

Carte Interactive

I- Map Box

Mapbox est une entreprise américaine spécialisée dans la cartographie en ligne. Elle fournit les cartes de sites, services et médias.

Nous utilisons une api fournis par MapBox afin d'intégrer grâce à du code javascript une carte interactive.

Afin de pouvoir intégré l'api, nous utilisons des fonctions javascript fourni par le bundle leaflet de symfony. Ce bundle nous permet de mieux gérer les cartes affiché en javascript.

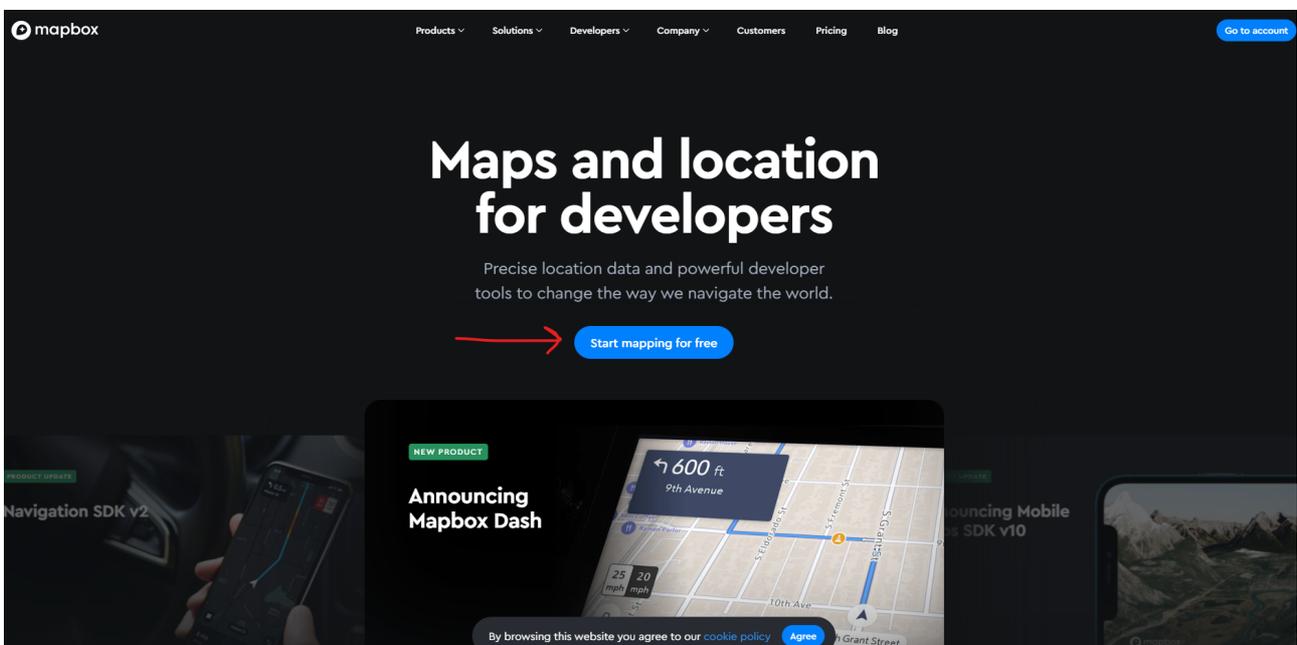
```
composer require leaflet/leaflet
```

Pour cela nous ajoutons un eventlistener dans le javascript de notre page html qui va afficher la carte. Nous avons mis par défaut les coordonnées du centre de la France Métropolitaine.

Nous devons générer sur le site de mapBox un token d'accès afin de pouvoir utiliser l'api.

1- Aller sur www.mapbox.com

2- cliquer sur « Start mapping for free »



3- S'inscrire

4- Cliquer sur « token » dans la navBar

5- Cliquer sur « + create a token »

Access tokens

You need an API access token to configure [Mapbox GL JS](#), [Mobile](#), and [Mapbox web services](#) like routing and geocoding. Read more about [API access tokens](#) in our documentation.

+ Create a token

6- Nommer le token puis cliquer sur « create token »

Create an access token

Token name

Choose a name to help associate it with a project.

Name

0 / 128

Token scopes

All tokens, regardless of the scopes included, are able to view styles, tilesets, and geocode locations for the token's owner. [Learn more](#).

