations

This option allows you to find a point on the calculated route from the start point. The Calculate locations value in seconds allows you to find this point from the start point of the itinerary.

Consolidate segments

This option allows you to regroup, in the result of the route calculation, those segments that will bear the same name and which are close together in a single segment (useful for navigation).

Route example: input coordinates in WGS 84, output coordinates in Lambert 2 Extended (the graph projection system), all fields are returned in French.


SmartRouting Admin - calculate route

datasource	navteq_maps_for_geoconcept_Q311_L2E.siti
Start point	-2.212,47.279
viapoint 1	-1.834,47.628
viapoint 2	-1.377,47.718
End point	-2.038,47.517
options	use default [✓ define]

Calculate route

Update options from graph features

search route criteria :	distance [✓ time]
max graph snap distance :	m
graph snap speed :	m/s
reference level :	
use meta graph	yes no [✓ *undefined*]
input coordinate system :	epsg:4326
output coordinate system :	
fields :	
language :	fr
calculate locations :	

Exclusion rules (reject flags)

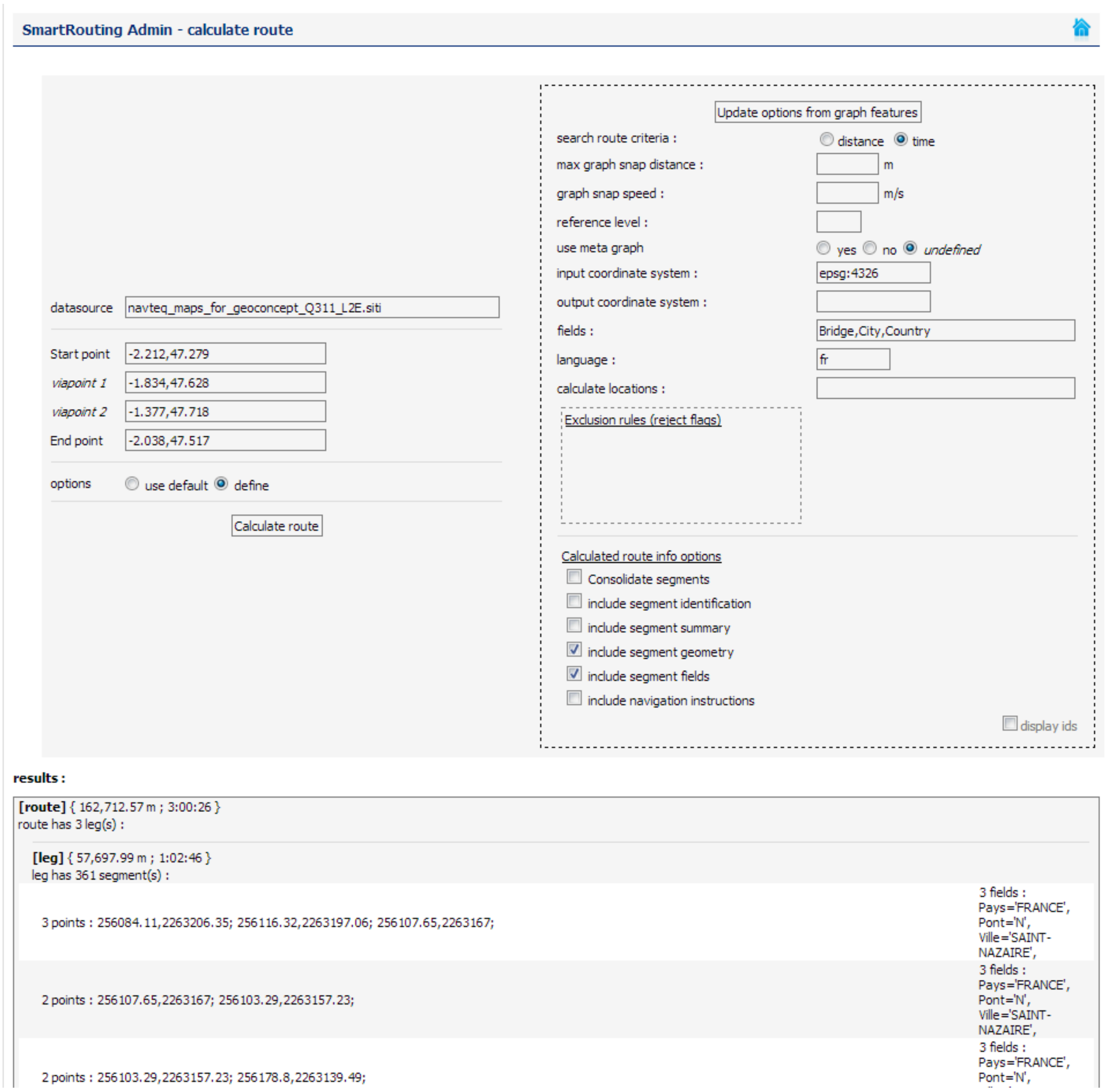
Calculated route info options
- ☐ Consolidate segments
- ☐ include segment identification
- ☐ include segment summary
- ☑ include segment geometry
- ☑ include segment fields
- ☐ include navigation instructions

