

Galigeo Open Map REST API

Table de matières

Table de matières	2
1 - Introduction	3
2 - Prérequis	4
3 – Architecture schema	5
4- Détail de l'appel REST openMap	6
5 – Format de données supporté	10
6 - Exemple d'appel REST openMap avec jQuery	12
7 – Autres appels REST utiles	12
8 – Javascript API	13
Premières étapes avec l'API Javascript	13
Aller plus loin.....	13

1 - Introduction

PUBLIC CONCERNE PAR CE GUIDE

Ce document est destiné aux développeurs, non spécialiste en webmapping, souhaitant intégrer des cartes dans leurs applications (BI, CRM, ERP,...), portail d'entreprise ou site web.

L'API Galigeo permet en quelques lignes de code, d'intégrer une App de webmapping out of the box, permettant de croiser, visualiser et analyser à la demande des données statiques et/ou dynamiques.

L'objectif est de visualiser et d'analyser dynamiquement des indicateurs BI localisés du portail d'Entreprise sur les cartes Galigeo.

L'API OpenMap Galigeo est disponible sous la forme d'un Webservice RESTful. Les échanges entre client et Webservice se font au format JSON:

- Appel de L'URL REST Galigeo avec passage des paramètres et données à visualiser sur la carte au format JSON selon la méthode POST
- Réponse du service REST au format JSON. La réponse contient l'URL du viewer Galigeo HTML5 à appeler depuis le client web du portail d'Entreprise, depuis par exemple une iframe HTML
- Several API samples are available at this url : /Galigeo/viewer/api

OBJECTIF DU DOCUMENT

Ce document présente :

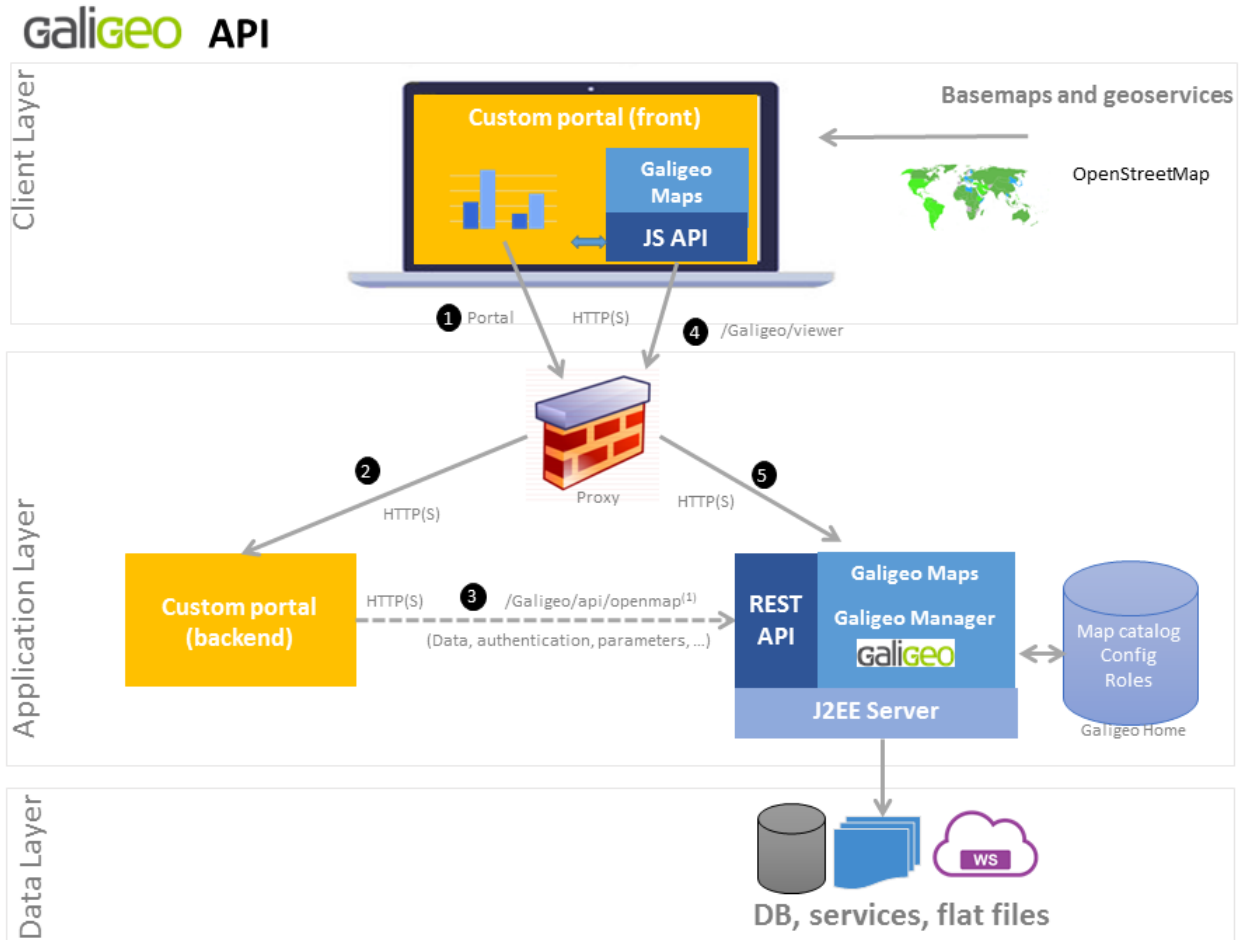
- Le détail de l'appel REST Galigeo:
 - URL d'appel du Webservice REST et exemple
 - Méthode d'appel: POST
 - Paramètres + données à passer, au format JSON et exemple
 - La réponse du Webservice REST au format JSON et exemple
- Et plus en détail, la structure JSON d'un dataset. Un dataset contient, à un instant t en fonction d'invites utilisateur par exemple, les indicateurs BI géolocalisés par une dimension géographique (ex: CODE_INSEE) et/ou des coordonnées géographiques (ex: latitude / longitude).

