

Cartographie web

avec Leaflet

par
Richard Carlier

Cartographie web • avec Leaflet
par Richard Carlier

ISBN (imprimé) : 978-2-8227-1142-5

Copyright © 2024 Éditions D-Booker
Tous droits réservés

Conformément au Code de la propriété intellectuelle, seules les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective ainsi que les analyses et les courtes citations dans un but d'exemple et d'illustration sont autorisées. Tout autre représentation ou reproduction, qu'elle soit intégrale ou partielle, requiert expressément le consentement de l'éditeur (art L 122-4, L 122-5 2 et 3a).

Publié par les Éditions D-Booker, 229 rue Solférino, 59000 Lille
www.d-booker.fr
contact@d-booker.fr

Les exemples (téléchargeables ou non), sauf indication contraire, sont propriété des auteurs.

Visuel de couverture : La Garenne-Colombes map.svg, anonyme (Wikipédia)
Mise en page : générée sous Calenco avec des XSLT développées par la société NeoDoc (www.neodoc.biz)
Diffusion : AFPU-D (<https://www.afpu-diffusion.fr>)
Distribution : DILISCO (<https://dilisco-diffusion.centprod.com>)

Dépôt légal : Octobre 2024
Édition : 1
Version : 1.0

Table des matières

1. Introduction	1
1.1. Notre propos	1
1.2. Plan du livre	3
1.3. Des hommes derrière les cartes	4
Volodymyr Agafonkin	4
Votre guide	5
1.4. Sources des exemples	7
Préparation au voyage	9
2. Qu'est-ce qu'une carte ?	11
2.1. Projection(s)	12
2.2. Orientation	13
2.3. Coordonnées	16
2.4. Notations	19
2.5. Autres notions	21
3. Cartographie et Web	23
3.1. Historique rapide	23
3.2. Intégrer des cartes ?	24
4. Afficher une carte sur le Web	27
4.1. Iframe et partage	27
4.2. Construction JavaScript	29
5. Équipement et choix techniques	33
5.1. Environnement de développement	33
5.2. Leaflet : local vs CDN	33
5.3. Organisation du code et sources	34
Géo localisation, regarder du côté de chez sai.....	35
6. Une première carte	37
6.1. Créer un projet	37
6.2. Créer nos dossiers	37
6.3. Mettre en place Leaflet	38

6.4. Préparer le fichier HTML	39
6.5. Préparer le fichier JavaScript	40
6.6. Système de notation	42
7. Marqueurs, popup	45
7.1. Ajouter un marqueur et une fenêtre d'information	45
7.2. Écouteur de clics, récupérer des coordonnées	49
7.3. Ajouter une série de marqueurs	52
7.4. Marqueurs personnalisés	54
7.5. Éléments cartographiques	57
8. Cas pratique : carte interactive	61
8.1. Préparation des données	61
8.2. Les marqueurs	62
8.3. Charger les données, tracer les points	63
8.4. Ajouter un contrôleur	64
9. Fonds de cartes	67
9.1. Changer le fond de carte	67
9.2. Utiliser ses propres fonds de cartes	70
9.3. Système de tuiles	74
10. Dessiner sur la carte	77
10.1. Cercles (de confinement), rectangles	77
10.2. Lignes, et course cycliste	80
10.3. Polygones, et découpage scolaire	82
10.4. Images	84
11. Cas pratique : cartes narratives	87
11.1. Story map	87
11.2. Récupérer les informations	88
11.3. HTML et CSS	89
11.4. JavaScript	89
Géo découvertes, obtenir les infos d'une carte.....	93
12. Nous voulons des informations !	95
12.1. Retrouver les coordonnées du clic	95
12.2. Déplacer le marqueur	95
12.3. Déplacer la carte	96

