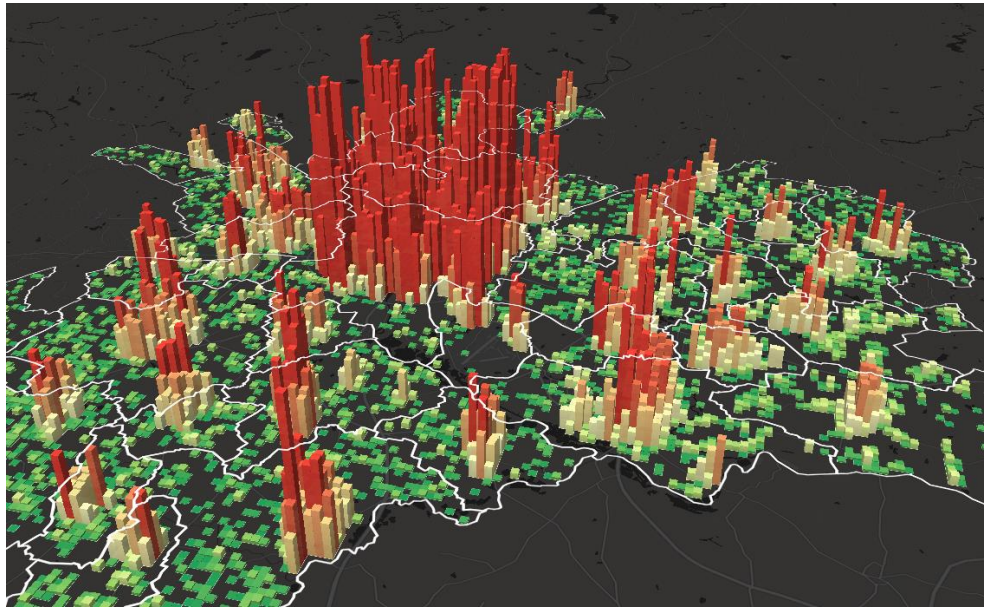
mapbox

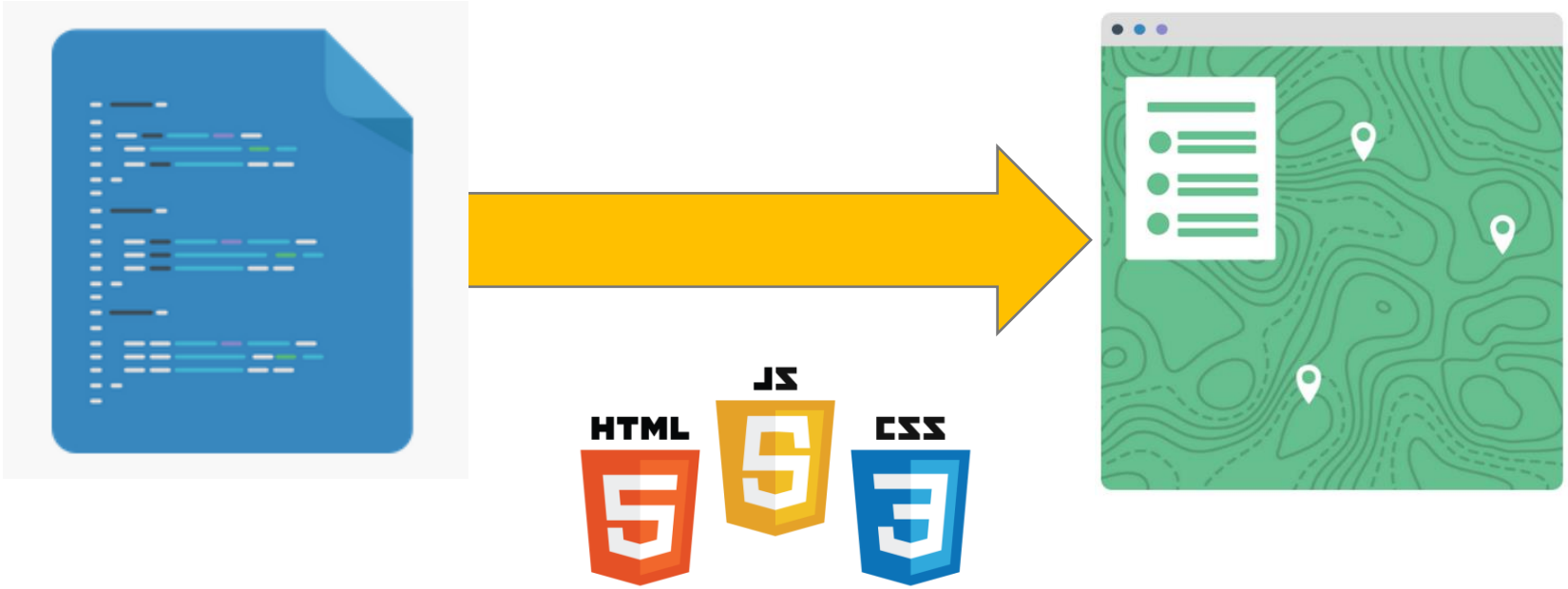
# Introduction à la bibliothèque JavaScript MapboxGL

@Boris Mericskay

# Objectifs atelier

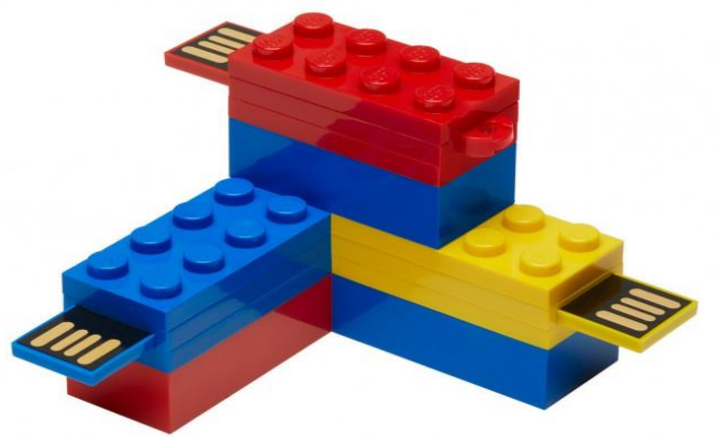
- Publication de données spatiales sur le Web
- De la page HTML à l'application en ligne
- Familiarisation avec le Javascript, l'HTML et le CSS



# Prise en main de Mapbox GL

# Coder = LEGO

- Vous aller à partir de maintenant « jouer » au LEGO en assemblant des lignes de codes pour construire des cartes sur le Web!



```
L.mapbox.accessToken = 'pk.eyJ1IjoibmluYW5vdW4iLCJhIjoiSkN4dndmTSJ9.6plStO7M5AuAbDa6O1m54A';
var map = L.mapbox.map('map', 'mapbox.light').setView([48.11,-1.66], 13);
// Ajouts des WMS

var orthophotographie = L.tileLayer.wms('http://geobretagne.fr/geoserver/photo/wms?', {
format: 'image/png',
transparent: true,
layers: 'ortho-ouverte'
}).addTo(map);

var quartiers = L.tileLayer.wms('http://geobretagne.fr/geoserver/rennesmetropole/wms?', {
format: 'image/png',
transparent: true,
layers: 'quartiers_vdr'
}).addTo(map);
```

# MapBoxGL

MapBoxGL est une bibliothèque JavaScript libre de cartographie en ligne *open-source* utilisant le WebGL pour l'affichage (tuiles vectorielles, affichage 3D)

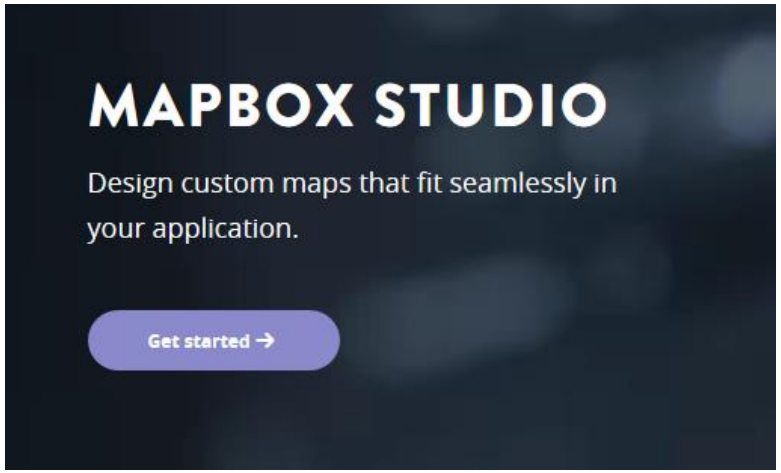
https://docs.mapbox.com/mapbox-gl-js/api/

Exemples: https://docs.mapbox.com/mapbox-gl-js/example/

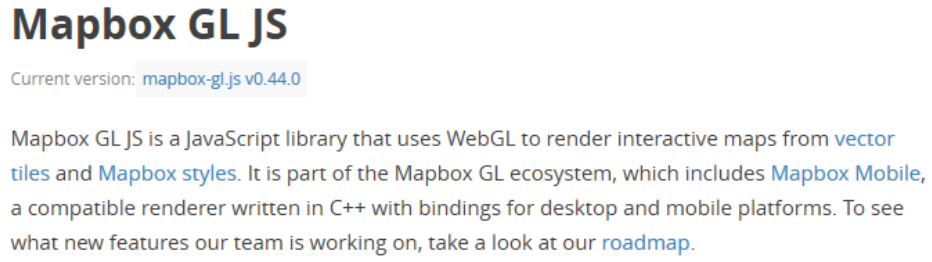
Tutoriels : https://docs.mapbox.com/help/tutorials/?product=Mapbox+GL+JS

# L’écosystème Mapbox

- Mapbox Studio
  - Créer des fonds de carte (*style*)
  - Héberger des jeux de données (*tilesets*)
    - Sous forme de tuiles vectorielles

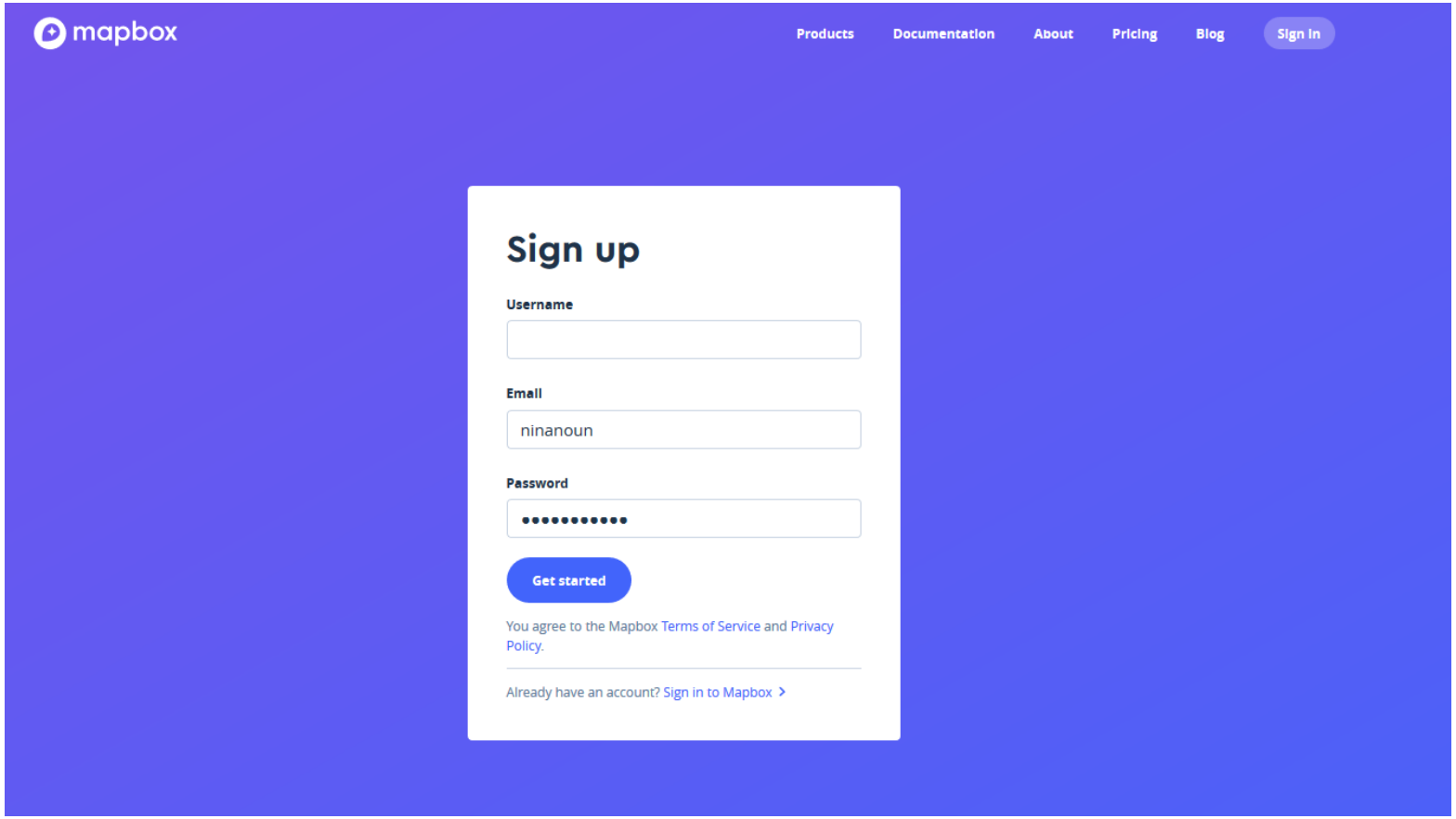



- API MapboxGL.js
  - Bibliothèque JavaScript pour créer des cartes Web
  - La mobilisation de MapboxGL nécessite une clef d’accès = besoin d’un compte Mapbox

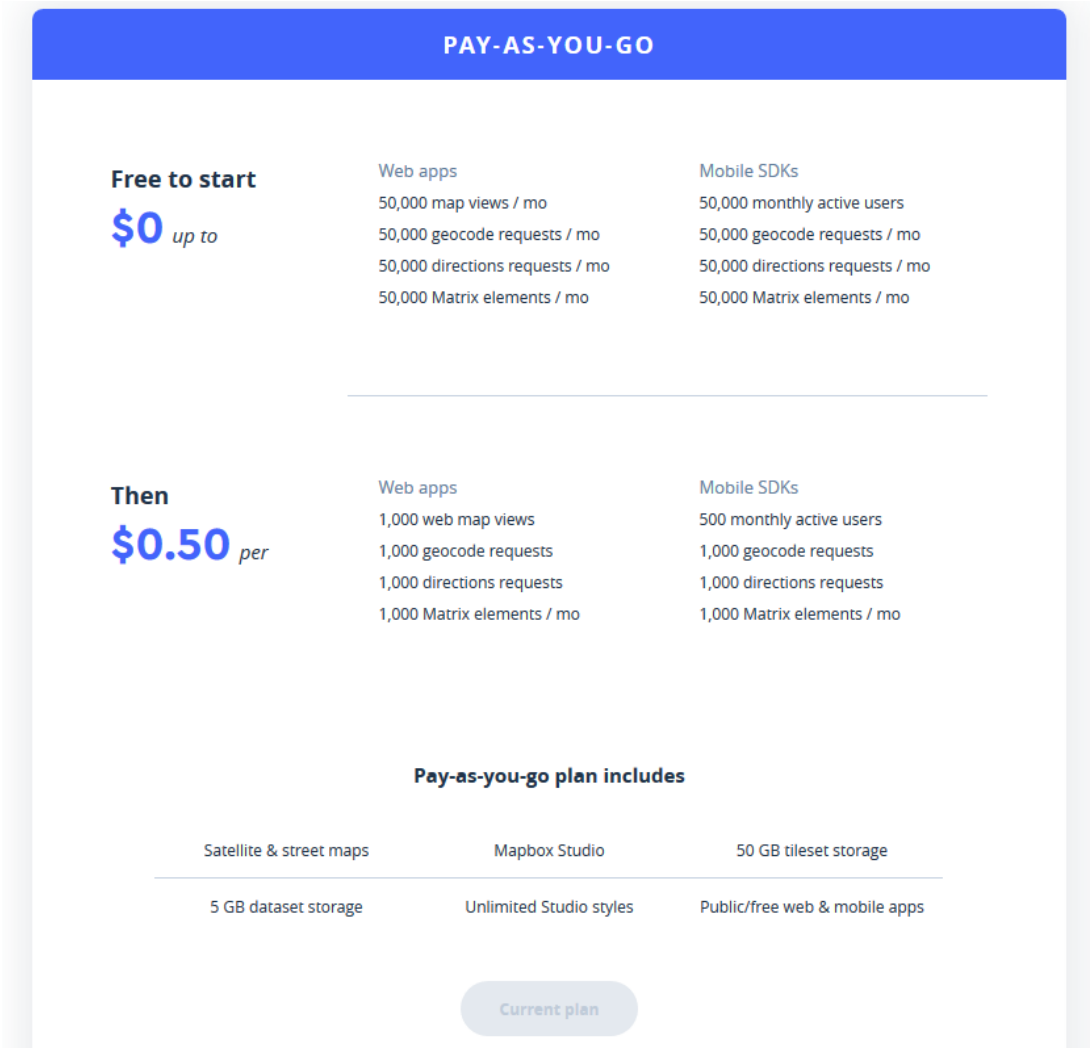


# Créer un compte Mapbox

https://www.mapbox.com/signup/



# Limitations d’un compte gratuit



**PAY-AS-YOU-GO**

**Free to start**

**$0** *up to*

| Web apps | Mobile SDKs |
| --- | --- |
| 50,000 map views / mo | 50,000 monthly active users |
| 50,000 geocode requests / mo | 50,000 geocode requests / mo |
| 50,000 directions requests / mo | 50,000 directions requests / mo |
| 50,000 Matrix elements / mo | 50,000 Matrix elements / mo |

**Then**

**$0.50** *per*

| Web apps | Mobile SDKs |
| --- | --- |
| 1,000 web map views | 500 monthly active users |
| 1,000 geocode requests | 1,000 geocode requests |
| 1,000 directions requests | 1,000 directions requests |
| 1,000 Matrix elements / mo | 1,000 Matrix elements / mo |

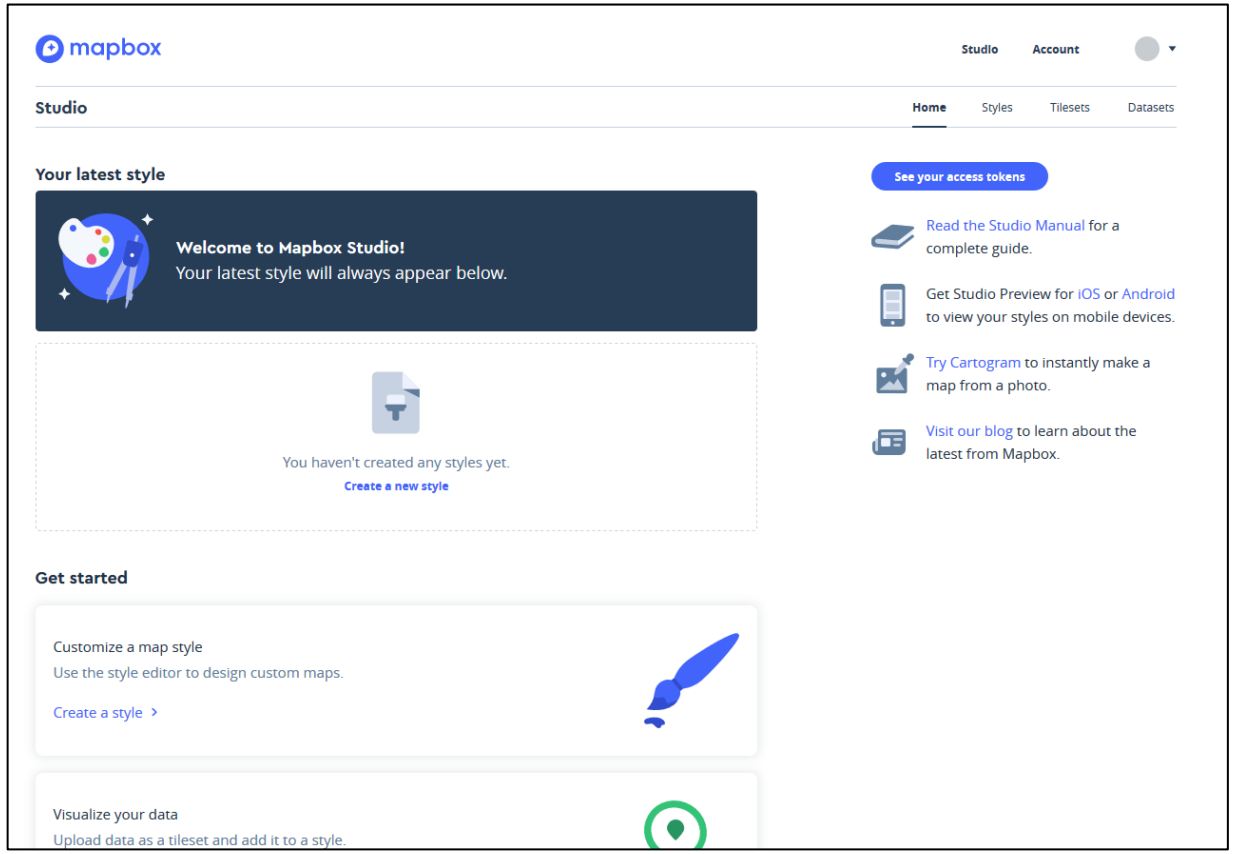
**Pay-as-you-go plan includes**

| | | |
| --- | --- | --- |
| Satellite & street maps | Mapbox Studio | 50 GB tileset storage |
| 5 GB dataset storage | Unlimited Studio styles | Public/free web & mobile apps |

Current plan

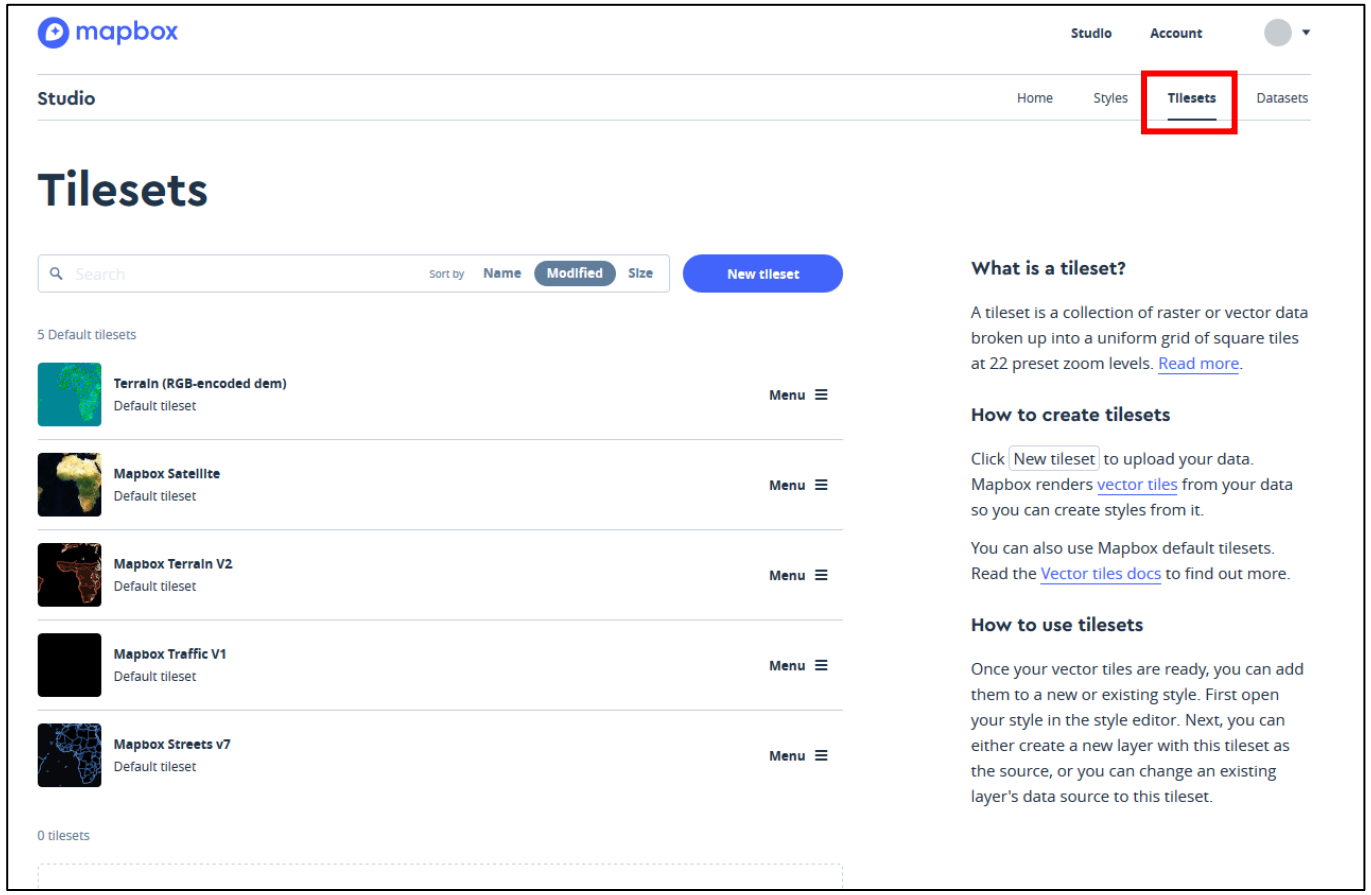
# Mapbox Studio

- Environnement en ligne de gestion des :
  - Fonds de carte (*Styles*)
  - Jeux de données (*Tilesets*)
  - Des clefs d'accès à l'API (*Account*)



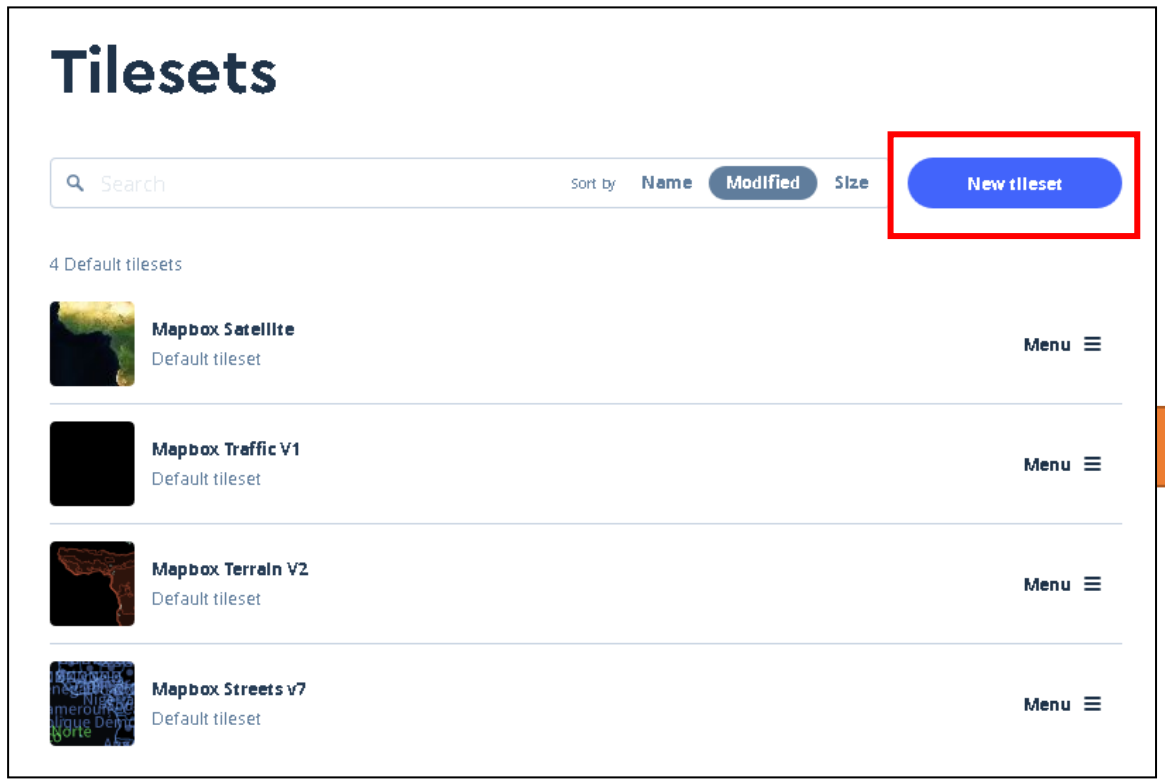
# Mapbox Studio

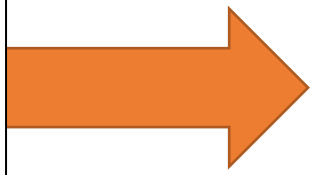
- Créer et gérer des jeux de données (Tilesets)



# Importer des données dans Mapbox Studio

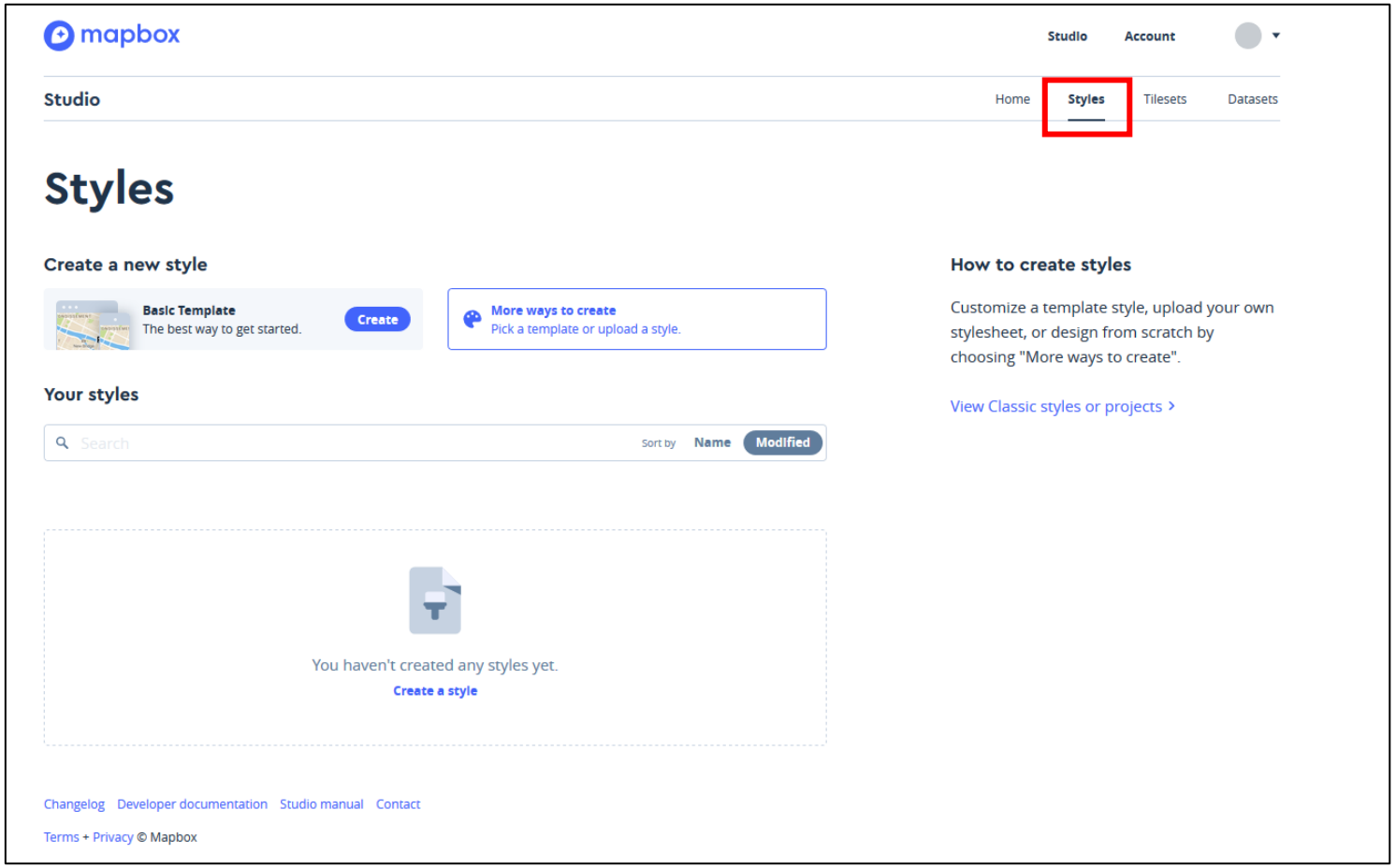
- Mapbox studio permet de stocker 50GO de données vectorielles et matricielles



