Galigeo openMap REST API

Table des matières

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'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

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 - Pré-requis

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 - Détail de l'appel REST openMap

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 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>",
        "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	• 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** (mais recommandé)	L'utilisateur BI courant consultant la carte	• 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
lang	**non**	fr_FR ou en_US	Si le paramètres est omis, l'application est en anglais
reportName	**non**	Le nom du rapport BI dans lequel la carte est insérée	• 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.	• 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	{
	"**status**": "200",
	"**url**": "<GGO-HTML5-WEB-CLIENT-URL>"
	}
		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	{
	"**status**": "400",
	"**message**": "mapId not found or empty"
	}
		Dans ce cas de figure le paramètre obligatoire "mapId" est manquant dans le POST JSON.
		Le processus d'affichage de la carte ne peut continuer.

KO	{
	"**status**": "400",
	"**message**": "data not found or empty"
	}
		Dans ce cas de figure les données, attribut "**data**" du POST JSON, n'ont pas été transmises .
		Le processus d'affichage de la carte ne peut continuer.

KO	{
	"**status**": "500",
	"**message**": "A JSONObject text must begin with \'{\' at character 1"
	}
		Dans cet exemple, le JSON qui a été posté a un problème de structure.

4 - Détail de la structure JSON d'un DataSet

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

STRUCTURE JSON D'UN DATASET

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
    }
   }
]
```

5 - 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/jsp/openMapRestTest.jsp**

6 - Exemple d'appel REST openMap avec ApiJs

On peut en utilisant l'api Javascript pour initier, agir et manipuler les événements dans la carte du l'extérieur de l'IFRAME, où GALIGEO et l'application appelante sont sur le même domaine

a- Ajouter une carte

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

```
<div id="mapId" ></div>

      var mapGaligeo = new Galigeo.Map('mapId',
        {
          id: 'UniqueIdOfMyMap',
          name: 'Name of my map',
          url: 'http://exemple.com/Galigeo',
          user: 'UserName',
          lang: 'fr or en or ...'
        });
      mapGaligeo.load(function () {
        // do something after map load
      }.bind(this), function () {
       // failer to load
      });
```

b- Ajout des donner avant le LOAD

```
mapGaligeo.addDataUrl('http://exemple.com/data_filter.json', 'DataName');
```

c- Ecouter un événement sur la Carte

```
mapGaligeo.load(function () {
        mapGaligeo.on('click', function (event) {
          console.log(event);
        }.bind(this));
      }.bind(this), function () {
       // failer to load
```

```
      });
```

d- Ecouter un événement sur une couche

```
mapGaligeo.load(function () {
        mapGaligeo.on('layer:select', function (event) {
         console.log(event);
        });
      }.bind(this), function () {
       // failer to load
      });
```

L'api existe dans le WAR de GALIGEO : viewer\api\js\galigeo-api-Version.js

Si l'application web Galigeo est déployée, des exemples sont accessible à l'URL suivante :
http://<TOMCAT-SERVER>:<TOMCAT-PORT>/Galigeo/viewer/api/