Geomajas Google layer plug-in

Geomajas Developers and Geosparc

Geomajas Google layer plug-in

by Geomajas Developers and Geosparc

2.0.0

Copyright © 2010-2012 Geosparc nv

Table of Contents

List of Figures

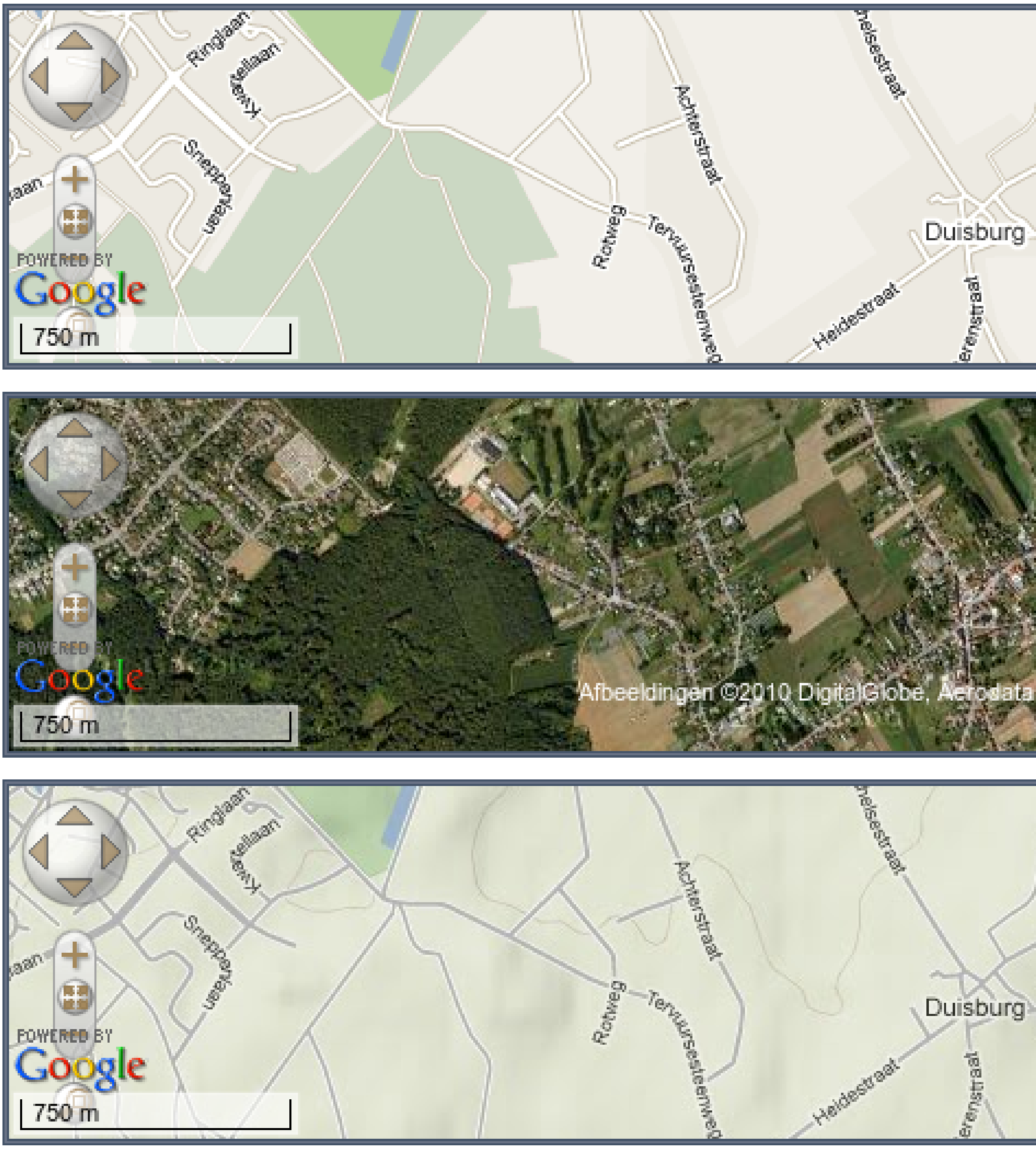
List of Tables

List of Examples

Chapter 1. Introduction

The Google layer allows you to use Google maps imagery as a raster layer in Geomajas. There are three possible types of images which can be displayed: the normal view displays street data, satellite view displays satellite images and physical view displays a hybrid of street and elevation information.