2 - Prérequis

L'application web Galigeo est déployée dans son serveur d'applications J2EE Tomcat/Java et a été initialisée. L'opération d'initialisation crée le répertoire **<GALIGEO_HOME>** dans lequel sont stockées les ressources nécessaires au fonctionnement de l'application Galigeo.

Pour plus de détails sur les pré-requis techniques, on se réfèrera au guide d'installation.

3 – Architecture schema



⁽¹⁾ Can optionally be called by the front-end

4- Détail de l'appel REST openMap

L'API OpenMap Galigeo est disponible sous la forme d'un Webservice RESTful. La communication entre client et Webservice se font au format JSON:

- Appel de L'URL REST openMap avec passage des paramètres et données à visualiser sur la carte au format JSON selon la méthode POST
- Réponse du service REST au format JSON. La réponse contient l'URL du viewer Galigeo HTML5 à appeler depuis le client web du portail d'Entreprise, depuis par exemple une iframe HTML.

URL D'APPEL REST OPENMAP

L'URL d'appel REST OpenMap à la forme suivante:

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

Où <TOMCAT-SERVER> et <TOMCAT-PORT> sont respectivement le nom (ou IP) et le port du serveur Tomcat hébergeant la webapp Galigeo.

Exemple:

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

LA METHODE D'APPEL

Seule la méthode **POST** est supportée.

LA STRUCTURE DU JSON A POSTER

On fournit avec le présent document un exemple complet de JSON à poster: **samples/POST-JSON-Sample.json**

Le JSON à poster lors de l'appel REST openMap a la structure suivante:

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

Exemple:

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

```

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

