Galigeo openMap REST API

Table of Contents

1 - Introduction

PUBLIC INTERESTED BY THIS GUIDE

This document is meant for the developers/integrators that are not specialized in webmapping, that wish to integrate maps in their applications (BI, CRM, ERP …), Enterprise portal of their enterprise, or their website.

The Galigeo API allows with just some lines of code, the integration of a webmapping APP out of the box, allowing crossing, visualizing and analysing on request static and/or dynamic data.

The objective is to dynamically visualize and to analyse the located BI indicators of the Enterprise portal on Galigeo maps.

The Galigeo OpenMap API is available in the form of a RESTful WebService. The exchanges between the client and the WebService are done in JSON format:

- Galigeo REST URL call with parameters and data, to visualize on the map, in JSON format according to the POST method
- Response of the REST service in JSON format. The response contains the Galigeo HTML5 viewer URL to be called from the web client of the Enterprise portal, from, for instance, an HTML iframe

PURPOSE OF THE DOCUMENT

This document presents:

- The detail of the Galigeo REST call:
    - The URL of the REST WebService call and example
    - Call method: POST
    - Parameters + data to be transferred, in JSON format and example
    - The REST WebService response in JSON format and example

- And in more detail, the JSON structure of a dataset. A dataset contains in a t moment, according to the user prompts for instance, the geolocalized BI indicators by a geographical dimension (e.g. FIPS code) and/or geographical coordinates (e.g. latitude/longitude).

2 - Prerequisites

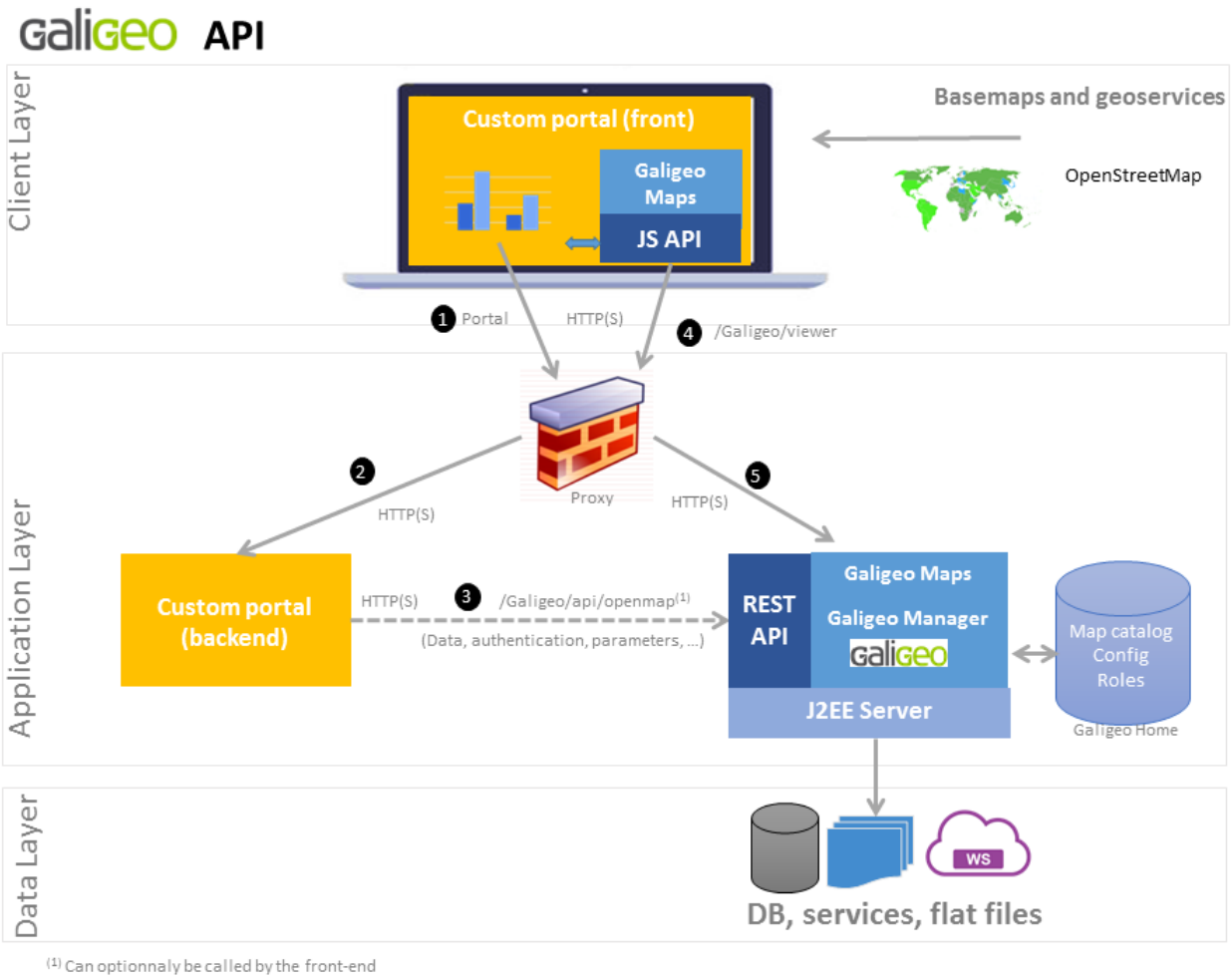
The Galigeo web application is deployed on its J2EE Tomcat/Java application server, and was initialized. The initialization operation creates the **<GALIGEO_HOME>** directory in which are stored the resources necessary for the Galigeo application functioning.

