

2020

Feuille de route TP : Développement d'une application cartographique avec l'API Leaflet

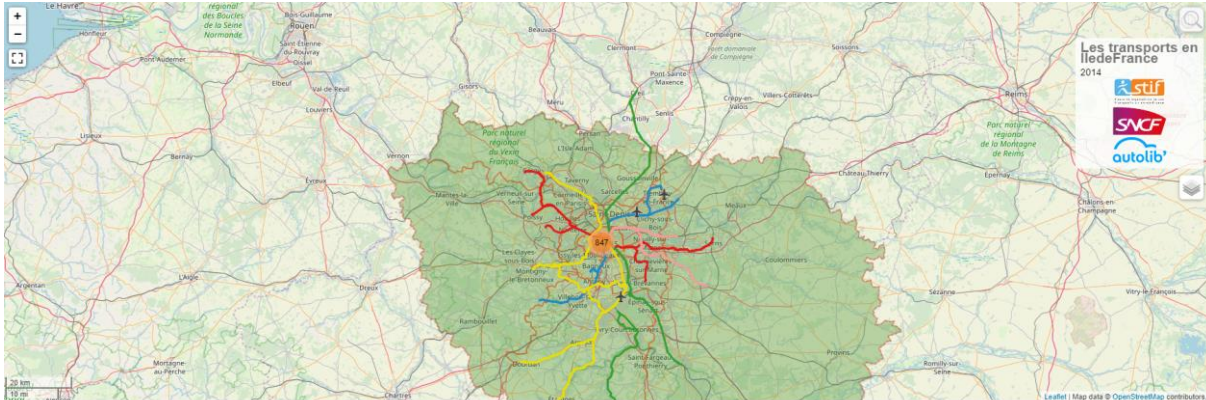
Serveur de données géographiques

Leaflet

Sujet : (zip) \Leaflet\doc\tp_leaflet_ESIPE-IG1

Qu'ai-je appris ?

- Développement avec une librairie js cartographique interactive



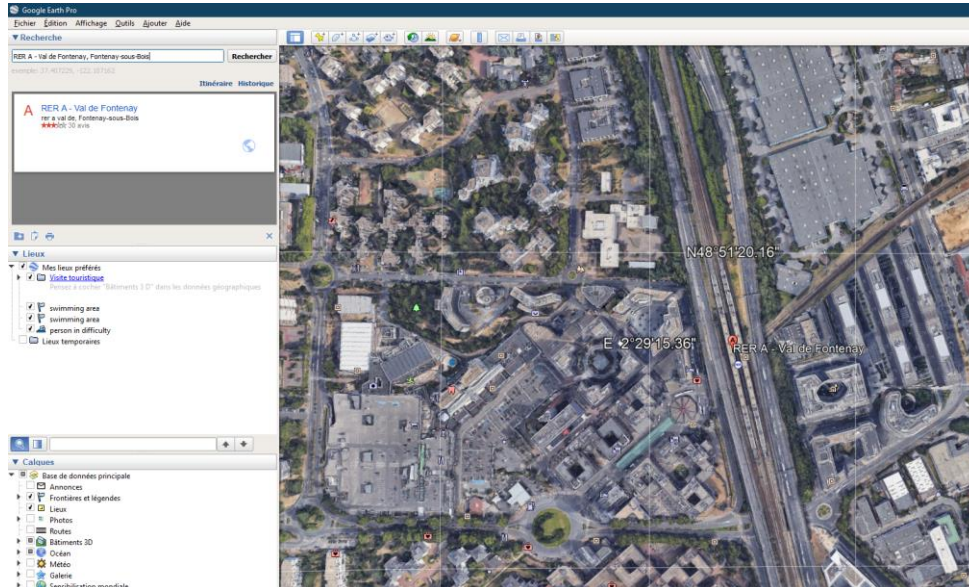
- Ajout de calques GeoJSON
- Le format Geojson en pratique
- Révision du langage javascript avec des notions cartographiques
- Développement en javascript ES6

```
// detect fullscreen toggling
map.on('enterFullscreen', () => {
  if (window.console) window.console.log('enterFullscreen');
});
map.on('exitFullscreen', () => {
  if (window.console) window.console.log('exitFullscreen');
});
```

- Import de librairie externes (OSMGeocoder)
<https://github.com/k4r573n/leaflet-control-osm-geocoder>
- Un peu de css.

