

Workshop: Google Map

Objectif :

Le but de ce workshop est de configurer l'API de Google Map.

Avant de commencer:

Supposons qu'on travaille sur un projet qui s'appelle **GoogleMaps**, le namespace **my** et sur un bundle **MapBundle**.

Une API est une spécification utilisée par des composants logiciels et de communiquer les uns avec les autres.

API = Application d'Interface de Programmation

Etapes :

1. Définir la route :

```
my_map_homepage:  
    path:      /map  
    defaults: { _controller: MyMapBundle:Default:index }
```

2. Modifier l'action index sous le Controller :

```

<?php

namespace My\MapBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;

class DefaultController extends Controller {

    public function indexAction() {
        $name = 'name';
        return $this->render('MyMapBundle:Default:index.html.twig', array('name' => $name));
    }

}

```

3. Modifier la page index.html.twig sous le répertoire MapBundle /ressource/views/Default :

```

{% extends "myMapBundle:Default:base.html.twig" %}
{% block container %}

{% endblock %}

```

4. Créer une nouvelle page base.html.twig sous le répertoire MapBundle /ressource/views/Default :

```

<html>
<head>
<style type="text/css">html, body, #map-canvas { height: 100%; margin: 0; padding: 0;}</style>
<script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?key=API_KEY">
</script>
<script type="text/javascript">
function initialize()
{
    var mapOptions = {
        center: {lat: -34.397, lng: 150.644},
        zoom: 8
    };
    var map = new google.maps.Map(document.getElementById('map-canvas'),
        mapOptions);
    var myLatLng = new google.maps.LatLng(-25.363882, 131.044922);
    var mapOptions = {
        zoom: 4,
        center: myLatLng
    }
    var map = new google.maps.Map(document.getElementById("map-canvas"), mapOptions);
}
google.maps.event.addDomListener(window, 'load', initialize);
</script>
</head>
<body>
<div id="map-canvas"></div>
</body>
</html>

```

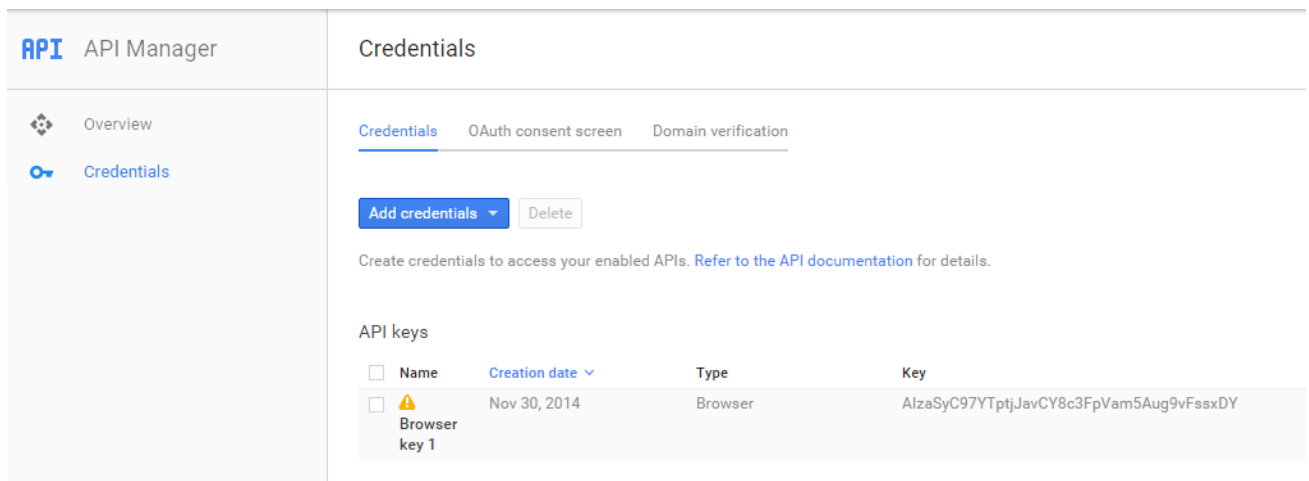
Le code suivant sert à charger l'API Map :

Il faut que la page soit complètement chargée (en utilisant `window.onload`) et en écrivant le code JavaScript dans une balise `<script>` dans la page twig.