Figure 1.1. Google layer formats, normal, satellite and hybrid



Note

Use of the Google Maps images is subject to Googles terms of use. For details, check with Google, though the following URLs may help you http://code.google.com/intl/nl/apis/maps/terms.html and http://www.google.com/intl/en_ALL/help/terms_maps.html.

Chapter 2. Configuration

1. Dependencies

Make sure your include the plug-in in your project. If you are using Maven and the geomajas-dep dependency to manage versions, include following dependency to your pom:

Example 2.1. Google layer dependency, using geomajas-dep

```
<dependency>
    <groupId>org.geomajas.plugin</groupId>
    <artifactId>geomajas-layer-google</artifactId>
</dependency>
```

If you are not using geomajas-dep to manage versions, then you need to mention the version explicitly.

Example 2.2. Google layer dependency, explicit version

```
<dependency>
    <groupId>org.geomajas.plugin</groupId>
    <artifactId>geomajas-layer-google</artifactId>
    <version>2.0.0</version
</dependency>
```

If you want to use these as part of an application using the GWT face, then the dependencies change to:

Example 2.3. Google layer dependency for GWT, using geomajas-dep

```
<dependency>
    <groupId>org.geomajas.plugin</groupId>
    <artifactId>geomajas-layer-google-gwt</artifactId>
</dependency>
```

If you are not using geomajas-dep to manage versions, then you need to mention the version explicitly.

Example 2.4. Google layer dependency for GWT, explicit version

```
<dependency>
    <groupId>org.geomajas.plugin</groupId>
    <artifactId>geomajas-layer-google-gwt</artifactId>
    <version>2.0.0</version
</dependency>
```

2. Google raster layer

2.1. Base configuration

Warning

When using the Google layer, it is your responsibility to comply with the Google's terms of use. Some sources of information about these terms include http://code.google.com/apis/maps/, http://code.google.com/intl/nl/apis/maps/terms.html and http://www.google.com/intl/en_ALL/help/terms_maps.html. Some of the things you need to do include adding the Google

API code in your application (using a Google API key when not running on localhost), and (from the GWT face), using the `GoogleAddon` class to assure the copyright notices are displayed on the map.

A base Google layer configuration looks as follows:

Example 2.5. Simple Google layer configuration

```
<bean name="google" class="org.geomajas.layer.google.GoogleLayer">
    <property name="tilesEnabled" value="true" />
    <property name="apiKey" value="" />
</bean>
```

Warning

If you are using 1.7.1 or earlier of the OpenStreetMap plug-in then this configuration will not work. In that version we simplified configuration by not forcing you to configure values you are not allowed to change anyway. For the old version, the configuration would look like this:

Example 2.6. Google layer configuration

```
<bean name="layerGoogle" class="org.geomajas.layer.google.GoogleLayer" >
    <property name="layerInfo" ref="layerGoogleInfo" />
    <property name="satellite" value="false" />
    <property name="maxZoomLevel" value="21" />
</bean>

<bean name="layerGoogleInfo" class="org.geomajas.configuration.RasterLayerIn
    <property name="crs" value="EPSG:900913"/>
    <property name="maxExtent">
        <bean class="org.geomajas.geometry.Bbox">
            <!--
            see http://cfis.savagexi.com/2006/05/03/google-maps-deconstructe
            -20037508.342789, -20037508.342789 to 20037508.342789, 20037508.
            -->
            <property name="x" value="-20026376.393709917"/>
            <property name="y" value="-20026376.393709917"/>
            <property name="width" value="40052752.787419834"/>
            <property name="height" value="40052752.787419834"/>
        </bean>
    </property>
    <property name="tileWidth" value="256"/>
    <property name="tileHeight" value="256"/>
 </bean>
```

Note that these older versions do not allow all of the other configurations.