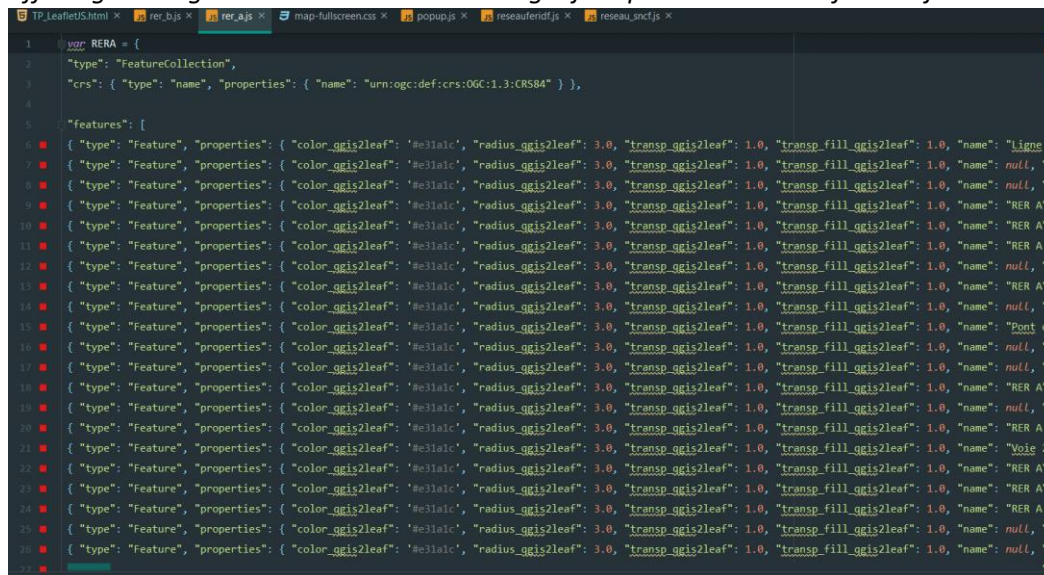
```
.info {
  padding: 6px 8px;
  font: 14px/16px Arial, Helvetica, sans-serif;
  background: white;
  background: rgba(255,255,255,0.8);
  boxshadow: 0 0 15px rgba(0,0,0,0.2);
  border-radius: 5px;
}
.info h2 {
  margin: 0 0 5px;
  color: #777;
}
```

- Navigation sur Google Earth Pro



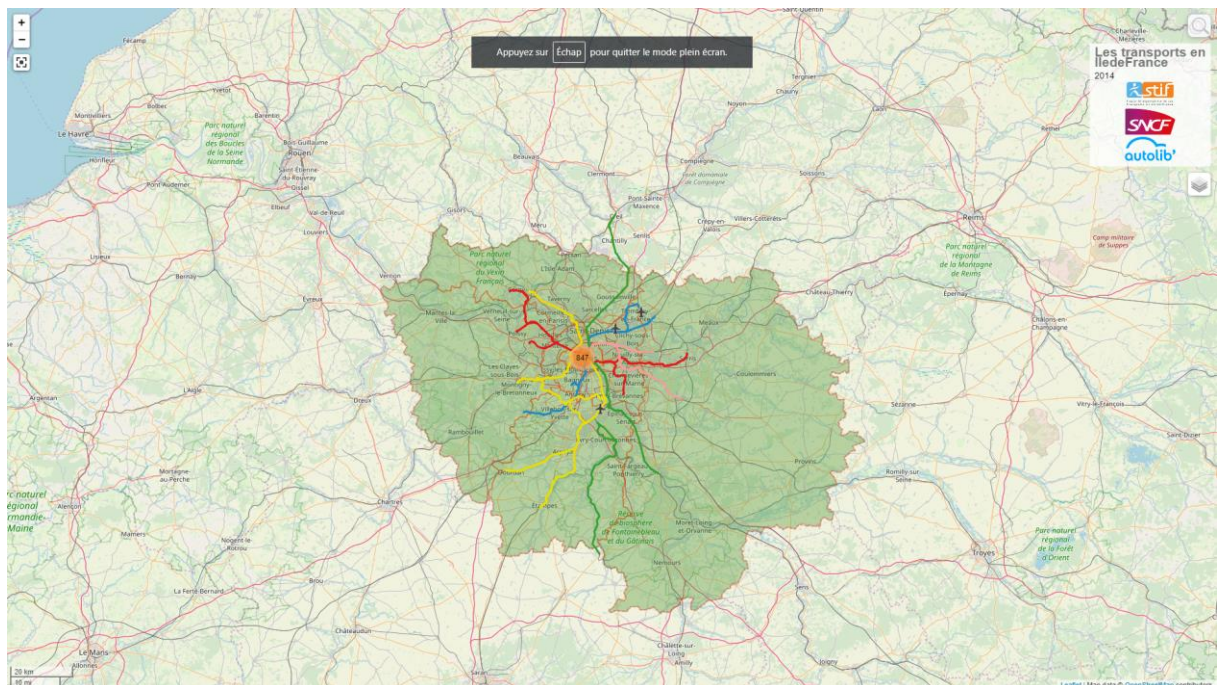
Ce qui **fonctionne** dans l'application