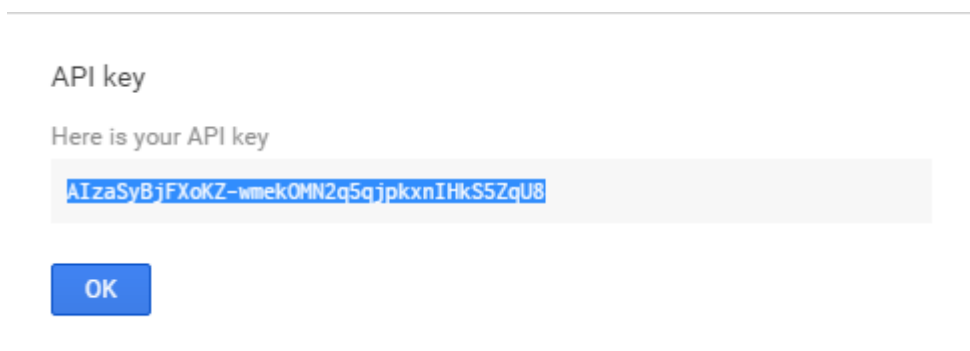
En outre, nous demandons l'API pour exécuter uniquement la fonction `initialize ()` ensuite l'API est entièrement chargée en passant `callback=initialize` à l'API Google Mapsbootstrap.

5. Utiliser une clé API, le mettre comme paramètre

Visitez la console API au <https://code.google.com/apis/console/> et connectez-vous avec votre compte Google.



6. Copier cette clé :



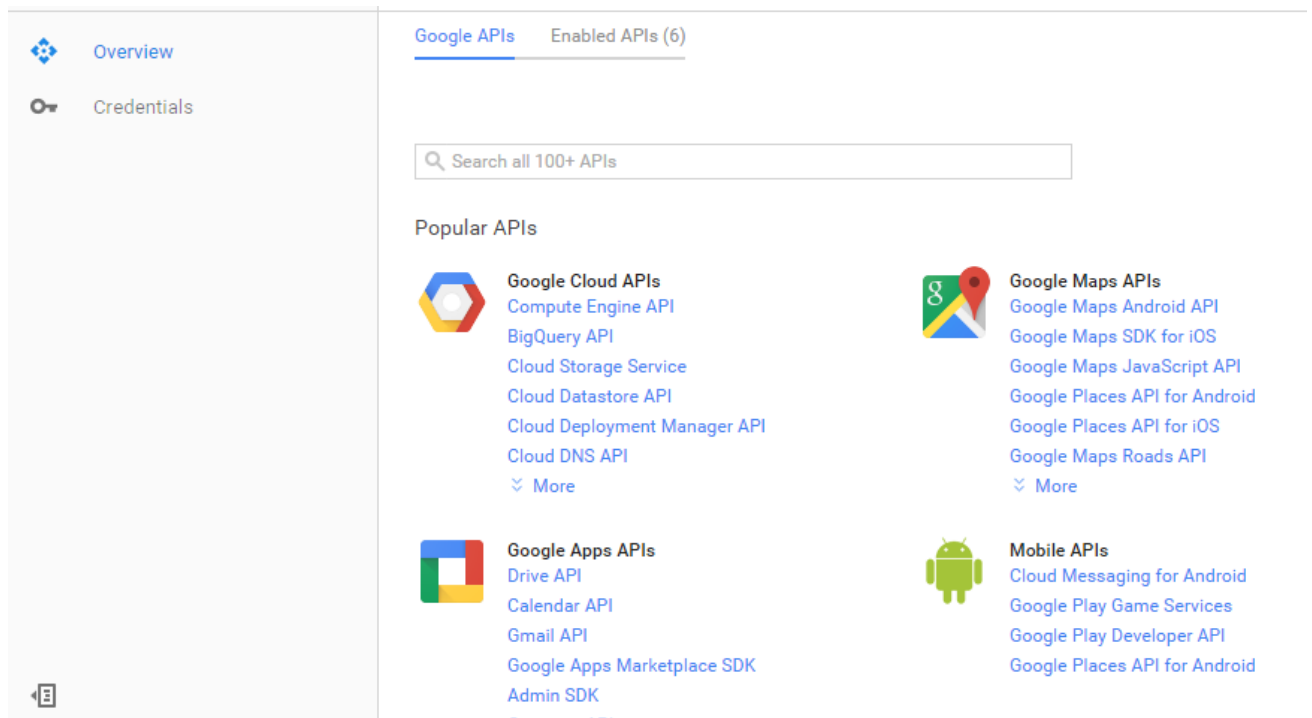
7. Mettre cette clé dans le fichier base.html.twig ;

```
<script src="http://maps.googleapis.com/maps/api/js?key=YOUR_KEY"></script>
```