```

Paramètre	Obligatoire ?	Description	Remarques
mapId	oui	Chaîne alphanumérique représentant l'ID unique de la carte Galigeo	<ul style="list-style-type: none"> • Une fois assigné, cet ID ne peut pas changer • Pas de caractère underscore, ni de caractères spéciaux, ni d'espace dans la chaîne • Le nombre de digit est sans importance. On doit juste s'assurer que chaque carte Galigeo a un ID unique
user	non	L'utilisateur BI courant consultant la carte	<ul style="list-style-type: none"> • Le rôle Galigeo (author, end user, etc.) assigné à cet utilisateur est défini dans la console Galigeo Manager • L'utilisateur "built-in" Administrator a tous les droits • Si le paramètre "user" est omis, l'accès est considéré comme anonyme. La carte n'est alors accessible qu'en consultation • Si le paramètre « user » est omis et un utilisateur est déjà authentifié dans le navigateur actuel, son nom d'utilisateur sera utilisé.
profile	non	Profile à utiliser pour charger la carte	<p>Lorsqu'un profile est défini, le profile par défaut de l'utilisateur est ignoré</p> <p>Il n'est pas possible de spécifier un profil administrateur à travers ce paramètre</p> <p>Le paramètre s'active lorsque la clé <code>apiAllowForceProfile=true</code> est présente dans <code>gaia.properties</code></p>
lang	non	fr_FR ou en_US	Si le paramètre est omis, l'application est en anglais

Paramètre	Obligatoire ?	Description	Remarques
reportName	non	Le nom du rapport BI dans lequel la carte est insérée	<ul style="list-style-type: none"> Facilite ultérieurement le transport des cartes entre environnements de développement et de production avec l'outil de transport Galigeo, accessible depuis Galigeo Manager L'outil de transport sait toutefois travailler sans cette informations
reportId	non	l'ID unique du rapport BI dans lequel la carte est insérée	Mêmes remarques que ci-dessus
data	oui	un tableau de DataSets. Chaque DataSet est destiné à être visualiser sur la carte.	<ul style="list-style-type: none"> Chaque DataSet peut par exemple représenter un niveau géographique différent. Voir chapitre "4 - Détail de la structure JSON d'un dataset" pour le détail de la structure d'un DataSet

LA STRUCTURE DE LA REPONSE JSON

Les différents réponses JSON possibles sont données dans le tableau suivant:

Réponse OK / KO	Exemple de réponse JSON	Commentaires
OK	<pre>{ "status": "200", "url": "<GGO-HTML5-WEB-CLIENT-URL>" }</pre>	le paramètre " url " rprésente l'URL du viewer Galigeo HTML5 à appeler depuis le client web du portail d'Entreprise, depuis par exemple une iframe HTML.
KO	<pre>{ "status": "400", "message": "mapId not found or empty" }</pre>	<p>Dans ce cas de figure le paramètre obligatoire "mapId" est manquant dans le POST JSON.</p> <p>Le processus d'affichage de la carte ne peut continuer.</p>

Réponse OK / KO	Exemple de réponse JSON	Commentaires
KO	<pre>{ "status": "400", "message": "data not found or empty" }</pre>	<p>Dans ce cas de figure les données, attribut "data" du POST JSON, n'ont pas été transmises .</p> <p>Le processus d'affichage de la carte ne peut continuer.</p>
KO	<pre>{ "status": "500", "message": "A JSONObject text must begin with '{' at character 1" }</pre>	<p>Dans cet exemple, le JSON qui a été posté a un problème de structure.</p>

5 – Format de données supporté

Un DataSet permet de définir la donnée qui sera visualisée dans la carte pour un utilisateur donné à un instant donné.

The data can be specified in two formats : CSV and inline JSON data.

DATASET PROVENANT D'UN LIEN (FORMAT CSV)

L'API Galigeo supporte données d'entrée utilisant le format CSV. Le fichier CSV doit être disponible tel que lien http visible depuis le serveur Galigeo.

Format JSON :

```
{
  "format": "link",
  "url": "link_to_csv_file",
  "name": "name of the dataset"
}
```

Le fichier CSV spécifié par l'attribut « url » sera alors chargé dans la carte utilisant le nom défini par « name ».

DATASET UTILISANT LA STRUCTURE INLINE JSON

Le JSON posté à l'API peut insérer les propres données utilisant le format JSON (reposé sur le format ESRI Geoservice JSON). Cette section décrit cette spécification JSON.