- Affichage global de la carte et de l'option de zoom
- Affichage des lignes RER à l'aide de chaînes geojson placées dans des fichiers js :





- Affichage des stations autolib.
- Affichage en mode plein écran à l'aide des fichiers css/js Leaflet.fullscreen



- Ajout des aéroports (marqueurs, à l'aide d'un LayerGroup et d'une image) :



- Ajout de titres et sous-titres directement sur la carte
- Interrogation des attributs à l'aide d'une fonction popup dans le fichier popup.js, et de l'appel de celle-ci dans le script principal :

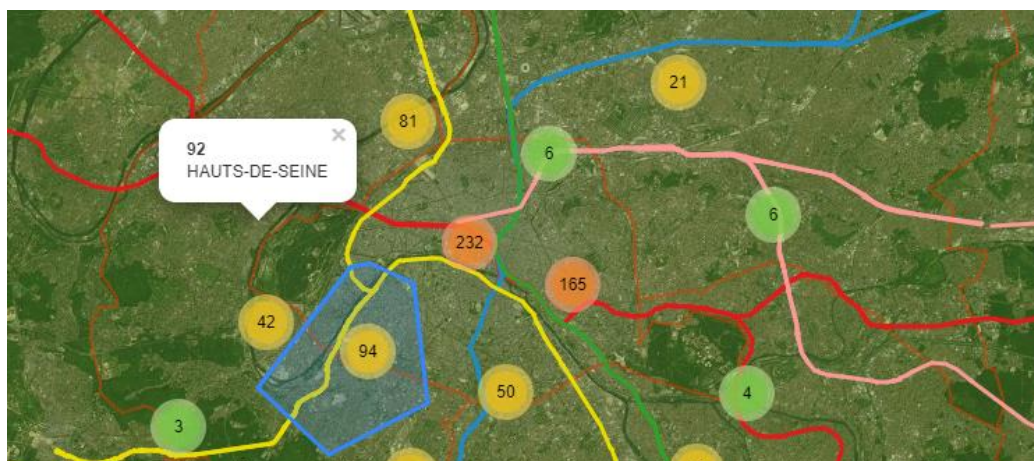
```
//fonction popup
function onEachFeature(feature, layer) {
  if (feature.properties) {
    layer.bindPopup("<b>" + feature.properties.CODE_DEPT + "</b><br>"
+
    feature.properties.NOM_DEPT);
  }
}
```

```
//Ajout du Layer GeoJSON Departements IDF
var deps_idf = new L.geoJson(deps_idf, {
  onEachFeature: onEachFeature,
  style: function (feature) {
    return {
      color:
        feature.properties.border_color_qgis2leaf,
      fillColor:
        feature.properties.color_qgis2leaf,
      weight:
        feature.properties.radius_qgis2leaf,
      opacity:
        feature.properties.transp_qgis2leaf,
      fillOpacity:
        feature.properties.transp_qgis2leaf
    };
  }
});
```

- Affichage de logo de transports idf :



Résultat au clic :



