

Institut Galilée Paris 13	LANGAGES WEB - JAVASCRIPT Exercices LEAFLET	Réf. : WEBC [0.9]
		Page : 1 - 1

1- Carte Nominatim

Dans cet exercice on utilise jQuery ainsi que la librairie Leaflet pour la cartographie.

Centrer au départ (dès le chargement de page) la carte sur votre pays préféré ou à défaut sur le Groenland ou au pire sur le point [0,0] de la planète. Associez ensuite au click sur la carte une fonction onClick() par exemple. Un paramètre e (par exemple sera automatiquement fourni. e contiendra la lat/lon (e.latlng).

A partir de ce point cliqué on devra interroger via un appel AJAX le service web de Nominatim qui à partir d'un point du globe est capable de vous donner des informations (pays, code ISO du pays, etc...).

L'url à invoquer est la suivante : <http://nominatim.openstreetmap.org/reverse>.

Documentez-vous sur : <https://nominatim.org/release-docs/develop/api/Overview/>

Il apparaîtra que la requête peut par exemple contenir les paramètres suivants : limit=1,lat=...,lon=...

Codez l'appel AJAX adéquat et affichez dans la <div> info le résultat de la requête au format json indenté proprement.

A partir de là, au niveau de la carte, on veut :

- Un marqueur standard au point cliqué,
- Un popup associé à ce marqueur (avec lat/lon le nom du pays et le code du pays)
- Un cercle dont le centre sera le point cliqué (porteur rouge, intérieur jaune, opacité 0.5 et d'un rayon de 30000m)
- Et que la carte soit centrée sur le marqueur !

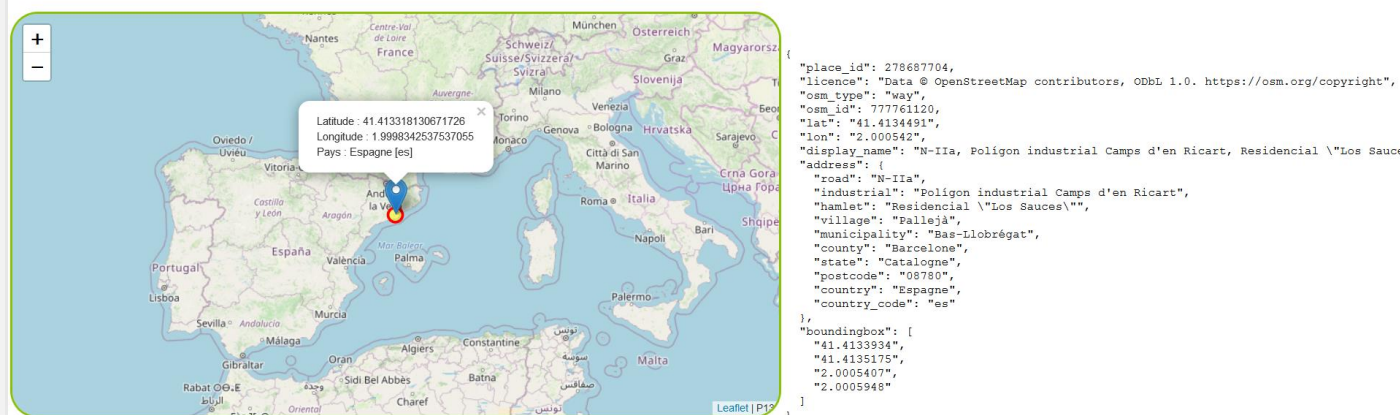
Voici à quoi ça ressemble (après avoir cliqué dans la région de Barcelone) :

jQuery + AJAX + JSON + Leaflet + Nominatim

Interaction avec la carte :

Cliquer sur la map pour faire apparaître un marqueur indiquant les coordonnées du point cliqué ainsi que le pays correspondant

Par ailleurs, à droite de la carte, sera affiché l'intégralité de la réponse JSON du service web Nominatim