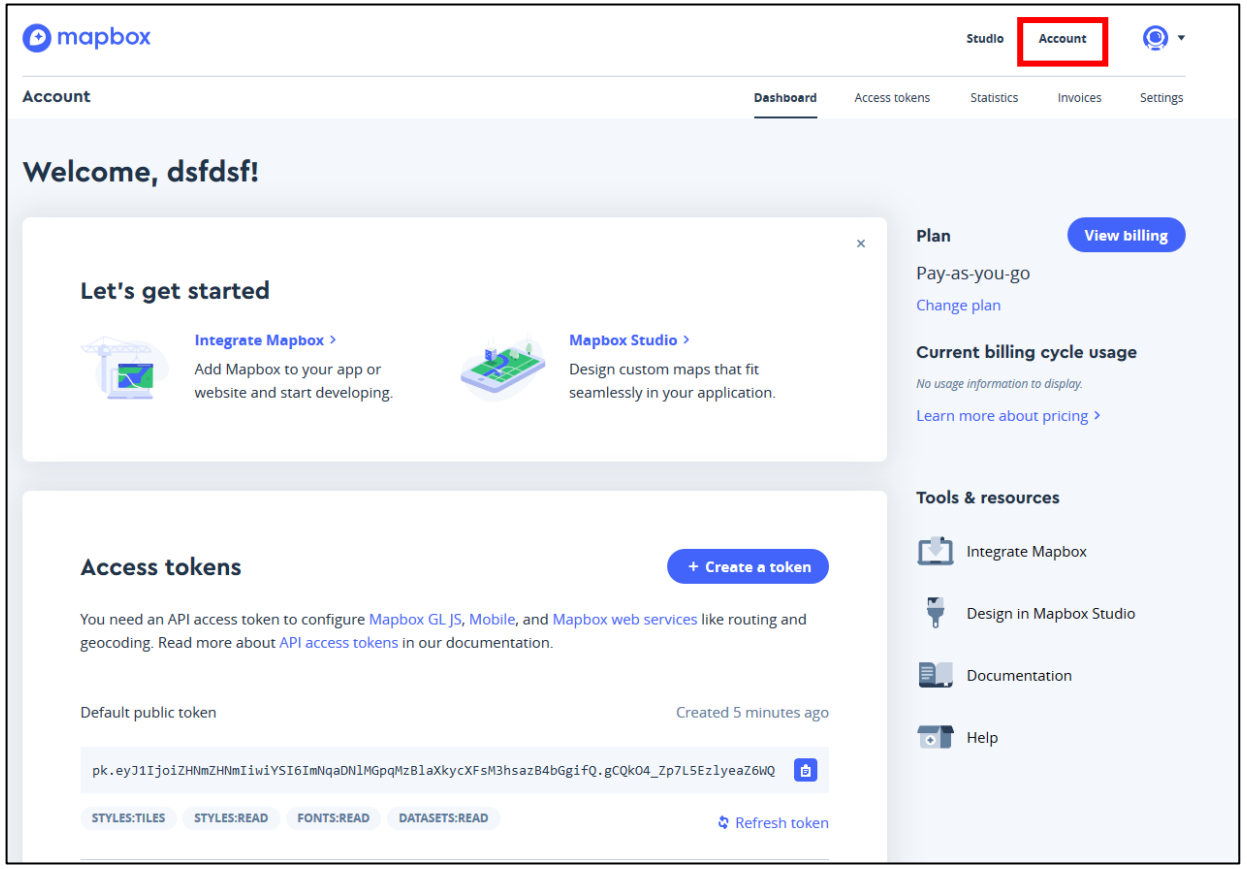
# Mapbox Studio

- Créer et gérer des fonds de cartes ( Styles)



# Mapbox Studio

- Créer et gérer ses clefs d’accès à l’API (Access tokens)



# Template de départ

https://gitlab.huma-num.fr/bmericskay/mapboxgl

```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8' />
    <title>MapboxGL</title>

<link href="https://api.mapbox.com/mapbox-gl-js/v3.1.2/mapbox-gl.css" rel="stylesheet">
<script src="https://api.mapbox.com/mapbox-gl-js/v3.1.2/mapbox-gl.js"></script>

<style>
#map {position:absolute; top:0; bottom:0; width:100%;}
</style>

</head>

<body>
<div id='map'></div>

<script>
    // AccesToken
mapboxgl.accessToken = 'pk.eyJ1IjoibWF1cmljaW9iYXF1ZXJvIiwiYSI6ImNscHNrZDd6aDA0NzcyaXJ6OTJiODVkMXYifQ.Sxnc-
50cCwIwe1Om58AHQA';

    // Configuration de la carte
var map = new mapboxgl.Map({
    container: 'map',
    style: 'mapbox://styles/mapbox/light-v9', // Fond de carte
    center: [-1.68, 48.12], // lat/long
    zoom: 15, // zoom
    pitch: 50, // Inclinaison
    bearing: -10 // Rotation
});

</script>

</body>
</html>
```

# Template de départ

```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8' />
    <title>MapboxGL</title>

<script src="https://api.mapbox.com/mapbox-gl-js/v2.0.1/mapbox-gl.js"></script>
<link href="https://api.mapbox.com/mapbox-gl-js/v2.0.1/mapbox-gl.css" rel="stylesheet" />

<style>
#map {position:absolute; top:0; bottom:0; width:100%;}
</style>

</head>

<body>
<div id='map'></div>

<script>
    // AccesToken
mapboxgl.accessToken =
'pk.eyJ1IjoibmluYW5vdW4iLCJhIjoiY2pjdHBoZGlzMnV4dDJxcGc5azJkbWRiYSJ9.o4dZRrdHcgVEKCveOXG1YQ';

    // Configuration de la carte
var map = new mapboxgl.Map({
    container: 'map',
    style: 'mapbox://styles/mapbox/light-v9', // Fond de carte
    center: [-1.68, 48.12], // lat/long
    zoom: 15, // zoom
    pitch: 50, // Inclinaison
    bearing: -10 // Rotation
});

</script>

</body>
</html>
```

Appel API MapboxGL

Style de la carte

Clef d'accès à l'API

Fond de carte
Niveau de zoom
Centrage de la carte (X,Y)
Inclinaison de la carte
Rotation de la carte

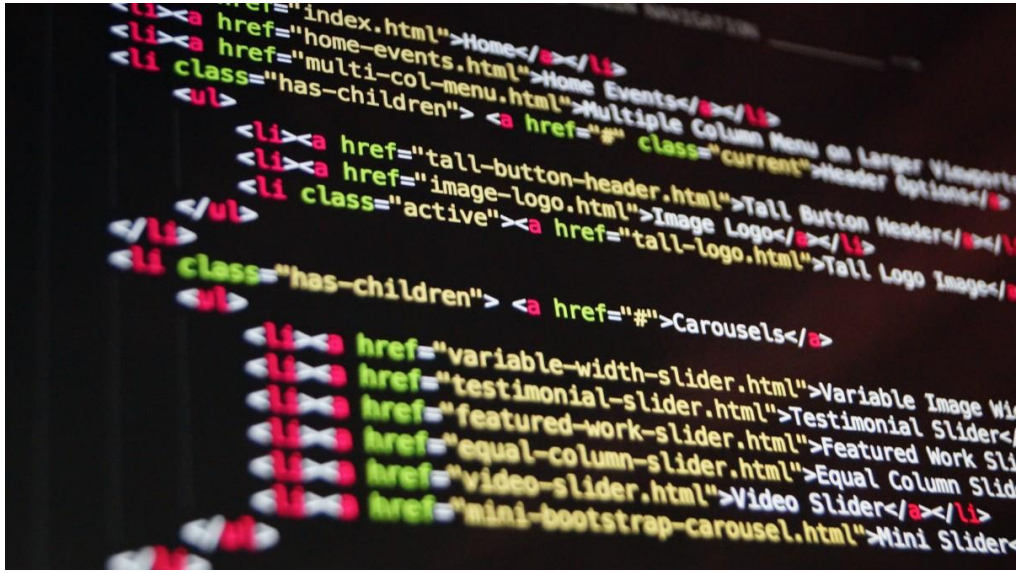
# Coder en ligne ou en local

- Utiliser un éditeur de code installé (notepad, ATOM,…)

OU

- Utiliser un éditeur de code en ligne

  https://codepen.io …

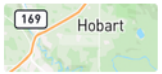
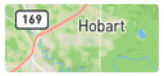
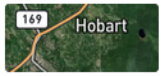
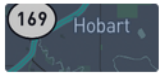


# Les fonds de carte

# Changer de fond de carte

- Les fonds de cartes de Mapbox > tuiles vectorielles

```
style: 'mapbox://styles/mapbox/dark-v11',
```

| Style name | Style URL | Style image |
|---|---|---|
| Mapbox Streets | mapbox://styles/mapbox/streets-v12 |  |
| Mapbox Outdoors | mapbox://styles/mapbox/outdoors-v12 |  |
| Mapbox Light | mapbox://styles/mapbox/light-v11 |  |
| Mapbox Dark | mapbox://styles/mapbox/dark-v11 |  |
| Mapbox Satellite | mapbox://styles/mapbox/satellite-v9 |  |
| Mapbox Satellite Streets | mapbox://styles/mapbox/satellite-streets-v12 |  |
| Mapbox Navigation Day | mapbox://styles/mapbox/navigation-day-v1 |  |
| Mapbox Navigation Night | mapbox://styles/mapbox/navigation-night-v1 |  |

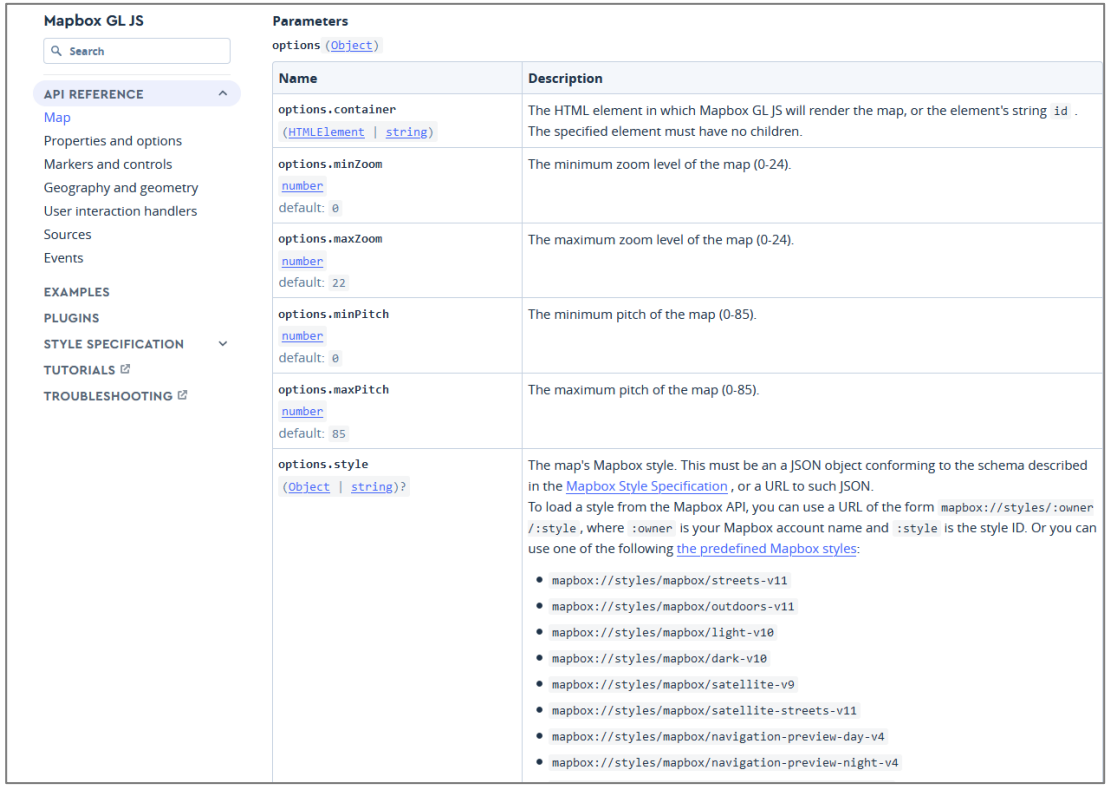
# Changer de fond de carte

- Aller plus loin avec la documentation

https://docs.mapbox.com/api/maps/styles/

https://docs.mapbox.com/mapbox-gl-js/guides/styles/

https://docs.mapbox.com/api/maps/styles/

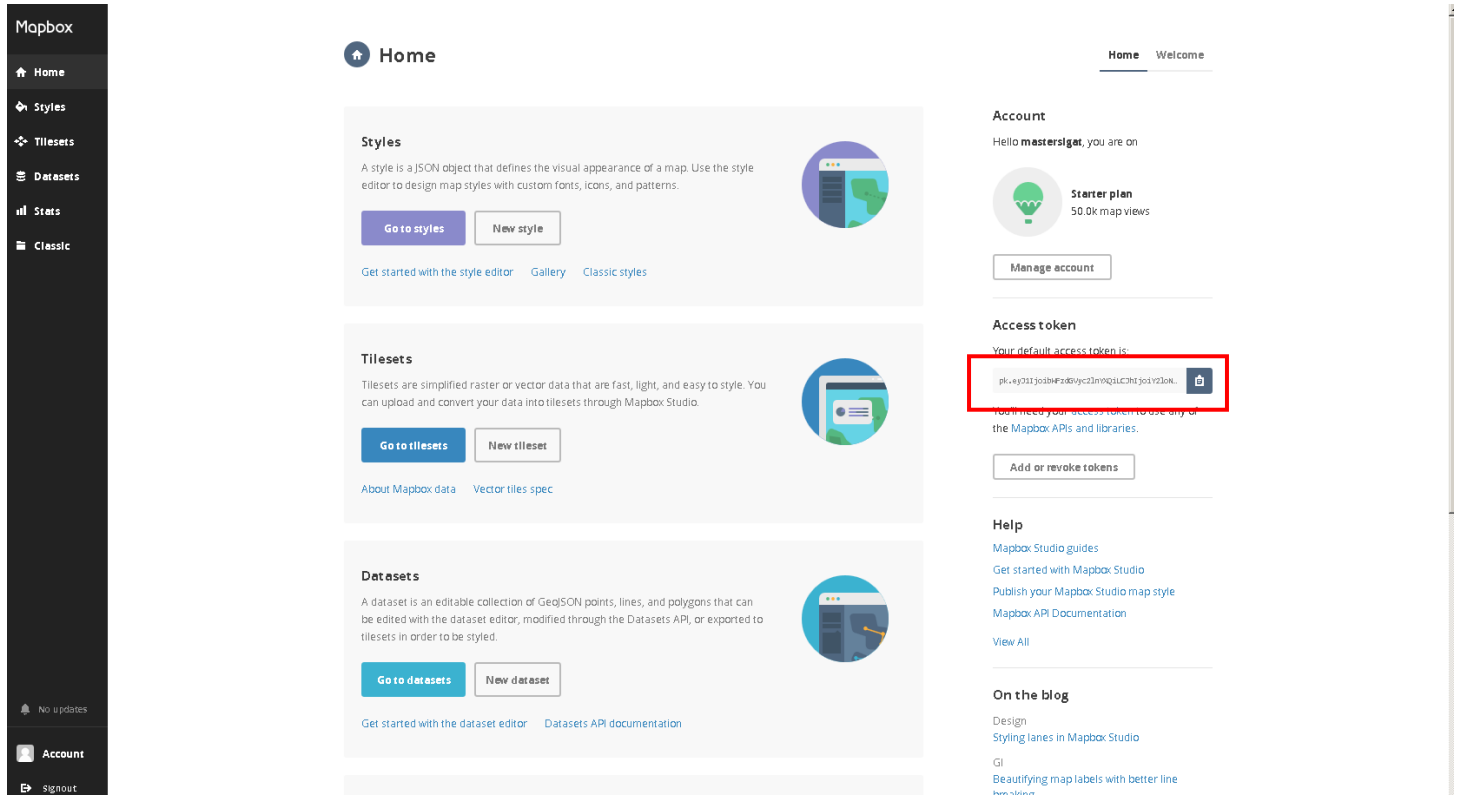


# Changer de fond de carte

```html
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8' />
    <title>MapboxGL</title>

<script src='https://api.tiles.mapbox.com/mapbox-gl-js/v0.44.0/mapbox-gl.js'></script>
<link href='https://api.tiles.mapbox.com/mapbox-gl-js/v0.44.0/mapbox-gl.css' rel='stylesheet' />

<style>
#map { position:absolute; top:0; bottom:0; width:100%; }
</style>

</head>

<body>
<div id='map'></div>

<script>
    // AccesToken
mapboxgl.accessToken = 'pk.eyJ1Ijoibm1uYW5vdW4iLCJhIjoiY2pjdHBoZGlzMnV4dDJxcGc5azJkbWRiYSJ9.o4dZRrdHcgVEKCveC

    // Configuration de la carte
var map = new mapboxgl.Map({
    container: 'map',
    style: 'mapbox://styles/mapbox/satellite-v9',
    center: [-1.68, 48.12], // lat/long
    zoom: 15, // zoom
    pitch: 50, // Inclinaison
    bearing: -10 // Rotation
});

</script>

</body>
</html>
```

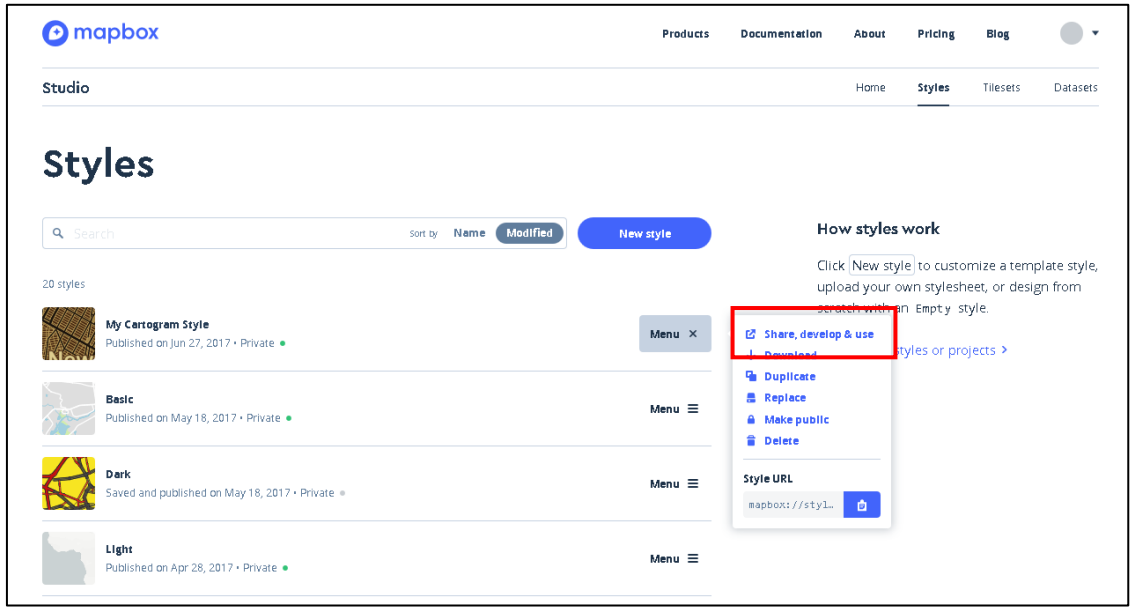
# Mettre un fond de carte personnel

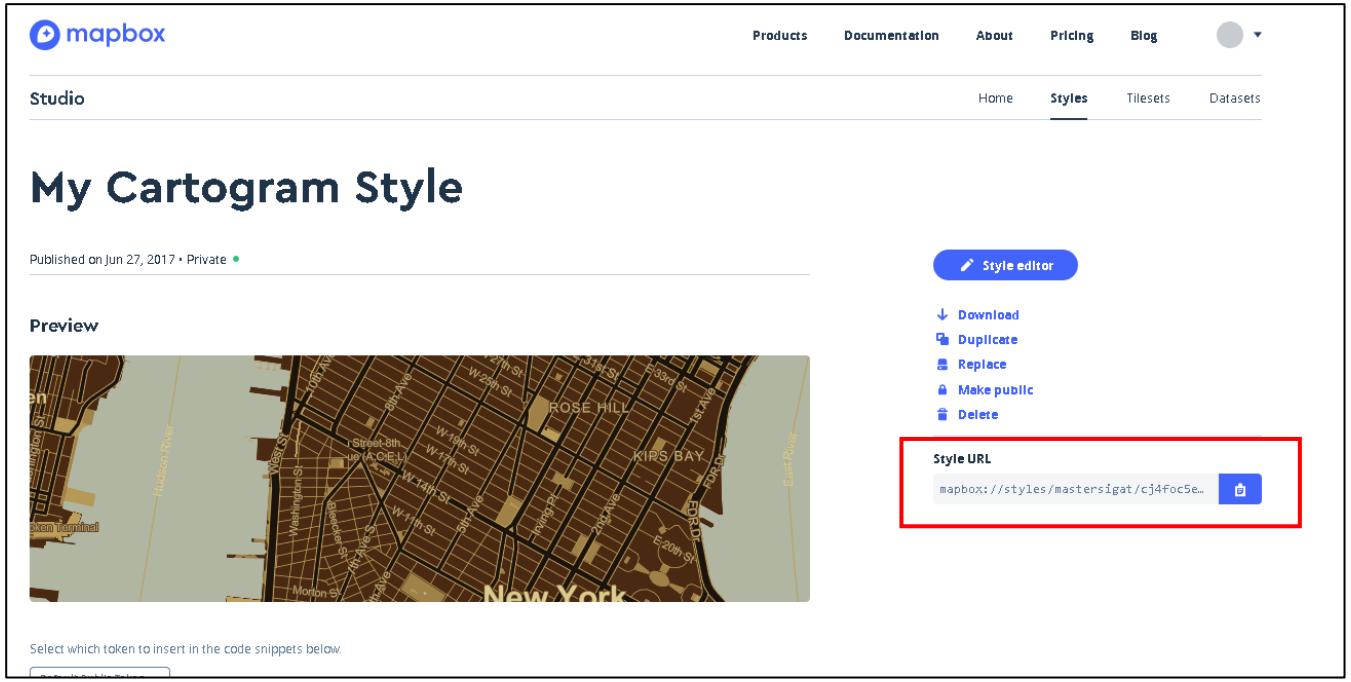
- Mettre votre clef d’accès personnelle à la place de celle fournie



# Incorporer un fond de carte personnel

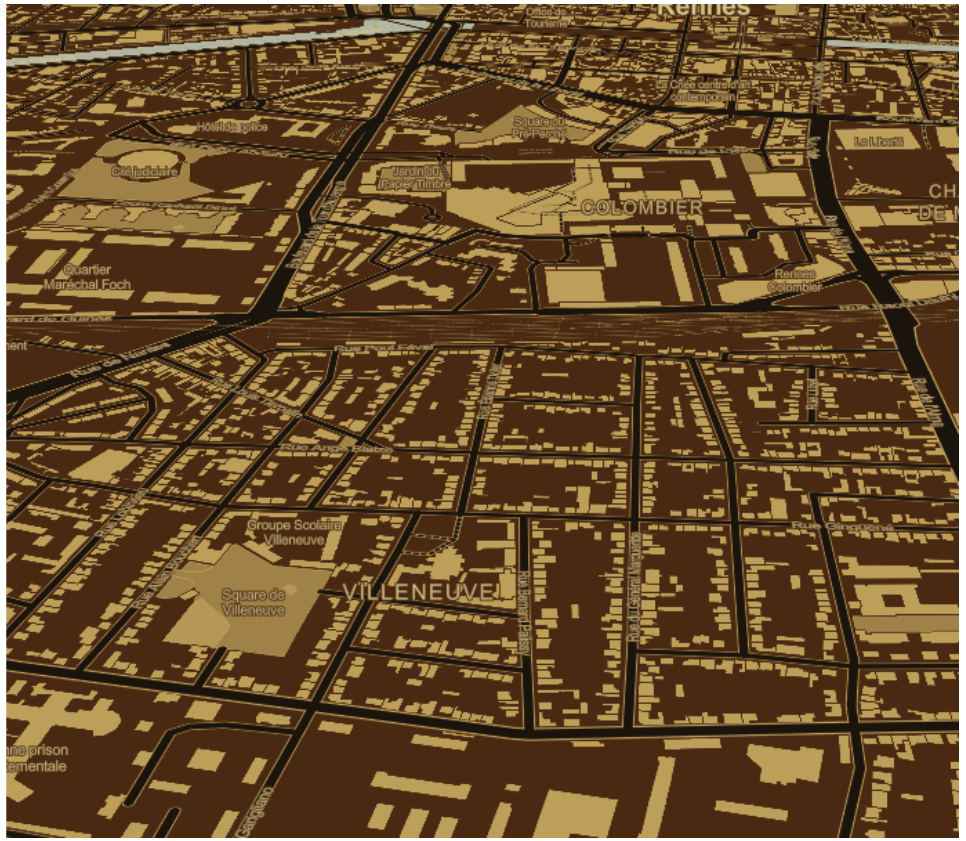
- Récupérer l'URL de votre fond de carte personnel



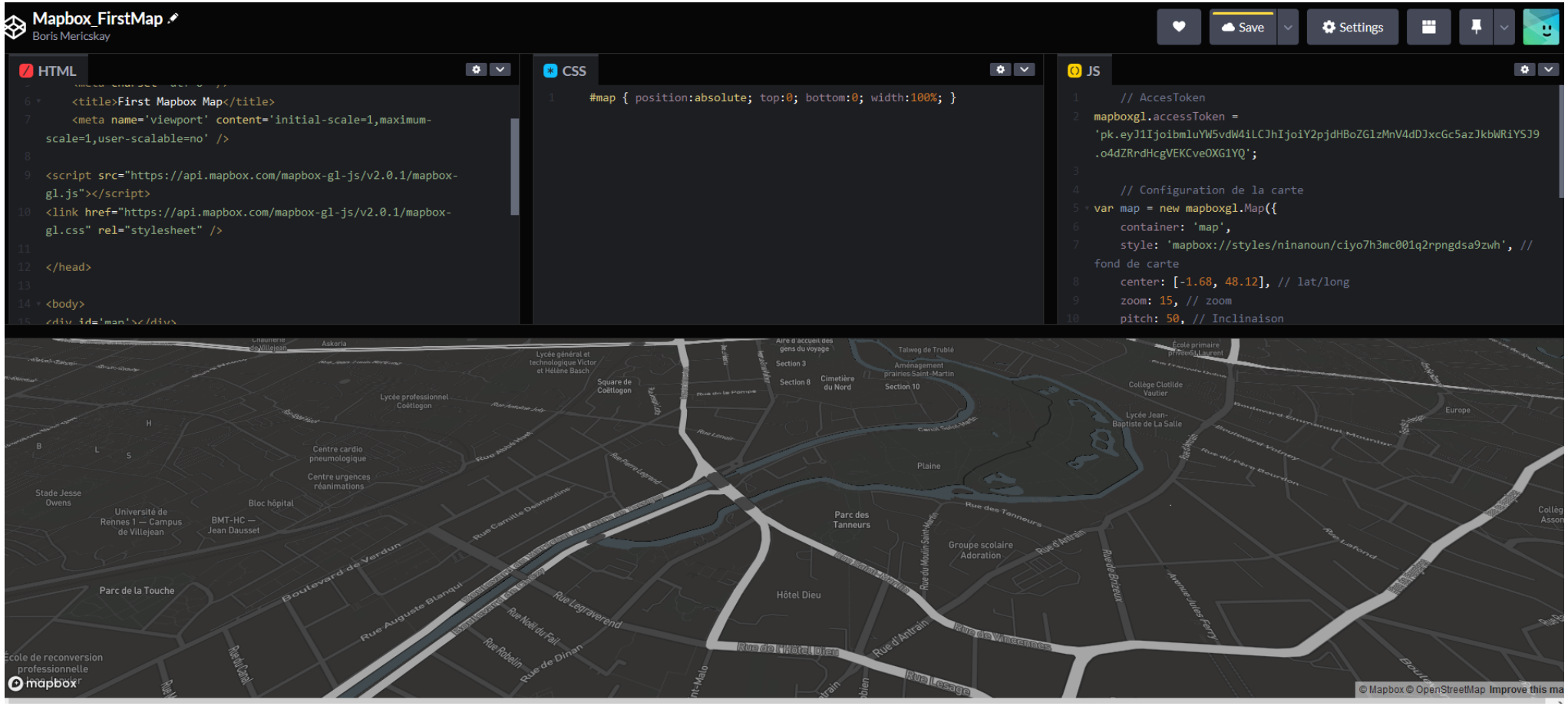



# Incorporer un fond de carte personnel

```html
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8' />
    <title>MapboxGL</title>

<script src='https://api.tiles.mapbox.com/mapbox-gl-js/v0.44.0/mapbox-gl.js'></script>
<link href='https://api.tiles.mapbox.com/mapbox-gl-js/v0.44.0/mapbox-gl.css' rel='stylesheet' />

<style>
#map { position:absolute; top:0; bottom:0; width:100%; }
</style>

</head>

<body>
<div id='map'></div>

<script>
    // AccesToken
mapboxgl.accessToken = 'pk.eyJ1IjoibWFzdGVyc2lnYXQiLCJhIjoiY2loNG9mamxwMHp2dHgxbTBjY2hlb2RteiJ9.dDYKXX9907pbT6sTAJ4fvA';

    // Configuration de la carte
var map = new mapboxgl.Map({
    container: 'map',
    style: 'mapbox://styles/mastersigat/cj4foc5eo3hsr2sqlomdgeuvd',
    center: [-1.68, 48.12], // lat/long
    zoom: 15, // zoom
    pitch: 50, // Inclinaison
    bearing: -10 // Rotation
});

</script>

</body>
</html>
```



# Première carte MapboxGL.js



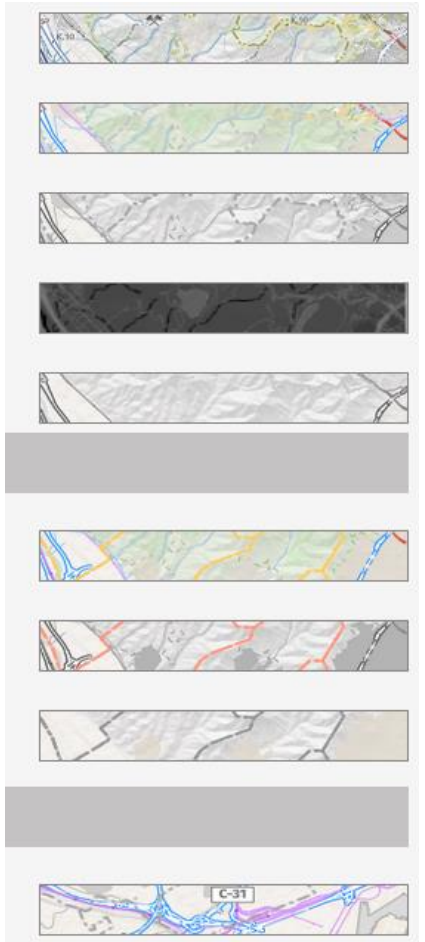

https://codepen.io/BorisMericskay/pen/qByJqZB

# Où trouver des fonds de carte libre

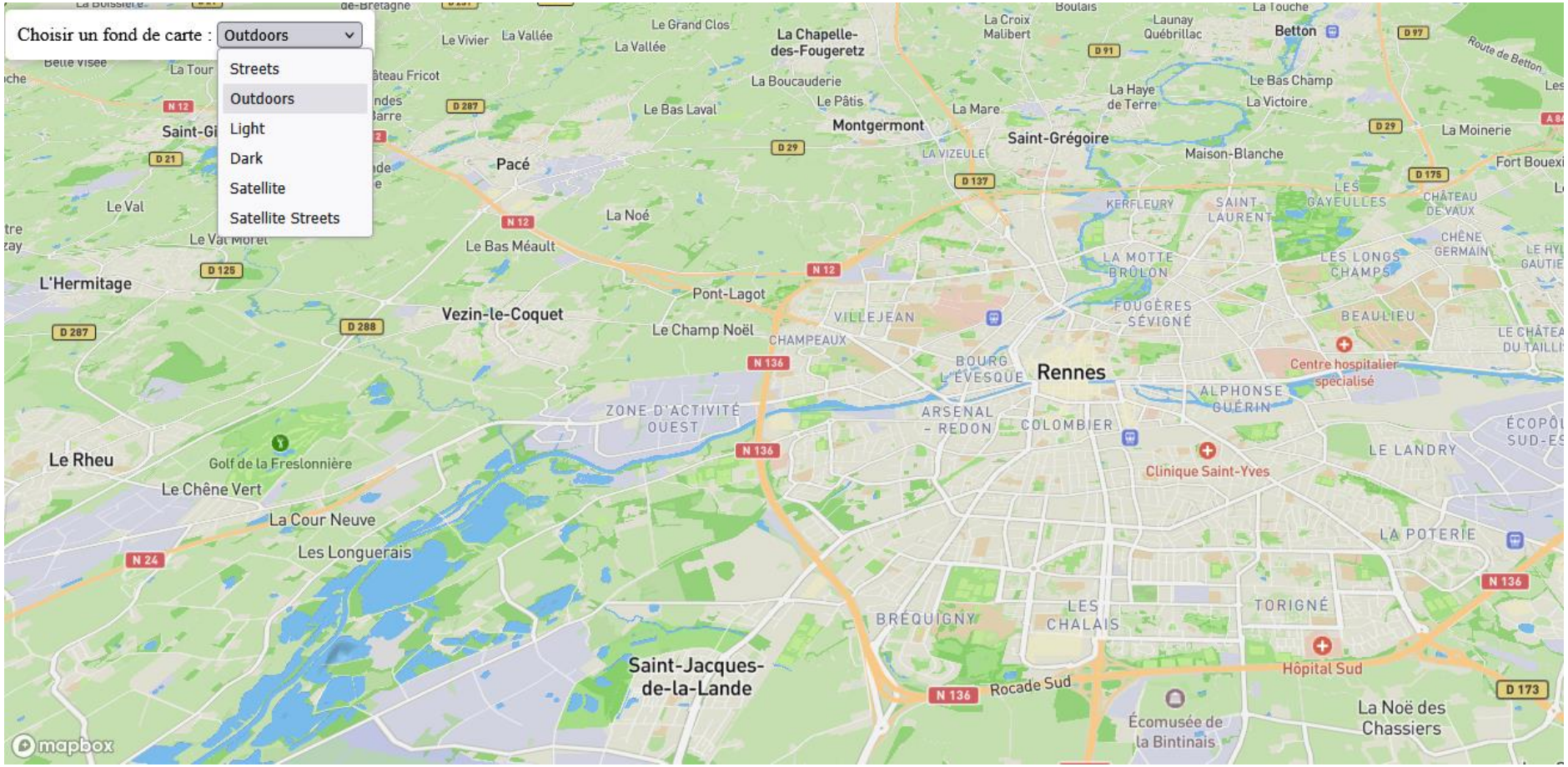
Il existe plusieurs flux de tuiles vectorielles en libre accès

Format .json / Basés sur les données OSM ou IGN

- https://deck.gl/docs/api-reference/carto/basemap
- https://openicgc.github.io/
- https://openmaptiles.geo.data.gouv.fr/
- https://github.com/Viglino/geoservice-style



# Ajouter des fonds de carte externes




https://codepen.io/BorisMericskay/pen/JojoXRq

# Ajout d’outils

**A placer à la fin du script !**

```
// Boutons de navigation

var nav = new mapboxgl.NavigationControl();
map.addControl(nav, 'top-left');
```

https://docs.mapbox.com/mapbox-gl-js/api/markers/#geolocatecontrol
https://docs.mapbox.com/mapbox-gl-js/api/markers/#navigationcontrol

```
// Ajout Echelle cartographique
map.addControl(new mapboxgl.ScaleControl({
    maxWidth: 120,
    unit: 'metric'}));
```

https://docs.mapbox.com/mapbox-gl-js/api/markers/#scalecontrol

```
// Bouton de géolocalisation

map.addControl(new mapboxgl.GeolocateControl
({positionOptions: {enableHighAccuracy: true},
trackUserLocation: true,
showUserHeading: true}));
```

https://docs.mapbox.com/mapbox-gl-js/api/markers/?size=n_10_n#geolocatecontrol

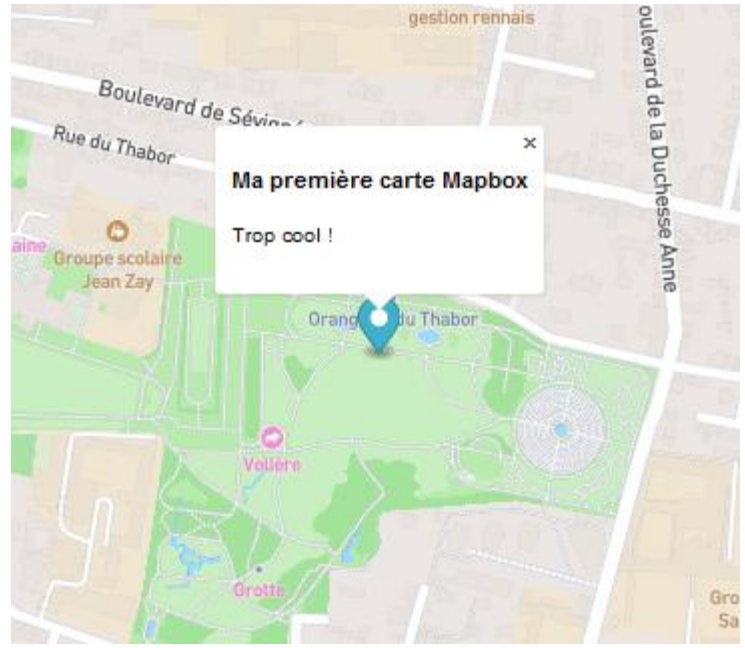
# Ajouter des données

OSM, données hébergées et données en local

# Ajouter un marqueur

- Ajouter marqueur + définir contenu et Popup

```
// Ajout Marqueur
const marker1 = new mapboxgl.Marker()
.setLngLat([-1.669099, 48.114799])
.addTo(map);

// Contenu de la popup du marqueur
  var popup = new mapboxgl.Popup({ offset: 25 })
    .setHTML("<h3>Ma première carte Mapbox</h3><p>Trop cool !</p>");

  // Associer Contenu et Marqueur
  marker1.setPopup(popup);
```




https://codepen.io/BorisMericskay/pen/XWGyqqx

# Ajout de données OSM

- MapboxGL permet aussi de mobiliser des données OSM
  - Le jeux de données (*tileset*) Mapbox Streets v8 propose un ensemble de couches OSM (routes, bâtiments, labels, hydrologie,…)


# Ajout de données OSM

- Documentation

https://docs.mapbox.com/vector-tiles/reference/mapbox-streets-v8/

# Ajout de données OSM

map.on('load', callback) est un événement dans Mapbox GL JS qui se déclenche lorsque la carte est entièrement chargée et prête à être utilisée. Cela inclut le chargement de la carte de base, des tuiles de carte, des sources de données, des couches et des styles.