13. Où suis-je ?	99
13.1. Géolocalisation de l'utilisateur	99
13.2. API Geolocation du JavaScript	100
13.3. Placement sur la carte	101
14. Géocodage	103
14.1. Principes	103
14.2. Nominatim	104
Récupérer ses coordonnées à partir d'une adresse	105
Récupérer une adresse à partir d'un point	106
14.3. Récupérer d'autres informations	108
15. Créer un plugin Leaflet	111
15.1. Travailler avec plusieurs points	111
15.2. Pourquoi passer par un plugin ?.....	112
15.3. Qui veut voyager loin, prépare sa monture... ..	112
15.4. L'extension MeasureDistance	113
15.5. En action	116
16. Cas pratique : autour d'un point	117
16.1. Cahier des charges	117
16.2. Organisation de la base de données	118
16.3. Installation d'un serveur Flask	120
16.4. Côté client, retour à Leaflet	122
16.5. Dans un rayon de... ..	124
Géo visualisation, donner du sens à des données.....	127
17. Visualisation de données géospatiales	129
17.1. Les précurseurs	129
17.2. John Snow et le choléra	129
17.3. Nombre de points et lisibilité	130
17.4. Type de représentation et lisibilité	132
17.5. L'art de la tranche... ..	133
18. Récupérer et utiliser des données.....	137
18.1. Formats de données géospatiales	137
18.2. Récupérer des données géolocalisées	140
18.3. GeoJSON et contours géographiques	141

Créer ses propres fichiers GeoJSON	142
Récupérer des contours géographiques	143
18.4. Afficher un GeoJSON dans Leaflet	144
18.5. Utilitaire : GeoJSON prévisualiseur	146
19. Dessiner à partir des données	149
19.1. Symboles proportionnels, et fromages	149
19.2. Cercles temporels, associations 1901 et séismes	153
20. Représentation graphique des données.....	157
20.1. Minicharts, croix et calvaires	157
20.2. ChartJS, et retour aux écoles	162
20.3. L'extension barChart	166
21. Cartes choroplèthes	169
21.1. Principe, et population européenne	169
21.2. Réalisation de la carte	170
22. Cartes thermiques/de densité	173
22.1. Principe, et incendies	173
22.2. Réalisation de la carte, commerces en Côte d'Ivoire	174
23. Cartes de regroupement (cluster)	177
23.1. Principe, et travailler à la Réunion	177
23.2. Réalisation de la carte, production viticole	178
24. Autres extensions	181
25. Cas pratique : désert médical	183
25.1. Multicartes	183
25.2. Préparation des données	184
25.3. Mise en place	184
25.4. JavaScript	185
25.5. Chaleur médicale... ..	187
Voyage au long cours	189
Annexes, aller plus loin	191
Leaflet, polyglotte	193
Leaflet et les frameworks JS	195

1. Vue	195
2. React	197
Leaflet et Python	199
1. Folium	199
2. Dash Leaflet, pour un tableau de bord	200
3. Autres applications	202
Leaflet et R	205
Leaflet et Wordpress	207
Leaflet et les autres...	209
Principales fonctions et méthodes Leaflet.....	211
Bibliographie sur les cartes	213
Index	215
À propos de l'auteur	219

3

Cartographie et Web

3.1. Historique rapide

L'histoire du *web mapping* (ou cartographie web) est relativement nouvelle et suit, logiquement, l'évolution d'Internet.

Dans les années 1990, le Web commence à se déployer, et les aspects cartographiques se retrouvent principalement sous la forme d'images statiques (parfois vectorielles), générées côté serveur. L'utilisateur ne pouvant guère agir, si ce n'est en rechargeant une autre carte. Les précurseurs sont Xerox PARC Map Viewer (1993), MapGuide (1995). Dès 1996, MapQuest propose le premier service de correspondance d'adresses et de routage en ligne populaire avec sortie cartographique.

L'Open Geospatial Consortium (OGC) va commencer à développer des services de cartographie web standardisés, comme WMS (service de cartographie web), WFS (service d'entités web) et WCS (service de couverture web), permettant d'accéder et d'échanger des données spatiales sur Internet, facilitant l'interaction et l'utilisation de données de provenance multiple par différentes applications de cartographie web.