You can add some properties to define the kind of images which are produced. By default, the layer displays road information, but this can be changed to either satellite or physical (roads+elevation) imagery. The view style modes are exclusive, last set will be valid

Table 2.1. GoogleLayer image type configuration

GoogleLayer configuration	
satellite	Set to true to use satellite view from Google. When this and physical are false (the default), the normal view (showing streets) will be used. Setting to true will reset physical to false.

GoogleLayer configuration	
physical	Set to true to use physical view from Google. When this and satellite are false, the normal view (showing streets) will be used. Setting to true will reset satellite to false.
maxZoomLevel	Set to a number for which maps are available in the region of interest, defaults to 19. The first zoom level has one tile for the entire world, the second has four tiles etc.

These are used in the following configuration.

Example 2.7. Configure to use satellite images and

```
    increase max zoom level -->
<bean name="satellite" class="org.geomajas.layer.google.GoogleLayer">
    <property name="tilesEnabled" value="true" />
    <property name="satellite" value="true" />
    <property name="maxZoomLevel" value="21" />
    <property name="apiKey" value="" />
</bean>
```

Alternatively, you can also configure the display style using the dataSourceName property in the layer info. This is shown in the example below. However, it is preferred to use the properties on the layer itself. This can be useful for the dojo face, which uses this property to detect Google layers. Note that the suffix "@GoogleLayer" will automatically be added. Possible values are "G_NORMAL_MAP", "G_SATELLITE_MAP" and "G_PHYSICAL_MAP".

Example 2.8. Configure to use satellite images

```
    and increase max zoom level -->
<bean name="satelliteDs" class="org.geomajas.layer.google.GoogleLayer">
    <property name="tilesEnabled" value="true" />
    <property name="layerInfo">
        <bean class="org.geomajas.configuration.RasterLayerInfo">
            <property name="dataSourceName" value="G_SATELLITE_MAP" />
            <property name="crs" value="EPSG:900913" /> <!-- required property
            <property name="tileHeight" value="640" />
            <property name="tileWidth" value="640" />
        </bean>
    </property>
    <property name="apiKey" value="" />
</bean>
```

2.2. Image source configuration

You can also force the set of URLs to choose from, as in the following example. Note that this may fail when setting the image type. You can define the different options using ${level}, ${x} and ${y} for the zoom level and tile coordinate. A simple configuration looks like this:

Example 2.9. Using custom URLs

```
<bean name="googleSingle" class="org.geomajas.layer.google.GoogleLayer">
    <property name="tilesEnabled" value="true" />
    <property name="tileUrl"
        value="http://maps.googleapis.com/maps/api/staticmap?center=${center}&a
    <property name="apiKey" value="" />
</bean>
```

The Google layer allows configuration of the strategy to choose the server to provide the tiles. Each of the image types can be served from four servers. The default behaviour is to select the server using a round robin strategy.

The strategy can also be configured. For example, the sample below selects the class to use for the URL selection strategy (this needs to implement UrlSelectionStrategy).

Example 2.10. Using a custom URL selection

```
    strategy -->
<bean name="googleStrategy" class="org.geomajas.layer.google.GoogleLayer">
    <property name="apiKey" value="" />
</bean>
```

2.3. Zoom level configuration

By default the maximum zoom level is 19, but this can be modified using the maxZoomLevel property if your data source supports a different level.

Example 2.11. Google layer max zoom level

```
    configuration -->
<bean name="googleMaxLevel" class="org.geomajas.layer.google.GoogleLayer">
    <property name="tilesEnabled" value="true" />
    <property name="maxZoomLevel" value="12" />
    <property name="apiKey" value="" />
</bean>
```

3. Using the Google map add-on

To enable the display of copyright information on the map (to comply with Google's terms of use), you should use the GoogleAddon. This can be done using a line of code like:

Example 2.12. Using the GoogleAddon

```
map.registerMapAddon(
    new GoogleAddon("google", map, GoogleAddon.MapType.SATELLITE, false));
```

This needs to be called on the MapWidget object. The first parameter is the DOM id for the object, the second the map widget itself, the third the type of imagery which needs to be used and the last indicates whether the Google map should be made visible.

Note that you will need to add the Google API key in your html source file to assure this can work (otherwise you will probably get an exception from Google when not accessing the page on localhost.