

# Architecture et Programmation du web

-

## TP. n. 3 : Affichage d'une carte avec OpenLayers et OpenStreetMaps

Pierre Pompidor

31 octobre 2018

### Introduction au TP :

Ce TP a pour but d'afficher une carte avec la bibliothèque **OpenLayers** interfaçant la base de données cartographiques **OpenStreetMaps**. Sur cette carte des marqueurs correspondront à des points d'intérêt. Ces points d'intérêt seront construits à partir d'informations prises dans des fichiers JSON et seront sélectionnables en faisant apparaître des popups sur la carte.

Vous devez faire ce TP en sept étapes :

- Première étape : création du projet et installation des modules JavaScript nécessaires
- Seconde étape : création de la page HTML
- Troisième étape : affichage d'une carte centrée sur Montpellier
- Quatrième étape : affichage de cases à cocher correspondant à des points d'intérêt
- Cinquième étape : création d'un accordéon pour afficher les cases à cocher
- Sixième étape : affichage de marqueurs sur la carte correspondant aux points d'intérêt
- Septième étape : affichage de popups sur la sélection des marqueurs

Ce TP sera aussi l'occasion de créer un projet JavaScript en utilisant des **modules JavaScript**.

## Première étape : installation des modules JavaScript :

Vous allez créer un projet JavaScript dans lequel des modules vont être installés avec la commande **npm** (ce qui implique que la plateforme Node.js ait été installée ce qui est le cas sur les postes de travail).

Outre le module *OpenLayers*, vous allez installer *JQuery* et sa bibliothèque de widgets graphiques *JQuery-UI*. Enfin vous installerez un module **parcel** qui permet de créer une page web "transportable" (avec ses ressources).

## Création du projet

Dans votre dossier *public.html*, créez un dossier *tp\_3\_ol\_osm*. Dans ce dossier, exécutez la commande suivante :

```
npm init
```

La commande *npm init* permet de créer le fichier *package.json* spécifiant les dépendances d'un projet JavaScript.

Dans votre cas, ce fichier *package.json* doit être modifié comme ceci (si le fichier HTML qui vous créerez dans l'étape suivante sera nommé *OL\_OSM.html* et le code JavaScript associé *OL\_OSM.js*) :

```
{
  "name": "tp_3_ol_osm",
  "version": "1.0.0",
  "description": "Basic example",
  "main": "OL_OSM.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "parcel OL_OSM.html",
    "build": "parcel build --public-url . OL_OSM.html"
  },
  "author": "<vous>",
  "license": "ISC",
  "devDependencies": {
    "parcel-bundler": "^1.10.3"
  }
}
```

## Installation des modules

Outre le module OpenLayers, vous allez installer *jQuery*, sa bibliothèque de widgets graphiques *jQuery-UI*, ainsi que le module **parcel** qui permet de créer une page web et ses ressources à partir des différents modules.

```
npm install ol
npm install jquery
npm install jquery-ui
npm install --save-dev parcel-bundler
```

Ces modules sont installés en local dans le dossier *node\_modules* (ils pourraient être installés globalement mais vous n'en avez pas le droit).

## Seconde étape : création de la page HTML :

Créez la page HTML *OL\_OSM.html* suivante :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
  </head>
  <body>
    <center> <h7> Essai OpenLayers avec OSM et JQUERY ! </h7> </center>
    <div id="points_interet" style="width:100%; height:25%"></div>
    <div id="map" class="map" style="width:100%; height:75%"></div>
    
    <div id="popupProto" style="display:none; font-size:18pt; color:black;
                                width:100px; height:100px"></div>

    <script src="OL_OSM.js"></script>
  </body>
</html>
```

La division *points\_interet*, l'image *markerProto* et la division *popupProto* seront utilisées dans les étapes suivantes.

## Troisième étape : affichage d'une carte centrée sur Montpellier

Créez un code nommé *OL\_OSM.js*. Dans ce code importez les objets et méthodes JavaScript nécessaires :

```
import 'ol/ol.css';
import {Map, View, Overlay} from 'ol';
import TileLayer from 'ol/layer/Tile';
import OSM from 'ol/source/OSM';
import {fromLonLat, transform} from 'ol/proj';
```

Complétez ce code pour afficher une carte centrée sur Montpellier (voir le polycopié).

Pour exécuter votre application via un serveur local de développement géré par *parcel*, exécutez la commande suivante :

