

Prise en main rapide

L'API Géoportail est un outil permettant d'afficher des cartes dynamiques personnalisées sur votre propre site Web.

Pour prendre en main rapidement l'API Géoportail, suivez les conseils ci- après.



[Premiers pas](#) | [Carte simple et points d'intérêts](#) | [Aller plus loin](#)

* Premiers pas

Pré-requis



Pour mettre en place l'API Géoportail sur votre site, les pré-requis sont :

- une clé d'utilisation délivrée sur le site <http://api.ign.fr>
 - un hébergement pour vos pages web OU un serveur web local (partage web personnel sur Mac, WAMPServer, EasyPHP ou autre sur PC...)
 - quelques notions de HTML
- une connaissance de JavaScript est souhaitable, mais pas indispensable.

Obtenir une clé d'utilisation (créer un contrat)



Etape 1 : Créer un compte

Renseignez les informations qui vous sont demandées notamment l'adresse mail et l'identifiant qui vous serviront à activer votre compte.

Etape 2 : Valider votre compte

Cliquez sur le lien présent dans le message pour activer votre compte.

Etape 3 : Créer un contrat gratuit

Cliquez sur « Créer un contrat » puis cliquez sur « Créez un contrat gratuit en ligne » :

Renseigner avec soin le champ URL lors de la création de votre contrat puisque ce champ déterminera l'URL associé à la validité de votre clé d'utilisation.

A savoir : Votre clé sera valable pour l'URL saisie et pour un serveur web local (http://localhost/...). Il est donc inutile de créer un autre contrat pour une adresse "localhost" si vous avez déjà une clé pour une URL du type : http://monsite.com

Étape 4 : Validation et exemple de page

Une fois les conditions d'utilisation acceptées, votre contrat est créé et vous pouvez afficher votre première carte en copiant et collant le code source fourni par l'IGN un fichier html.

Nommez par exemple ce nouveau fichier « carte.html ». Placez ce fichier de manière à le rendre accessible sur votre serveur local ou chez votre hébergeur de page web.

Important : pour que l'affichage des cartes soit possible, le page html contenant l'appel à l'API Géoportail doit être appelée via un serveur local ou un hébergeur.

Si la page html n'est pas accédée depuis un serveur, la carte ne s'affichera pas.

* Carte simple et points d'intérêts



Carte centrée sur une commune

Cette première carte est centrée sur la commune de *Pougne-Hérisson* (0.4 O, 46.65 N).

La seule ligne de code ajoutée à l'exemple de base est la suivante :

```
VISU.getMap().setCenterAtLonLat(-.4 , 46.65, 12); //lon, lat, zoom
```

Elle permet de centrer la carte sur le point de longitude 0.4° Ouest et de latitude 46.65° Nord à l'échelle 12 (valeur comprise entre 4 et 17).

> [Voir l'exemple : carte centrée sur une commune](#)

Ajouter un point d'intérêt et une info bulle



Dans cet exemple, le point d'intérêt (dans la suite du texte, abrégé par POI : point of interest) se compose : d'une nouvelle couche de données, d'un point et d'une info bulle.

Etape 1 : Création d'un point et transformation dans la projection Géoportail

```
// Creation d'un point
var mon_point = new OpenLayers.Geometry.Point(-0.4, 46.65);

// Passage en projection Géoportail
var sourceproj = OpenLayers.Projection.CRS84; // WGS84
var destproj = VISU.projection; // projection Geoportail
mon_point.transform(sourceproj, destproj);
```

Etape 2 : Création de l'objet qui représentera le POI selon un style prédéfini

```
// Création du POI à partir du point
var stylepoi =
{ externalGraphic:'http://api.ign.fr/geoportail/api/js/1.0beta4/theme/geoportail/img/picto_zoom
ToMapExtent_off.gif', graphicWidth:50, graphicHeight:50 };
var mon_poi = new OpenLayers.Feature.Vector(mon_point, null, stylepoi);
```

Etape 3 : Création de la couche de données sur laquelle sera représenté le POI

```
// Ajout d'une couche d'objets vectoriels
var vector_layer = new OpenLayers.Layer.Vector('Mes POIs');
```

```
// Ajout du POI à la couche créée et affichage de la couche
vector_layer.addFeatures([mon_poi]);
// Ajout de la couche à la carte
VISU.getMap().addLayer(vector_layer);
```

Etape 4 : Associer une info bulle au POI

```
// Ajout de l'info bulle
var popup = new OpenLayers.Popup('id',
    new OpenLayers.LonLat(mon_point.x, mon_point.y),
    new OpenLayers.Size(200,50),
    'Ma Première info bulle',
    true
);

// Ajout de l'info bulle
VISU.getMap().addPopup(popup, true);
```

> [Voir l'exemple : afficher un POI et une info bulle](#)

*** Aller plus loin**

Ajouter un fichier KML ou GPX

Il est très facile d'ajouter des données aux formats KML ou GPX sur votre carte en plus des données fournies par l'IGN :

L'utilisation de la fonction addLayer permet d'ajouter des données dans la plupart des formats standard d'internet tels que le KML, le GPX, le WMS, WFS.

```
VISU.getMap().addLayer( "KML",
    "POIs", "mes_poi.kml",
    {
        minZoomLevel:0,
        maxZoomLevel:20
    }
);
```

Attention : Il est important de noter que dans le cas où le fichier de données à ajouter n'est pas hébergé sur le même serveur que la page contenant l'appel à l'API Geoportail, il est nécessaire d'utiliser un proxy et de le déclarer grâce au code suivant :

```
VISU.getMap().setProxyUrl('http://localhost/proxy.php?url=');
```

Si vous ne savez pas ce qu'est un proxy ni comment en créer un vous pouvez vous reporter à la [documentation](#) du site de l'API Geoportail.