For more details on the technical prerequisites, we will refer to the installation guide.

Note in case of migration to G22 and greater

For those using Galigeo on premise, the migration to G22 will require to add an api key to the map parameters. You will find this key on the licence page of the Galigeo administration. If no API key is displayed, please ask us to get a new licence.

3 – Architecture schema



4 – REST openMap REST call detail

The Galigeo openMap API is available in the form of a RESTful WebService. The communication between the client and the WebService are done in JSON format:

- The REST URL call with the transfer of parameters and data to visualize on the map in JSON format according to the POST method
- The REST service response in JSON format. The response contains the GALIGEO HTML5 viewer URL to be called from the web client of the Enterprise portal, from an HTML iframe for instance.

THE REST OPENMAP CALL URL

The REST openMap call URL has the following format:

http://<TOMCAT-SERVER>:<TOMCAT-PORT>**/Galigeo/api/openMap**

Where <TOMCAT-SERVER> and <TOMCAT-PORT> are respectively the name (or IP) and the port of the Tomcat server that hosts the Galigeo webapp.

Example:
http://ggo-srv:8080**/Galigeo/api/openMap**

THE CALL METHOD

Only the POST method is supported.

THE STRUCTURE OF THE JSON TO BE POSTED

We are offering with the present document a complete example of a JSON to be posted:
samples/**POST-JSON-Sample.json**

The JSON to be posted from the REST openMap call has the following structure:

```
{
        "mapId":"<MAP-UNIQUE-ID>",
        "user":"<CURRENT-BI-USER>",
        "profile":"<CURRENT-BI-PROFILE>",
        "lang":"en_US",
        "reportName":"<BI-REPORT-NAME>",
        "reportId":"<BI-REPORT-UNIQUE-ID>",
        "data":[{<dataset-1>},{<dataset-2>},..]
}
```

Example:

```
{
        "mapId":"322A5787L2",
        "user":"Administrator",
        "lang":"en_US",
```

```
    "reportName":"171011 - Embedded LI Test",
    "reportId":"BAE31DFE04D249539E551102",
    "data":[{"fields": [{...},{...},...],"features": [{...},{...},...]}]
}
```


Parameter	Mandatory?	Description	Remarks
mapId	**yes**	Alphanumeric string representing the unique ID of the Galigeo map	• Once assigned, this ID cannot be changed • **No underscore characters**, nor special characters, nor spaces in the string • The number of digits is w/o importance. One must only ensure that each Galigeo map has an unique ID
user	**no**	The actual BI user consulting the map	• The Galigeo role (author, end user, etc.) assigned to this user is defined in the Galigeo Manager console • The Administrator built-in user has all the rights • If the "user" parameter is omitted, the access is considered as anonymous. The map is thus available only in read mode • If the "user" parameter is missing and a user is already authenticated in the current browser, its username will be used.
lang	**no**	fr_FR or en_US	If this parameter is omitted, the application is in English
profile	**non**	Profile to be used in order to load the map	• When a profile is defined, the user default profile is ignored • It is not possible to specify an administrator profile using this parameter • The parameter is activating when the key `apiAllowForceProfile=true` is present in the gaia.properties

Parameter	Mandatory?	Description	Remarks
reportName	**no**	The BI report name in which the map is inserted	• Enables later the transport of maps between development and production environments with the Galigeo transport tool, available from the Galigeo Manager • The transport tool can however work w/o this information
reportId	**no**	The BI report unique ID in which the map is inserted	Same remarks as above
data	**yes**	A Datasets table. Each Dataset is meant to be visualized on the map.	• Each Dataset can for instance represent a different geographical level. • See chapter "4 – Detail of the JSON structure of a dataset" for the detail of the structure of a DataSet.

THE STRUCTURE OF THE JSON RESPONSE

The different possible JSON responses are given in the following table:

Response OK / KO	JSON Response Example	Comments

OK	{
	"**status**": "200",
	"**url**": "<GGO-HTML5-WEB-CLIENT-URL>"
	}
		The "**url**" parameter represents the Galigeo HTML5 Viewer URL to be called from the web client of the Enterprise portal, from for instance an HTML iframe.

KO	{
	"**status**": "400",
	"**message**": "mapId not found or empty"
	}
		In this instance the mandatory parameter "mapId" is missing in the POST JSON.
		The display process of the map cannot continue.

KO	{
	"**status**": "400",
	"**message**": "data not found or empty"
	}
		In this instance the "**data**" attribute of the POST JSON was not sent.
		The display process of the map cannot continue.