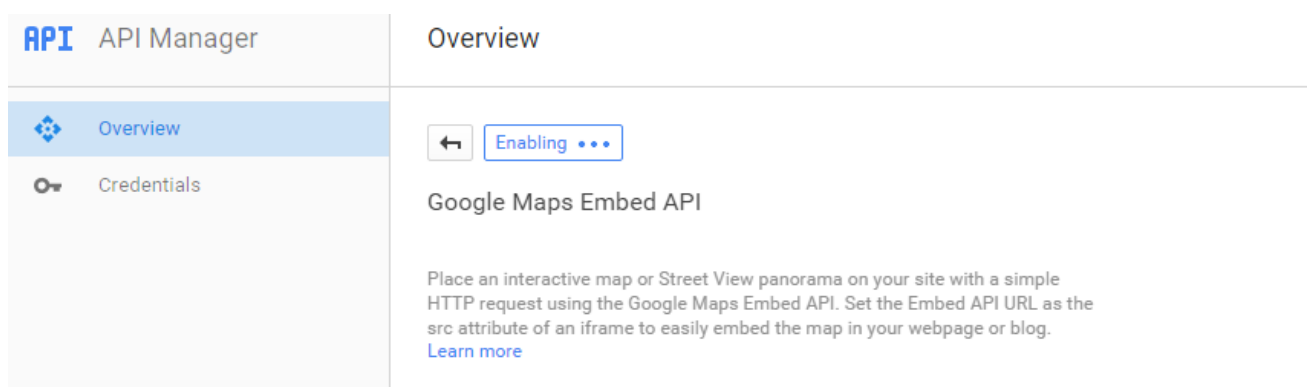
Comme indique l'image ci-dessous :

```
<html>
<head>
  <script
    src="http://maps.googleapis.com/maps/api/js?key=AIzaSyDEuWhaOCsjBU7Bigyf3dCxClnw0ltwHfQ">
  </script>
```

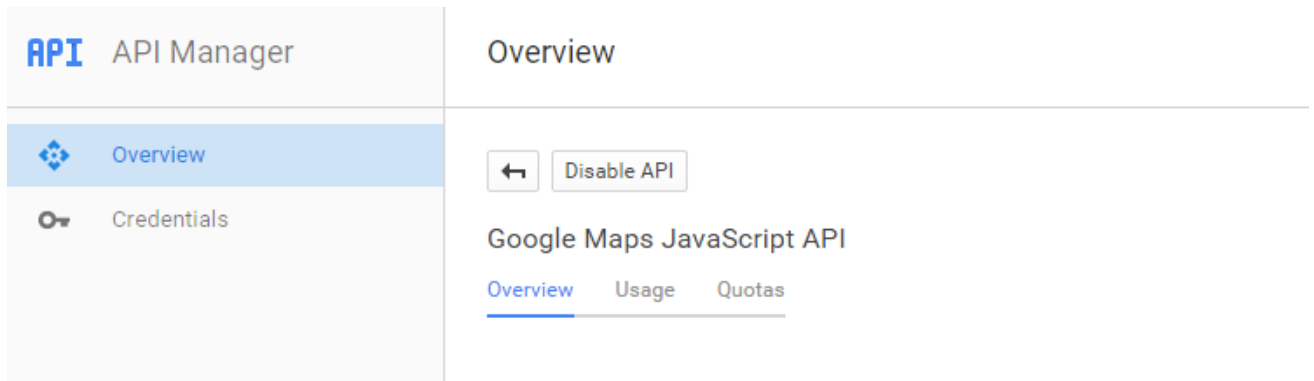
8. Activez le service API de Google Maps :