Vous pouvez également utiliser le code suivant qui constitue un proxy programmé en PHP (l'installation de la librairie CURL est nécessaire) :

```
<?php
// Get the REST call path from the AJAX application
// Is it a POST or a GET?
$url = ($_POST['url']) ? $_POST['url'] : $_GET['url'];

// Open the Curl session
$session = curl_init($url);

// If it's a POST, put the POST data in the body
if ($_POST['url']) {
    $postvars = "";
    while ($element = current($_POST)) {
        $postvars .= key($_POST).'='.$element.'&';
        next($_POST);
    }
    curl_setopt($session, CURLOPT_POST, true);
    curl_setopt($session, CURLOPT_POSTFIELDS, $postvars);
}

// Don't return HTTP headers. Do return the contents of the call
curl_setopt($session, CURLOPT_HEADER, false);
curl_setopt($session, CURLOPT_RETURNTRANSFER, true);

//test
curl_setopt($session, CURLOPT_HTTPPROXYTUNNEL, 1);
// curl_setopt($session, CURLOPT_PROXY, 'http://www.monserveur.com/monproxy.pac');
curl_setopt($session, CURLOPT_PROXY, 'monproxy.monserveur.com:3128');
//fin

// Make the call
$xml = curl_exec($session);

// The web service returns XML. Set the Content-Type appropriately
header("Content-Type: text/xml");

echo $xml;
curl_close($session);
?>
```

> [Voir l'exemple : ajouter un fichier KML ou GPX](#)

Utiliser les événements : obtenir les coordonnées de la souris



L'API Geoportail permet d'utiliser un certain nombre d'événements tels que clic, survol, déplacement, etc.

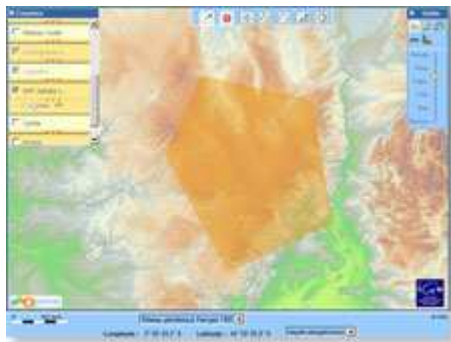
Récupérer les coordonnées de la souris

```
VISU.getMap().events.register("click",  
    VISU.getMap(), function(e)  
    { // A tout moment on récupère les coordonnées de la souris  
      var pos = VISU.getMap().getLonLatFromViewPortPx(e.xy);  
      alert(pos);  
    }  
);
```

Les coordonnées sont données dans la projection de la carte (projection Geoportail par défaut).

> [Voir l'exemple : obtenir les coordonnées de la souris](#)

Utiliser les contrôles prédéfinis : édition (dessin), mesure, etc.



L'API Geoportail permet d'intégrer très simplement des outils prédéfinis tels que l'outil d'affichage des couches, les outils de navigation, zoom et des outils de saisie de données vectorielles ou encore des outils de mesure.

Les contrôles d'édition de sélection et de modification sont associés à une couche de type « OpenLayers.Layer.Vector ».

```
// Ajout des outils d'édition d'objet vectoriel
// Ajout d'une couche "dessin"
var vector_layer = new OpenLayers.Layer.Vector('Dessin');
VISU.getMap().addLayer(vector_layer);

// Ajout des outils de saisie
var editing_tool= new OpenLayers.Control.EditingToolbar (vector_layer)

// Ajout de l'outil de déplacement
var drag_feature = new OpenLayers.Control.DragFeature(vector_layer);
editing_tool.addControls(drag_feature);

// Ajout de l'outil de modification
modify_feature = new OpenLayers.Control.ModifyFeature(vector_layer)
editing_tool.addControls(modify_feature);

// Ajout de l'outil de selection
selectControl = new OpenLayers.Control.SelectFeature(vector_layer);
editing_tool.addControls(selectControl);

// Ajout du panneau de contrôle à la carte
VISU.getMap().addControl(editing_tool);
```

Il est peut être utile d'ajouter quelques lignes de définition du style (CSS) du contrôle :

```
#GeoportalVisuDiv .olControlEditingToolbar
{
background-color:#99ccff;
border:1px solid white;
right:210px;
width:250px;
}
```

De même on définit le style des boutons. Ci-dessous un exemple pour le contrôle de modification d'objet

```
.olControlEditingToolbar .olControlModifyFeatureItemActive
{
background-image:
url("http://www.openlayers.org/dev/theme/default/img/overview_replacement.gif");
```

```
background-repeat: no-repeat;
background-color:#f1c02a;
background-position:center;
}
.olControlEditingToolbar .olControlModifyFeatureItemInactive
{
background-image: url
("http://www.openlayers.org/dev/theme/default/img/overview_replacement.gif");
background-repeat: no-repeat;
background-color:#e9e8e3;
background-position:center;
border: 1px solid white;
}
```

Le comportement de chacun des contrôles ajoutés ici est le comportement par défaut. Il est modifiable et paramétrable suivant vos besoins particuliers.

> [Voir l'exemple : utiliser les outils prédéfinis](#)