On fournit avec le présent document un exemple complet de JSON à poster: [samples/POST-JSON-Sample.json](#)

Voici les éléments qui doivent figurer dans un DataSet:

La liste des champs avec leurs alias:

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

- Pas de caractères spéciaux ni d'espace dans l'attribut "name". Il est à usage interne à l'application.
- Les champs alphanumériques sont toujours de type esriFieldTypeString ; les champs numériques sont toujours de type esriFieldTypeDouble.
- C'est l'attribut "alias" qui est présenté dans l'interface utilisateur de l'application. Il peut contenir des caractères spéciaux et des espaces.

- L'attribut "dimension" désigne une dimension s'il est à true, un indicateur s'il est à false.

La liste des features ou chaque feature est une map définissant les valeurs d'attributs pour chaque champs:

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

A TITRE D'INFORMATION

Ce fichier est au format Esri Geoservice et correspond à la réponse JSON de l'opération query sur Arcgis Server.

Voici un exemple de fichier:

http://sampleserver1.arcgisonline.com/ArcGIS/rest/services/Louisville/LOJIC_LandRecords_Louisville/MapServer/2/query?text=&geometry=&geometryType=esriGeometryPoint&inSR=&spatialRel=esriSpatialRelIntersects&relationParam=&objectIds=&where=ZONING_CODE%3D%27CM%27&time=&returnCountOnly=false&returnIdsOnly=false&returnGeometry=false&maxAllowableOffset=&outSR=&outFields=*&f=pjson

6 - Exemple d'appel REST openMap avec jQuery

On fournit avec le présent document un exemple de code source d'appel REST openMap avec jQuery: **samples/Code-Sample-jquery-rest-OpenMap.jsp**

Si l'application web Galigeo est déployée, cet exemple est accessible à l'URL suivante:

<http://<TOMCAT-SERVER>:<TOMCAT-PORT>/Galigeo/viewer/api/sample.html?url=sampleRestServer.jsp>

7 – Autres appels REST utiles

Galigeo se base sur les services complets REST et fournit des interfaces variées pour gérer le produit de l'API. Cette sélection détaille quelques appels utiles.

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" }	Se connecter comme un utilisateur spécifique à l'application. La réponse fournit un cookie de session appelé GALIGEO_TOKEN. Des appels suivants à l'API doivent fournir ce cookie pour certifier une authentification valide
/Galigeo/feature/portal/login	GET	Comme ci-dessus	Appeler/Se connecter avec la méthode GET, est utilisée pour vérifier si un utilisateur est authentifié ou pas
/Galigeo/feature/portal/documents	GET	[{ "docId": "DOC_ID", "docName": "DOC_NAME", "isPrivate": false, "isShared": false, "application": { DOCUMENT METADATA } }, { ... }]	Liste toutes les cartes qui ont été créées. Nécessite d'être authentifié.
/Galigeo/feature/portal/duplicateDocuments	GET docId={docId}&docName={docName}	{ "docId": "NEW_DOC_ID", "docName": "NEW_DOC_NAME", Etc... }	Duplique un document existant spécifié par {docId} et s'appelle {docName}. Nécessite d'être authentifié avec la capacité « author ». Note: {docName} est facultatif

8 – Javascript API

L'API Galigeo expose quelques composants Javascript utilisés pour dialoguer avec la carte.

Premières étapes avec l'API Javascript

1. Importer l'API Javascript

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

2. Ajouter un div sur votre page HTML pour héberger la carte Galigeo

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

3. In Javascript, créer une instance de la carte Galigeo. C'est l'objet basé pour commencer l'interaction avec Galigeo

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

Documentation Javascript

La documentation des fonctions Javascript est disponible depuis ce lien :

https://doc.galigeo.com/G22_0/api/

Aller plus loin

Cette documentation est encore incomplète (nous essayons à l'améliorer !). En tout cas, vous pouvez accéder nos exemples a cet URL :

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