Latitude: 41.413318130671726
Longitude: 1.9998342537537055
Pays: Espagne [es]

```
{
  "place_id": 276687704,
  "licence": "Data © OpenStreetMap contributors, ODbL 1.0. https://osm.org/copyright",
  "osm_type": "way",
  "osm_id": 77761120,
  "lat": "41.4134491",
  "lon": "2.000542",
  "display_name": "N-IIa, Poligon industrial Camps d'en Ricart, Residencial \"Los Sauces\"",
  "address": {
    "road": "N-IIa",
    "industrial": "Poligon industrial Camps d'en Ricart",
    "hamlet": "Residencial \"Los Sauces\"",
    "village": "Pallejà",
    "municipality": "Bas-Llobregat",
    "county": "Barcelone",
    "state": "Catalogne",
    "postcode": "08780",
    "country": "Espagne",
    "country_code": "es"
  },
  "boundingbox": [
    41.4133934,
    41.4135175,
    2.0005407,
    2.0005946
  ]
}
```

2- Carte des régions de France vs geojson

Dans cet exercice on utilise jQuery ainsi que la librairie Leaflet pour la cartographie. On va manipuler le format geojson (un json orienté cartographie). Nous allons écrire du code permettant de sélectionner une région de la France pour en afficher les contours sur une carte.

Pour cela, vous avez à votre disposition un dossier regions qui contient tous les contours de toutes les régions de France. Il ya un fichier spécifique à chaque région et il est au format geojson.

Par ailleurs vous avez à votre disposition un fichier au format json (regions.json) indiquant pour chaque région le nom du fichier correspondant. C'est juste un tableau de correspondance pour faciliter le travail.

Et enfin, vous disposez d'un fichier HTML dans lequel la carte de la France s'affiche.

Réaliser l'appel AJAX permettant de récupérer dans un premier temps le tableau contenu dans regions.json. Vous aurez ainsi à votre disposition la liste des noms de fichiers geojson pour chacune des régions.

Ensuite, on surveille l'évènement « change » sur le select pour déclencher un nouvel appel AJAX récupérant les contours de la région qui vient d'être sélectionnée.

Pour créer une couche (layer) geoJson en Leaflet on utilise L.geoJson(data,{style: monStyle}). Ici, l'objet data contient les données au format geoJson. Le style est optionnel. Ici, il peut nous servir à personnaliser l'affichage. Par exemple :

```
var monStyle = {"color": "#ff7800", "weight": 5, "opacity": 0.65};
```

couleur orange, contour d'épaisseur 5 et un peu d'opacité pour l'intérieur de la zone.

Après avoir « ajouter » la couche geoJson à la carte il faudra veiller à centrer la carte sur la région sélectionnée. Enfin on aimerait que lorsque l'on clique sur la région un popup indiquant « nom de la région :J'arrive sur zone ! » s'affiche.