- Ajouter cette commande à la fin du script

On appel ici les routes issues d'OSM

```
map.on('load', function () {

 map.addSource('mapbox-streets-v8', {
  type: 'vector',
  url: 'mapbox://mapbox.mapbox-streets-v8'});

 map.addLayer({
      "id": "Routes",
      "type": "line",
      "source": "mapbox-streets-v8",
      "layout": {'visibility': 'visible'},
      "source-layer": "road",
      "paint": {"line-color": "#FF7F50", "line-width": 1}
   });

  });
```

# Ajout de données OSM

- Ajouter cette commande à la fin du script

On appel ici les routes

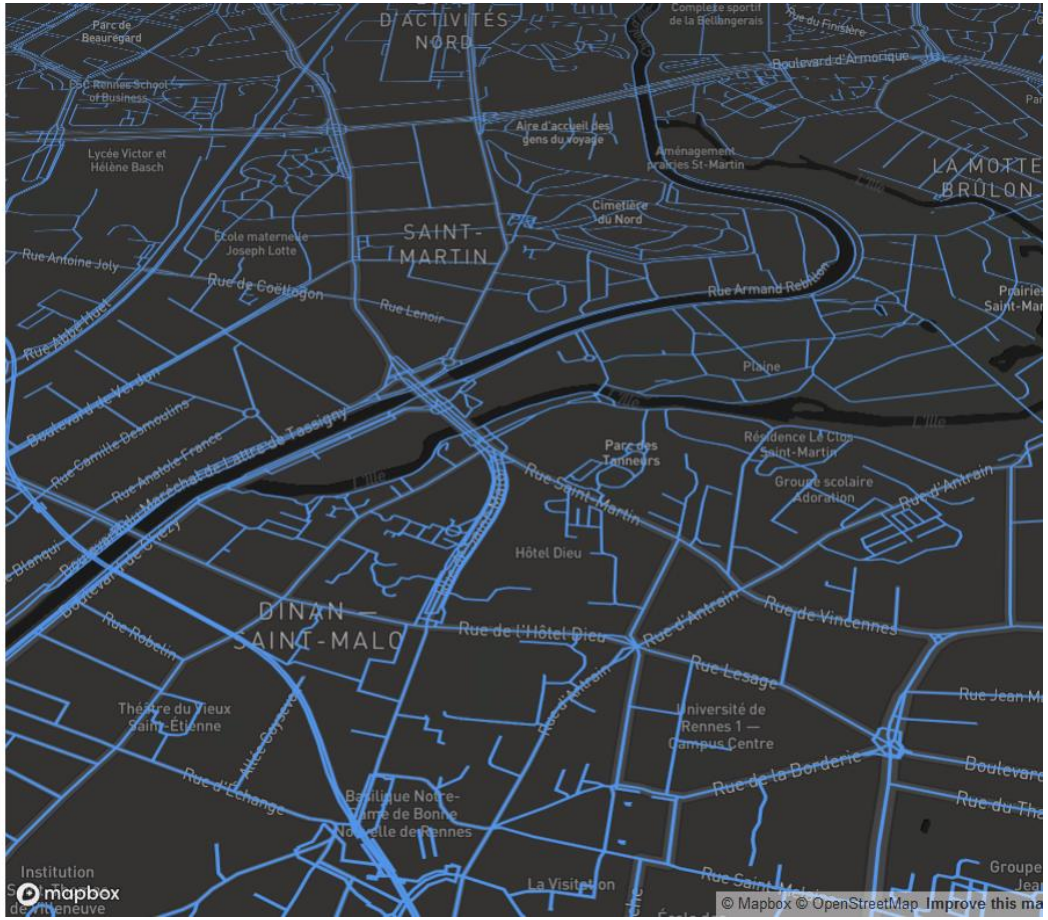
```
map.on('load', function () {

map.addSource('mapbox-streets-v8', {
 type: 'vector',
 url: 'mapbox://mapbox.mapbox-streets-v8'});

 map.addLayer({
      "id": "Routes",
      "type": "line",
      "source": "mapbox-streets-v8",
      "layout": {'visibility': 'visible'},
      "source-layer": "road",
      "paint": {"line-color": "#FF7F50", "line-width": 1}
  });
  });
```

Définition de la source de données

Couche de données

# Ajout de données OSM

```
<script>
    // AccesToken
mapboxgl.accessToken =
'pk.eyJ1IjoibmluYW5vdW4iLCJhIjoiY2pjdHBoZGlzMnV4dDJxcGc5azJkbWRiYSJ9.o4dZRrdHcgVEKCveOXG1YQ';

    // Configuration de la carte
var map = new mapboxgl.Map({
    container: 'map',
    style: 'mapbox://styles/mapbox/dark-v9', // Fond de carte
    center: [-1.68, 48.12], // lat/long
    zoom: 15, // zoom
    pitch: 50, // Inclinaison
    bearing: -10 // Rotation
});

  map.on('load', function () {

 map.addSource('mapbox-streets-v8', {
   type: 'vector',
   url: 'mapbox://mapbox.mapbox-streets-v8'});

  map.addLayer({
        "id": "Routes",
        "type": "line",
        "source": "mapbox-streets-v8",
        "layout": {'visibility': 'visible'},
        "source-layer": "road",
        "paint": {"line-color": "#5496eb", "line-width": 2}
    });

   });

</script>

</body>
</html>
```



# Ajout de données OSM

- Ajout du réseau hydrographique
  - Ajouter à la suite de l'appel de la couche des routes juste un appel de couche car la source est la même que pour les routes (*mapbox-streets-v7*)

```
// Hydrologie

map.addLayer({"id": "hydrologie",
                        "type": "line",
                        "source": "mapbox-streets-v8",
                        "source-layer": "waterway",
                        "paint": {"line-color": "#4dd2ff",
                        "line-width": 6}
    });
```

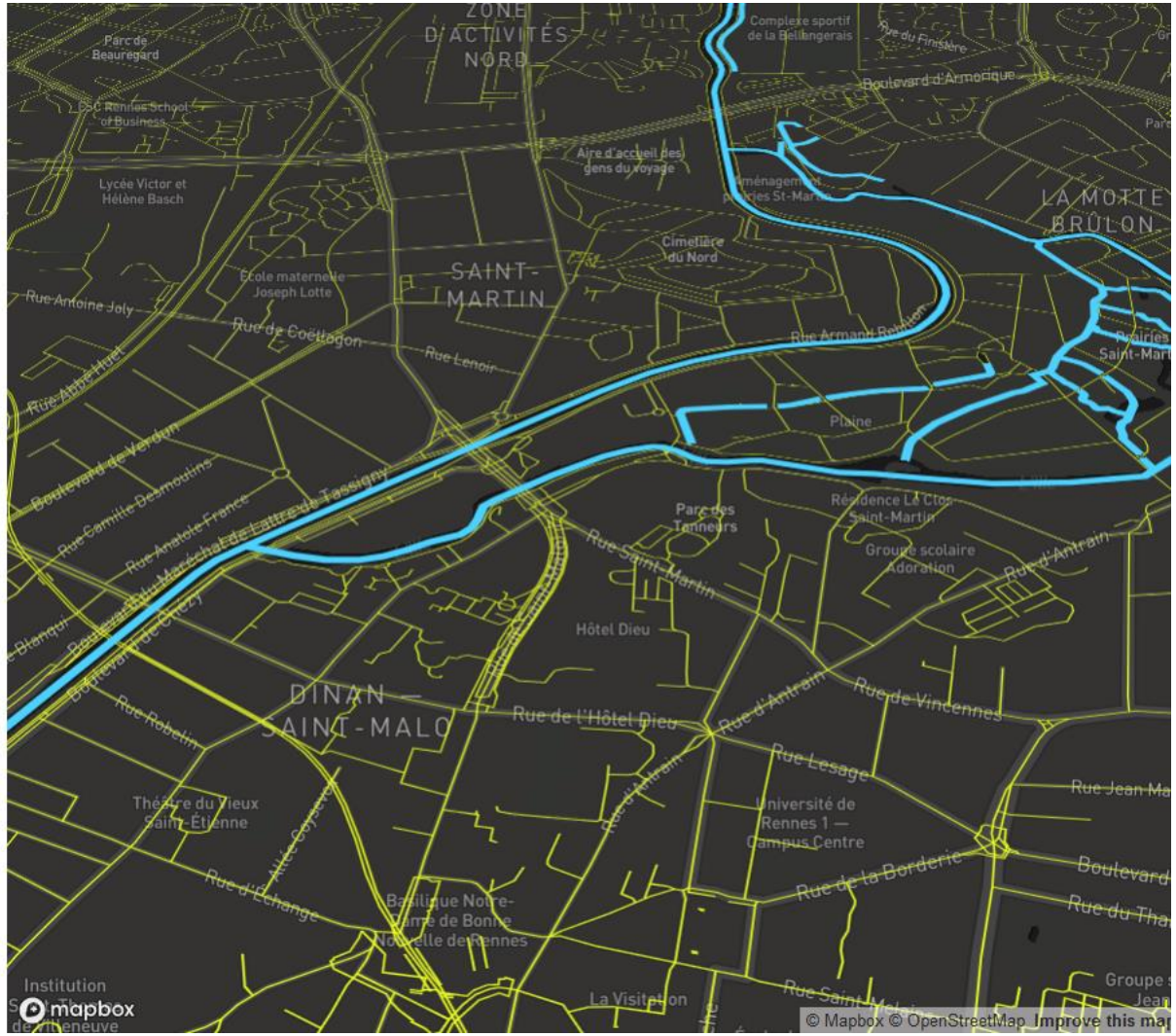
# Gérer fonds de carte ET données !

- Demander de régler le problème à l'IA


- Solution : https://codepen.io/BorisMericskay/pen/xbxbVLo

# Ajout de données OSM

```
    center: [-1.68, 48.12], // lat/long
    zoom: 15, // zoom
    pitch: 50, // Inclinaison
    bearing: -10 // Rotation
});

  map.on('load', function () {

 map.addSource('mapbox-streets-v8', {
   type: 'vector',
   url: 'mapbox://mapbox.mapbox-streets-v8'});

  map.addLayer({
        "id": "Routes",
        "type": "line",
        "source": "mapbox-streets-v8",
        "layout": {'visibility': 'visible'},
        "source-layer": "road",
        "paint": {"line-color": "#ebff1e", "line-width": 1}
    });

    // Hydrologie

   map.addLayer({"id": "hydrologie",
                        "type": "line",
                        "source": "mapbox-streets-v8",
                        "source-layer": "waterway",
                        "paint": {"line-color": "#4dd2ff",
                        "line-width": 7}
    });

   });

</script>

</body>
</html>
```



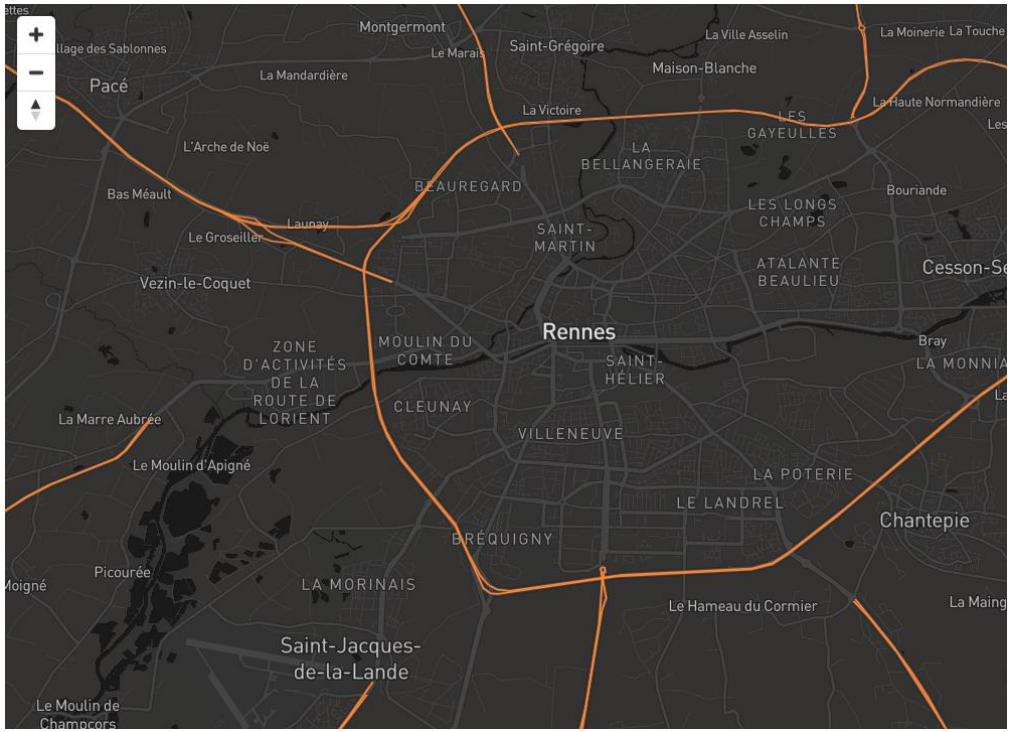
# Filtrer des données OSM

- Il est possible de filtrer les données pour l'affichage
  - On peux par exemple filtrer les routes selon leur classe

# Filtrer des données OSM

- Je ne veux afficher que les routes à double sens

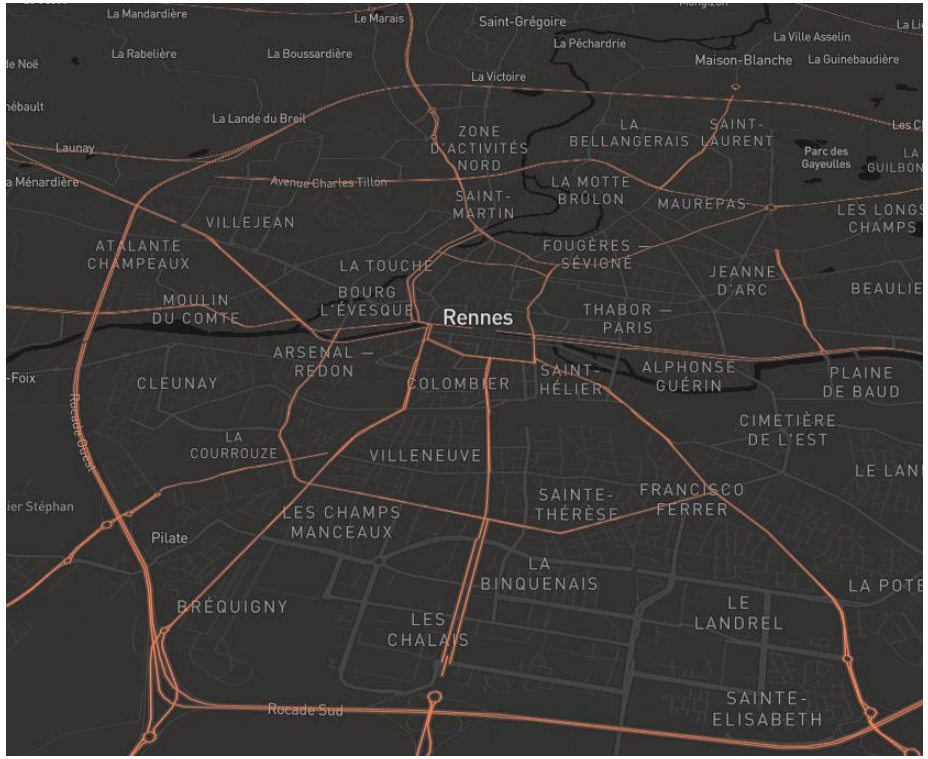
```
map.addLayer({
        "id": "routes",
        "type": "line",
        "source": "mapbox-streets-v7",
        "source-layer": "road",
        "filter": ['==', 'class', 'trunk'],
        "layout": {'visibility': 'visible'},
        "paint": {"line-color": "#ff8533", "line-width": 1.3}
    });
```



# Filtrer des données OSM

- Je ne veux afficher que les routes principale (double sens, principale,…)

```
map.addLayer({
        "id": "routes",
        "type": "line",
        "source": "mapbox-streets-v7",
        "source-layer": "road",
        "filter": ["all",  ["in", "class", "motorway", "trunk", "primary"]],
        "layout": {'visibility': 'visible'},
        "paint": {"line-color": "#ff8533", "line-width": 1.3}
    });
```



# Ajout de données OSM

- Ajout les bâtiments (ajouter juste un appel de couche car la source est la même que pour les routes)

```
// Batiments

map.addLayer({
        "id": "batiments",
        "type": "fill",
        "source": "mapbox-streets-v8",
        "source-layer": "building",
        "paint": {"fill-color": "#FEFEFE",
                  "fill-opacity": 0.8}
    });
```

# Exemple

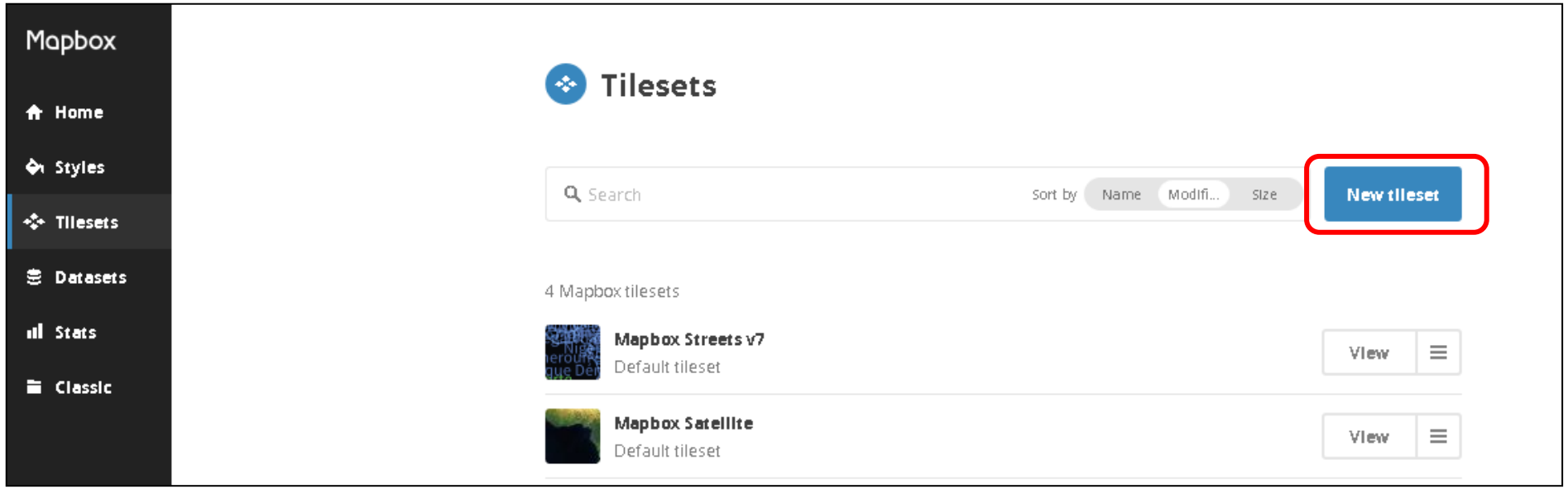
#MapboxGL / Afficher et filtrer des données d'OSM


Built with blockbuilder.org Open

https://codepen.io/BorisMericskay/pen/abjRBav

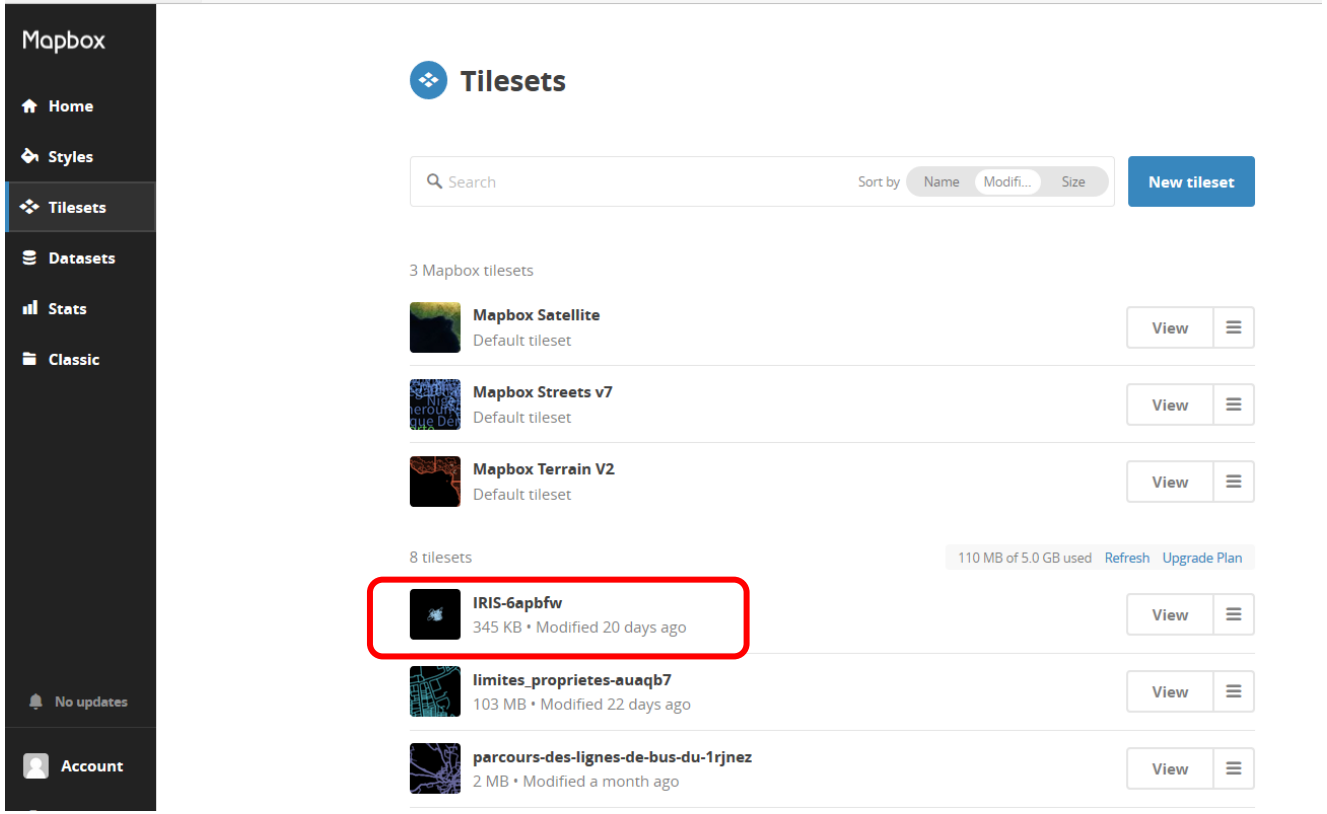
# Ajout de données personnelles

- Première étape charger des données comme des Tilesets dans le Studio de Mapbox (csv, geojson, gpx, kml, shapefile zippé)
  - Intégrer le jeu de données des arrêts de bus et celui de la base équipements



# Ajout de données personnelles

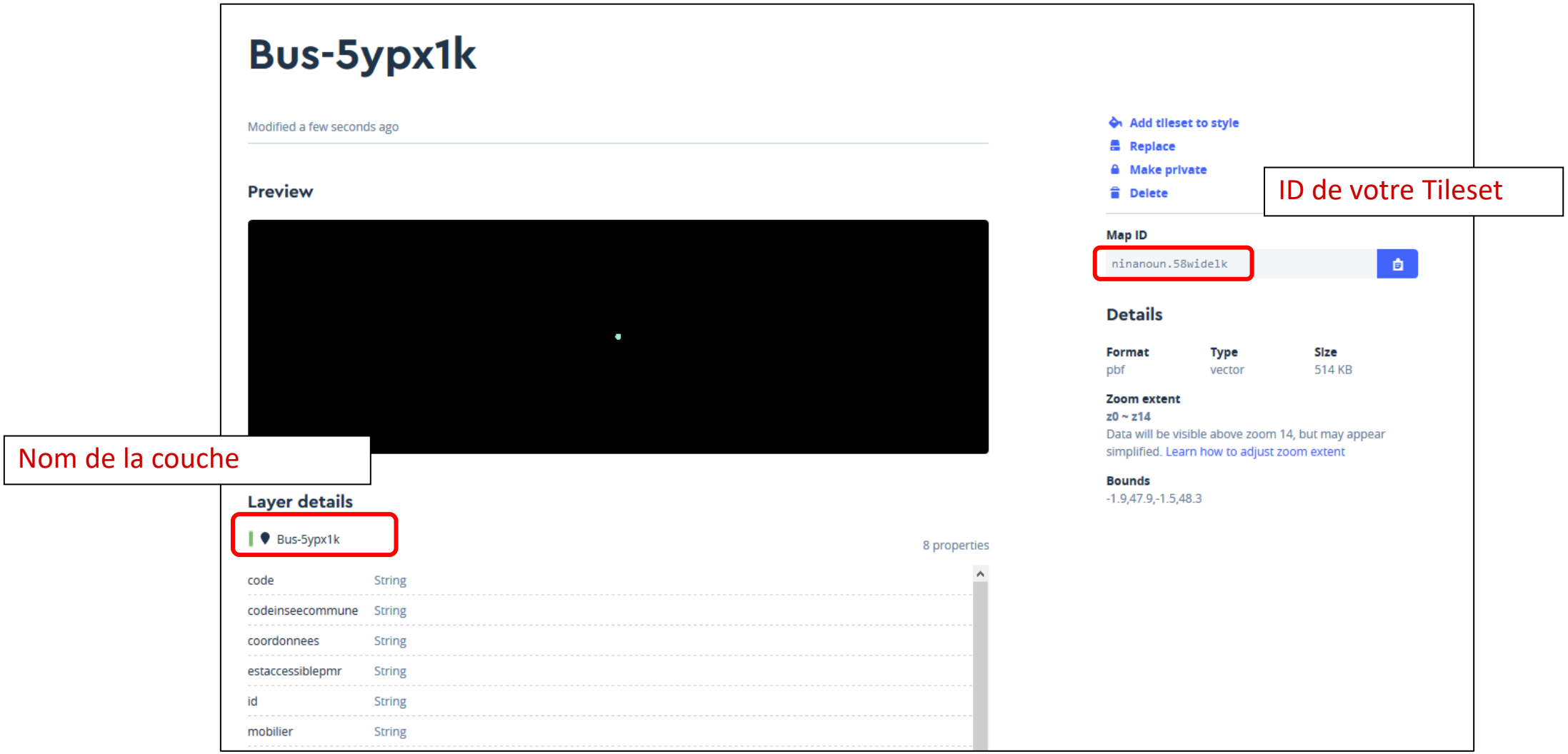
- Aller chercher les infos dans le studio de Mapbox (Tilesets)



# Ajout de données personnelles

```
//Mes datas

map.addSource('nomdelasource', {
        type: 'vector',
        url: 'mapbox://' iddutileset'});

map.addLayer({
        'id': 'nomgenerique',
        'type': 'circle',
        'source': 'nomdelasource ',
        'source-layer': 'nomdelacouche',
        'layout': {'visibility': 'visible'},
        'paint': {'circle-radius': {'base': 1.5,'stops': [[13, 2], [22, 60]]}, 'circle-color': '#f31616',}, minzoom:10
});
```

# Ajout de données personnelles



# Ajout des arrêts de bus

```
// Arret de bus

map.addSource('Arrets', {
      type: 'vector',
      url: 'mapbox://ninanoun.58widelk'});

map.addLayer({
      'id': 'Arrets',
      'type': 'circle',
      'source': 'Arrets',
      'source-layer': 'Bus-5ypx1k',
      'layout': {'visibility': 'visible'},
      'paint': {'circle-radius': {'base': 1.5,'stops': [[13, 2], [22, 60]]}, 'circle-color': '#f31616',}, minzoom:12
   });
```

Bien renseigner l'ID de votre Tilesets

Bien renseigner le nom la couche du tileset

# Ajout des arrêts de bus

```
<script>
    // AccesToken
mapboxgl.accessToken = 'pk.eyJ1Ijoibmluyw5vdW4iLCJhIjoiY2pjdHBoZG1zMnV4dDJxcGc5azJkbWRiYSJ9.o4dZRrdHcgVEKCveOXG1Y

    // Configuration de la carte
var map = new mapboxgl.Map({
    container: 'map',
    style: 'mapbox://styles/mapbox/dark-v9',
    center: [-1.68, 48.12], // lat/long
    zoom: 15, // zoom
    pitch: 50, // Inclinaison
    bearing: -10 // Rotation
});

map.on('load', function () {

 map.addSource('mapbox-streets-v7', {
   type: 'vector',
   url: 'mapbox://mapbox.mapbox-streets-v7'});

  map.addLayer({
        "id": "routes",
        "type": "line",
        "source": "mapbox-streets-v7",
        "layout": {'visibility': 'visible'},
        "source-layer": "road",
        "paint": {"line-color": "#FF7F50", "line-width": 1}
    });

    map.addSource('Arrets', {
        type: 'vector',
        url: 'mapbox://ninanoun.58widelk'});

map.addLayer({
        'id': 'Arrets',
        'type': 'circle',
        'source': 'Arrets',
        'source-layer': 'Bus-5ypx1k',
        'layout': {'visibility': 'visible'},
        'paint': {'circle-radius': 5, 'circle-color': '#f5f60d',}
    });


  });

</script>
```