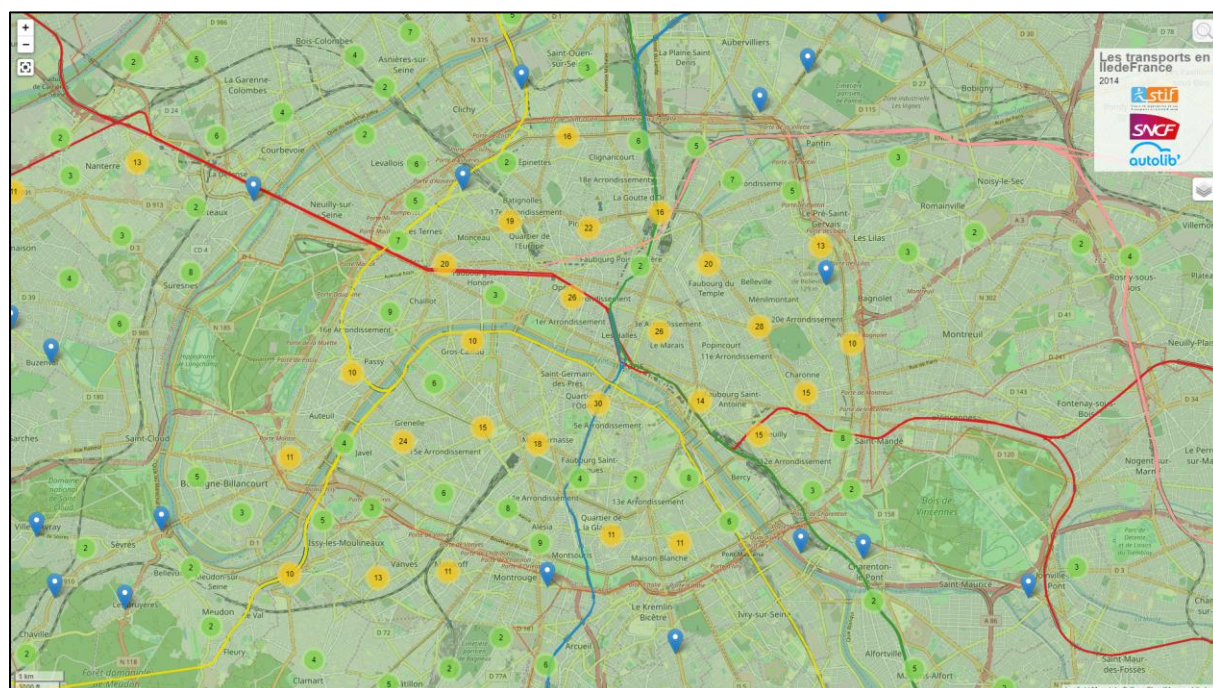
- Groupement de marqueurs

- Géo-Recherche :

A l'aide d'une fonction de géocodage du script Control.OSMGeocoder.js et de la feuille de style Control.OSMGeocoder.js.

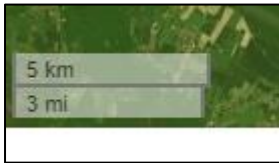


Résultat :