Response OK / KO	JSON Response Example	Comments

KO	{
	"**status**": "500",
	"**message**": "A JSONObject text must begin with \'{\' at character 1"
	}
		In this instance, the posted JSON has a structure problem.

5 – Supported data formats

A Dataset allows the defining of the data that will be visualized on the map for a user at a certain instant.
The data can be specified in two formats : CSV and inline JSON data.

DATASET FROM A LINK (CSV FORMAT)

The Galigeo API supports incoming data using the CSV format. The CSV file must be available as a HTTP link visible from the Galigeo server.

JSON format :

```
{
  “format”: “link”,
  “url”: “link_to_csv_file”,
  “name”; “name of the dataset”
}
```

The CSV file specified by the attribute “url” will then be loaded in the map using the name defined by “name”.

DATASET USING THE INLINE JSON STRUCTURE

The JSON posted to the API can embed its own data using the JSON format (based on the Esri Geoservice JSON format). This section describes this JSON specification.

With the present document we give a complete example of a JSON to be posted: samples/**POST-JSON-Sample.json**

Here are the elements that have to appear in a Dataset:

The fields list with their alias:

```
"fields" : [
   {
    "name" : "NAME",
    "type" : "esriFieldTypeString",
    "alias" : "Name",
    "dimension": true
   },
   {
    "name" : "INDICATOR",
    "type" : "esriFieldTypeDouble",
    "alias" : "My indicator",
    "dimension": false
   }
 ]
```

- No special characters or spaces in the “name” attribute. It will be internally used by the application.
- The alphanumerical fields are always of esriFieldTypeString type; the numerical fields are always of esriFieldTypeDouble type.

- It is the "alias" attribute that is presented in the application user interface. It can contain special characters and spaces.
- The "dimension" attribute designates a dimension if it is set to true, and an indicator if it is set to false.

The features list, where each feature is a map defining the attribute values for each field:


```
"features" : [
   {
    "attributes" : {
     "NAME" : "COMM MANUFACTURING",
     "INDICATOR" : 432351.0
    }
   },
   {
    "attributes" : {
     "NAME" : "COMM SAFETY",
     "INDICATOR" : 503096.0
    }
   }
]
```

6 - REST openMap Call example with jQuery

We supply with the present document an example of the source code of the REST openMap call with jQuery: samples/**Code-Sample-jquery-rest-OpenMap.jsp**
If the Galigeo web application is deployed, this example is available at the following URL:
http://<TOMCAT-SERVER>:<TOMCAT-PORT>**/Galigeo/viewer/api/sample.html?url=sampleRestServer.jsp**

7 – Other useful REST calls

Galigeo is based on REST full services and provides various interfaces to manage the product from the API. This section details some useful calls.

REST URL	Parameters	Result	Description

/Galigeo/feature/portal/login	POST x-www-form-urlencoded
	
	Form parameters:
	userName: {username}
	password: {password}
		{
		"userName": USERNAME",
		"role": "author",
		"token": "JWT_TOKEN",
		"lang": "fr",
		"capabilities": [...],
		"productName": "Galigeo API"
		}
			Login a specific user to the application.
			The response provides a session cookie called GALIGEO_TOKEN. Further calls to the API must provide this cookie certify a valid authentication

/Galigeo/feature/portal/login	GET	Same as above	Calling /login with the GET method is used to check if a user is authenticated or nor

/Galigeo/feature/portal/documents	GET	[
		{
		"docId": "DOC_ID",
		"docName": "DOC_NAME",
		"isPrivate": false,
		"isShared": false,
		"application": {
		DOCUMENT METADATA
		}
		},
		{
		...
		}
		]
			List all the maps that have been created.
			Requires to be authenticated.

/Galigeo/feature/portal/duplicateDocuments	GET
	docId={docId}&docName={docName}
		{
		"docId": "NEW_DOC_ID",
		"docName": "NEW_DOC_NAME",
		Etc…
		}
			Duplicates an existing document specified by {docId} and call it {docName}.
			Requires to be authenticated with the capability "author".
			Note: {docName} is optional

8 – Javascript API

The Galigeo API exposes some Javascript components used to interact with the map.

First steps with the Javascript API

1. Import the Javascript API

```
<script type="text/javascript" src="/Galigeo/viewer/api/js/galigeo-api-0.1.js"></script>
```

2. Add a div on your HTML page to hold the Galigeo Map

```
<div id="ggoMapId" style="position:absolute; height:100%;width:100%"></div>
```

3. In Javascript, create an instance of Galigeo.Map. This is the based object to start interacting with Galigeo

```
var ggoMap = new Galigeo.Map('ggoMapId',
{
      id: 'sampleFilterCSV',
      name: 'FrCities',
      url: location,
      data: []
});
ggoMap.load()
```

Javascript Documentation

The Javascript documentation is available from this link:
https://doc.galigeo.com/G23_0/api/

Going further

This documentation is still incomplete (we are working to improve it!). However, you can access our samples page at this url:

http://<TOMCAT-SERVER>:<TOMCAT-PORT>**/Galigeo/viewer/api**