# Ajout la couche équipements publics

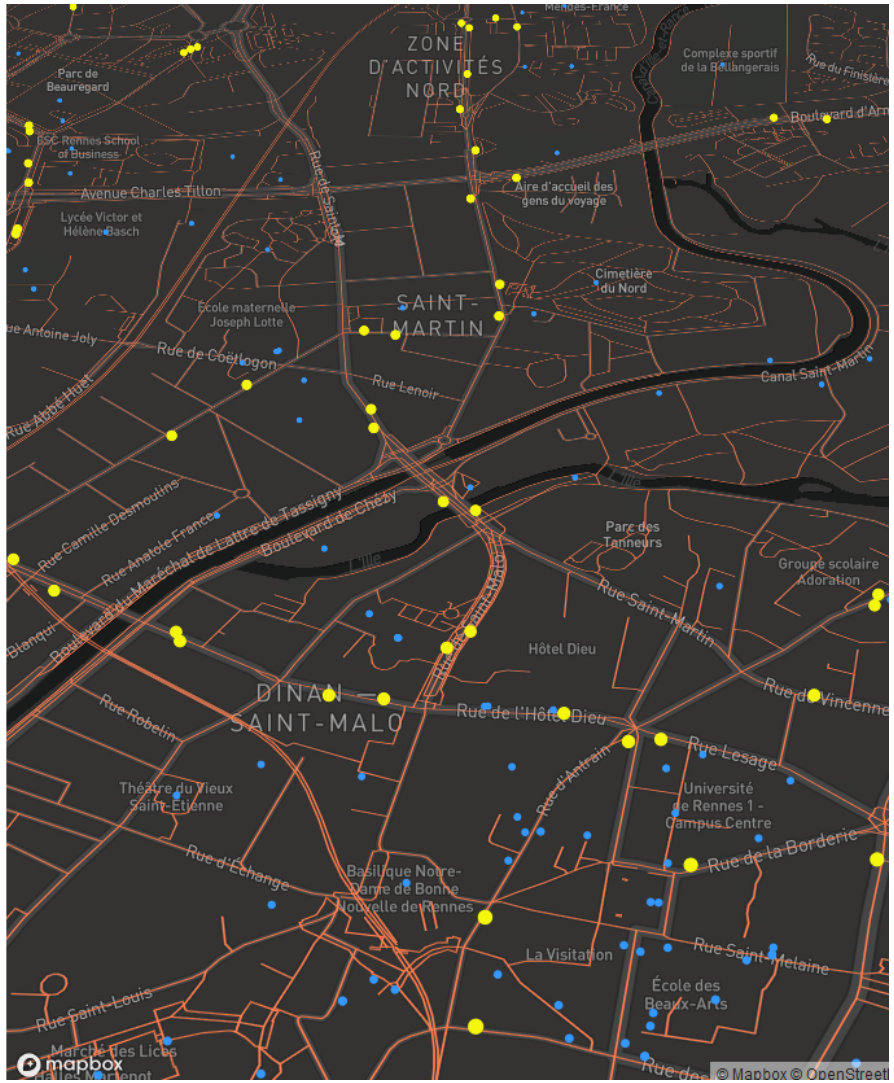
```
// Equipements publics

map.addSource('Equipements', {
      type: 'vector',
      url: 'mapbox://ninanoun.4xcn5ude'});

map.addLayer({
      'id': 'Equipements',
      'type': 'circle',
      'source': 'Equipements',
      'source-layer': 'base-orga-var-6k0zky',
      'layout': {'visibility': 'visible'},
      'paint': {'circle-radius': {'base': 1.5,'stops': [[13, 2], [22, 60]]}, 'circle-color': '#16f337'}, minzoom:14
   });
```

Bien renseigner l'ID de votre Tilesets

Bien renseigner le nom de la couche a afficher

# Ajout la couche équipements

```
map.on('load', function () {

 // Ajout routes OSM

 map.addSource('mapbox-streets-v7', {
   type: 'vector',
   url: 'mapbox://mapbox.mapbox-streets-v7'});

  map.addLayer({
        "id": "routes",
        "type": "line",
        "source": "mapbox-streets-v7",
        "layout": {'visibility': 'visible'},
        "source-layer": "road",
        "paint": {"line-color": "#FF7F50", "line-width": 1}
    });

// Ajout arrets de bus

map.addSource('Arrets', {
        type: 'vector',
        url: 'mapbox://ninanoun.58widelk'});

map.addLayer({
        'id': 'Arrets',
        'type': 'circle',
        'source': 'Arrets',
        'source-layer': 'Bus-5ypx1k',
        'layout': {'visibility': 'visible'},
        'paint': {'circle-radius': 5, 'circle-color': '#f5f60d'}
    });

// Ajout couche équipements

map.addSource('Equipements', {
        type: 'vector',
        url: 'mapbox://ninanoun.4xcn5ude'});

map.addLayer({
        'id': 'Equipements',
        'type': 'circle',
        'source': 'Equipements',
        'source-layer': 'base-orga-var-6k0zky',
        'layout': {'visibility': 'visible'},
        'paint': {'circle-radius': 3, 'circle-color': '#3399ff'}
    });

  });

</script>
```



# Mettre en forme les données

- Pour personnaliser la symbologie des données se référer à la documentation

  https://www.mapbox.com/mapbox-gl-js/style-spec/#layers

Type d'objets géographiques dans MapboxGL :

- circle (point)
- symbol (point avec pictogramme)
- line (ligne)
- fill (polygone)
- fill-extrusion (polygone 3D)
- …

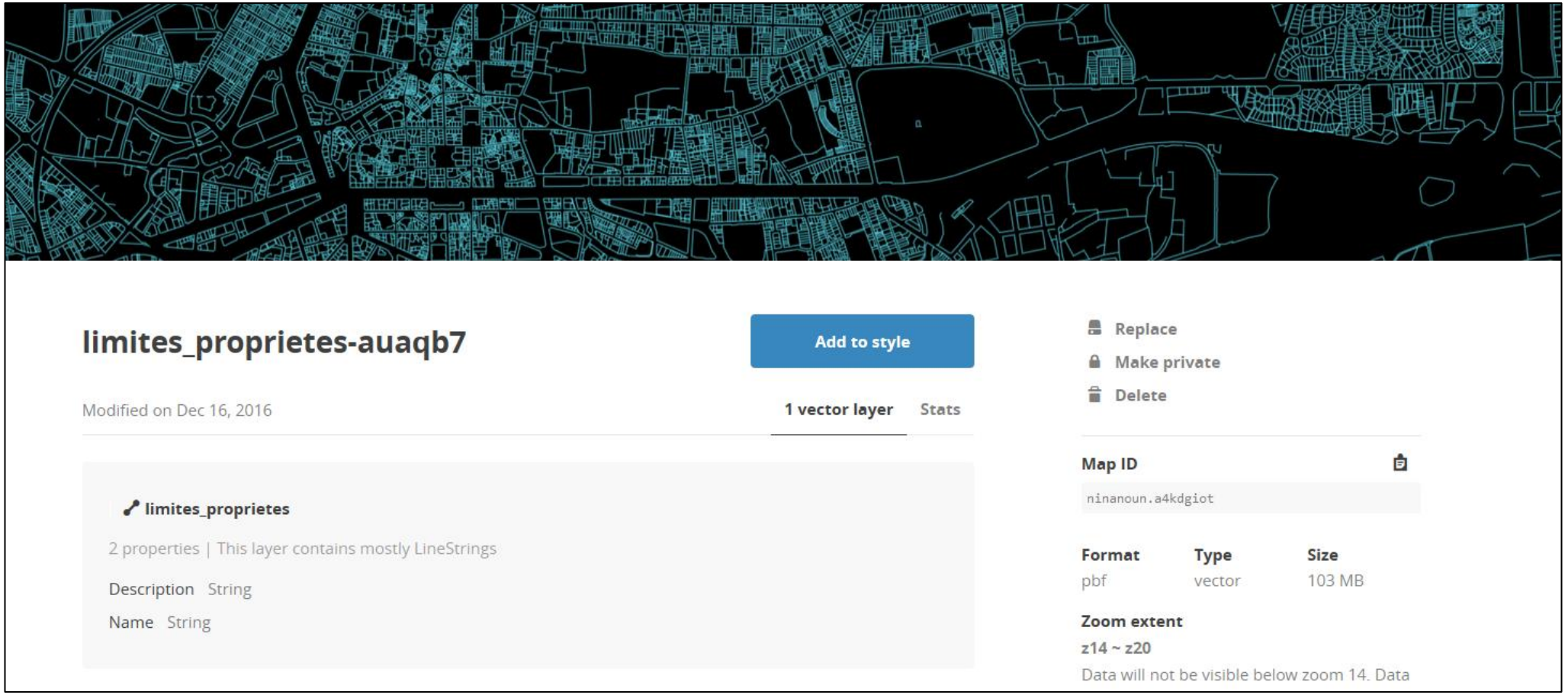
# Mettre en forme les données

- Changer la taille
- Changer la couleur

  http://www.code-couleur.com/
- Définir des niveaux de zoom (max/min)

```
'paint': {'circle-radius': {'base': 1.5,'stops': [[13, 2], [22, 60]]}, 'circle-color': '#3399ff'}, minzoom:14
```

# Ajouter les limites de propriétés

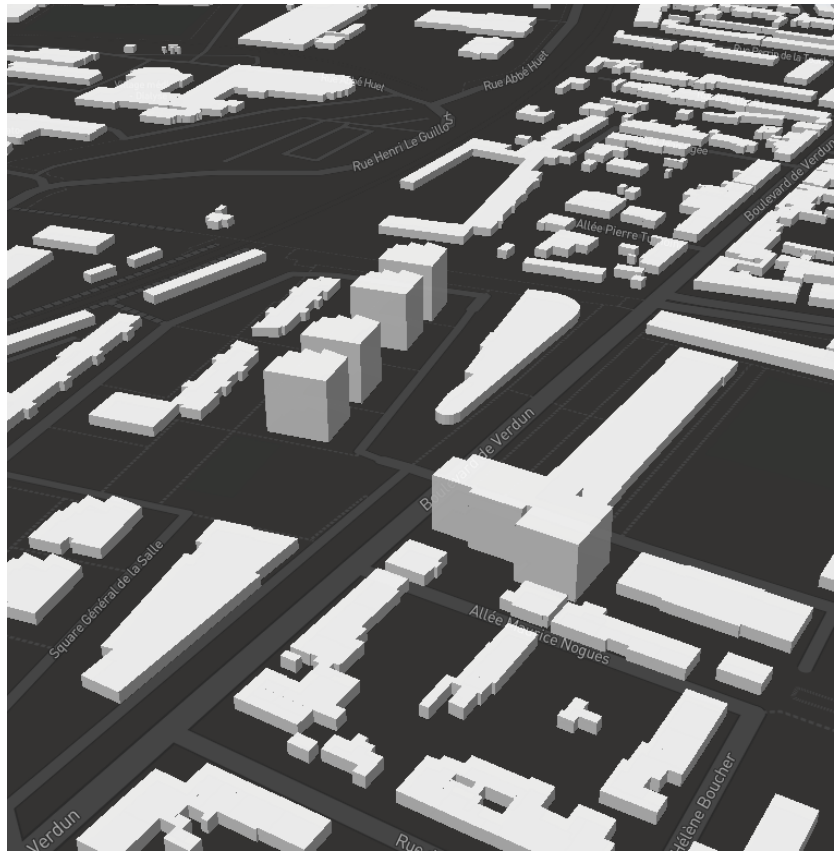


# Ajouter les limites de propriétés

```
//Proprietes

map.addSource('Proprietes', {
        type: 'vector',
        url: 'mapbox://ninanoun.a4kdgiot'
    });

map.addLayer({
        'id': 'Proprietes',
        'type': 'line',
        'source': 'Proprietes',
        'source-layer': 'limites_proprietes',
        'layout': {'visibility': 'visible',
        'line-join': 'round','line-cap': 'round'},
        'paint': {'line-color': '#FFFFFF', 'line-width': 0.5}
});
```

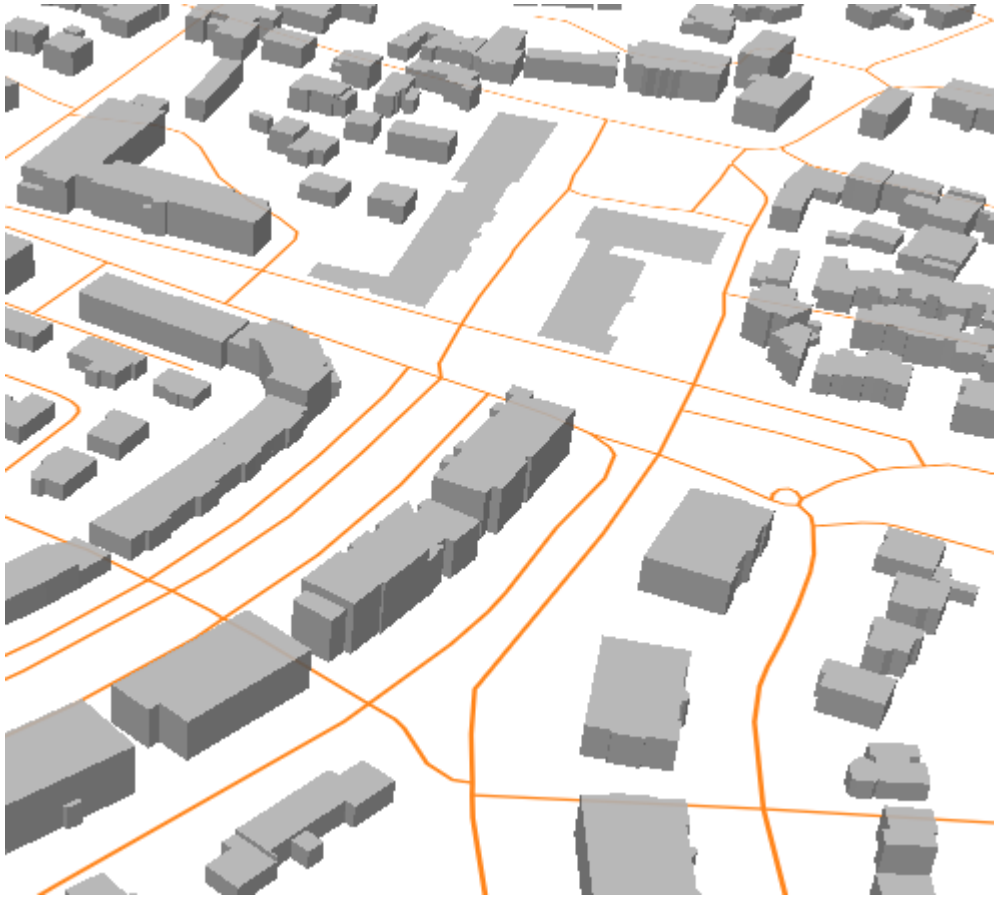
# Ajouts de données 3D – OSM Bâti

```
// Ajout batiments 3D OSM

map.addLayer({
        'id': 'Batiments_3D',
        'source': 'composite',
        'source-layer': 'building',
        'filter': ['==', 'extrude', 'true'],
        'type': 'fill-extrusion',
        'minzoom': 15,
        'paint': {'fill-extrusion-color': '#555555', 'fill-extrusion-height':
        {'type': 'identity','property': 'height'},
        'fill-extrusion-base': {'type': 'identity','property': 'min_height'},
        'fill-extrusion-opacity': 0.8
        }
    });
```

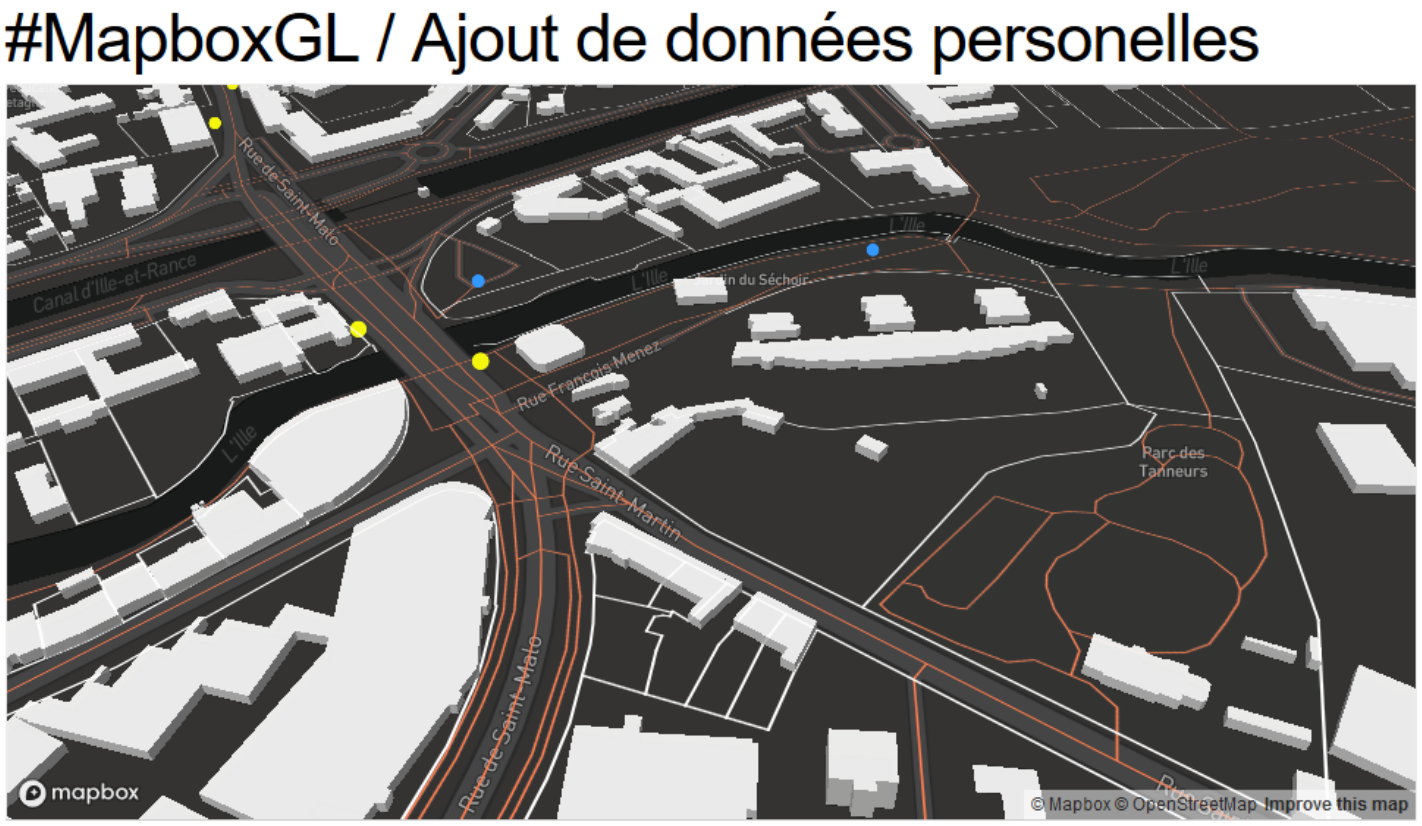


# Ajouts de données 3D –BDTOPO Bâti

```
 //BATIMENTS

map.addSource('Batiments', {
        type: 'vector',
        url: 'mapbox://mastersigat.a4h4ovrl'
    });



  map.addLayer({
    'id': 'Batiments',
    'type': 'fill-extrusion',
    'source': 'Batiments',
    'source-layer': 'batiIGN-8zf03o',
    'layout': {'visibility': 'visible'},
                  'paint':
                                  {'fill-extrusion-color': '#A9A9A9',
                                   'fill-extrusion-height':{'type': 'identity','property':
'HAUTEUR'},
                                   'fill-extrusion-opacity': 0.90,
           'fill-extrusion-base': 0}
    });
```



# Exemple




https://codepen.io/BorisMericskay/pen/YzjJpdQ

# Ajouter des données en local / en ligne

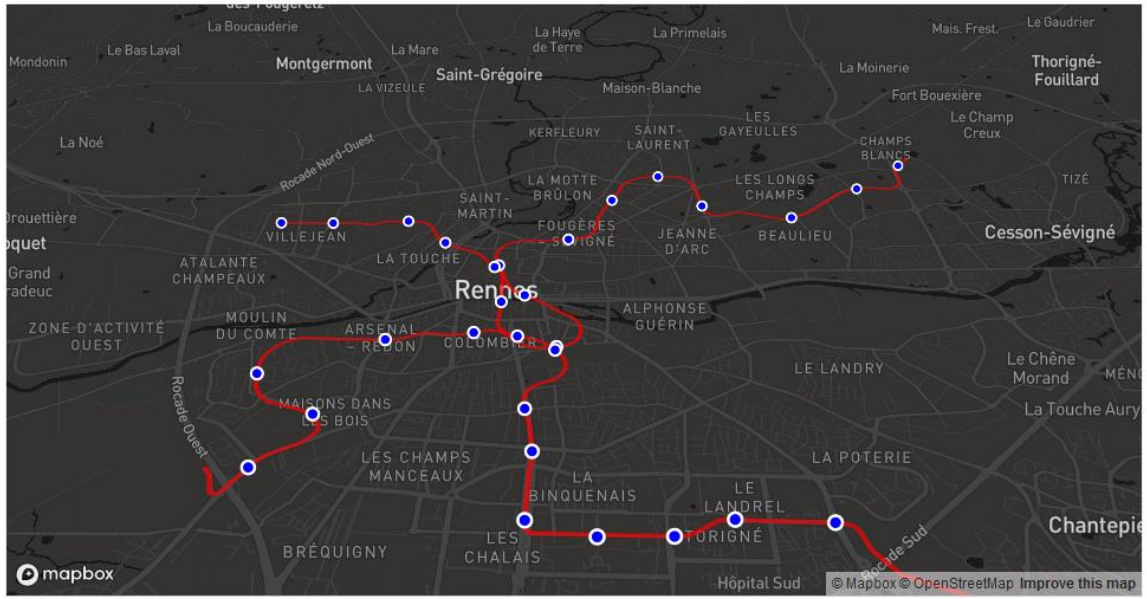
- Il est possible de mobiliser des jeux de données (Geojson) stockés en local (même dossier que la page html) ou accessible via une URL

```
map.on("load", function() {
//Couche EPCI
map.addLayer({
        id: "epci",
        type: "line",
        source: {type: "geojson",
                    data: "./epci.geojson" },

        paint: {'line-color': '#000000',
                    'line-width':1}
        });
 });
```

```
map.on("load", function() {
//Couche EPCI
map.addLayer({
        id: "epci",
        type: "line",
        source: {type: "geojson",
                    data: 'URL'
        paint: {'line-color': '#000000',
                    'line-width':1}
        });
 });
```

# Ajouter des données issues de GitHub/Dropbox

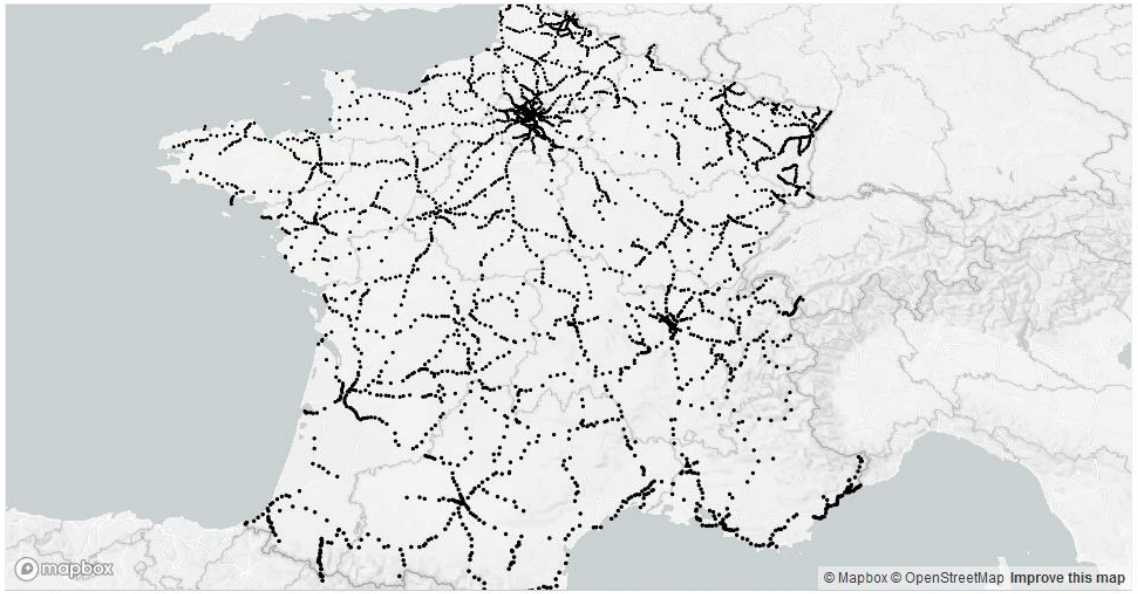
## #MapboxGL / Ajouter des Geojson - hébergés sur Github




Built with blockbuilder.org Open

https://codepen.io/BorisMericskay/pen/jOpeVJd

## #MapboxGL / Afficher des Geojson hébérgés sur une Dropbox

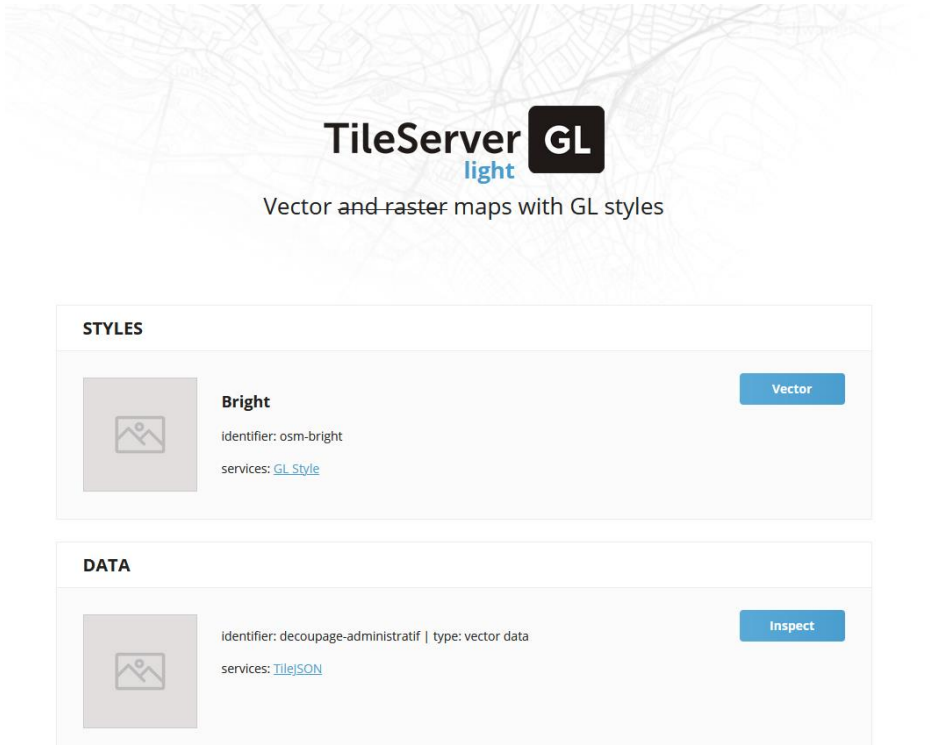



Built with blockbuilder.org Open

https://codepen.io/BorisMericskay/pen/gOjBgpK

# Ajout de tuiles vectorielles externes

- https://openmaptiles.geo.data.gouv.fr/



# Ajouter des données externes

- Données sur un portail opendata (Geojson)

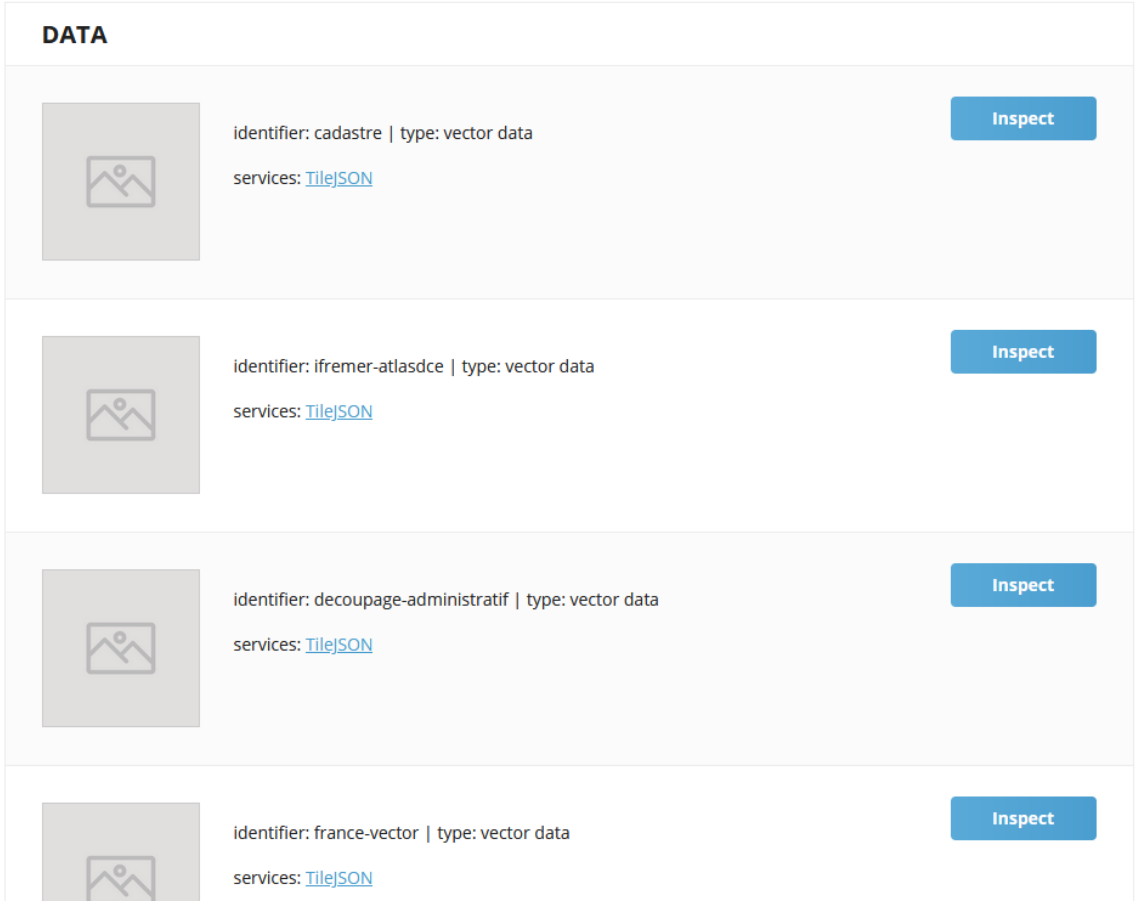
```
// Ajout lignes de metros

    map.addSource('lignes', {
          type: 'geojson',
          data: 'https://data.rennesmetropole.fr/api/explore/v2.1/catalog/datasets/metro-du-reseau-star-traces-de-laxe-des-lignes/exports/geojson'
        });

    map.addLayer({
          'id': 'lignesmetros',
          'type': 'line',
          'source': 'lignes',
          'paint': {'line-opacity': 0.7, 'line-width': 3.5,
                    'line-color': 'red'}
    });
```

# Ajouter des données externes

- Données sur Github (Geojson)

```
// Ajout lignes de metros

    map.addSource('lignes', {
          type: 'geojson',
          data: 'https://raw.githubusercontent.com/mastersigat/data/main/metro-du-reseau-star-traces-de-laxe-des-lignes.geojson'
       });

    map.addLayer({
          'id': 'lignesmetros',
          'type': 'line',
          'source': 'lignes',
          'paint': {'line-opacity': 0.7, 'line-width': 3.5,
                    'line-color': 'red'}
    });
```

# Ajout du cadastre (flux de tuiles vectorielles)

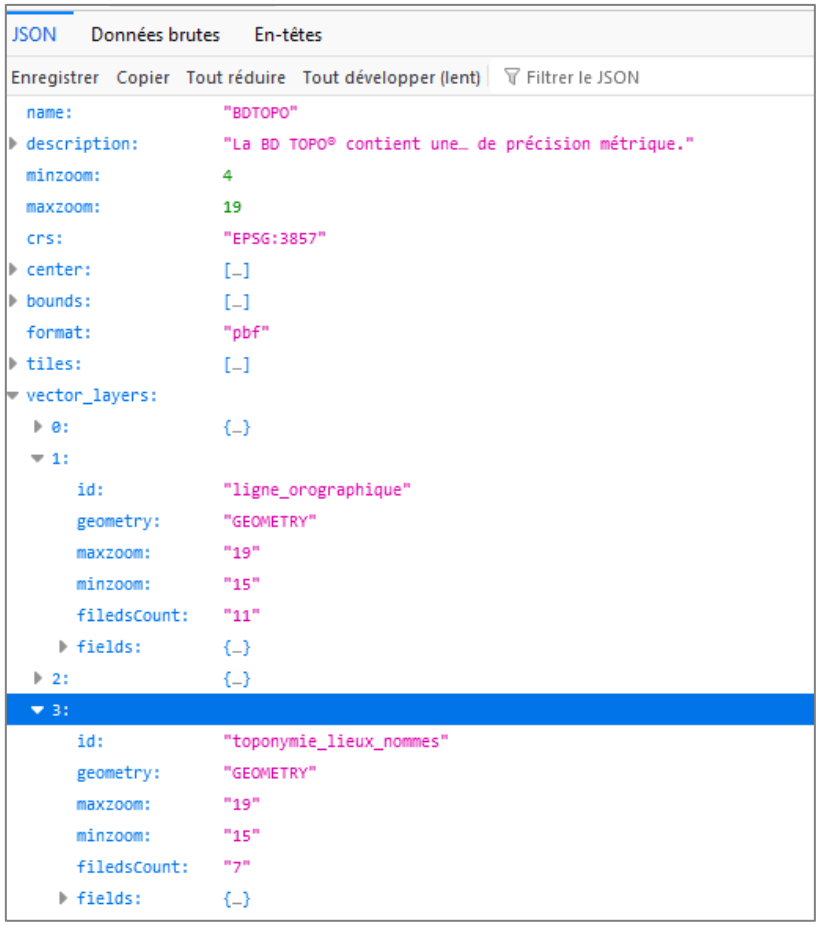



https://openmaptiles.geo.data.gouv.fr/

# Ajout du cadastre (flux de tuiles vectorielles)

```
 // AJOUT DU CADASTRE ETALAB

map.addSource('Cadastre', {
                        type: 'vector',
                        url: 'https://openmaptiles.geo.data.gouv.fr/data/cadastre.json'    });



map.addLayer({
                       'id': 'Cadastre',
                       'type': 'line',
                       'source': 'Cadastre',
                       'source-layer': 'parcelles',
                       'layout': {'visibility': 'visible'},
                       'paint': {'line-color': '#000000'},
                       'minzoom':16, 'maxzoom':19 });

map.setPaintProperty('communeslimites', 'line-width', ["interpolate",["exponential",1],["zoom"],16,0.3,18,1]);
```

# Ajouter des données externes / IGN


https://geoservices.ign.fr/documentation/services/api-et-services-ogc/tuiles-vectorielles-tmswmts/connexion