- Barre d'échelle

```
L.control.scale().addTo(map);
```

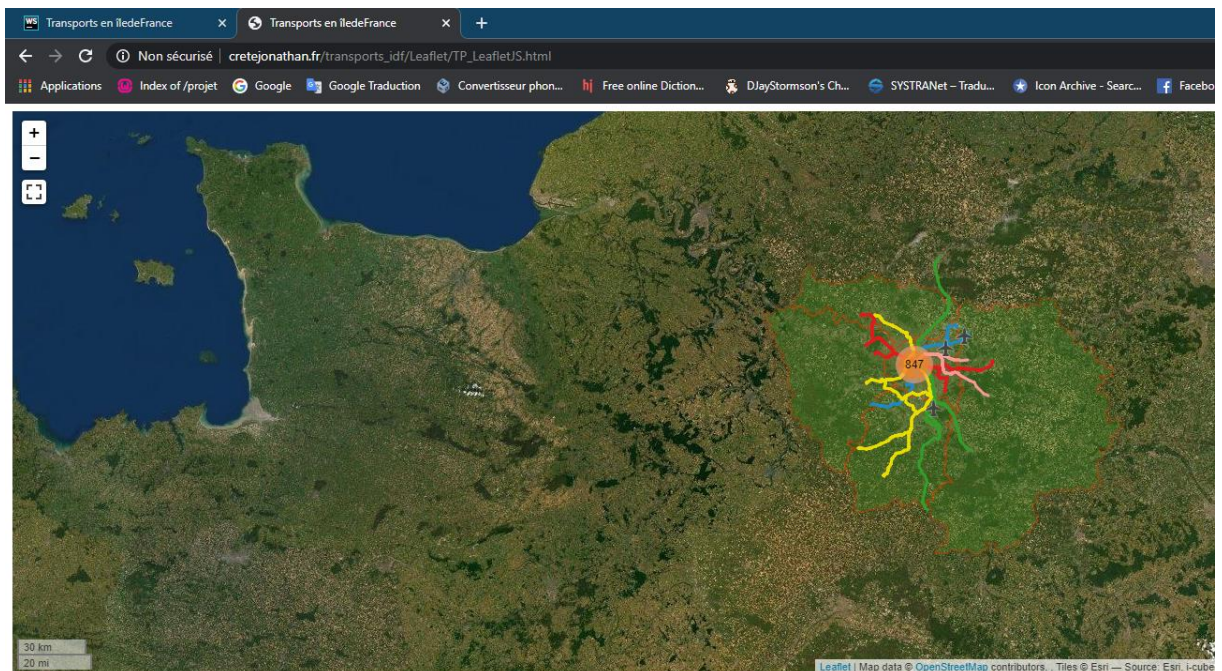


- Affichage de deux couches (fonds OSM) et contrôle :



- Hébergement de l'application sur mon hébergeur hostinger :

http://www.cretejonathan.fr/transports_idf/Leaflet/TP_LeafletJS.html



(Utilisation du client ftp FileZilla)

- Versionning de l'application :
Sur mon repository gitHub (branche master) :
https://github.com/JonathanCrt/INFO_Geomatic/tree/master

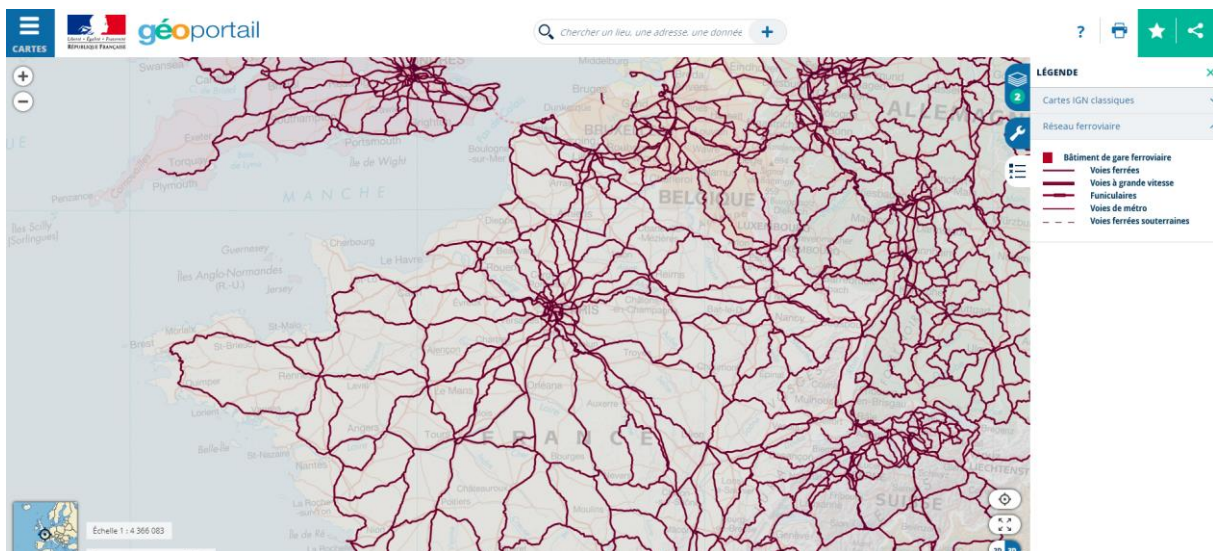
Ce qui ne **fonctionne pas** dans l'application

- Affichage du fond BING Maps Satellite :
Je n'ai pas réussi à récupérer une couche sur le site BING, je suppose qu'il s'agit d'une auquel nous devons posséder une clé/jeton pour pouvoir nous identifier et récupérer la couche.

Conclusion

Ce TP sur LeafletJS est très intéressant dans la mesure où il mélange des compétences de développement web (que je possède) et des notions de cartographies (Que je suis en train d'apprendre, WebMapping). J'ai pu manipuler un Framework avec de riches fonctionnalités, me permettant de construire une carte interactive intéressante. Si j'ai le temps, je pourrai essayer d'en faire une autre, par exemple pour afficher une carte du réseau ferroviaire de France:

Dans cette idée:



<https://www.geoportail.gouv.fr/donnees/reseau-ferroviaire>