```
npm start
```

et ouvrez votre navigateur avec l'URL : *localhost :1234*

### Quatrième étape : affichage de cases à cocher correspondant à des points d'intérêt

Des cases à cocher doivent être créées dans la division *points\_interet*. Les informations concernant ces points d'intérêt sont contenues dans le dossier *JSON* du moodle.

Créez un dossier *JSON* dans votre dossier *tp\_3\_ol\_osm* et recopiez dans celui-ci les fichiers JSON provenant du Moodle.

Comme dans le TP précédent, nous allons utiliser la bibliothèque JQuery pour récupérer le contenu de ces fichiers. Installez JQuery sous forme de modules :

```
const jquery = require("jquery");
window.$ = window.jQuery = jquery;
```

Modifiez votre code JavaScript pour faire apparaître autant de cases à cocher que de points d'intérêts. La lecture des informations contenus dans les fichiers JSON doit se faire "récursivement" à partir du fichier *themes.json*.

Vous utiliserez `$.getJSON()` pour importer ces informations. Par exemple pour le premier fichier :

```
$.getJSON("http://localhost/~votreLogin/tp_3_ol_osm/JSON/themes.json",
    function(data) { ... });
```

Attention : à partir de cette étape, utilisez la commande suivante pour générer un dossier nommé *dist* contenant la page web et ses ressources :

```
npm run build
```

et invoquez la page dans votre navigateur ainsi :

```
http://localhost/~votreLogin/tp_3_ol_osm/dist/OL_OSM.html
```

### Cinquième étape : création d'un accordéon pour afficher les cases à cocher

Les cases à cocher doivent être gérées par un **accordéon** qui est une widget (composant graphique) de la bibliothèque **jquery-ui**.

Importez les objets et méthodes JavaScript nécessaires comme suit :

```
require("jquery-ui");  
require("jquery-ui/ui/unique-id");  
require("jquery-ui/ui/widgets/accordion");
```

Complétez votre page HTML pour associer les styles relatifs aux accordéons.

```
<link rel="stylesheet" href="jquery-ui-1.12.1.custom/jquery-ui.min.css"></link>
```

Modifiez votre code pour que les thèmes des points d'intérêt (ici *hôtels* et *restaurants*) soient encadrés par des balises *h3*, et que les cases à cocher soient encadrées par des divisions. Enfin appliquez la méthode à la division *points.interet* comme suit :

```
$('#points_interet').accordion({collapsible: true, heightStyle: 'content'});
```

### Sixième étape : affichage de marqueurs sur la carte correspondant aux points d'intérêt

Rajouter le code nécessaire pour qu'autant de marqueurs (sous OpenLayers, des **overlays**) soient affichés sur la carte que de points d'intérêt (voir le polycopié pour l'affichage d'un marqueur). Chaque marqueur doit être associé à une image clonée à partir de celle présente dans la page HTML initiale.

```
let image = $("#markerProto").clone();  
image.attr("id", "marker"+nomPI); // nomPI contient le nom du point d'intérêt  
$("#body").append(image);
```

Il est remarquer que les informations relatives aux marqueurs devraient être plutôt formatées dans le format **geojson**.

La sélection d'une case à cocher pour faire apparaître ou disparaître un marqueur peut se faire ainsi :

```
$( 'body' ).on( "change", "input[type=checkbox]", function() {  
    let  valeur = $(this).attr('name');  
    console.log("sélection de la case à cocher "+valeur);  
    if ($(this).is(':checked')) {  
        ...  
    }  
    else {  
        ...  
    }  
});
```

### Septième étape : affichage de popups sur la sélection des marqueurs

Rajouter le code nécessaire pour qu'autant de messages (en fait une chaîne de caractère dans une division) soient affichés sur la carte que de points d'intérêt (voir le polycopié pour l'affichage d'une popup). Chaque nouvelle popup est clonée à partir de celle présente dans la page HTML initiale.

```
let popup = $("#popupProto").clone();  
popup.attr("id", "popup"+nomPI);  
popup.append("<p>"+nomPI+"</p>");  
$( "body" ).append(popup);
```