# Ajouter des données externes / IGN

- Données en tuiles vectorielles

```
// Ajout BDTOPO

map.addSource('BDTOPO', {
type: 'vector',
url: 'https://wxs.ign.fr/topographie/geoportail/tms/1.0.0/BDTOPO/metadata.json',
minzoom: 15,
maxzoom: 19
    });

 map.addLayer({
'id': 'batiments',
'type': 'fill-extrusion',
'source': 'BDTOPO',
'source-layer': 'batiment',
'paint': {'fill-extrusion-color': '#A9A9A9',
          'fill-extrusion-height':{'type': 'identity','property': 'hauteur'},
          'fill-extrusion-opacity': 0.90,
          'fill-extrusion-base': 0}
});
```

# Ajouter des données externes

https://codepen.io/BorisMericskay/pen/MWLjJqx

# Ajouter des données externes via des API’s

- URL des API’s
  - Contour d’une commune: https://api.gouv.fr/documentation/api_carto_cadastre
  - RPG : https://api.gouv.fr/documentation/api_carto_rpg
  - PLU : https://apicarto.ign.fr/api/doc/gpu#/
  - Parkings relais : https://data.explore.star.fr/explore/dataset/tco-parcsrelais-star-etat-tr/api/
  - VLS : https://data.explore.star.fr/explore/dataset/vls-stations-etat-tr/api/

- Besoin de jQuery (https://fr.wikipedia.org/wiki/JQuery)

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js" ></script>
```

# API Carto Cadastre

- Contours commune: https://api.gouv.fr/documentation/api_carto_cadastre

```
dataCadastre = 'https://apicarto.ign.fr/api/cadastre/commune?code_insee=35238';

 jQuery.when( jQuery.getJSON(dataCadastre)).done(function(json) {
   for (i = 0; i < json.features.length; i++) {
     json.features[i].geometry = json.features[i].geometry;
   };

map.addLayer(
{'id': 'Contourcommune',
 'type':'line',
 'source': {'type': 'geojson','data': json},
 'paint' : {'line-color': 'black',
            'line-width':2.5},
 'layout': {'visibility': 'visible'},
   });

 });
```

# API Carto RPG

- RPG : https://api.gouv.fr/documentation/api_carto_rpg

```
dataRPG =
'https://apicarto.ign.fr/api/rpg/v2?annee=2021&geom=%7B%22type%22%3A%20%22Point%22%2C%22coordinates%22%3A%5B-
1.647%2C48.146%5D%7D&_limit=100';

 jQuery.when( jQuery.getJSON(dataRPG)).done(function(json) {
   for (i = 0; i < json.features.length; i++) {
     json.features[i].geometry = json.features[i].geometry;
   };

   map.addLayer(
   { 'id': 'RPG',
     'type':'fill',
     'source': {'type': 'geojson','data': json},
     'paint' : {'fill-color': 'red'},
     'layout': {'visibility': 'none'}
   });

 });
```

# API Carto GPU

- PLU : https://apicarto.ign.fr/api/doc/gpu#/

```
dataPLU = 'https://apicarto.ign.fr/api/gpu/zone-urba?partition=DU_243500139';

jQuery.when(jQuery.getJSON(dataPLU)).done(function(json) {
  // Filtrer les entités pour ne garder que celles avec typezone = 'U'
  var filteredFeatures = json.features.filter(function(feature)
  {return feature.properties.typezone === 'N';});

  // Créer un objet GeoJSON avec les entités filtrées
  var filteredGeoJSON = { type: 'FeatureCollection', features: filteredFeatures};

  map.addLayer({
    'id': 'PLU',
    'type': 'fill',
    'source': {'type': 'geojson',
               'data': filteredGeoJSON},
    'paint': {'fill-color': 'green',
              'fill-opacity': 0.5},
  });

 });
```

# API STAR

- Parkings relais : https://data.explore.star.fr/explore/dataset/tco-parcsrelais-star-etat-tr/api/

```
$.getJSON('https://data.rennesmetropole.fr/api/explore/v2.1/catalog/datasets/tco-parcsrelais-star-etat-tr/records?limit=20',
function(data) {var geojsonData4 = {
            type: 'FeatureCollection',
            features: data.results.map(function(element) {
               return {type: 'Feature',
                     geometry: {type: 'Point',
                     coordinates: [element.coordonnees.lon, element.coordonnees.lat]},
                     properties: { name: element.nom,
                                 capacity: element.jrdinfosoliste}};
            })
        };

  map.addLayer({ 'id': 'Parcrelais',
              'type':'circle',
              'source': {'type': 'geojson',
                           'data': geojsonData4},
             'paint': {'circle-color': 'orange'}
   });
 });
```

# API STAR

- VLS : https://data.explore.star.fr/explore/dataset/vls-stations-etat-tr/api/

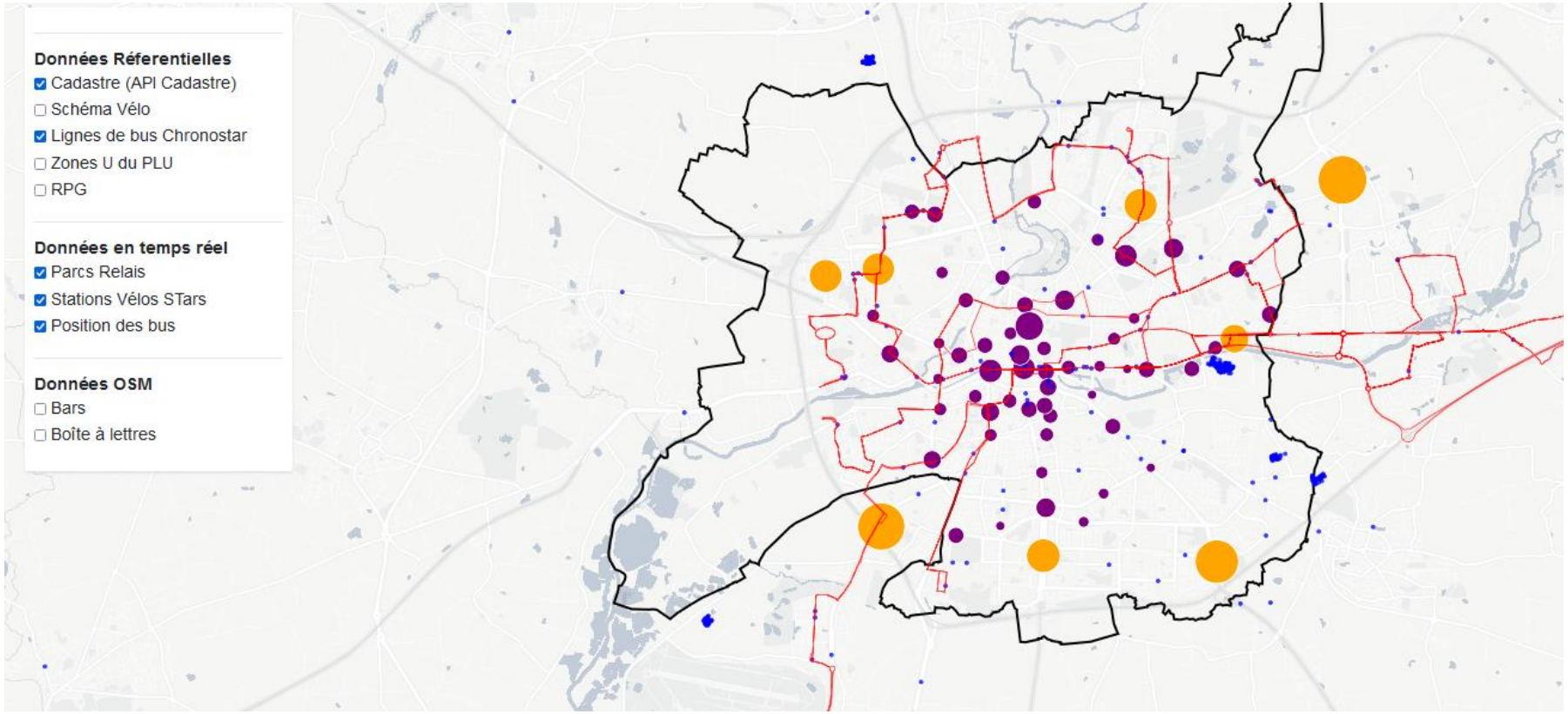
- Position des bus en temps réel: https://data.explore.star.fr/explore/dataset/tco-bus-vehicules-position-tr

# Ajouter des données OSM via l'API

```
const ville = "Rennes";

$.getJSON(`https://overpass-api.de/api/interpreter?data=[out:json];area[name="${ville}"]->.searchArea;(node["amenity"="bar"](area.searchArea););out center;`,

 function(data) {var geojsonData = {
   type: 'FeatureCollection',
   features: data.elements.map(function(element) {
     return {type: 'Feature',
       geometry: { type: 'Point',coordinates: [element.lon, element.lat] },
       properties: {}};
   })
 };

 map.addSource('customData', {
   type: 'geojson',
   data: geojsonData
 });

 map.addLayer({
   'id': 'pubs',
   'type': 'circle',
   'source': 'customData',
   'paint': {'circle-color': 'green',
             'circle-radius': 5},
   'layout': {'visibility': 'none'}
 });

});
```

# Ajouter des données externes

- Via des API




https://codepen.io/BorisMericskay/pen/GRXYjJr

# Interactivité avec les données

Hover et Click

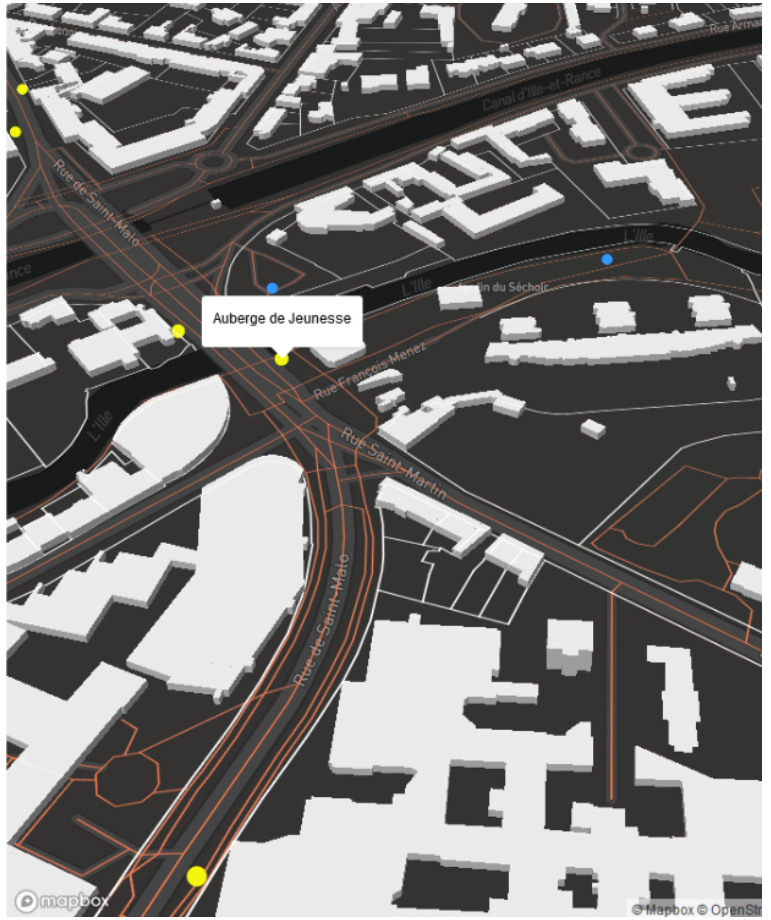
# Interactivité avec les données / Hover

- Hover de d'une couche (survol) = couche arrets
  - Cette commande doit être placée à la fin du script

```
//Interactivité HOVER

var popup = new mapboxgl.Popup({
   closeButton: false,
   closeOnClick: false });

map.on('mousemove', function(e) {
   var features = map.queryRenderedFeatures(e.point, { layers: ['Equipements'] });
   // Change the cursor style as a UI indicator.
   map.getCanvas().style.cursor = (features.length) ? 'pointer' : '';

   if (!features.length) {
      popup.remove();
      return; }

   var feature = features[0];
      popup.setLngLat(feature.geometry.coordinates)
      .setHTML(feature.properties.organom)
      .addTo(map);

});
```

# Interactivité avec les données / Hover

```
        'type': 'line',
        'source': 'Proprietes',
        'source-layer': 'limites_proprietes',
        'layout': {'visibility': 'visible',
        'line-join': 'round','line-cap': 'round'},
        'paint': {'line-color': '#FFFFFF', 'line-width': 1.5}
});

  // Ajout batiments 3D

map.addLayer({
        'id': 'Batimenst_3D',
        'source': 'composite',
        'source-layer': 'building',
        'filter': ['==', 'extrude', 'true'],
        'type': 'fill-extrusion',
        'minzoom': 15,
        'paint': {'fill-extrusion-color': '#FFFFFF', 'fill-extrusion-height':
        {'type': 'identity','property': 'height'},
        'fill-extrusion-base': {'type': 'identity','property': 'min_height'},
        'fill-extrusion-opacity': 0.9
        }
    });

  });

//Interactivité HOVER

var popup = new mapboxgl.Popup({
    closeButton: false,
    closeOnClick: false });

map.on('mousemove', function(e) {
    var features = map.queryRenderedFeatures(e.point, { layers: ['Arrets'] });
    // Change the cursor style as a UI indicator.
    map.getCanvas().style.cursor = (features.length) ? 'pointer' : '';

    if (!features.length) {
        popup.remove();
        return; }

    var feature = features[0];
        popup.setLngLat(feature.geometry.coordinates)
        .setHTML(feature.properties.nom)
        .addTo(map);

});

</script>
```



# Interactivité avec les données / Hover

- Hover de deux couches (survol)

```
//Interactivité HOVER

var popup = new mapboxgl.Popup({
   closeButton: false,
   closeOnClick: false });

map.on('mousemove', function(e) {
   var features = map.queryRenderedFeatures(e.point, { layers: ['Arrets'] ['Equipements'] });
   // Change the cursor style as a UI indicator.
   map.getCanvas().style.cursor = (features.length) ? 'pointer' : '';

   if (!features.length) {
      popup.remove();
      return;
   }

   var feature = features[0];

      popup.setLngLat(feature.geometry.coordinates)
      .setHTML(feature.properties.nom)
      .addTo(map);

    var feature = features[1];

       popup.setLngLat(feature.geometry.coordinates)
      .setHTML(feature.properties.organom)
      .addTo(map);
});
```

# Interactivité avec les données / Hover

```
115
116  //Interactivité HOVER
117
118  var popup = new mapboxgl.Popup({
119      closeButton: false,
120      closeOnClick: false });
121
122  map.on('mousemove', function(e) {
123      var features = map.queryRenderedFeatures(e.point, { layers: ['Arrets'] ['Equipements'] });
124      // Change the cursor style as a UI indicator.
125      map.getCanvas().style.cursor = (features.length) ? 'pointer' : '';
126
127      if (!features.length) {
128          popup.remove();
129          return;
130      }
131
132      var feature = features[0];
133
134          popup.setLngLat(feature.geometry.coordinates)
135          .setHTML(feature.properties.nom)
136          .addTo(map);
137
138        var feature = features[1];
139
140          popup.setLngLat(feature.geometry.coordinates)
141          .setHTML(feature.properties.organom)
142          .addTo(map);
143  });
144
145
146  </script>
147
```

# Modifier le style d'une Popup

**Ajouter dans le Style**

```
.Mypopup .mapboxgl-popup-content {
    background-color: black;
    color : white;
    opacity : 0.7;
}
```

**Ajouter l'appel du CSS dans l'appel de la Popup**

```
var popup = new mapboxgl.Popup({
   className: "Mypopup",
   closeButton: false,
   closeOnClick: false });
```

# Modifier le style du contenu d’une popup

- Reprendre la main avec une syntaxe en html dans le contenu à afficher
  - Soyez très vigilent avec les '

```
.setHTML('<b> Nom de l’arrêt</b>' + '<br>' +  feature.properties.nom )
```

```
.setHTML('<h3> Nom l’organisme</h3>' + feature.properties.organom )
```

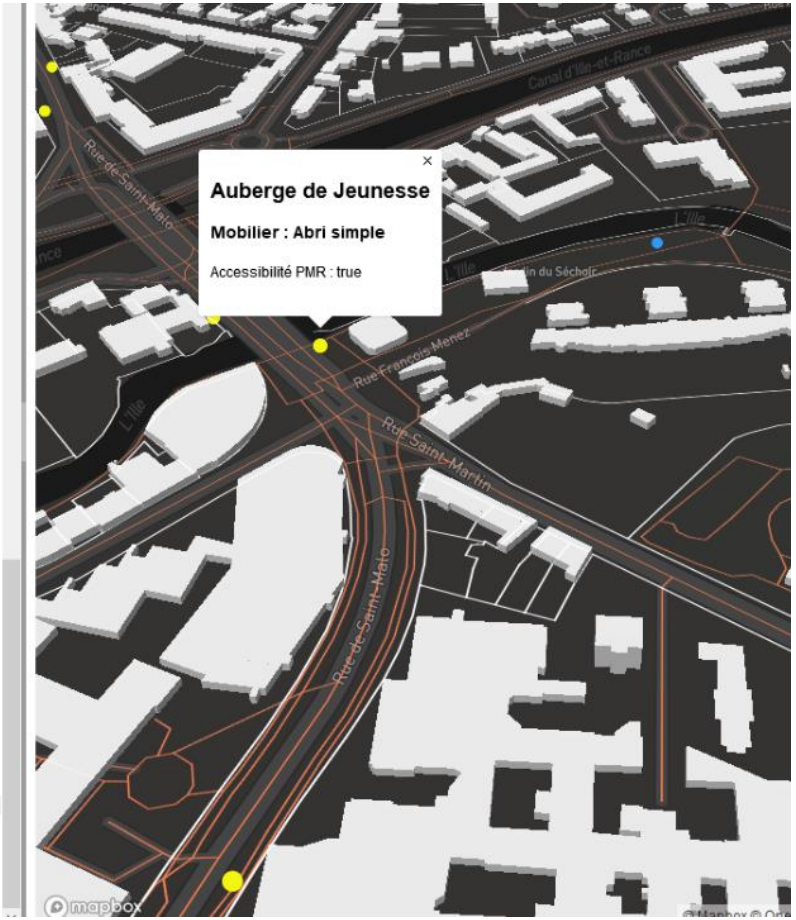
# Interactivité avec les données / Click

- Click d'une couche (popup) = couche arrets

```
//Interactivité CLICK

map.on('click', function (e) {
   var features = map.queryRenderedFeatures(e.point, { layers: ['Arrets'] });

   if (!features.length) {
      return;
   }

   var feature = features[0];
   var popup = new mapboxgl.Popup({ offset: [0, -15] })
      .setLngLat(feature.geometry.coordinates)
      .setHTML('<h2>' + feature.properties.nom + '</h2><h3>'
                                    +"Mobilier : " + feature.properties.mobilier + '</h3><p>'
                                    +"Accessibilité PMR : " + feature.properties.estaccessiblepmr + '</p>' )
                  .addTo(map);
});

map.on('mousemove', function (e) {
   var features = map.queryRenderedFeatures(e.point, { layers: ['Arrets'] });
   map.getCanvas().style.cursor = (features.length) ? 'pointer' : '';
});
```

# Interactivité avec les données / Click

```
});

  // Ajout batiments 3D

map.addLayer({
        'id': 'Batimenst_3D',
        'source': 'composite',
        'source-layer': 'building',
        'filter': ['==', 'extrude', 'true'],
        'type': 'fill-extrusion',
        'minzoom': 15,
        'paint': {'fill-extrusion-color': '#FFFFFF', 'fill-extrusion-height':
        {'type': 'identity','property': 'height'},
        'fill-extrusion-base': {'type': 'identity','property': 'min_height'},
        'fill-extrusion-opacity': 0.9
        }
    });

   });

//Interactivité CLICK

map.on('click', function (e) {
    var features = map.queryRenderedFeatures(e.point, { layers: ['Arrets'] });

    if (!features.length) {
        return;
    }

    var feature = features[0];
    var popup = new mapboxgl.Popup({ offset: [0, -15] })
        .setLngLat(feature.geometry.coordinates)
        .setHTML('<h2>' + feature.properties.nom + '</h2><h3>'
    +"Mobilier : " + feature.properties.mobilier + '</h3><p>'
    +"Accessibilité PMR : " + feature.properties.estaccessiblepmr + '</p>' )
        .addTo(map);
});

map.on('mousemove', function (e) {
    var features = map.queryRenderedFeatures(e.point, { layers: ['Arrets'] });
    map.getCanvas().style.cursor = (features.length) ? 'pointer' : '';
});


</script>

</body>
</html>
```



# Interactivité sur le cadastre Etalab

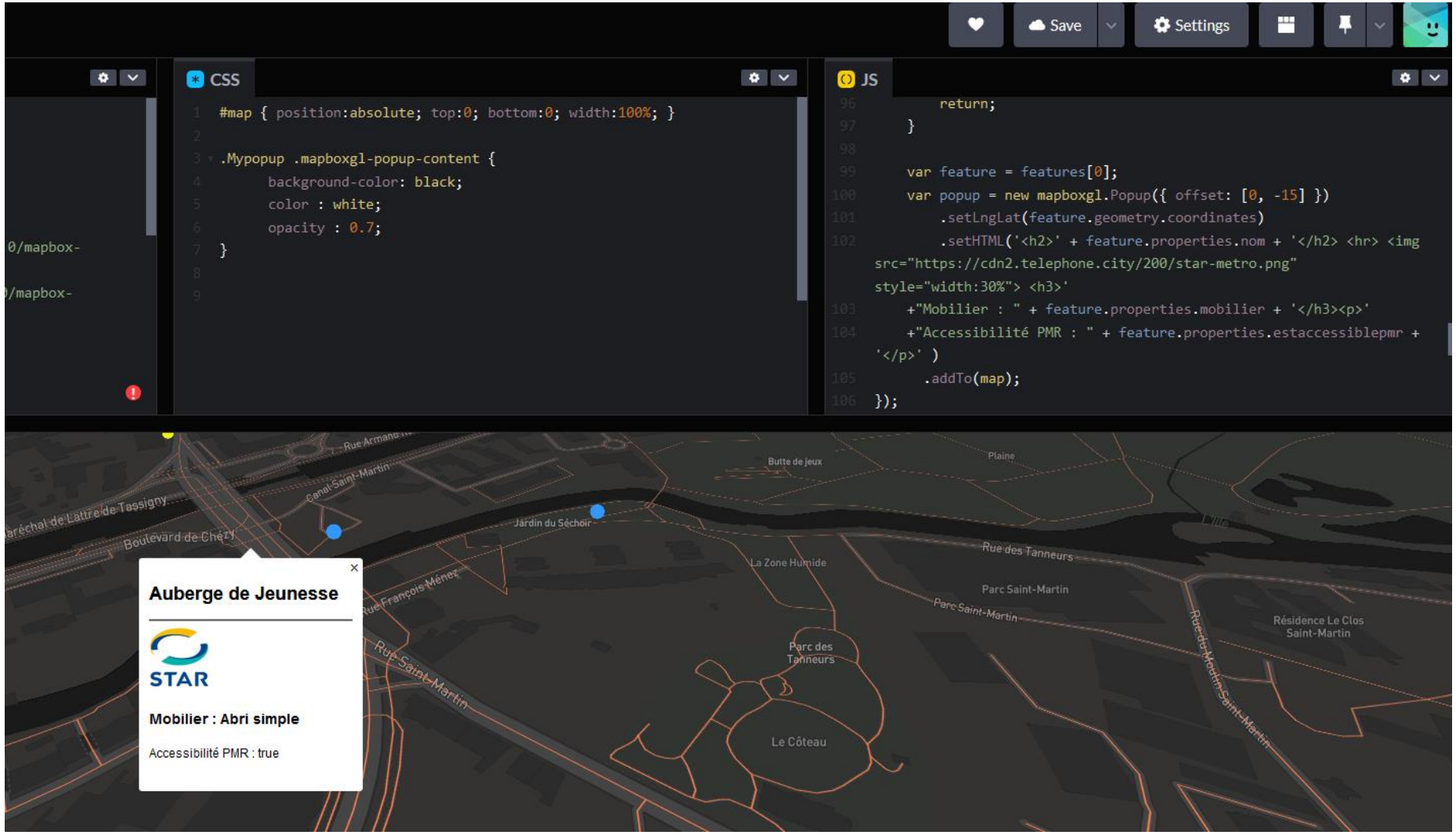
```
//Interactivité CLICK sur cadastre

map.on('click', function (e) {
var features = map.queryRenderedFeatures(e.point, { layers: ['Cadastre'] });
if (!features.length) {
return;
}
var feature = features[0];
var popup3 = new mapboxgl.Popup({ className: "Mypopup2", offset: [0, -15] })
.setLngLat(e.lngLat)
.setHTML(feature.properties.id + '<br>' + 'Numéro:' + feature.properties.numero + '<br>' +
feature.properties.contenance + '  m2')
.addTo(map);
});
map.on('mousemove', function (e) {
var features = map.queryRenderedFeatures(e.point, { layers: ['Cadastre'] });
map.getCanvas().style.cursor = (features.length) ? 'pointer' : '';
});
```

# Ajouter une image à la Popup

<img src="https://cdn2.telephone.city/200/star-metro.png" style="width:30%">

```
//Interactivité CLICK

map.on('click', function (e) {
    var features = map.queryRenderedFeatures(e.point, { layers: ['Arrets'] });

    if (!features.length) {
        return;
    }

    var feature = features[0];
    var popup = new mapboxgl.Popup({ offset: [0, -15] })
        .setLngLat(feature.geometry.coordinates)
        .setHTML('<h2>' + feature.properties.nom +  '</h2> <img
src="https://cdn2.telephone.city/200/star-metro.png"  style="width:30%">  <h3>'
        +"Mobilier : " + feature.properties.mobilier + '</h3><p>'
        +"Accessibilité PMR : " + feature.properties.estaccessiblepmr + '</p>' )
        .addTo(map);
});

map.on('mousemove', function (e) {
    var features = map.queryRenderedFeatures(e.point, { layers: ['Arrets'] });
    map.getCanvas().style.cursor = (features.length) ? 'pointer' : '';
});
```

# Ajouter une image à la Popup



# Exemple


https://codepen.io/BorisMericskay/pen/JjBmEwp

# Menu de gestion des couches

# Ajouter un menu pour gérer les couches

- Première étape: définir le style (CSS) de votre menu

```
#menu {
  width: 20%;
  Z-index: 1;
  top: 10px;
  left: 20px;
  position: absolute;
  opacity: 0.7;
  font-size: 14px;
  font-family: 'Helvetica Neue', Arial, Helvetica, sans-
serif;
}
#menu a {
  border-radius: 5px;
  display: block;
  color: #000000;
  margin: 5px;
  padding: 10px 10px;
  text-align: center;
  font-weight: bold;
  border: solid 2px;
  background-color: #FFFFFF;
  text-decoration: none;
}
#menu a.active {
  background-color: #CC6600;
  color: #FFFFFF;
}
#menu a:hover:not(.active) {
  background-color: #CC6600;
  color: #FFFFFF;
}
```

```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8' />
    <title>MapboxGL</title>

<script src='https://api.tiles.mapbox.com/mapbox-gl-js/v0.44.0/mapbox-gl.js'></script>
<link href='https://api.tiles.mapbox.com/mapbox-gl-js/v0.44.0/mapbox-gl.css' rel='stylesheet' />

<style>
#map { position:absolute; top:0; bottom:0; width:100%; }
#menu {
    width: 20%; margin-right: auto; margin-left: auto;
    Z-index: 1; top: 10px; right: 10px; position: absolute;
    border-color: #FFFFFF; background-color: #808080 ;
    font-size: 12px; font-family: 'Helvetica Neue', Arial, Helvetica, sans-serif; }

#menu a {
    display: block; color: #FFFFFF; padding: 8px 16px;
    text-align: center;  font-weight: bold;
    border-style: solid; border-color: #000000;}

#menu a.active { background-color: #CC6600;
    color: #FFFFFF;}

#menu a:hover:not(.active) {
    background-color:  #CC6600;
    color: #FFFFFF;}

</style>

</head>

<body>
<div id='map'></div>

<script>
    // AccesToken
```

# Ajouter un menu pour gérer les couches

- Deuxième étape: créer un Div pour votre menu et placer la dans la Div de la carte

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

```
81      <button id='Rennes2'>Université Rennes 2</button>
82  </div>
83
84  <div id="map"><div id="menu"></div></div>
85
86  <script>
87
88  // Appel de la carte
```

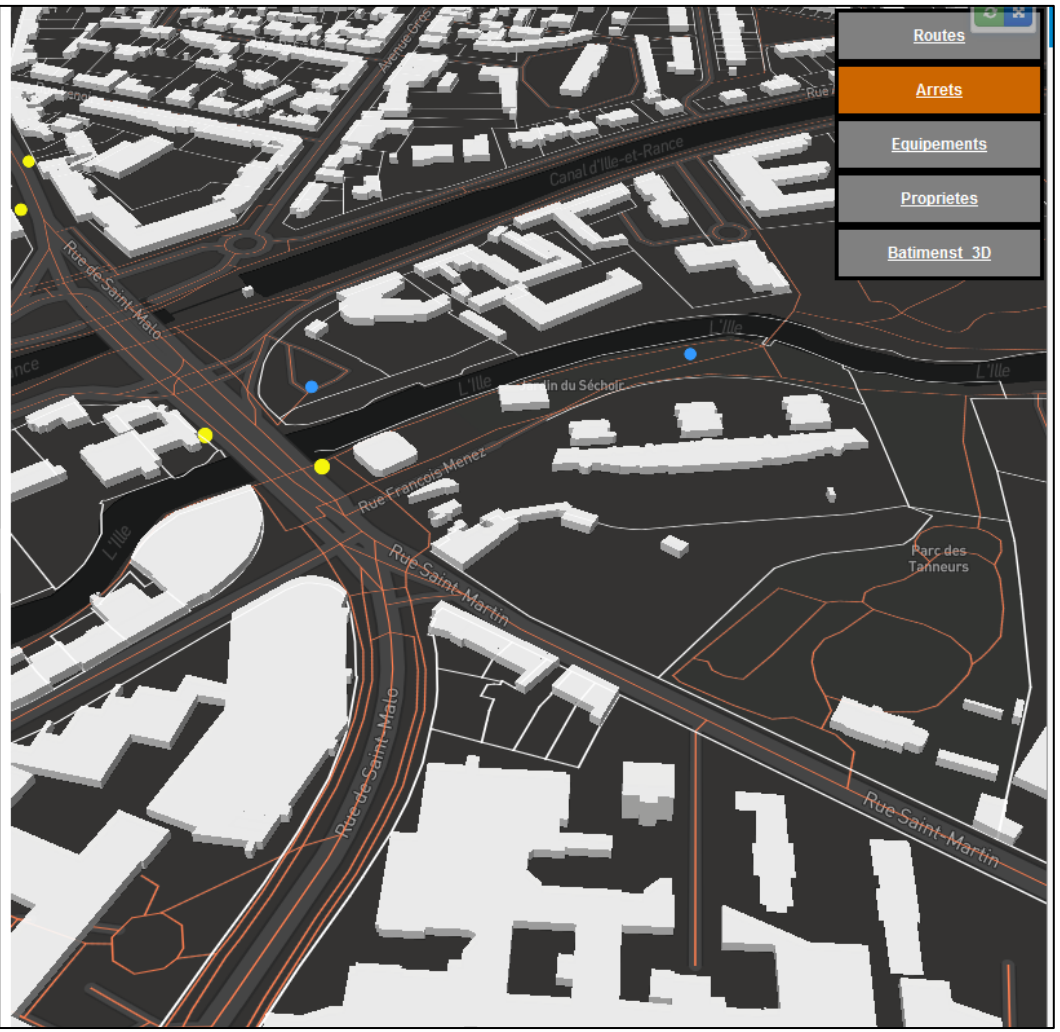
# Ajouter un menu pour gérer les couches

- Dernière étape : Ajouter à la fin du script la commande pour configurer votre menu

```
// Gestion du menu des couches

var toggleableLayerIds = ['Routes', 'Arrets', 'Equipements', 'Proprietes',  'BatimentsIGN' ];

for (var i = 0; i < toggleableLayerIds.length; i++) {var id = toggleableLayerIds[i];

    var link = document.createElement('a');
    link.href = '#';
    link.className = 'inactive';
    link.textContent = id;

    link.onclick = function (e) {var clickedLayer = this.textContent;
        e.preventDefault();
        e.stopPropagation();
        var visibility = map.getLayoutProperty(clickedLayer, 'visibility');
        if (visibility === 'visible') {
            map.setLayoutProperty(clickedLayer, 'visibility', 'none');
            this.className = '';} else {this.className = 'active';
            map.setLayoutProperty(clickedLayer, 'visibility', 'visible');} };

var layers = document.getElementById('menu');  layers.appendChild(link); }
```

# Ajouter un menu pour gérer les couches

