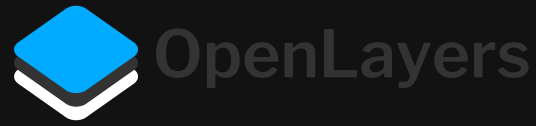
OpenLayers

OpenLayers

Librairies JavaScript de Webmapping

Deux librairies principales pour du webmapping 2D open source:

Leaflet	OpenLayers
Simple, mais extensible par des plugins	Beaucoup de fonctionnalités de base + plugins
Grande communauté, beaucoup d'exemples	Très à jour sur les nouveaux standards
Projection suisse peu supportée	Très présente sur le marché Suisse

D’autres librairies existent:

- MapLibre: Ensembles de fonctions avec tuiles vectorielles
- Cesium: Visualisation 3D
- Here Maps API, Google Maps API: Propriétaires

Dans ce cours, nous utiliserons OpenLayers

OpenLayers

Exemples d’utilisation

OpenLayers est très populaire en Suisse mais aussi ailleurs dans le monde, il sert de base à de nombreux géoportails.

- GeoAdmin : https://map.geo.admin.ch
- SuisseMobile : https://map.wanderland.ch
- Luxembourg : https://map.geoportail.lu
- EPFL : https://plan.epfl.ch
- SITN : https://sitn.ne.ch
- Transports publics en temps réel : https://tracker.geops.ch
- Saint-Pierre de la Réunion : https://geo.saintpierre.re
- Région de Nyon : https://map.cartolacote.ch

OpenLayers

Ressources

OpenLayers est très bien documenté :

- Site officiel : https://openlayers.org/
- Quickstart : https://openlayers.org/doc/quickstart.html
- API complète : https://openlayers.org/en/latest/apidoc/
- La longue liste d'exemples : https://openlayers.org/en/latest/examples/

Nous allons voir les bases d'OpenLayers, mais il est attendu que vous appreniez à utiliser ces ressources. Les exemples sont le meilleure moyen d'apprendre rapidement, OpenLayers en contient un vaste catalogue.

Comment ça marche?

Le HTML

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Quick Start</title>
  </head>
  <body>
    <div id="map"></div>
    <script type="module" src="./main.js"></script>
  </body>
</html>
```

OpenLayers a besoin d'un `div` dans lequel sera placé la carte, il a ici l'id *map*.
Le script principal est ensuite inclus, comme dans un projet Vite de base.

OpenLayers - La Map

Principes de base

C’est l’objet de base, il représente la carte affichée à l’écran.

Propriétés :

- *target* : id de l’élément dans lequel créer la *Map*
- *view* : Objet *View* qui décrit la vue à afficher dans la carte
- *layers* : Tableau d’objets *Layers* qui décrivent les couches présentes sur la carte

De manière facultative :

- *controls* : Tableau d’objets *Control* décrivant les interactions avec la carte (zoom, rotation, plein écran, etc)
- *interactions* : Tableau d’actions pour lesquelles la carte doi réagir (clic, double clic, drag-drop, etc)
- D’autres propriétés existent, voir https://openlayers.org/en/latest/apidoc/module-ol_Map-Map.html

Comment ça marche? - La Map

Le JavaScript

```
import './style.css';
import Map from 'ol/Map.js';

const map = new Map({
    target: 'map',          // 'target' est l'id de la div à utiliser comme carte
    view: ???,
    layers: ???
});
```

Comment ça marche? - La Map

Le JavaScript

```
import './style.css';
import Map from 'ol/Map.js';

const map = new Map({
    target: 'map',          // 'target' est l'id de la div à utiliser comme carte
    view: ???,
    layers: ???
});
```

Dans OpenLayers, les paramètres sont toujours passés à l'aide d'un objet. On écrira :

```
new Map({target:'map', view:???, layers:???})
```

Et non pas

```
new Map(target='map', view=???, layers=???)
```

OpenLayers - La View

Principes de base

Objet qui décrit la résolution, la rotation, la position, la projection, etc. de la *Map*.

Propriétés :

- *center*: tableau contenant les coordonnées du centre de la carte
- *zoom*: nombre décrivant le niveau de zoom

Autres propriétés :

- *resolution* : alternative à *zoom*, qui décrit la résolution de la carte
- *projection* : la projection de la vue
- D'autres propriétés existent : https://openlayers.org/en/latest/apidoc/module-ol_View-View.html

Comment ça marche? - La View

Le JavaScript

```
import './style.css';
import Map from 'ol/Map.js';
import View from 'ol/View.js';

const map = new Map({       // La Map est l'objet principal, c'est lui qui contiendra tous les composants OpenLayers
    target: 'map',          // 'target' est l'id de la div à utiliser comme carte
    view: new View({        // La View est le composant qui indique comment la Map doit être affichée
        center: [0, 0],     // Ici centrée en 0;0
        zoom: 2,            // Et avec un zoom à 2
    }),
    layers: ???,
});
```

OpenLayers - Les Layers

Principes de base

Il existe plusieurs types de layers (module `ol/layer`), mais ils peuvent être divisés en 2 catégories:

- Raster (par ex: `TileLayer`, `ImageLayer`)
- Vectoriel (par ex: `VectorLayer`, `VectorTileLayer`)

La *source* de la donné (module `ol/source`) est une propriété d'un `layer` et on peut à nouveau les séparer en 2 catégories:

- Raster (par ex: `TileSource`, `ImageSource`)
- Vectoriel (par ex: `VectorSource`, `VectorTile`)

Un exemple de `TileSource` est un objet `OSM` qui indique une couche OpenStreetMap.

Exemple minimal d’OpenLayers

Code complet de 'script.js'

```
import './style.css';
import Map from 'ol/Map.js';
import OSM from 'ol/source/OSM.js';
import TileLayer from 'ol/layer/Tile.js';
import View from 'ol/View.js';

const map = new Map({       // La Map est l'objet principal, c'est lui qui contiendra tous les composants OpenLayers
    target: 'map',          // 'target' est l'id de la div à utiliser comme carte
    view: new View({        // La View est le composant qui indique comment la Map doit être affichée
        center: [0, 0],     // Ici centrée en 0;0
        zoom: 2,            // Et avec un zoom à 2
    }),
    layers: [               // Liste des layers à afficher
        new TileLayer({     // Dans ce cas il y a un layer tuilé, d'OpenStreetMap (OSM)
            source: new OSM(),
        }),
    ],
});
```

OpenLayers - Les Contrôles

Les contrôles sont des éléments permettant de manipuler la carte ou d'afficher une information.

Par défaut, `Map` en charge 3:

- `Zoom`
- `Rotate` : Orientation (apparaît dès que la carte est tournée)
- `Attribution` : Copyright des données

Il en existe d'autres

- Barre d'échelle : `ScaleLine`
- Carte d'aperçu : `OverviewMap`
- Position curseur : `MousePosition`
- Plein écran : `FullScreen`
- Zoom sur étendue max : `ZoomToExtent`
- Curseur de zoom : `ZoomSlider`

Exemples : https://openlayers.org/en/latest/examples/?q=control