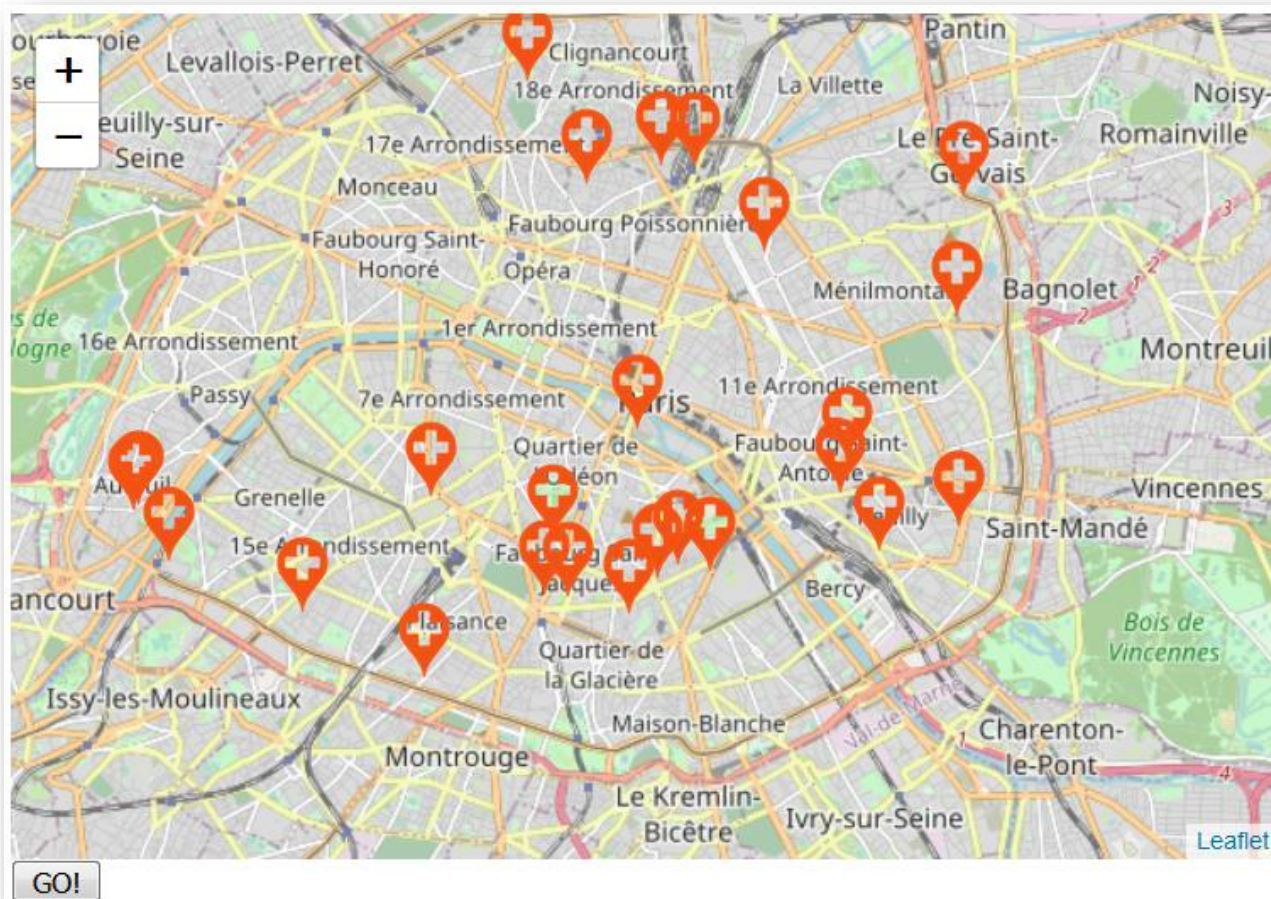
Ca donne à peu près ca lorsque l'on choisit l'Ile de France et qu'on clique dessus:



3- Hopitaux la suite – Etape 4

On reprend là où on était avec les CHR de la ville de Paris.

Quand on clique sur GO on veut désormais ca :



Utilisez ce marqueur personnalisé (icône fournie) :

```
var monIcone = L.icon({
  iconUrl: 'marker-icon-58.png', /* Image à FOND TRANSPARENT !! */
  shadowUrl: '', /* Pas obligatoire, mais avec une ombre, ça fait plus pro */
  iconSize: [36, 36],
  //shadowSize: [50, 20],
  iconAnchor: [18, 36],
  //shadowAnchor: [10, 20],
  //popupAnchor: [0, -24]
});
```

3- Hopitaux la suite – Etape 5

Quand on clique sur un des marqueurs on veut quelque chose comme ca (un popup) :