```
160
161
162 //Interactivité CLICK
163
164 map.on('click', function (e) {
165     var features = map.queryRenderedFeatures(e.point, { layers: ['Arrets'] });
166
167     if (!features.length) {
168         return;
169     }
170
171     var feature = features[0];
172     var popup = new mapboxgl.Popup({ offset: [0, -15] })
173         .setLngLat(feature.geometry.coordinates)
174         .setHTML('<h2>' + feature.properties.nom + '</h2><h3>'
175     +"Mobilier : " + feature.properties.mobilier + '</h3><p>'
176     +"Accessibilité PMR : " + feature.properties.estaccessiblepmr + '</p>' )
177       .addTo(map);
178 });
179
180 map.on('mousemove', function (e) {
181     var features = map.queryRenderedFeatures(e.point, { layers: ['Arrets'] });
182     map.getCanvas().style.cursor = (features.length) ? 'pointer' : '';
183 });
184
185
186
187 var toggleableLayerIds = ['Routes', 'Arrets', 'Equipements', 'Proprietes',  'Batimenst_3D' ];
188
189 for (var i = 0; i < toggleableLayerIds.length; i++) {var id = toggleableLayerIds[i];
190
191     var link = document.createElement('a');
192     link.href = '#';
193     link.className = 'inactive';
194     link.textContent = id;
195
196     link.onclick = function (e) {var clickedLayer = this.textContent;
197         e.preventDefault();
198         e.stopPropagation();
199         var visibility = map.getLayoutProperty(clickedLayer, 'visibility');
200         if (visibility === 'visible') {
201             map.setLayoutProperty(clickedLayer, 'visibility', 'none');
202             this.className = '';} else {this.className = 'active';
203             map.setLayoutProperty(clickedLayer, 'visibility', 'visible');} };
204
205 var layers = document.getElementById('menu');  layers.appendChild(link); }
206
207
208 </script>
```



# Ajouter un menu pour gérer les couches

- Pour personnaliser si les couches sont active ou pas il suffit de le préciser dans la commande de visibilité *layout*
  - *visible* = couche active
  - *none* = couche non visible

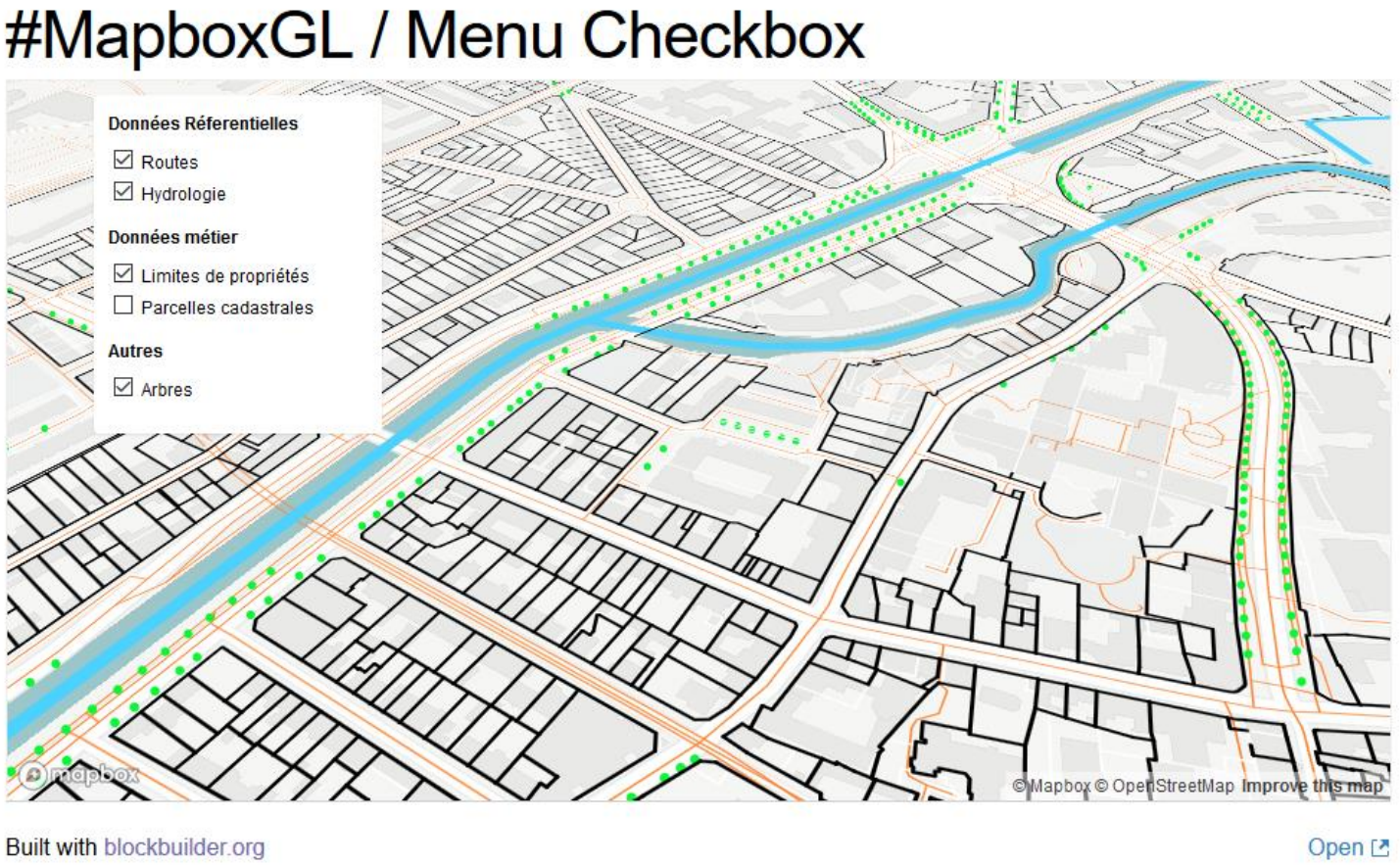
```
map.addLayer({'id': 'Arrets',
                    'type': 'circle',
                    'source': 'Arrets',
                    'source-layer': 'Bus-5ypx1k',
                    'layout': {'visibility': 'visible'},
                    'paint': {'circle-radius': 7, 'circl
```

```
map.addLayer({'id': 'Arrets',
                    'type': 'circle',
                    'source': 'Arrets',
                    'source-layer': 'Bus-5ypx1k',
                    'layout': {'visibility': ‘none'},
                    'paint': {'circle-radius': 7, 'circle-color': '#f5f60d'}    });
```

# Ajouter un menu pour gérer les couches


https://codepen.io/BorisMericskay/pen/NWBORVv

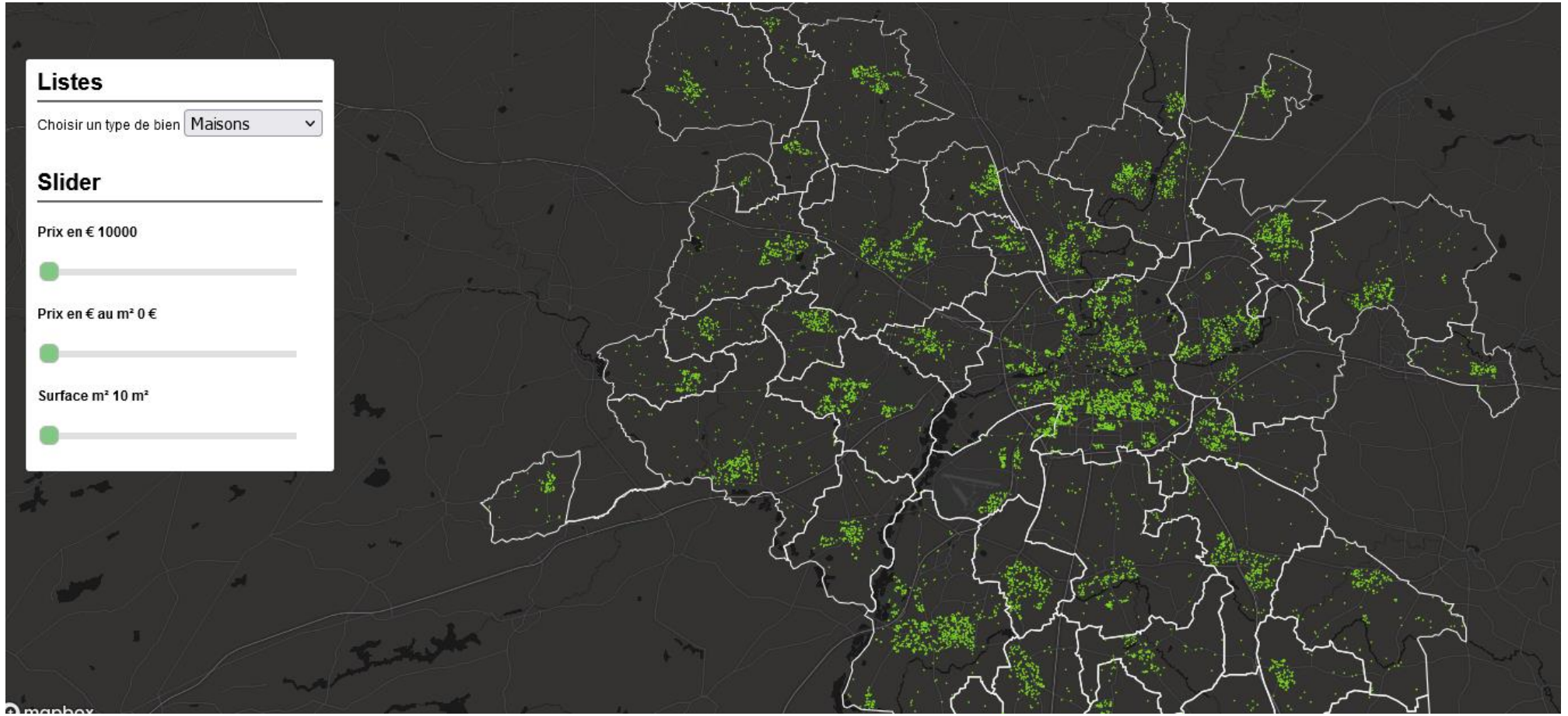
# Menu en mode CheckBox




https://codepen.io/BorisMericskay/pen/ZEjmyqQ

# Filtrer des données

# Exemple




https://codepen.io/BorisMericskay/pen/poZqweG

# Les onglets géographiques

# Onglets géographiques

- L'idée est de proposer des boutons pour aller directement à un endroit sur la carte

  - Première étape rajouter une boutons dans la div *map*

```
<div>
        <button id='Gare'>Quartier Gare-Centre</button>
        <button id='Rennes1'>Université Rennes 1</button>
        <button id='Rennes2'>Université Rennes 2</button>
</div>
```

```
<body>
<div id='map'>

 <div>
    <button id='Gare'>Quartier Gare-Centre</button>
    <button id='Rennes1'>Université Rennes 1</button>
    <button id='Rennes2'>Université Rennes 2</button>
</div>

<div id="menu"></div>
</div>

<script>
```

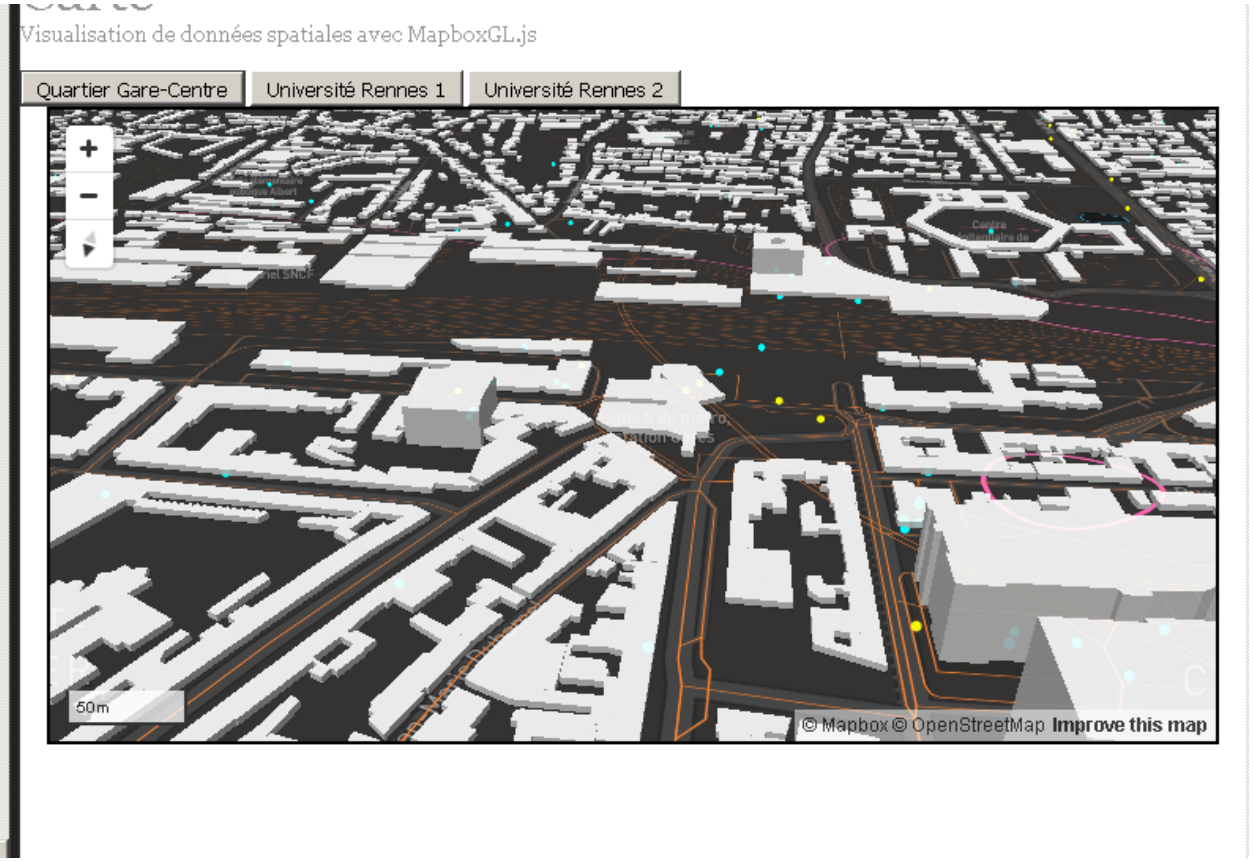
# Onglets géographiques

- Seconde étape, rajouter dans le script (à la fin) la configuration des onglets géographiques

```
// Configuration onglets geographiques

document.getElementById('Gare').addEventListener('click', function ()
{ map.flyTo({zoom: 16,
                    center: [-1.672, 48.1043],
                     pitch: 145,
                      bearing: -197.6 });
});
```

# Onglets géographiques

```
        .setLngLat(feature.geometry.coordinates)
        .setHTML(feature.properties.nom)
        .addTo(map);
});

map.on('mousemove', function (e) {
    var features = map.queryRenderedFeatures(e.point, { layers: ['Arrets'] });
    map.getCanvas().style.cursor = (features.length) ? 'pointer' : '';
});

var nav = new mapboxgl.NavigationControl();
map.addControl(nav, 'top-left');
map.addControl(new mapboxgl.ScaleControl({
    maxWidth: 120,
    unit: 'metric'}));


// Configuration onglets géographiques

document.getElementById('Gare').addEventListener('click', function ()
{ map.flyTo({zoom: 16,
             center: [-1.672, 48.1043],
             pitch: 145,
             bearing: -197.6 });
});

</script>
```

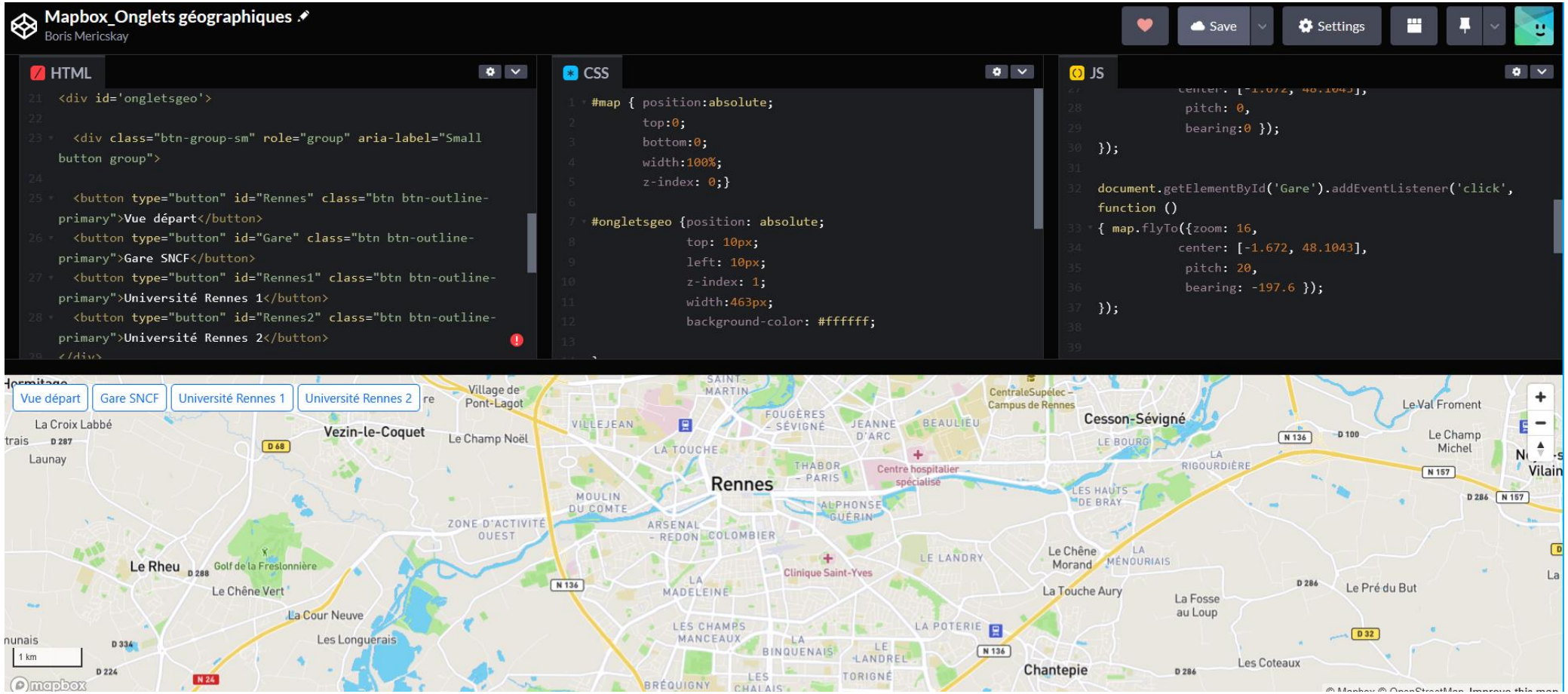


# Onglets géographiques

- Configurer les deux autres onglets géographique (Rennes 2 et Rennes1)

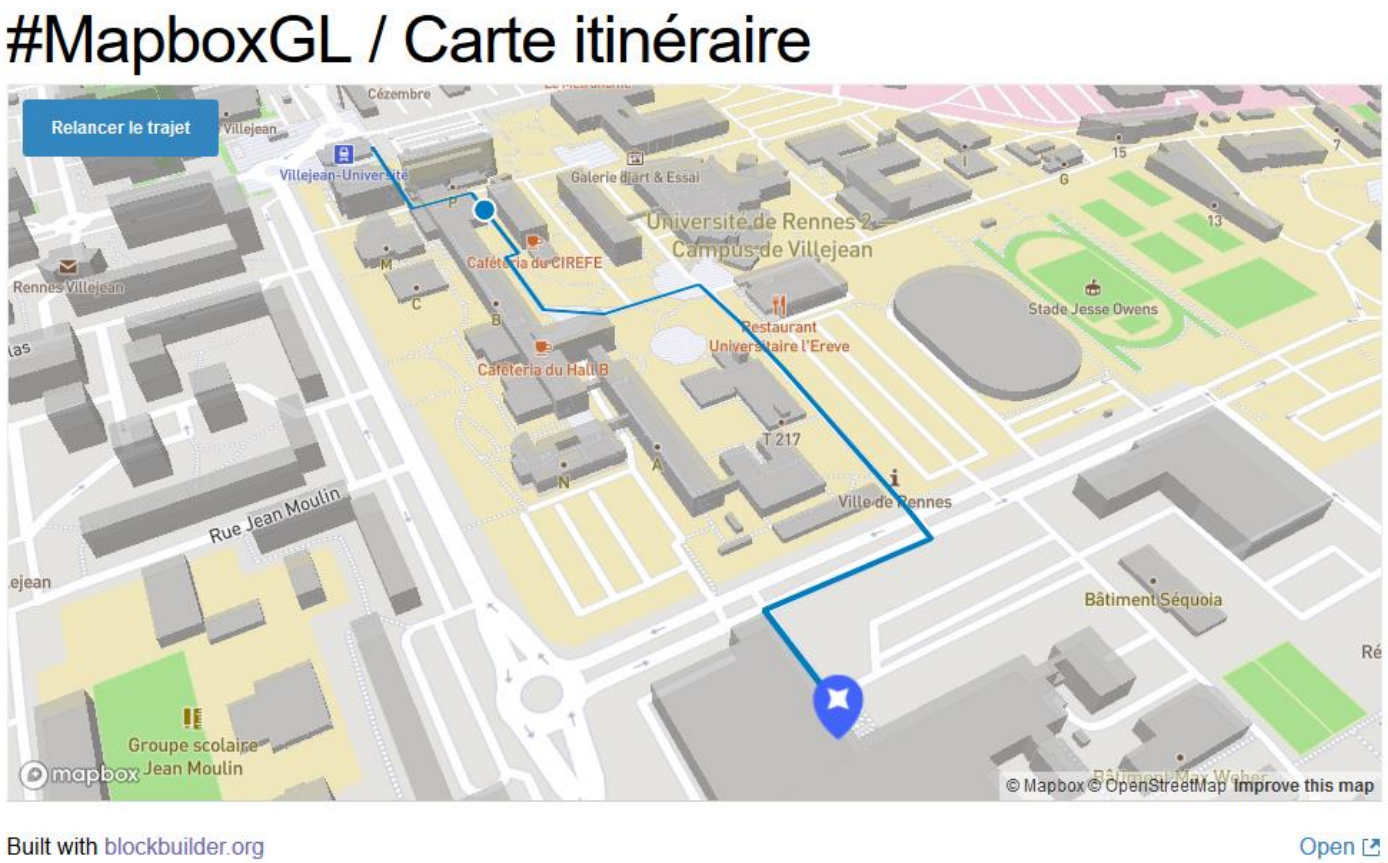
```
// Configuration onglets géographiques

document.getElementById('Gare').addEventListener('click', function () {
    map.flyTo({zoom: 16,
    center: [-1.672, 48.1043],
    pitch: 145,
    bearing: -197.6
    });
});

document.getElementById('Rennes1').addEventListener('click', function () {
    map.flyTo({zoom: 16,
    center: [-1.6396, 48.1186],
     pitch: 145,
    bearing: 197.6
    });
});

document.getElementById('Rennes2').addEventListener('click', function () {
    map.flyTo({zoom: 16,
    center: [-1.7023, 48.1194],
     pitch: 45,
    bearing: 50
    });
});


</script>
```

# Onglets géographiques




https://codepen.io/BorisMericskay/pen/GRXNQQv

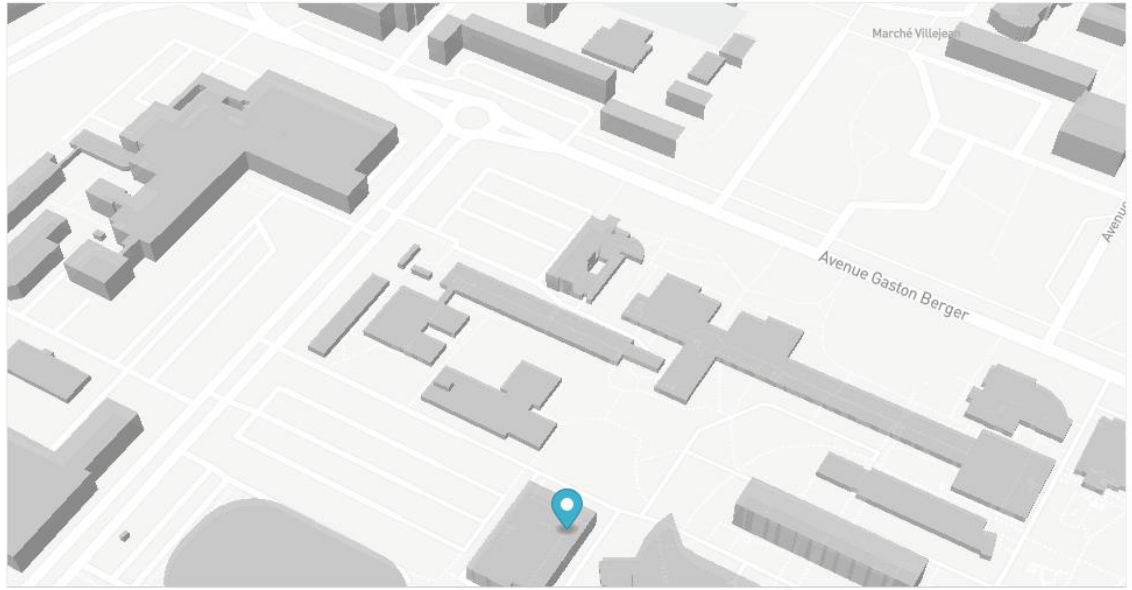
# Exemple itinéraire




https://codepen.io/BorisMericskay/pen/dyqOmYz

# Animation

#MapboxGL / Animation 2




https://codepen.io/BorisMericskay/pen/dyjQRLm

#MapboxGL / Animation carte

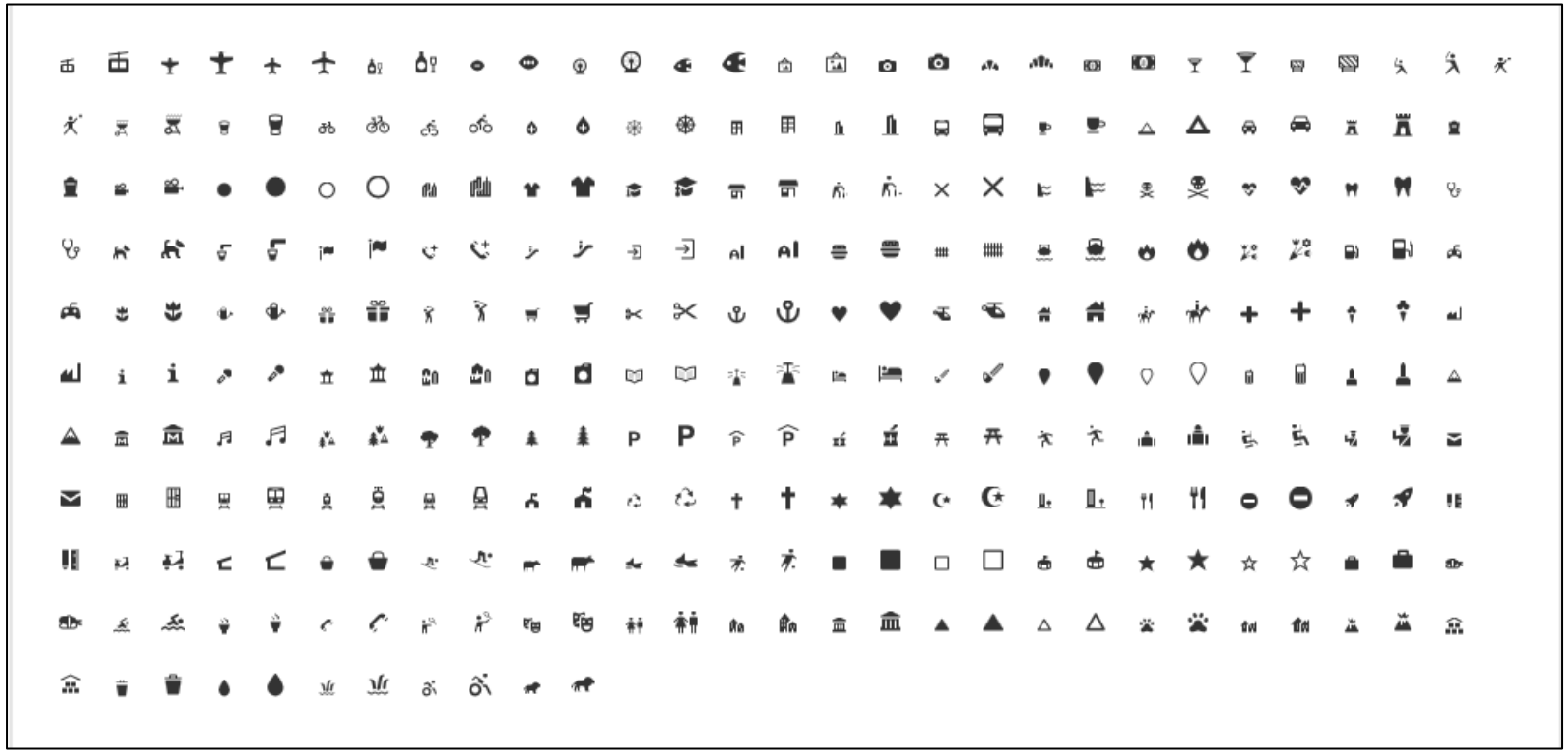

https://codepen.io/BorisMericskay/pen/MWBzoMy

# Mise en forme poussée de données spatiales

Pictogrammes, catégorisation, graduation, cercles gradués, extrusion 3D, combinaison de deux variables, cluster, heatmap,…

# Mise en forme de données personnelles

- Mobiliser la bibliothèque vectorielle Maki
  https://www.mapbox.com/maki-icons/

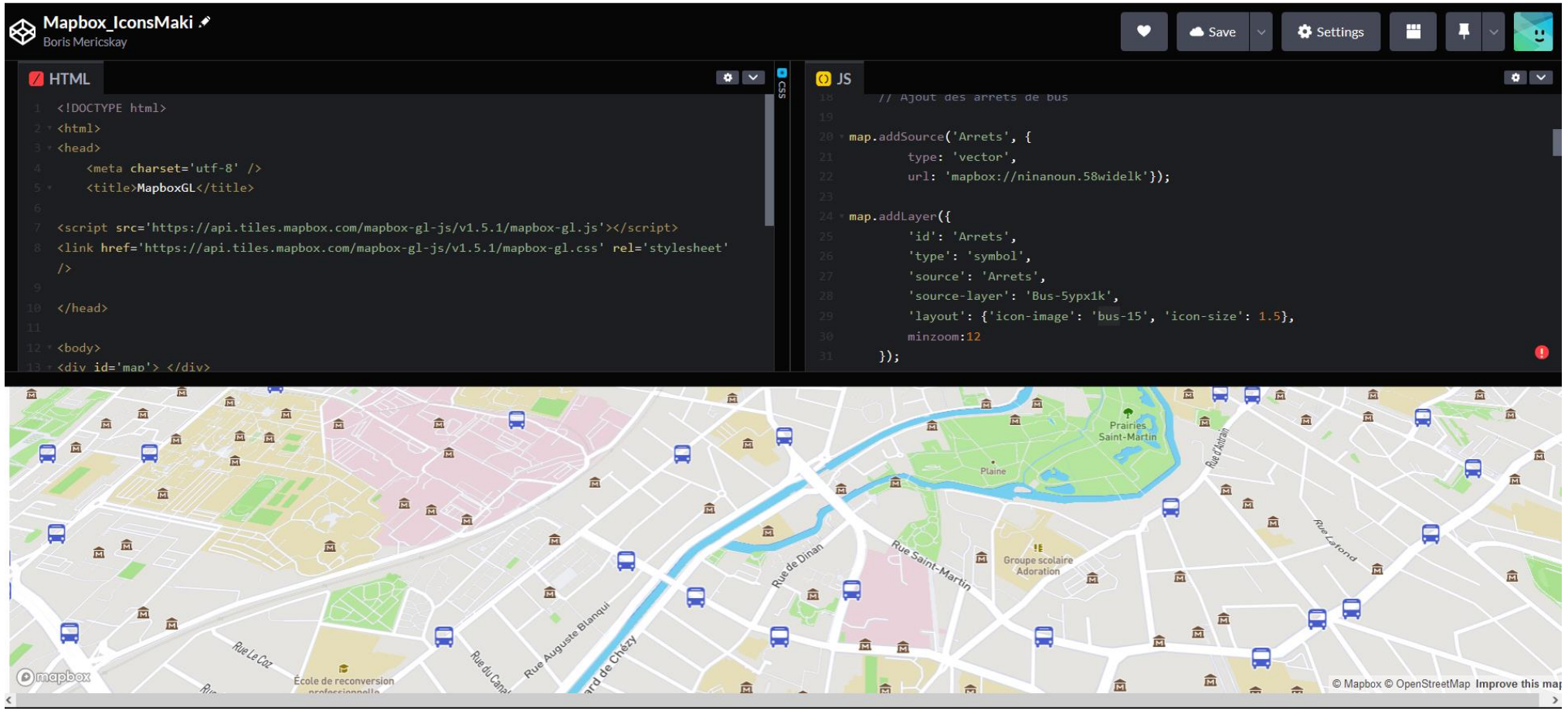


# Utiliser des pictogrammes

- Utiliser la symbologie des symboles (pictos)

```
map.addSource('Arrets', {
        type: 'vector',
        url: 'mapbox://ninanoun.7mtp5buo'});

  map.addLayer({
            "id": "Arrets",
            "type": "symbol",
            "source": "Arrets",
            "source-layer": "topologie-des-points-darret-d-9ya955",
            "layout": { "icon-image": "bus-15",
                        "icon-size": 1.5}
        });
```

# Utiliser des pictogrammes




https://codepen.io/BorisMericskay/pen/OJobvWa

# Graduation couleur

- Configurer les options de mise en forme
  - Il faut spécifier la variable mobilisée, le type d'échelle, les valeurs des bornes et les couleurs

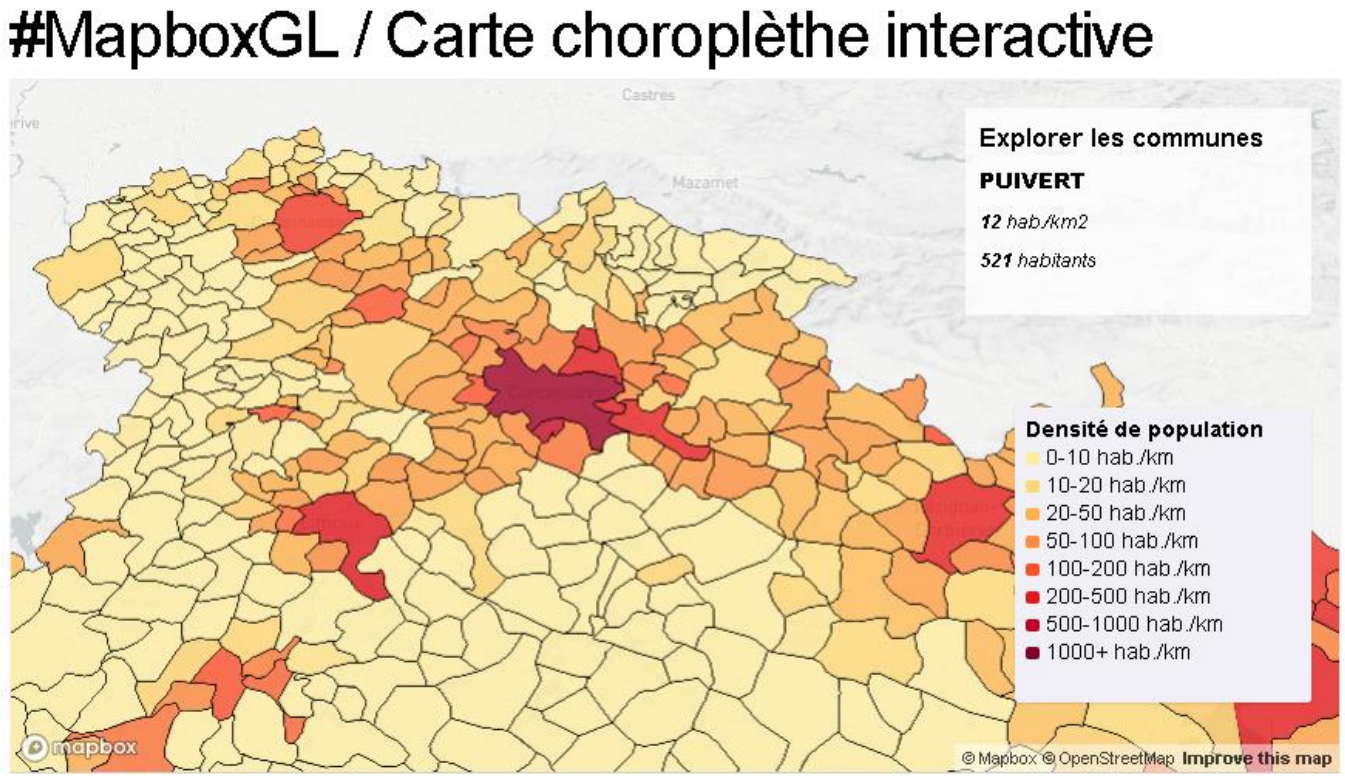
**Cercles**