Combiné à l'arrivée de JavaScript dans les navigateurs, les années 2000 vont voir apparaître des cartes dynamiques, interactives, mêlant données et fonctionnalités. C'est l'époque où vont se mettre en place des services que nous connaissons. Pour n'en citer que quelques-uns :

- 2004 : OpenStreetMap une carte du monde open-source, à contenu ouvert, fondée par Steve Coast ;
- 2005 : Google Maps, Bing Maps, Baidu Maps, Google Earth... ainsi qu'OpenLayers (la première version de la bibliothèque JavaScript open-source, qui sera utilisée par OpenStreetMap jusqu'à son remplacement par Leaflet) ;

- 2008 : Leaflet, qui s'inspire de la précédente (voir l'interview par Minds Behind Maps de Volodymyr Agafonkin sur Youtube [<https://d-booker.io.my/lf-va>] qui revient entre autres sur sa création [en anglais]) ;
- 2012 : Apple Maps (première application de cartographie basée sur des tuiles vectorielles) ;
- 2016 : Pokemon Go, un des premiers jeu vidéo mobile fondé sur la localisation (du moins à cette échelle – Ingress l'avait précédé en 2013) ;
- 2017 : HERE et son service de trafic en temps réel.

Les années ultérieures verront se développer les services existants, avec entre autres l'arrivée des images satellites sur les cartes et l'ajout de la 3D.

3.2. Intégrer des cartes ?

Pour mettre en place une carte sur un site web, nous disposons de plusieurs solutions selon ce que nous voulons obtenir.

Si notre choix se porte sur une carte simple, statique, il est parfois plus simple de mettre une capture écran d'une carte, assortie d'un marqueur indiquant l'emplacement d'un point (une boutique par exemple). On peut également utiliser une carte basée sur du JavaScript (le thème de cet ouvrage, développé au fil des chapitres), ou plus facilement un `iframe` que l'on intègre via un copier-coller (nous y revenons au chapitre suivant [p. 27]).

Une carte dynamique permettra d'afficher la carte, mais également d'y ajouter des points en fonction de certains critères comme la proximité de l'utilisateur (la boutique la plus proche de là où l'utilisateur se trouve).

La carte interactive ajoutera la possibilité à l'utilisateur de filtrer les affichages que l'on désire, en fonction de divers critères (les boutiques d'alimentation ou de vêtements les plus proches ainsi que le chemin le plus court pour y parvenir).

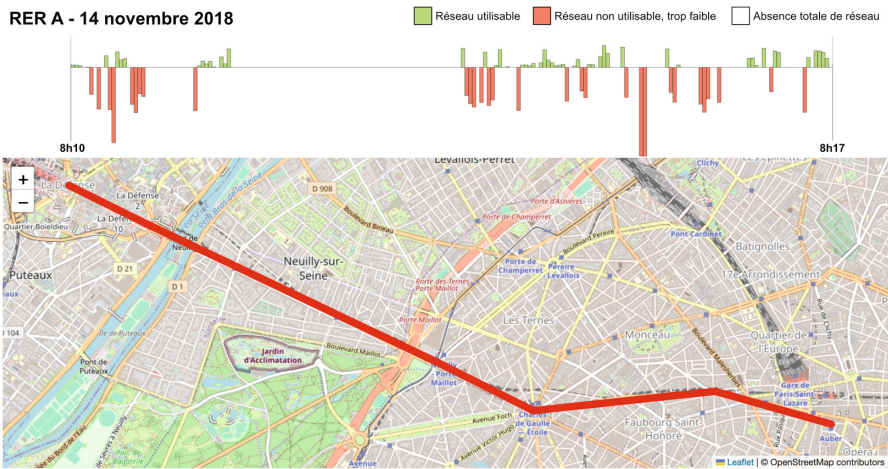
Pour ces deux dernières hypothèses, il faudra passer par l'utilisation d'une bibliothèque JavaScript.

Mettre en place ce genre de carte n'est pas négligeable en terme d'expérience utilisateur. Techniquement, le poids d'une carte dynamique est généralement plus conséquent que celui d'une simple image, et son impact sur les temps de chargement est souvent perceptible. Du fait du système de tuiles [p. 74], une carte dy-

namique va charger de nombreux fichiers, et ce à chaque consultation, déplacement, niveau de zoom.