☐ display ids

results :

```
[route] { 162,712.57 m ; 3:00:26 }
route has 3 leg(s) :

  [leg] { 57,697.99 m ; 1:02:46 }
  leg has 361 segment(s) :

    3 points : 256084.11,2263206.35; 256116.32,2263197.06; 256107.65,2263167;
        11 fields : Bretelle='N', Classement administratif='', Code Postal='44600',
        Contre-allée='N', Pays='FRANCE', Pont='N', Péage='N', Tunnel='N', Urbain='Y',
        Ville='SAINT-NAZAIRE', automobiles='Y',
    2 points : 256107.65,2263167; 256103.29,2263157.23;
        11 fields : Bretelle='N', Classement administratif='', Code Postal='44600',
        Contre-allée='N', Pays='FRANCE', Pont='N', Péage='N', Tunnel='N', Urbain='Y',
        Ville='SAINT-NAZAIRE', automobiles='Y',
    2 points : 256103.29,2263157.23; 256178.8,2263139.49;
        11 fields : Bretelle='N', Classement administratif='', Code Postal='44600',
        Contre-allée='N', Pays='FRANCE', Pont='N', Péage='N', Tunnel='N', Urbain='Y',
```

Example route: input coordinates in WGS 84, output coordinates in Lambert 2 Extended (the graph projection system), fields returned: Bridge, City, Country in French.



SmartRouting Admin - calculate route

datasource	navteq_maps_for_geoconcept_Q311_L2E.siti
Start point	-2.212,47.279
viapoint 1	-1.834,47.628
viapoint 2	-1.377,47.718
End point	-2.038,47.517
options	use default [✓ define]

Calculate route

Update options from graph features

search route criteria :	distance [✓ time]
max graph snap distance :	m
graph snap speed :	m/s
reference level :	
use meta graph	yes no [✓ *undefined*]
input coordinate system :	epsg:4326
output coordinate system :	
fields :	Bridge,City,Country
language :	fr
calculate locations :	

Exclusion rules (reject flags)

Calculated route info options

- ☐ Consolidate segments
- ☐ include segment identification
- ☐ include segment summary
- ☑ include segment geometry
- ☑ include segment fields
- ☐ include navigation instructions

☐ display ids

results :

[route] { 162,712.57 m ; 3:00:26 }
route has 3 leg(s) :

[leg] { 57,697.99 m ; 1:02:46 }
leg has 361 segment(s) :

3 points : 256084.11,2263206.35; 256116.32,2263197.06; 256107.65,2263167;	3 fields : Pays='FRANCE', Pont='N', Ville='SAINT-NAZAIRE',
2 points : 256107.65,2263167; 256103.29,2263157.23;	3 fields : Pays='FRANCE', Pont='N', Ville='SAINT-NAZAIRE',
2 points : 256103.29,2263157.23; 256178.8,2263139.49;	3 fields : Pays='FRANCE', Pont='N',

Advanced options

The `%smartrouting%/smartrouting/conf/service.xml` file allows you to specify the configuration to be used for all graphs.

You will find the configuration for the graphs you currently possess in the Datasources configuration page, by clicking on the info link. The utilisation or not of a metagraph in the graph is displayed at the bottom of the page.

SmartRouting Command Line

SmartRoutingCommandLine, or SRCmdLine is a standalone application inspired by the principle of UGCCmdLine, that allows calculation of a series of routes in a single operation.