```
 'paint': {'circle-radius': {'base': 1.5,'stops': [[13, 2], [22, 60]]},
          'circle-color': {property: 'hauteur',
          type: 'exponential',
          stops: [[0, '#edf8e9'],
                  [5, '#c7e9c0'],
                  [10, '#a1d99b'],
                  [15, '#74c476'],
                  [20, '#006d2c']]
}
```

**Polygones**

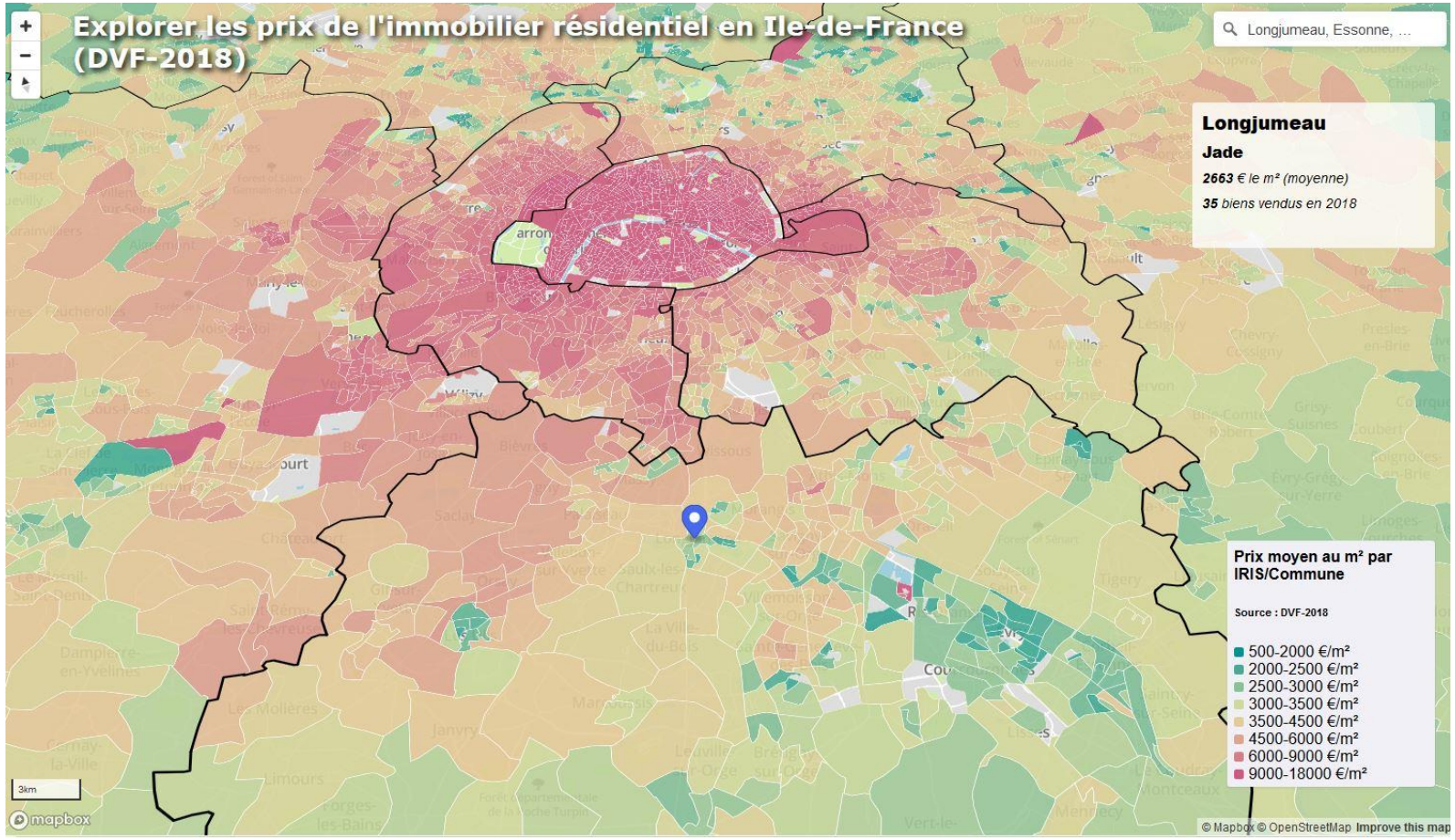
```
'paint': {'fill-color': {'property': ‘densite',
          'stops': [[1, '#1a9850'],
                    [10, '#91cf60'],
                    [20, '#d9ef8b'],
                    [50, '#ffffbf'],
                    [100, '#fee08b'],
                    [150, '#fc8d59'],
                    [200, '#d73027']]},
          'fill-opacity': 0.9}
```

# Exemple




https://codepen.io/BorisMericskay/pen/abjQydV

# Exemple




https://www.sites.univ-rennes2.fr/mastersigat/Webmapping/DVF_IDF.html

# Catégorisation

- Configurer les options de mise en forme
  - Il faut spécifier la variable mobiliser, les valeurs des catégories et les couleurs

```
'paint': {  'circle-radius': {'base': 1.75,'stops': [[12, 2], [22, 180]]},
            'circle-color': [
                        'match',['get', 'ethnicity'],
                         'White', '#fbb03b',
                         'Black', '#223b53',
                         'Hispanic', '#e55e5e',
                         'Asian', '#3bb2d0',
                         '#ccc']
          }
```

# Exemple

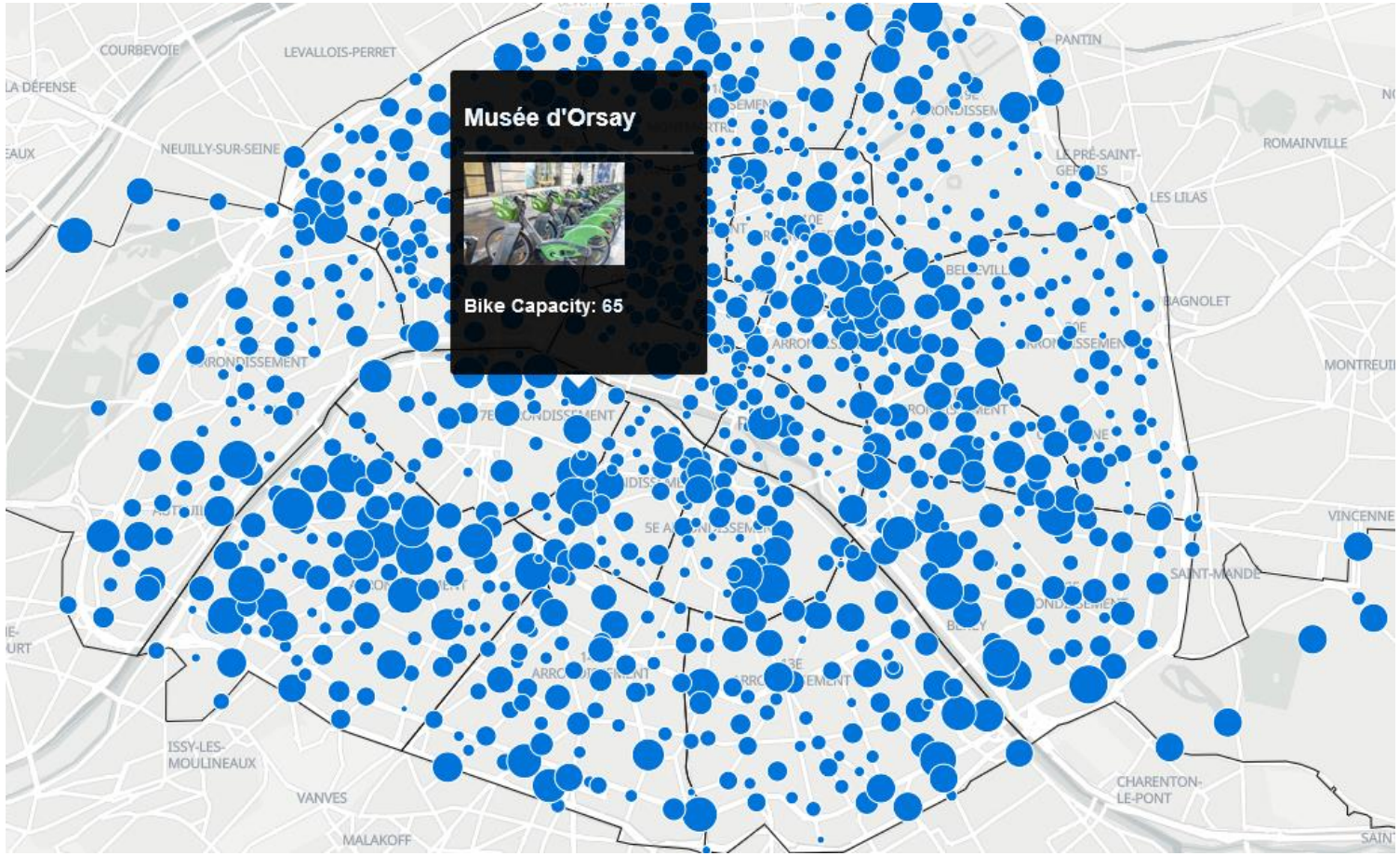

https://codepen.io/BorisMericskay/pen/VwBqKjx

# Cercles gradués

- Configurer les options de mise en forme
  - Il faut spécifier la variable mobiliser et les valeurs des bornes *(valeur, taille du cercle)*

```
paint: {'circle-color': '#D49A66',
        'circle-radius': {property: ‘population',
         type: 'exponential',
         stops: [[10, 1],[2000, 20]]},
         'circle-opacity': 0.8}
```

# Exemple



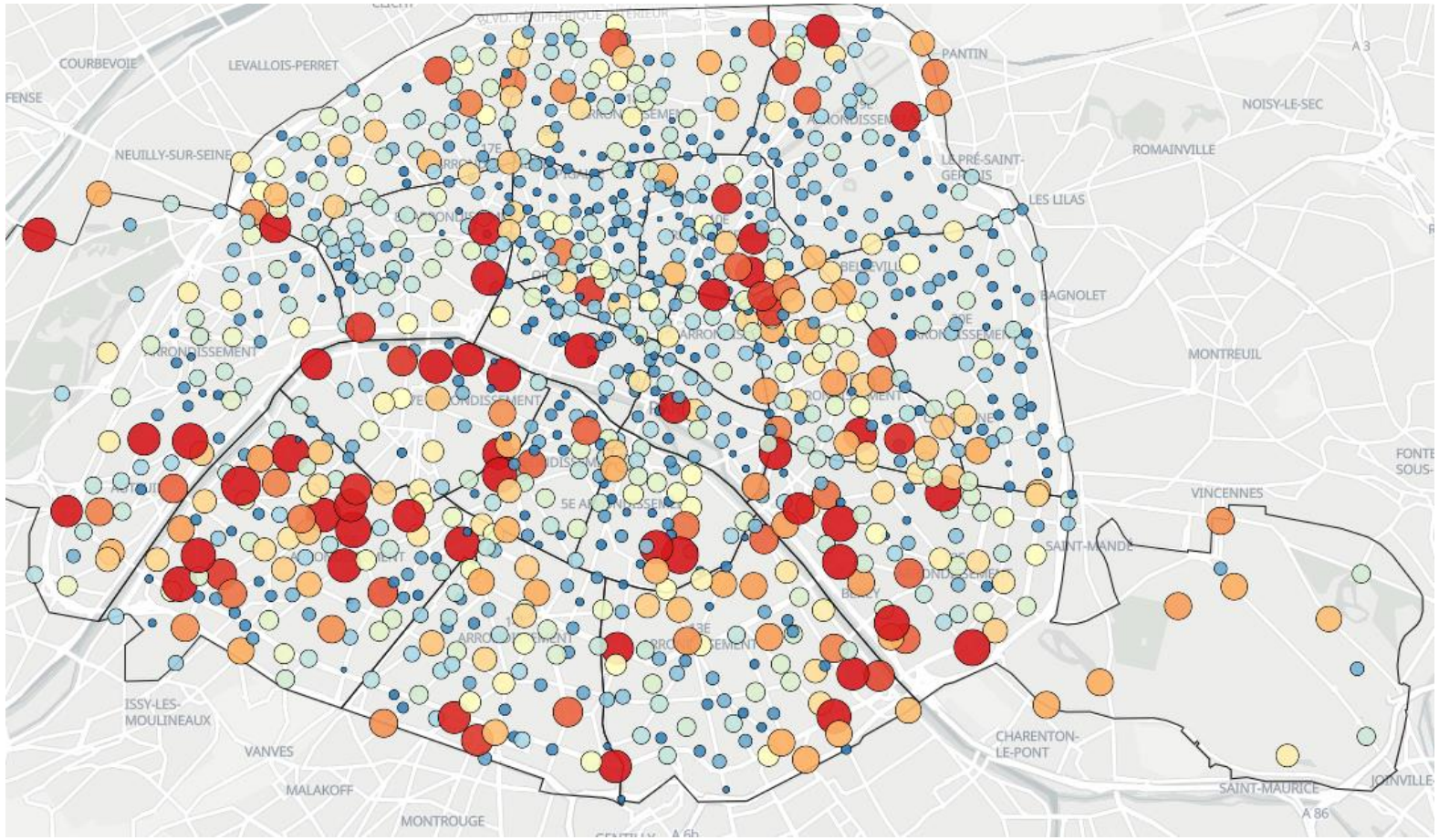

https://codepen.io/BorisMericskay/pen/MWBZbVX

# Cercles gradués avec graduation de couleur

- Configurer les options de mise en forme
  - Il faut spécifier la variable mobiliser et les valeurs des bornes *(valeur, taille du cercle)*

```
 'paint': {'circle-stroke-color': '#000000',
            'circle-stroke-width': 0.6,
            'circle-radius': {property: 'capacity', type: 'exponential', stops:
[[0, 0],[100, 20]]},
            'circle-color': {property: 'capacity',stops: [
                                [20, '#2c7bb6'],
                                [30, '#abd9e9'],
                                [40, '#ffffbf'],
                                [50, '#fdae61'],
                                [60, '#d7191c'],]},
             'circle-opacity':0.9}}
);
```

# Exemple

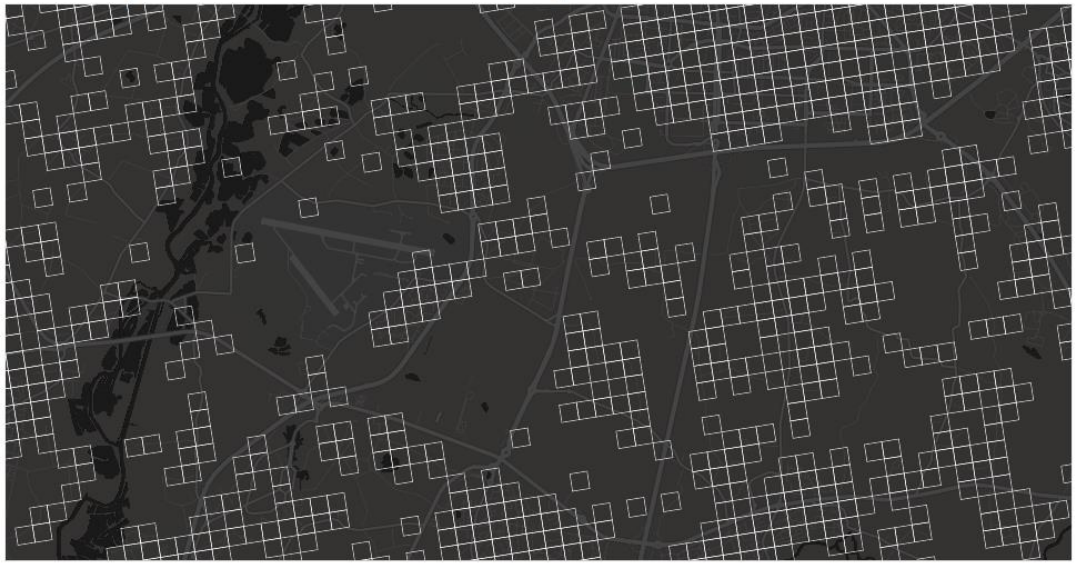



https://codepen.io/BorisMericskay/pen/MWBZbRP

# Extrusion 3D

- Récupérer le template

#Template / Extrusion MapboxGL



Built with blockbuilder.org Open

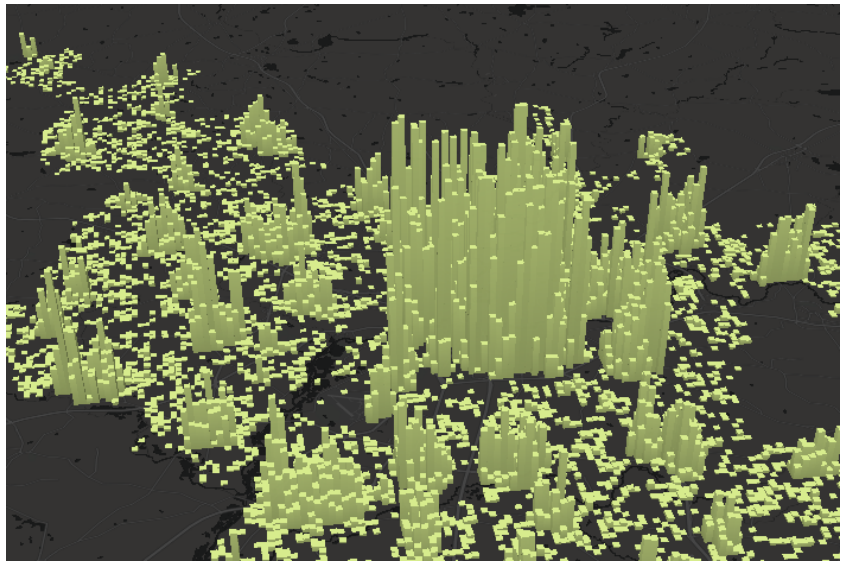
# index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset='utf-8' />
<title>Display a map</title>
```

https://bl.ocks.org/mastersigat/64af1a273f155037214d96406cb4777a/7c3f5c36742ec2dcf231059e903b144118c79d42

# Extrusion 3D

- Configurer les options de mise en forme
  - Il faut spécifier la variable mobilisée et les modalités de l'extrusion *(valeur, taille de l'extrusion)*



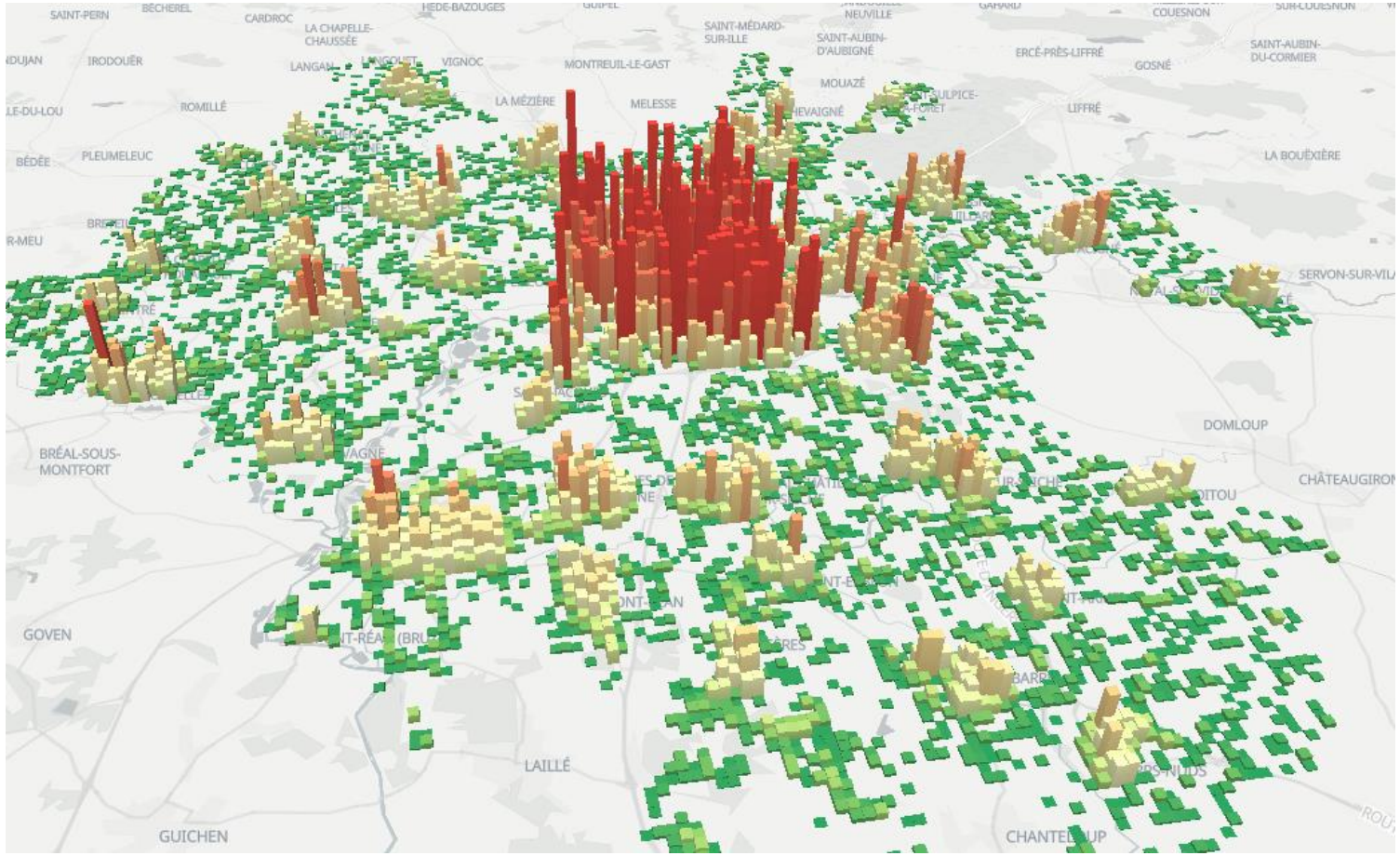
```
map.addLayer({
   'id': 'extrude',
   'type': 'fill-extrusion',
   'source': 'Carro',
   'source-layer': 'karook-dcnhdj',
   'layout': {'visibility': 'visible'},
   'paint': { 'fill-extrusion-color': '#d9ef8b',
              'fill-extrusion-height': {
              'property': 'Individus',
                          'stops': [[1, 0],
                                    [10, 100],
                                    [700, 7000]]},
     'fill-extrusion-opacity': 0.95,
     'fill-extrusion-base': 0 }
});
```

# Extrusion 3D

- Ajouter des couleurs

```
'paint': {
    'fill-extrusion-color': {
      'property': 'Individus',
      'stops': [
        [1, '#1a9850'],
        [10, '#91cf60'],
        [20, '#d9ef8b'],
        [50, '#ffffbf'],
        [100, '#fee08b'],
        [150, '#fc8d59'],
        [200, '#d73027']]
    },
    'fill-extrusion-height': {
      'property': 'Individus',
                              'stops': [[1, 0],
            [10, 100],
            …
```

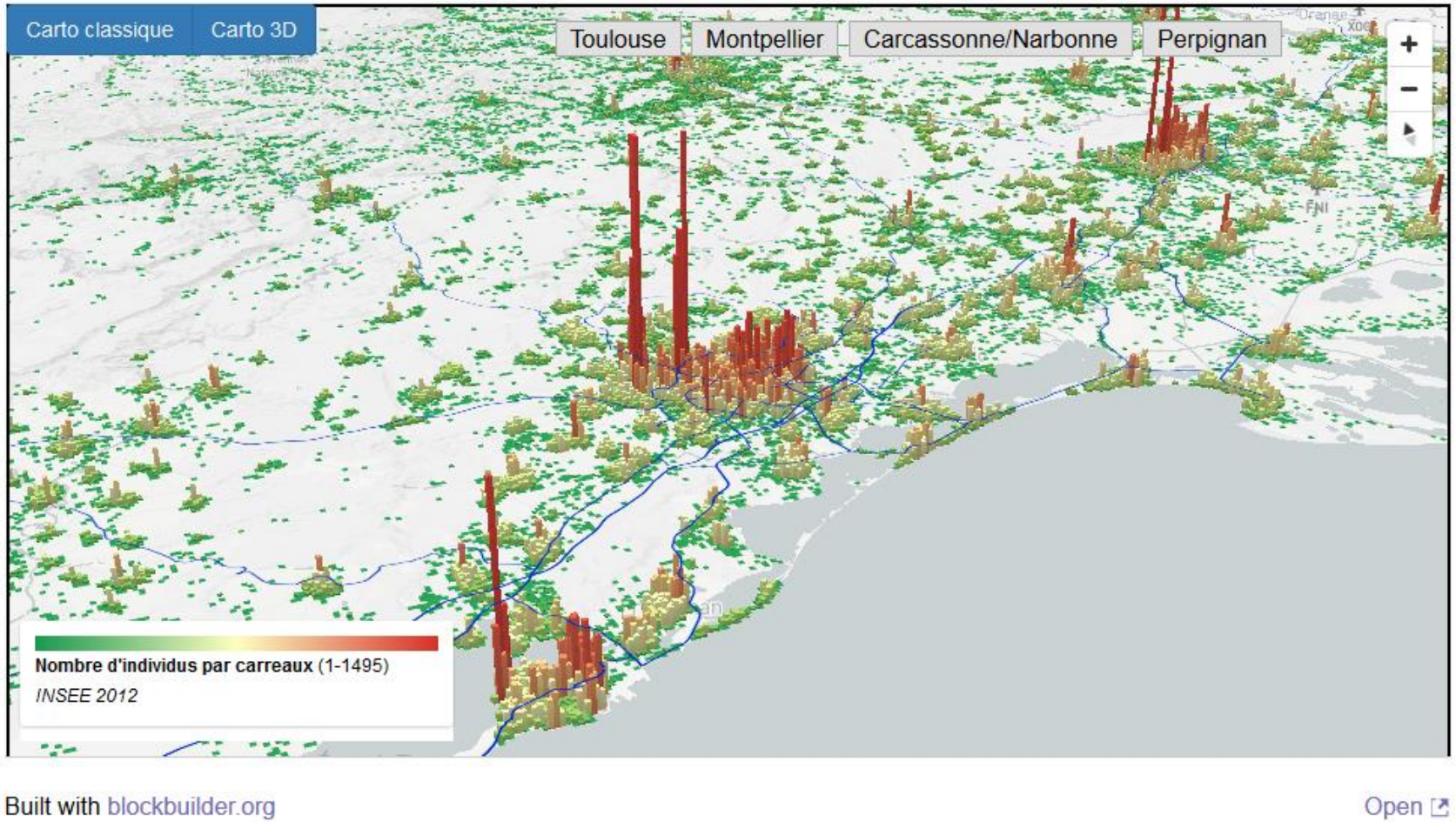
# Exemple




https://codepen.io/BorisMericskay/pen/wvxRgjE

# Exemple

## #MapboxGL / Extrusion carreaux




https://sites-formations.univ-rennes2.fr/mastersigat/WebMaps/Extrusion_Occitanie.html

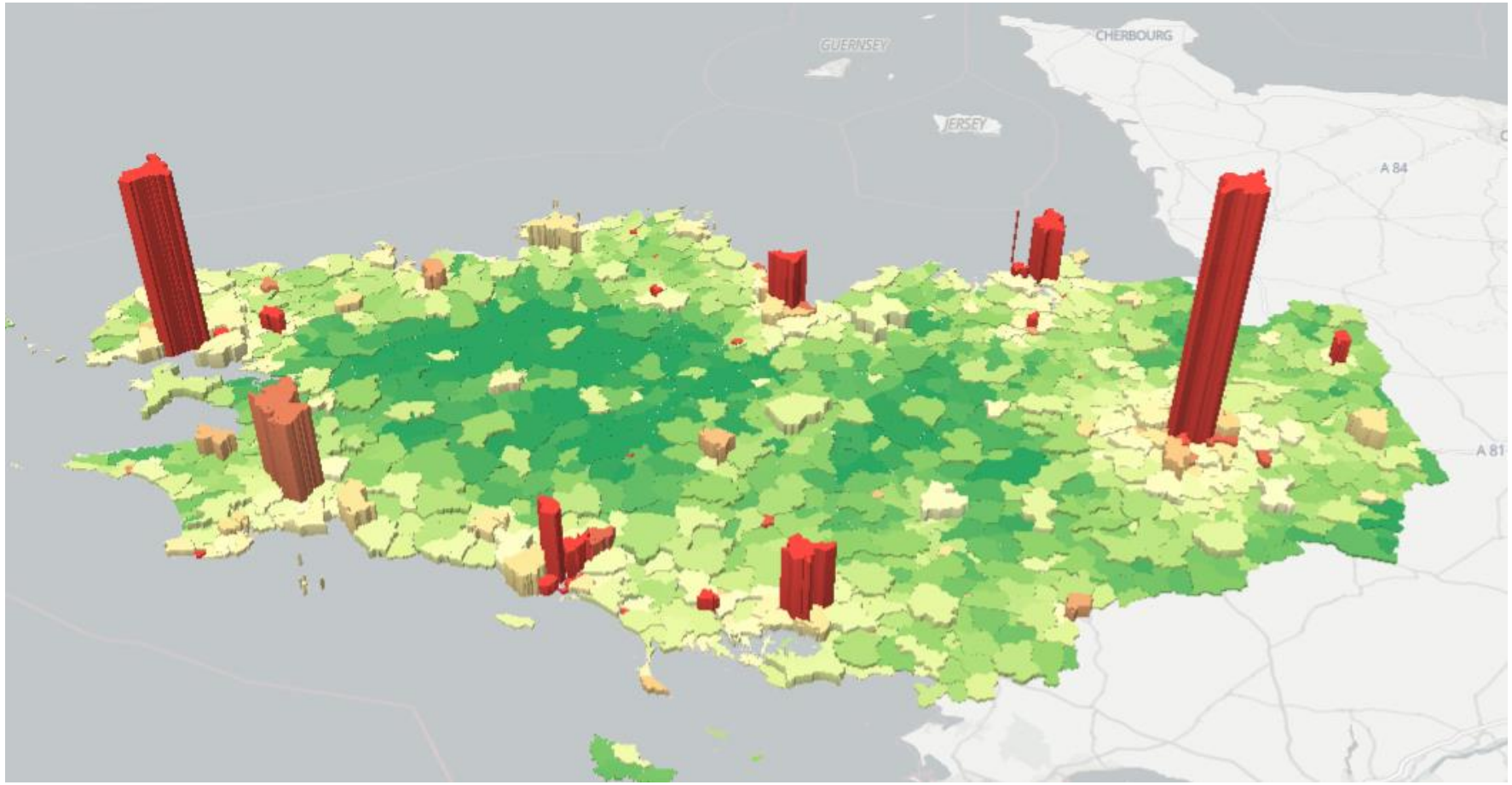
# Deux variables

```
map.addLayer({
  'id': 'extrudecommunes',
  'type': 'fill-extrusion',
  'source': 'communes',
  'source-layer': 'TD1_Data-3kid81',
  'layout': {'visibility': 'visible'},
  'paint': {'fill-extrusion-color': {
            'property': 'densite',
            'stops': [[20, '#1a9850'],
                      [50, '#91cf60'],
                      [100, '#d9ef8b'],
                     [200, '#ffffbf'],
                     [500, '#fee08b'],
                      [1000, '#d73027']]},
           'fill-extrusion-height': {'property': 'popOK',
           'stops': [[100, 10],
                     [1000, 100],
                     [200000, 100000]] },
            'fill-extrusion-opacity': 0.8,
            'fill-extrusion-base': 0
           }
});
```

Symbologie graduation de couleur (densité)