3- Hopitaux la suite – Etape 6

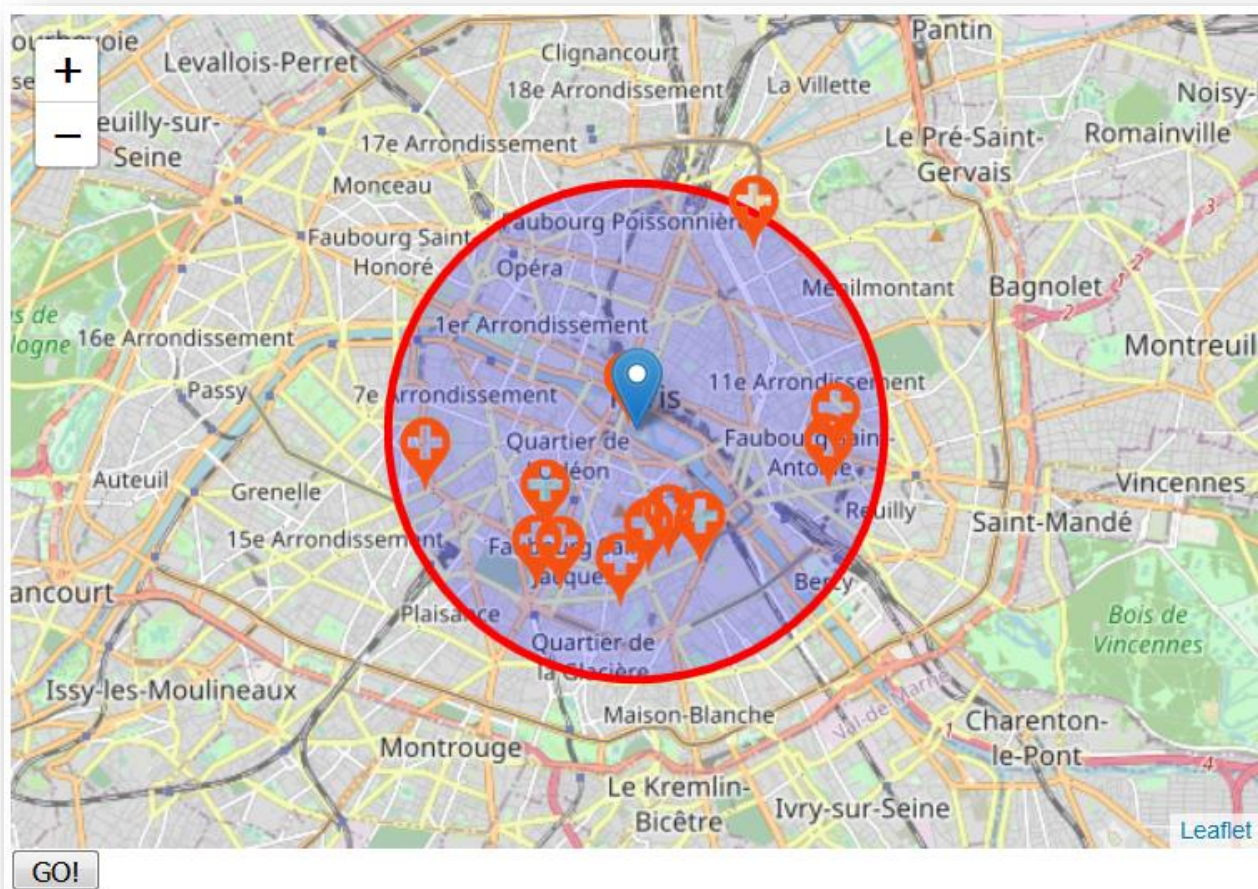
Plus compliqué cette fois ! On veut en plus dessiner un cercle autour d'un marqueur (indiquant notre Dame de paris). Ce marqueur est standard (le bleu de Leaflet). Le cercle a donc pour centre ce marqueur et un rayon (radius) de 3000m. Pour le cercle on a une bordure rouge d'épaisseur 4 et de couleur intérieure bleue.

Et on doit en plus filter les marqueurs c'est à dire que l'on affiche que ceux qui sont au sein même du cercle (pas les autres).

Voici une fonction qui pourra sans doute vous aider :

```
function estDansLeCercle(point,cercle) {
  var rayon = cercle.getRadius(); //on récupère le rayon en mètres
  var centreCercle = cercle.getLatLng(); //on récupère les coordonnées du centre
  du cercle
  var booleen = Math.abs(centreCercle.distanceTo(point)) <= rayon;
  return booleen ;
}
```


Voici ce que cela peut donner :