Input data

The routes to be calculated are present in an input file (txt):

- One line per itinerary, with, in the following order:
  - X Start
  - Y Start
  - X Finish
  - Y Finish
  - TypeDeCalcul(“1” or “Shortest” for a calculation of the shortest route, “2” or “Fastest” for a calculation of the fastest route)
- Each of these values must be separated by a ‘ ;’ character
- The last value, TypeOfCalcul, is optional (if not specified, SRCmdLine takes the value specified in its config: see below)

Result file

The results of the route calculations are written to a result file:

- One line per route, with, in the following order:
  - The route length (distance) in metres,
  - The route travel time in seconds,
  - The route calculation time in milliseconds (the treatment time)
  - If it has not been possible to calculate any route, the values will be replaced by “N.A.”.
- Each of these values is separated by a “;”

Run the processing operation

SRCmdLine is a tool that runs in command line mode from the command line.

- To display the help facility: «SRCmdLine.exe –help»
- To display the product version number: « SRCmdLine.exe –version »
- Where possible, you can specify a series of parameters (some of which are mandatory, and others of which are optional) that will be used to configure SmartRouting, and therefore all route calculations in effect.
- The values of these parameters can EITHER be defined in a configuration file in XML format, OR defined directly in the command line.
- You should specify the config file in the following way: SRCmdLine –config <CheminCompletVersLeFichier>
- Here is the list of parameters that can be specified:

- Input file (mandatory):
  - Command line; -in <CheminCompletVersLeFichierDEntrée>
  - XML: <smartrouting-configuration><smartrouting-input><path>
- Output file (or results file) (mandatory):
  - Command line: -out <CheminCompletVersLeFichierDeSortie>
  - XML: <smartrouting-configuration><smartrouting-output><path>
- Graph (mandatory):
  - Command line: -graph <CheminCompletVersLeFichierGraphe>
  - XML: <smartrouting-configuration><smartrouting><connection-distance>
- Maximum search for snap-to distance (optional):
  - Command line: -connectionDistance <value>
  - XML: <smartrouting-configuration><smartrouting><connection-distance>
- Snap speed (optional):
  - Command line: -connectionSpeed <value>
  - XML: <smartrouting-configuration><smartrouting><connection-speed>
- Reference level (optional):
  - Command line: -refLevel <value>
  - XML: <smartrouting-configuration><smartrouting><reference-level>
- Type of route calculation (optional):
  - Command line: -itiMode <value>
  - XML: <smartrouting-configuration><smartrouting><iti-mode>
  - For a *shortest* calculation: value=1 or value=Shortest
  - For a *fastest* calculation: value=2 or value=Fastest
- Full loading of the graph (optional):
  - Command line: -fullLoadGraph <value>
  - XML: <smartrouting-configuration><smartrouting><full-load-graph>
  - Full loading: value=1
  - Partial loading: value=0
- Coordinates system as input (optional):
  - Command line: -inputProj <codeEPSG>
  - XML: <smartrouting-configuration><smartrouting><input-projection>

Example of a call to the programme:

- SRCmdLine -graph "data/graph.siti" -in "data/input.csv" -out "output.csv" -config "config.xml"
  - If a parameter is defined both in the config file and in the command line, the value specified in the command line will be taken.
  - If the type of route calculation (shortest or fastest) is defined both in the input file and in the config file, or the command line, the value specified in the input file will be the one taken as valid.

Example of config.xml configuration file

```
<?xml version="1.0" encoding="utf-8"?>
<smartrouting-configuration>
  <!-- Smart Routing configuration -->
  <smartrouting>
    <!-- Graph file path-->
    <!-- <graph-path>E:\SmartRouting-cmdline\SRCmdLine\navteq_maps_for_geoconcept_Q311_france_v2.siti</
graph-path> -->
        <graph-path>E:\SmartRouting-cmdline\graphe.siti</graph-path>
    <!-- Mode of routing. 1:Distance, 2:Time -->
    <iti-mode>1</iti-mode>
    <!-- Set connection distance -->
    <connection-distance>150.</connection-distance>
        <full-load-graph>1</full-load-graph>
        <reference-level>3</reference-level>
  </smartrouting>
  <!-- Input configuration (text file format) -->
  <smartrouting-input>
    <!-- Path of the input file. -->
    <path>E:\SmartRouting-cmdline\SRCmdLine\input.csv</path>
  </smartrouting-input>
  <!-- Ouput configuration (text file format) -->
  <smartrouting-output>
    <!-- Path of the output file.  -->
    <path>E:\SmartRouting-cmdline\SRCmdLine\output.csv</path>
  </smartrouting-output>
</smartrouting-configuration>
```