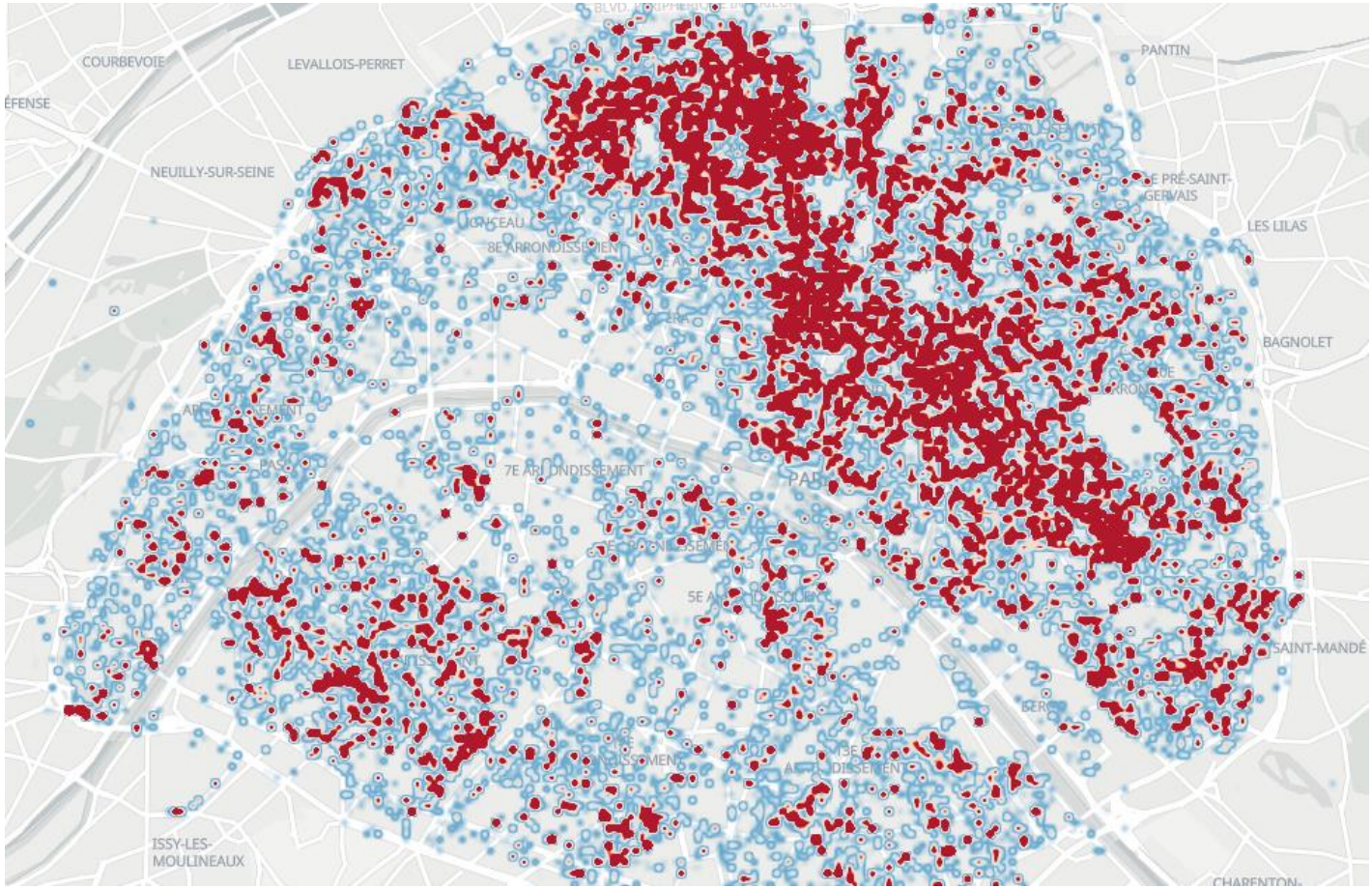
Symbologie extrusion 3D de la population

# Exemple



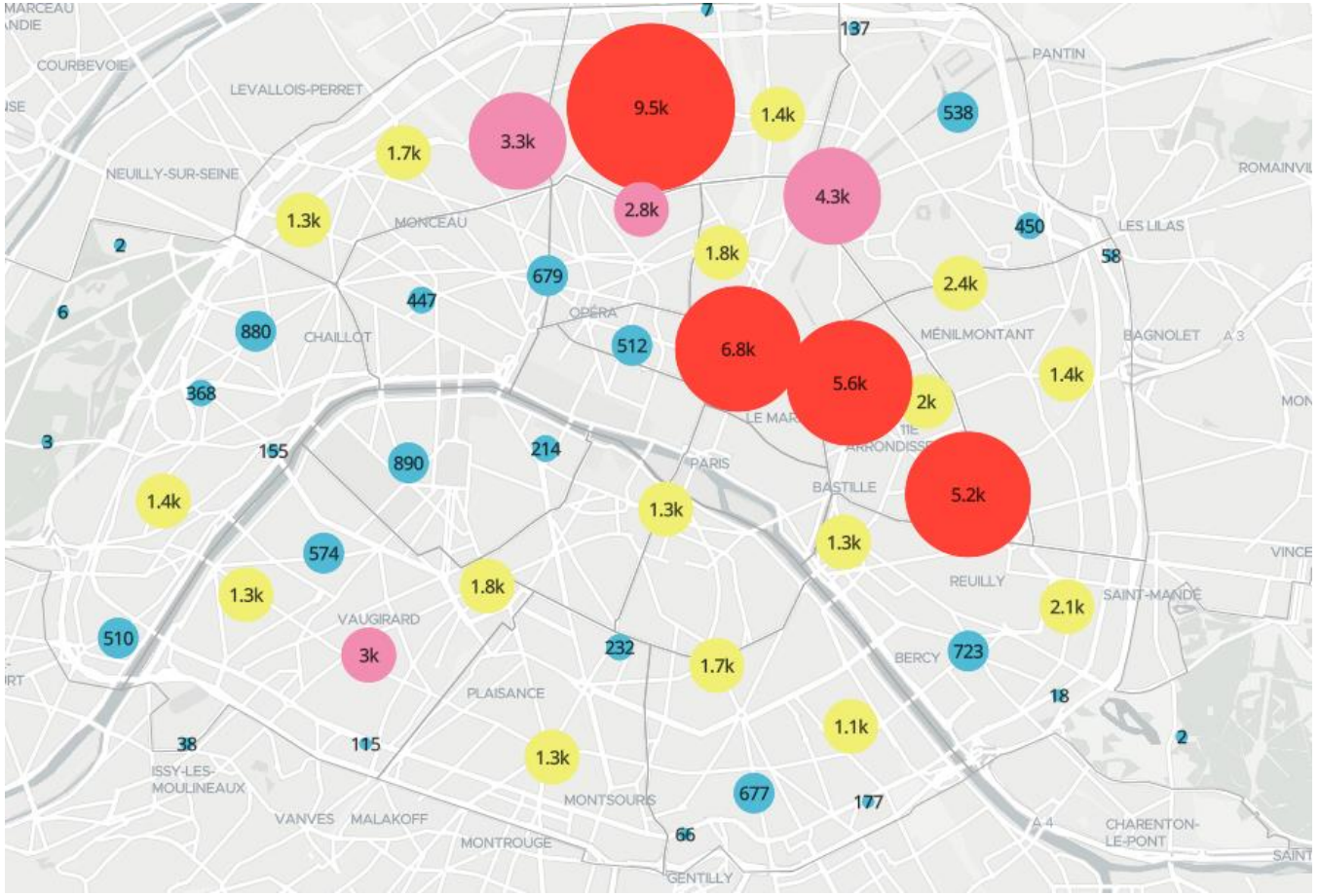

https://codepen.io/BorisMericskay/pen/eYLBMjB

# Heatmap



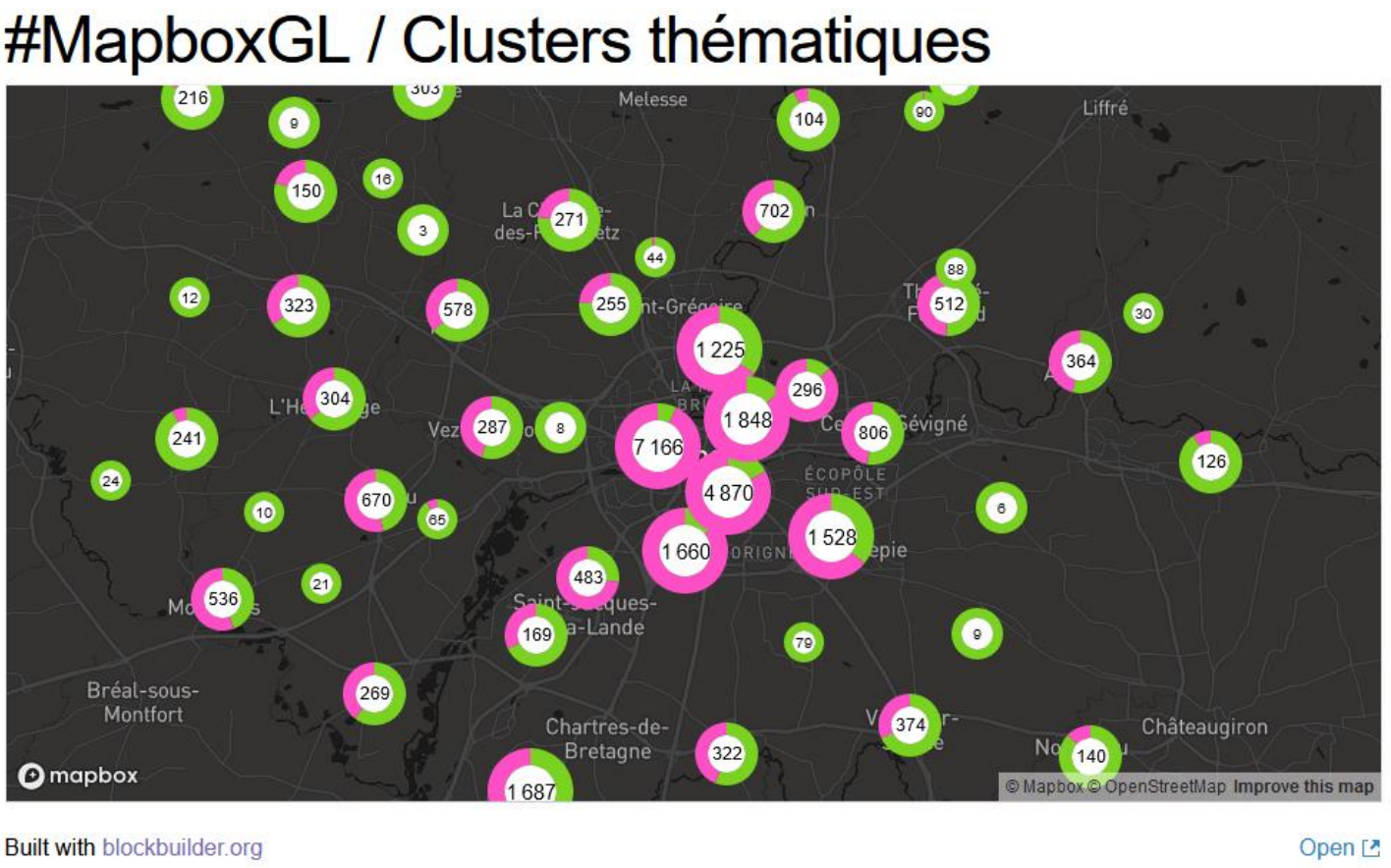

https://codepen.io/BorisMericskay/pen/gOjZmYB

# Cluster



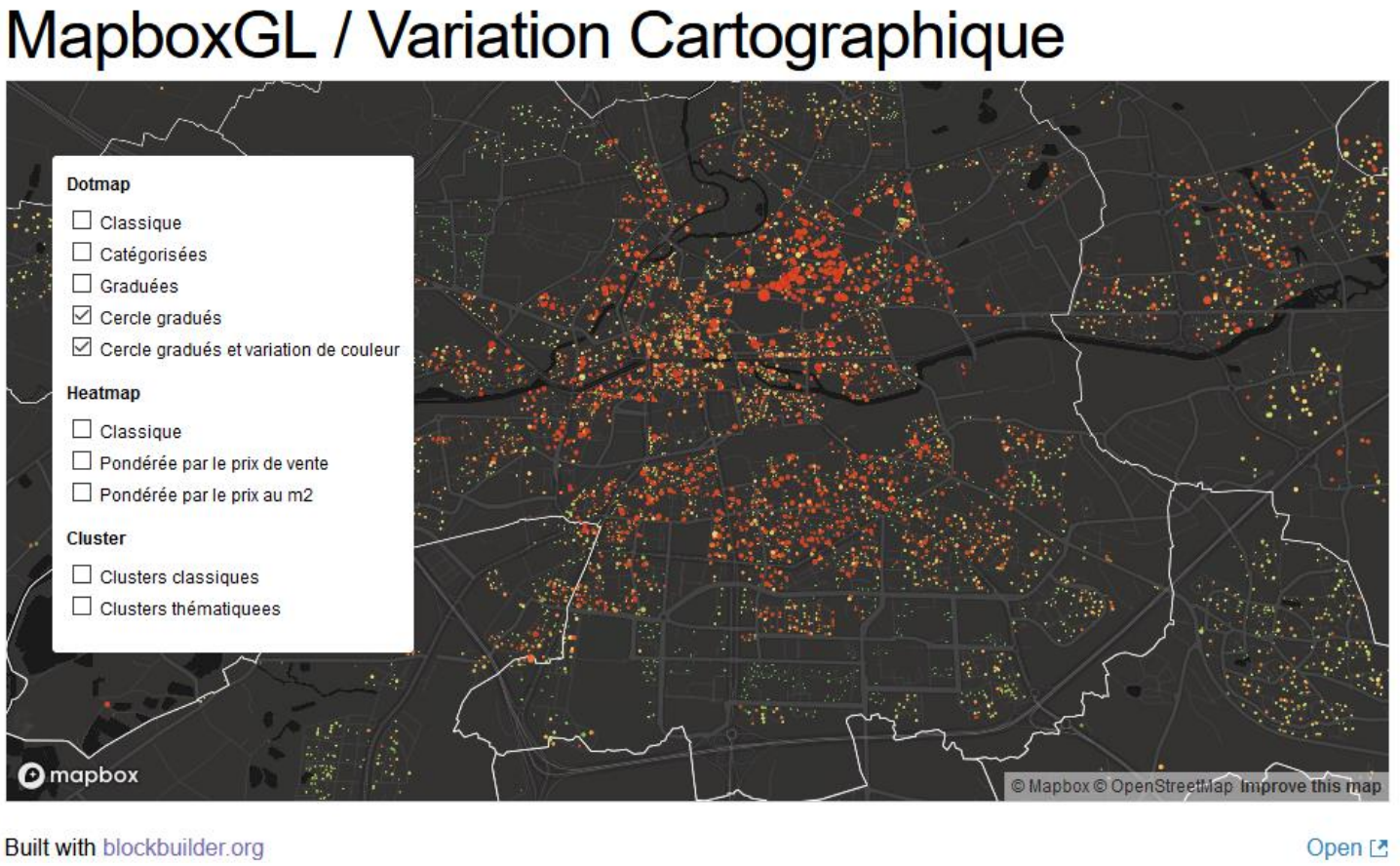

https://codepen.io/BorisMericskay/pen/OJwrpXq

# Cluster thématique




https://codepen.io/BorisMericskay/pen/qByLrpo

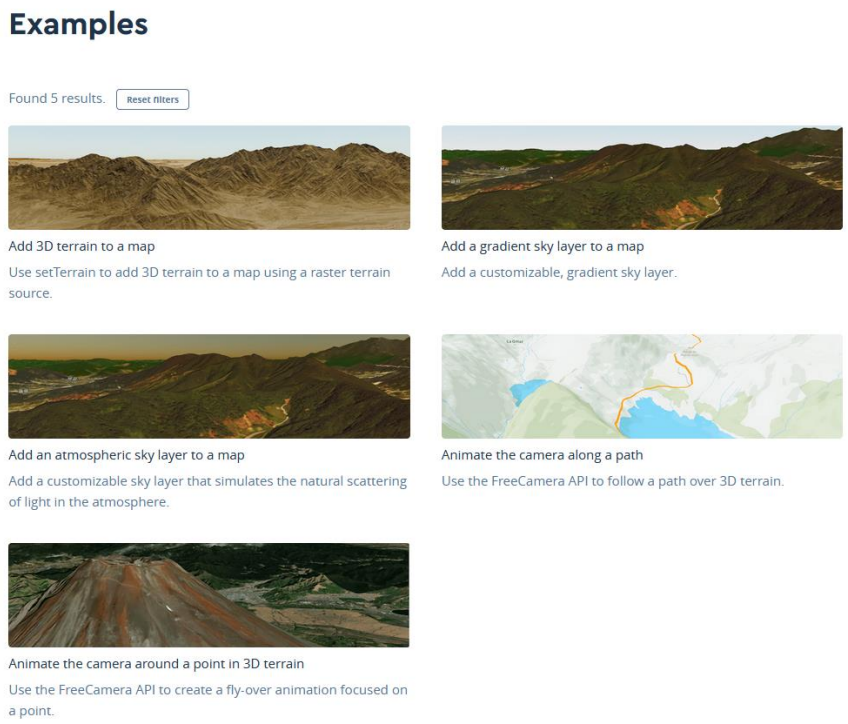
# Synthèse des modes de visualisation




https://sites-formations.univ-rennes2.fr/mastersigat//WebMaps/Variation_Mapbox.html

# MNT en 3D

- Depuis la version 2.x il est possible de mobiliser un MNT mondial et de l'extruder en 3D + texture du ciel + caméra (animation)



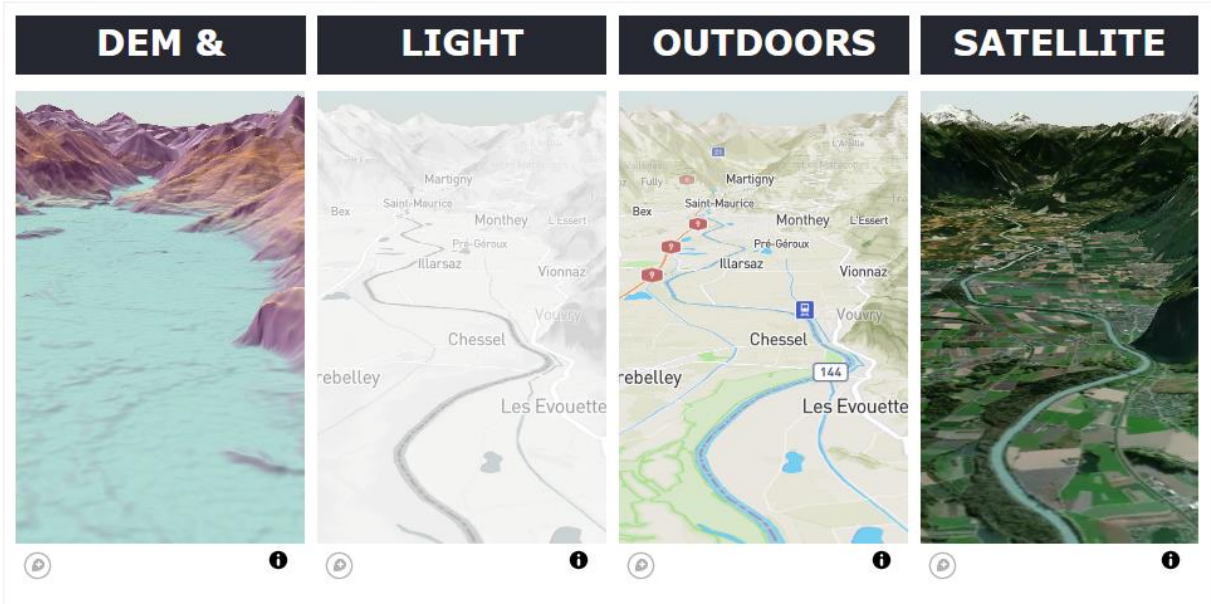

https://docs.mapbox.com/mapbox-gl-js/example/?topic=3D

# MNT en 3D

#MapboxGL / DEM 3D (MapboxGL 2.x)


https://codepen.io/BorisMericskay/pen/vYaVger

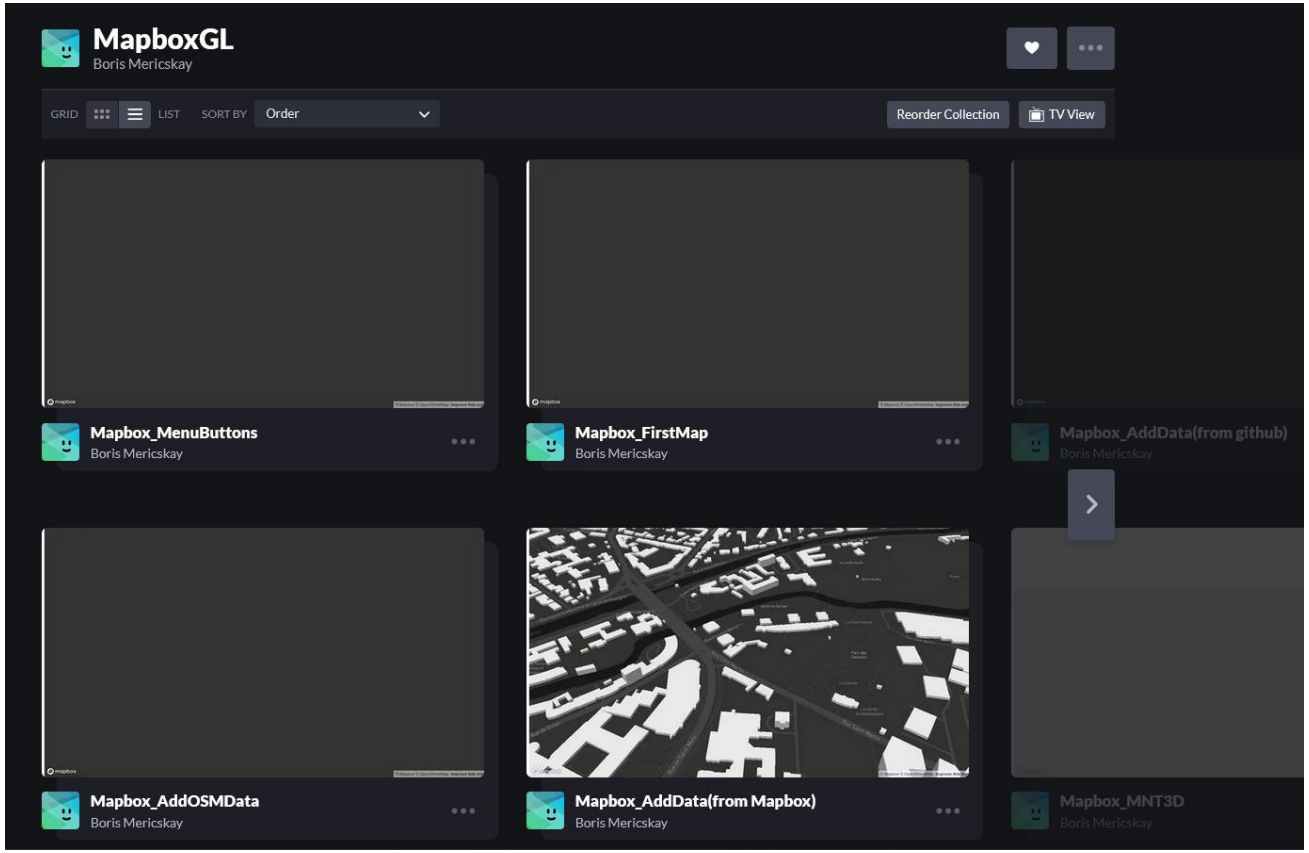
MapboxGL / Sync 3D maps



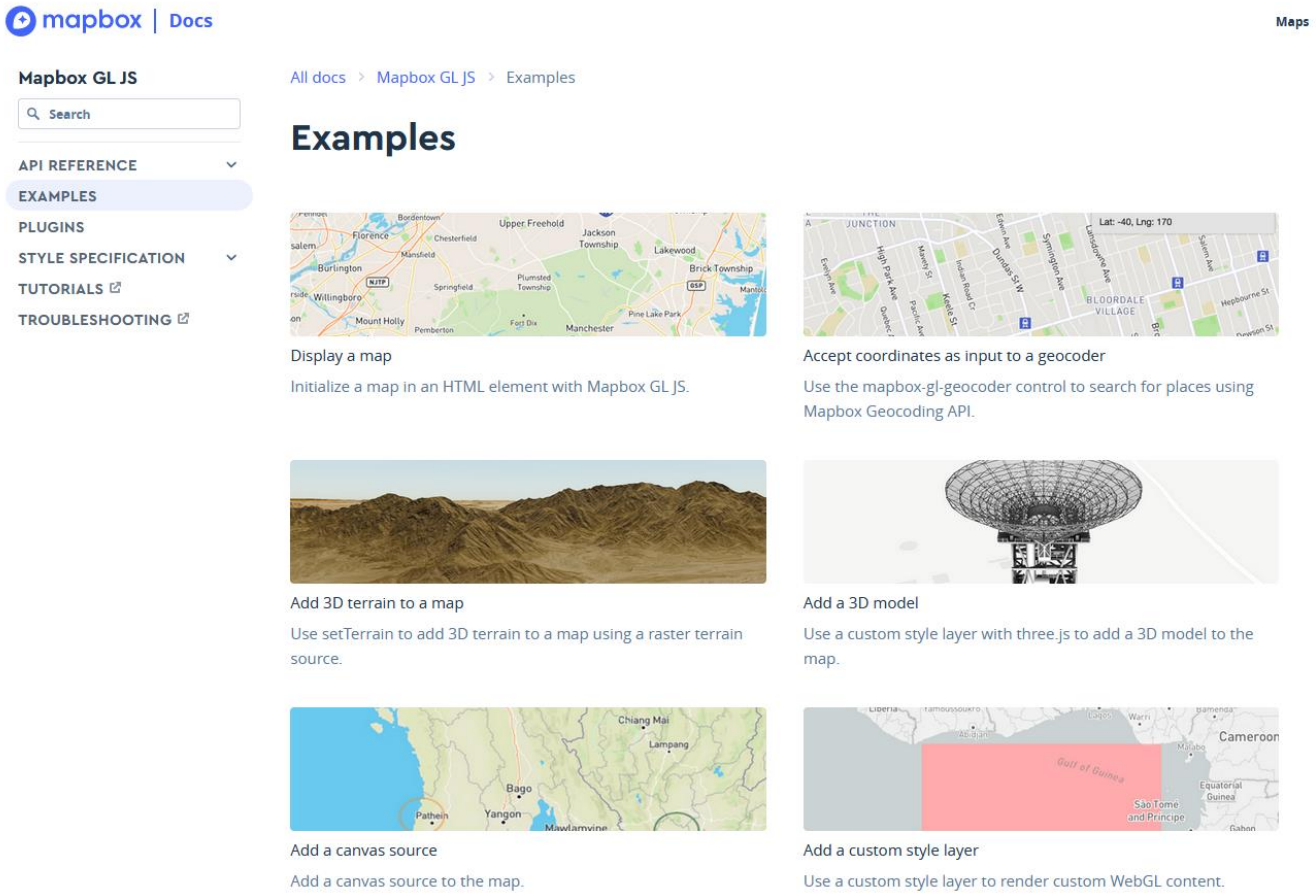

https://codepen.io/BorisMericskay/pen/xxJygjX

# Exemples divers

- https://codepen.io/collection/wayZNV



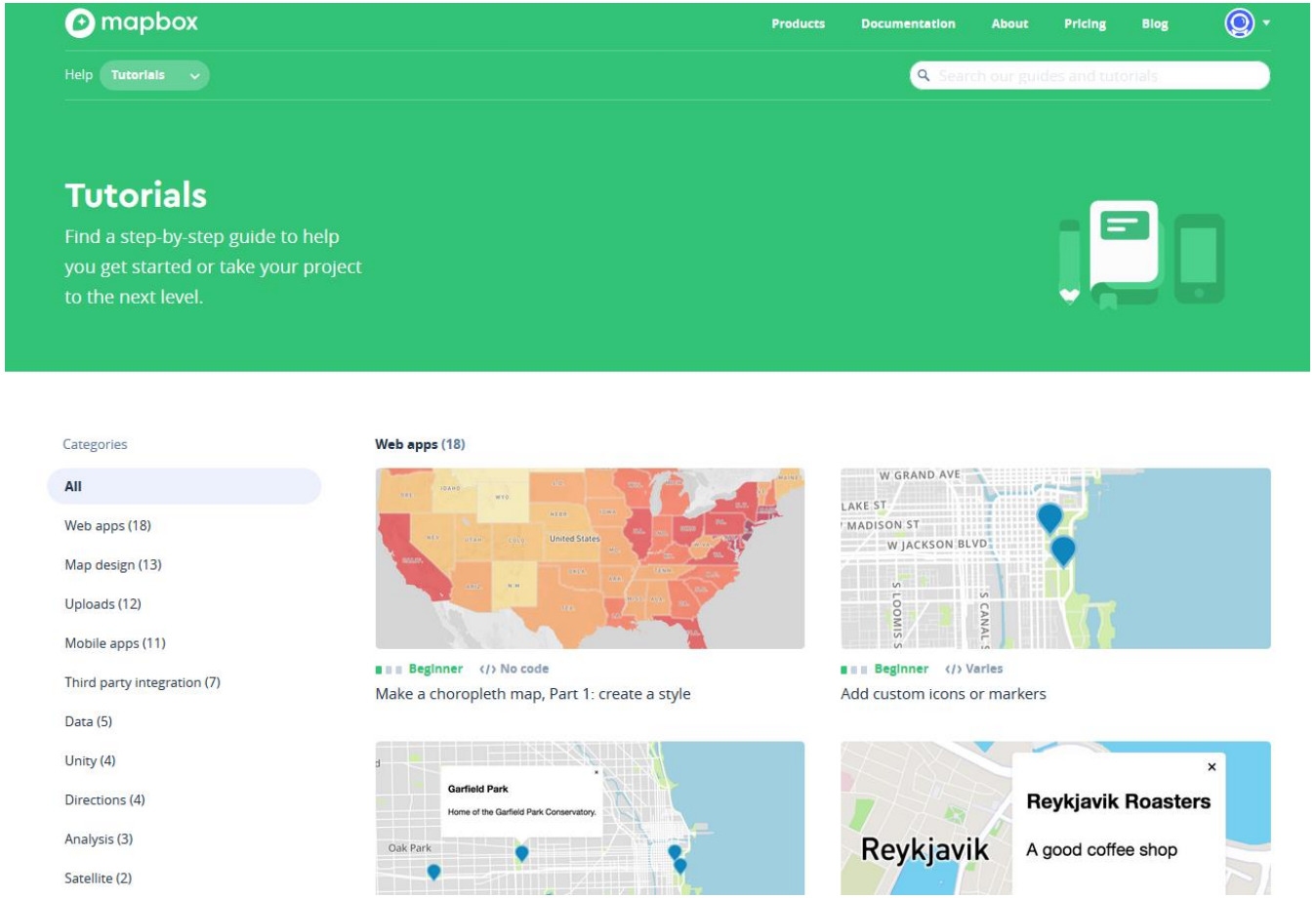
# Exemples Mapbox




https://docs.mapbox.com/mapbox-gl-js/example/

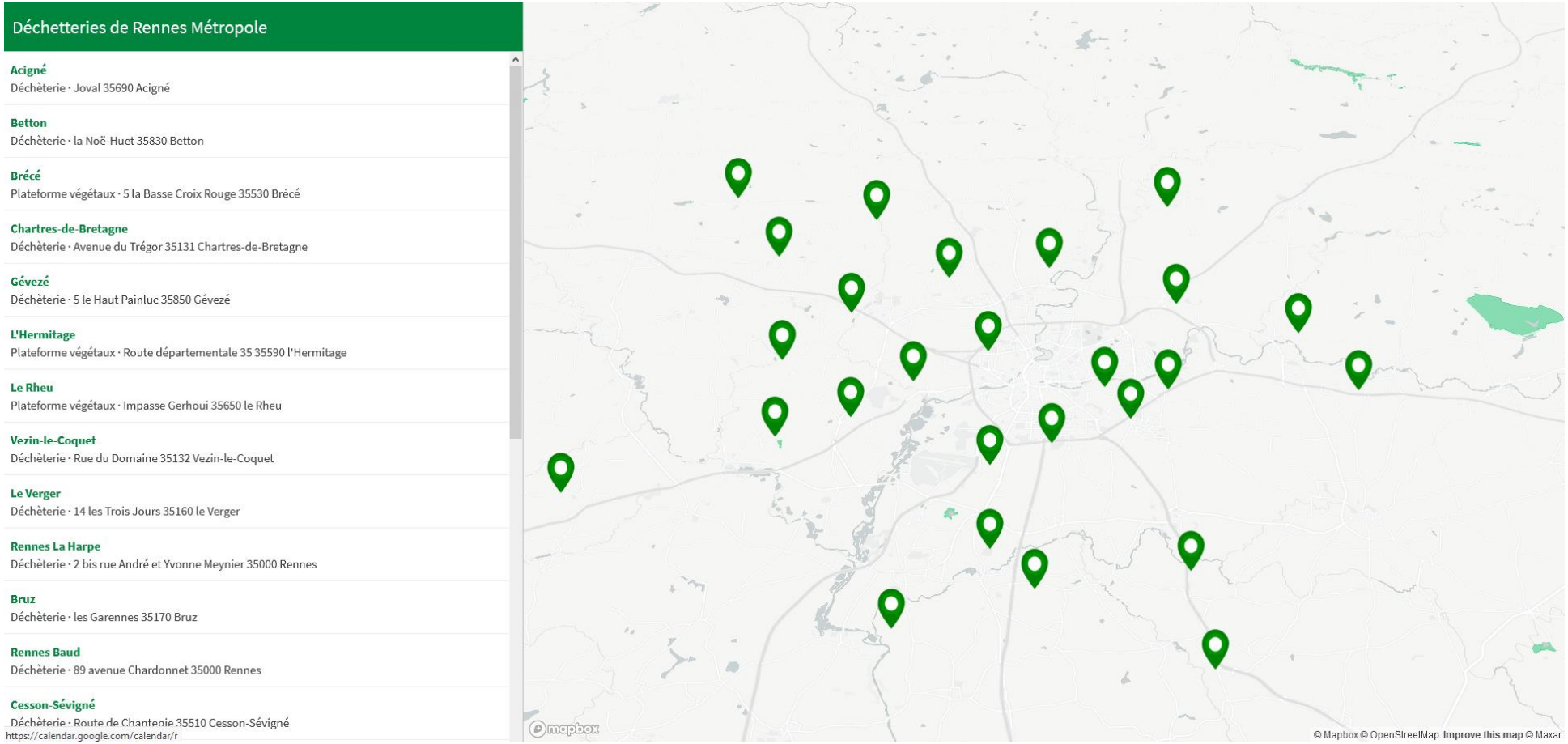
# Tutoriels

https://docs.mapbox.com/help/tutorials/?product=Mapbox+GL+JS



# Aller plus loin

https://docs.mapbox.com/help/tutorials/building-a-store-locator/




https://codepen.io/BorisMericskay/pen/dyqVoBQ