Public scopes

- | | | | |
|--|---|--|---|
| <input checked="" type="checkbox"/> STYLES:TILES | <input checked="" type="checkbox"/> STYLES:READ | <input checked="" type="checkbox"/> FONTS:READ | <input checked="" type="checkbox"/> DATASETS:READ |
| <input checked="" type="checkbox"/> VISION:READ | | | |

Secret scopes

- | | | | |
|--|---|--|--|
| <input type="checkbox"/> SCOPES:LIST | <input type="checkbox"/> MAP:READ | <input type="checkbox"/> MAP:WRITE | <input type="checkbox"/> USER:READ |
| <input type="checkbox"/> USER:WRITE | <input type="checkbox"/> UPLOADS:READ | <input type="checkbox"/> UPLOADS:LIST | <input type="checkbox"/> UPLOADS:WRITE |
| <input type="checkbox"/> FONTS:LIST | <input type="checkbox"/> FONTS:WRITE | <input type="checkbox"/> STYLES:WRITE | <input type="checkbox"/> STYLES:LIST |
| <input type="checkbox"/> STYLES:DOWNLOAD | <input type="checkbox"/> STYLES:PROTECT | <input type="checkbox"/> TOKENS:READ | <input type="checkbox"/> TOKENS:WRITE |
| <input type="checkbox"/> DATASETS:LIST | <input type="checkbox"/> DATASETS:WRITE | <input type="checkbox"/> TILESETS:LIST | <input type="checkbox"/> TILESETS:READ |
| <input type="checkbox"/> TILESETS:WRITE | <input type="checkbox"/> DOWNLOADS:READ | <input type="checkbox"/> VISION:DOWNLOAD | <input type="checkbox"/> NAVIGATION:DOWNLOAD |
| <input type="checkbox"/> OFFLINE:READ | <input type="checkbox"/> OFFLINE:WRITE | | |

Token restrictions

Make your access tokens more secure by adding URL restrictions. When you add a URL restriction to a token, that token will only work for requests that originate from the URLs you specify. Tokens without restrictions will work for requests originating from any URL.

URLs

Restrict this token to specific URLs. You can add URLs one at a time or as a comma-separated list. Your URL's format is important. [Learn more about how to format your URL on our access token documentation page](#). This feature is compatible with many Mapbox tools with some limitations. For web applications using Mapbox GL JS, it requires version 0.53.1 and higher. It is not currently compatible with Mapbox native SDKs.

URL

Add URL

0 URLs

This token will work for requests originating from any URL.

Cancel

Create token

```
188 window.addEventListener("load", function (event) {
189   var mymap = L.map('map').setView([46.925539, 2.409112], 6);
190   L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token={accessToken}', {
191     attribution: 'Map data © OpenStreetMap contributors, Imagery © Mapbox',
192     'maxZoom': 18,
193     id: 'mapbox/streets-v11',
194     tileSize: 512,
195     zoomOffset: -1,
196     accessToken: 'pk.eyJ1IjoieG9yY2V2ZiIwIWI6ImNrdzZlMG9zYjJsZDAyb25vMWNidmdnb2gifQ.20frcp606JrWKD42dLzVeA'
197   }).addTo(mymap);
198 }
```



7- Copier le token créé

mapbox | Account Dashboard Tokens Statistics Invoices Settings

Access tokens

You need an API access token to configure [Mapbox GL JS](#), [Mobile](#), and [Mapbox web services](#) like routing and geocoding. Read more about [API access tokens](#) in our documentation.

[+ Create a token](#)

Name	Token	Last modified	URLs
gsbMap	pk.eyJ1IjoieG9yY2V2ZiIwIWI6ImNrdzZlMG9zYjJsZDAyb25vMWNidmdnb2gifQ.20frcp606JrWKD42dLzVeA	5 months ago	0

8- Coller le dans « accessToken » de votre javascript

```
accessToken: 'pk.eyJ1IjoieG9yY2V2ZiIwIWI6ImNrdzZlMG9zYjJsZDAyb25vMWNidmdnb2gifQ.20frcp606JrWKD42dLzVeA'
```

9- Saisir les coordonnées ainsi que le zoom par défaut

```
var mymap = L.map('map').setView([46.925539, 2.409112], 6);
```

Le premier paramètre est un tableau de deux nombre. Le premier nombre est la latitude, le second est la longitude. Le second paramètre est le zoom de la carte.