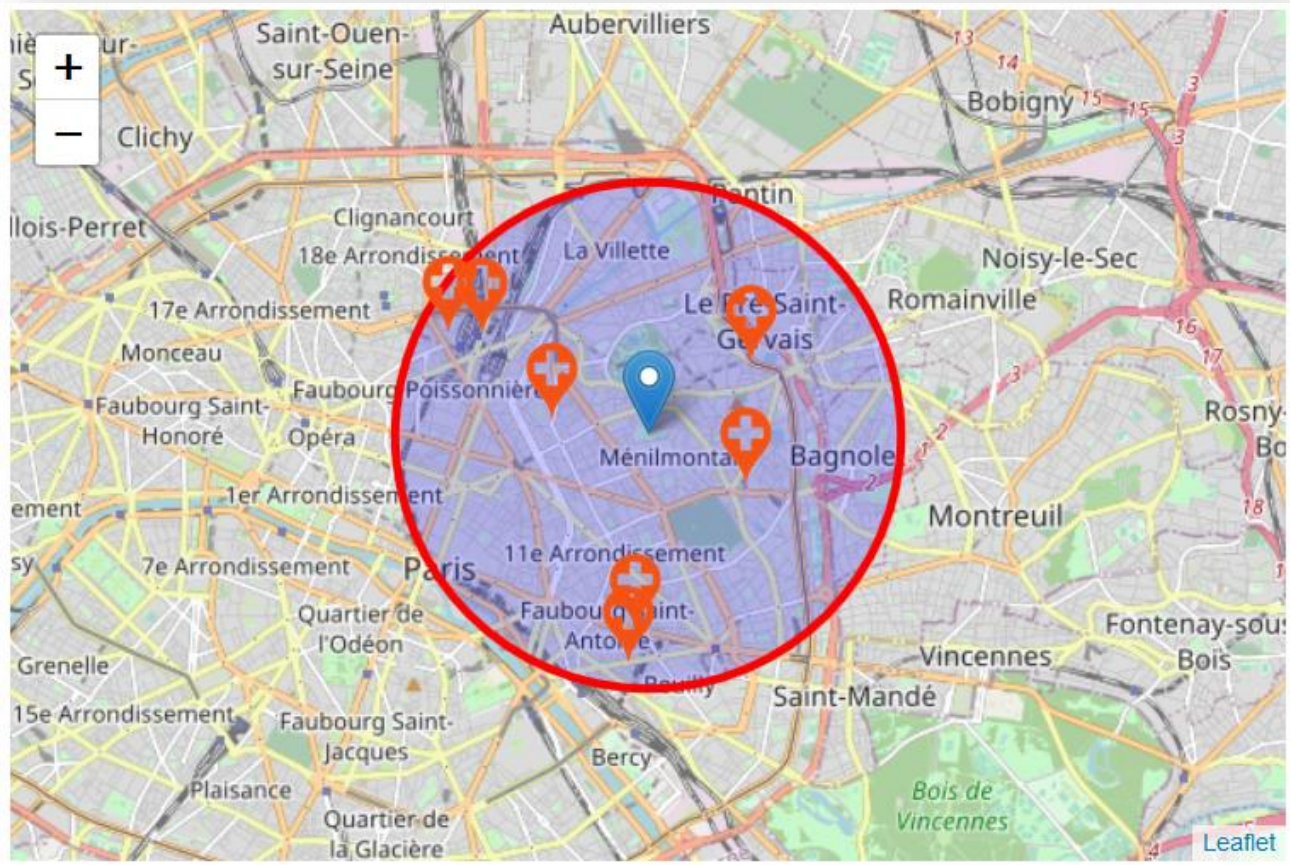


La carte reste centrée sur Notre Dame de Paris !

3- Hopitaux la suite – Etape 7

Il n'y a plus de bouton GO. Quand on double-clique n'importe où, sur la carte : le marqueur standard s'affiche (au point choisi) et s'affichent tout autour les hopitaux de Paris qui sont à moins de 3000m de rayon !

Ca peut donner ça (en double-cliquant sur un point nord-est de Paris) :



La carte est centrée sur le point double-cliqué. Pour désactiver l'effet de zoom au double-click utilisez : `map.doubleClickZoom.disable();`

Attention, si je double-clique ailleurs sur la carte, celle-ci doit être « nettoyée » avant d'afficher les nouveaux marqueurs. Pour nettoyer, ce code peut vous aider :

```
map.eachLayer(function (layer) {
  if (layer !== tile) {
    map.removeLayer(layer);
  }
});
```

Ce code nettoie (méthode `removeLayer()`) tout SAUF le fond de carte (disponible dans la variable nommée ici « `tile` »)

Institut Galilée	LANGAGES WEB - JAVASCRIPT Exercices LEAFLET	Réf. : WEBC [0.9]
Paris 13		Page : 1 - 7

3- Hopitaux une suite encore ?

Un `<input>` où l'on puisse indiquer le rayon du cercle (de recherche) en mètres ! J'ai la flemme ! Je ne l'ai pas fait cet exercice. A vous de jouer si ca vous amuse !
 Une géolocalisation (via le navigateur) aussi, ca serait bien.