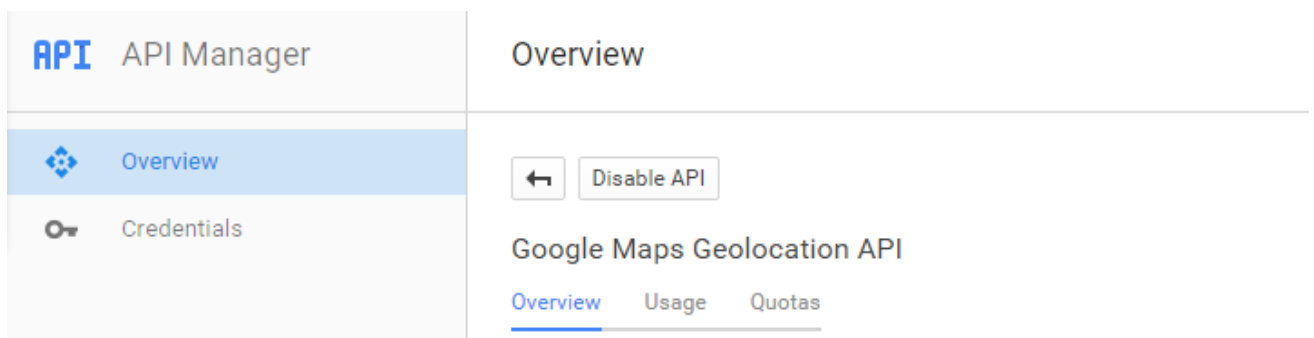
9. Activez le service API de Google Maps Embed :



10. Activez le service API de Google Maps JavaScript.



11. Activez le service API de Google Maps JavaScript.



12. Modifier la page **index.html.twig** sous le répertoire **MapBundle** /ressource/views/Default:

```
{% extends "myMapBundle:Default:base.html.twig" %}
1 {% block container %}
    <div id="googleMap" style="width:500px;height:380px;"></div>
- {% endblock %}
```

13. Modifier la balise body dans la page **base.html.twig** :

```

<html>
  <head>
    <script src="http://maps.googleapis.com/maps/api/js"></script>
    <script>
      function initialize() {
        var mapProp = {
          center: new google.maps.LatLng(51.508742, -0.120850),
          zoom: 5,
          mapTypeId: google.maps.MapTypeId.ROADMAP
        };
        var map = new google.maps.Map(document.getElementById("googleMap"), mapProp);
      }
      google.maps.event.addDomListener(window, 'load', initialize);
    </script>
  </head>

  <body>
    {& block container &}
    {& endblock &}
  </body>

```

Visualiser le résultat :

http://localhost/GoogleMaps/web/app_dev.php/map



14. Centrer la carte sur un point précis, nous créons un objet Lat et Lng afin de tenir cet endroit en passant les coordonnées de l'emplacement dans l'ordre : latitude, longitude dans le **Controller**

```
php
namespace My\MapBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;

class DefaultController extends Controller {

    public function indexAction() {
        $Latitudes = '-24';
        $Longitudes = '142';
        return $this->render('MyMapBundle:Default:index.html.twig', array('Latitudes' => $Latitudes, 'Longitudes' => $Longitudes));
    }
}
```

15. Ajouter les deux champs inputs dans la page **index.html.twig** :

```
{% extends "MyMapBundle:Default:base.html.twig" %}
{% block container %}
    <input id="lat" value= {{Latitudes}}/>
    <input id="lng" value= {{Longitudes}}/>
    <div id="googleMap" style="width:500px;height:380px;"></div>
{% endblock %}
```

➔ Si vous voulez mettre les 2 inputs invisibles, on ajoute le type hidden comme suit :

```
{% extends "mytestBundle::base.html.twig" %}
{% block container %}
    <input id="lat" type="hidden" value= {{Latitudes}}/>
    <input id="lng" type="hidden" value= {{Longitudes}}/>
    <div id="map-canvas"></div>
{% endblock %}
```

16. Créer et animer le Marker, on écrit ce bout de code dans la page **base.html.twig** :

Nous allons utiliser l'objet Marker de l'API Google Map. Son constructeur prend un seul paramètre, un tableau statique d'options :

- **position** définit la position du Marker sur la carte et prend un objet de type **LatLng**.


```

<head>
  <script
    src="http://maps.googleapis.com/maps/api/js?key=AIzaSyDEuWhaOCsjBU7Biqyf3dCxClwOltwHf0">
  </script>
  <script>
    var myCenter = new google.maps.LatLng(51.508742, -0.120850);
    var marker;
    function initialize()
    {
      var mapProp = {
        center: myCenter,
        zoom: 5,
        mapTypeId: google.maps.MapTypeId.ROADMAP
      };
      var map = new google.maps.Map(document.getElementById("googleMap"), mapProp);

      var marker = new google.maps.Marker({
        position: myCenter,
        animation: google.maps.Animation.BOUNCE
      });
      marker.setMap(map);
    }
    google.maps.event.addDomListener(window, 'load', initialize);
  </script>
</head>
<body>
  {% block container %}
  {% endblock %}
</body>

```

Visualiser le résultat :

http://localhost/GoogleMaps/web/app_dev.php/map