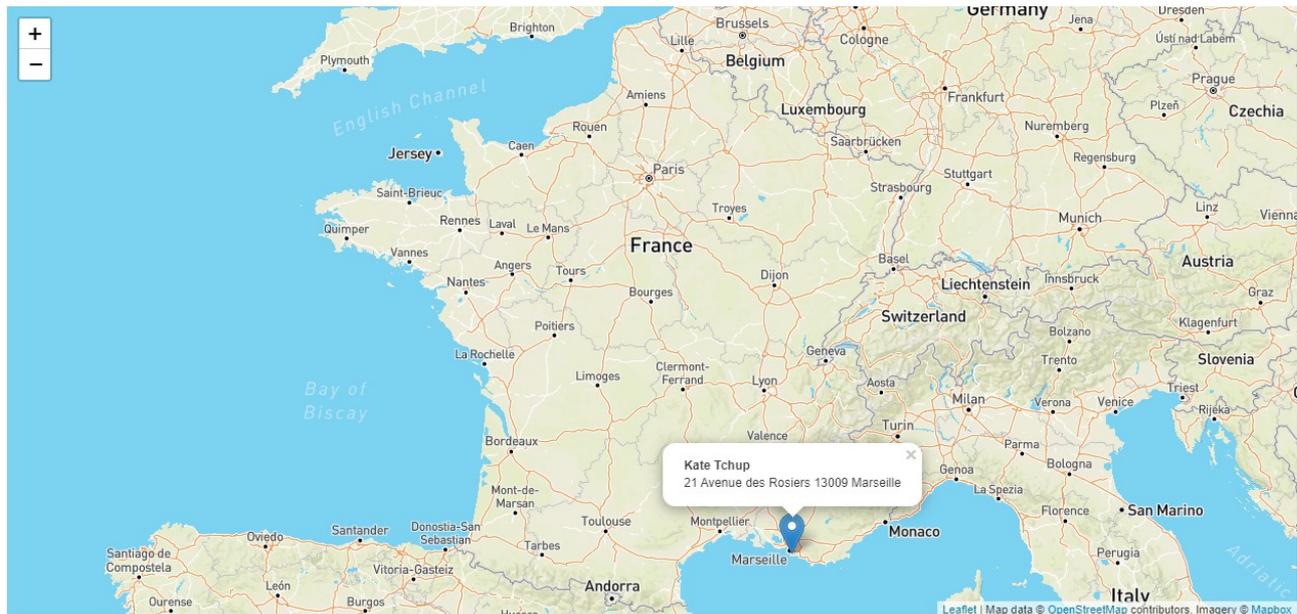
Voici le résultat :

```
188 window.addEventListener("load", function (event) {
189   var mymap = L.map('map').setView([46.925539, 2.409112], 6);
190   L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token={accessToken}', {
191     attribution: 'Map data ©&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors, Imagery © ' +
192       '<a href="https://www.mapbox.com/">Mapbox</a>',
193     maxZoom: 18,
194     id: 'mapbox/streets-v11',
195     tileSize: 512,
196     zoomOffset: -1,
197     accessToken: 'pk.eyJ1IjoieG9yY2VzIiw1IjI6ImNrdzZlMG9zYjJsZDAyb25vHWNlcm9udm92g1fQ.20frcp606JrWKD42dLzVeA'
198   }).addTo(mymap);
```



Par la suite grâce à leaflet nous pouvons ajouter des élément sur notre carte comme des pin contenant une description.

```
var marker = L.marker([pinLat, pinLon]).addTo(mymap);  
marker.bindPopup("<b>" + formatedName + "</b><br>" + pinAdress);
```



II- Coordonnées Gps

Afin de pouvoir afficher les « pin » sur la carte nous avons besoin des coordonnées gps de la position. Les pins correspondes aux adresses que l'ont veut rendre visible sur la carte.

Pour obtenir les coordonnées GPS de ces adresses nous avons utilisé l'api fourni par data.gouv, qui fournit pour toutes adresses du territoire français une coordonnées gps.

Voici le javascript.

```
window.fetch('https://api-adresse.data.gouv.fr/search/?q=' + formatedAddress).then(function (response) {  
    return response.json();  
});
```

Le json renvoyé nous fourni les coordonnées GPS de l'adresse saisie.

```
var coord = data.features[0].geometry.coordinates;  
var pinLon = coord[0];  
var pinLat = coord[1];
```

Puis nous les ajoutons à la carte :

```
var marker = L.marker([pinLat, pinLon]).addTo(mymap);
```