Il peut être intéressant de ne pas forcer le chargement d'une carte dynamique lourde, qui s'avérerait désagréable pour l'utilisateur. Par exemple lorsque celui-ci se trouve avec un réseau dégradé dans un transport en commun parisien (qui passe sous la Seine). C'est du vécu, j'ai même pris des mesures.

Figure 3.1 : Qualité du réseau mesurée par l'auteur entre La Défense et Auber



La solution, si envisageable, est de proposer une image statique avec un lien vers la version dynamique accessible au choix de l'utilisateur.

7

Marqueurs, popup

7.1. Ajouter un marqueur et une fenêtre d'information

Notre carte est magnifique, mais encore bien vide. Elle est certes centrée sur un point, mais nous ne voyons pas ce point. Nous allons donc ajouter un marqueur. Pour y aller progressivement, dupliquons notre fichier `101.html` et son `101.js` en `102` (attention à modifier le `src` du JavaScript), et ajoutons ce marqueur.

Pour cela, à la suite de notre construction et après avoir ajouté le `layer`, il suffit de créer une instance de la classe `marker`, en précisant sa position (tableau ou objet), sans oublier de l'ajouter à la carte !

```
let center = [48.9068, 2.2464];

// création de la carte, code que nous ne reprendrons pas
// de façon systématique
let map = L.map("map");
map.setView(center, 17);
let layer = L.tileLayer(
  "https://tile.osm.org/{z}/{x}/{y}.png",
  { attribution: `&copy; OpenStreetMap contributors` }
);
layer.addTo(map);

// ajout de marqueur
const marker = L.marker(center);
marker.addTo(map);
```

Créer une variable `layer` ne sert pas réellement, à moins de vouloir agir sur ce `layer` (le cacher pour en montrer un autre par exemple). Je le mets assez souvent, plus par habitude. Pour `marker` non plus, si vous utilisez le chaînage, la variable n'est pas indispensable (mais rend le code souvent plus clair).

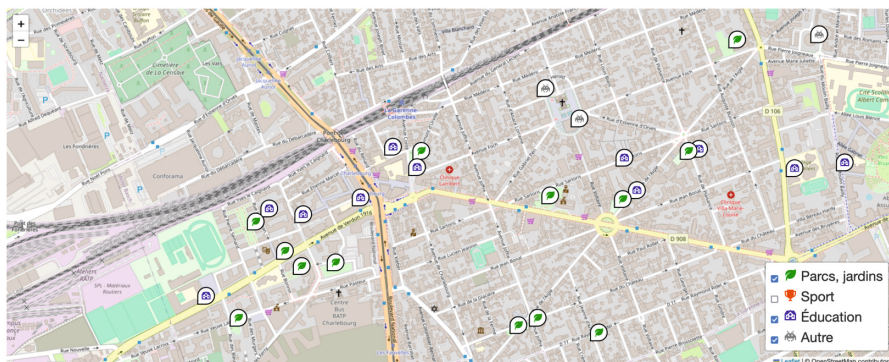
8

Cas pratique : carte interactive

Dans ce projet, nous allons construire une carte interactive, qui va permettre de :

- mettre en place notre carte ;
- charger un fichier de données ;
- tracer des marqueurs personnalisés selon le type d'endroit désigné ;
- ajouter un filtrage pour ne voir que certaines catégories d'endroits.

Figure 8.1 : Le résultat final, avec les sports décochés



8.1. Préparation des données

Nous allons pour ce projet utiliser les informations de la ville de La Garenne-Colombes, et positionner les points d'intérêt selon quatre catégories: les écoles (et collèges), les équipements sportifs, les parcs et jardins et une catégorie autre (pour la médiathèque, les marchés, etc.).

Nous allons créer un fichier JSON contenant les informations, et pour chacun d'entre eux nous allons les décrire ainsi :

10

Dessiner sur la carte

10.1. Cercles (de confinement), rectangles

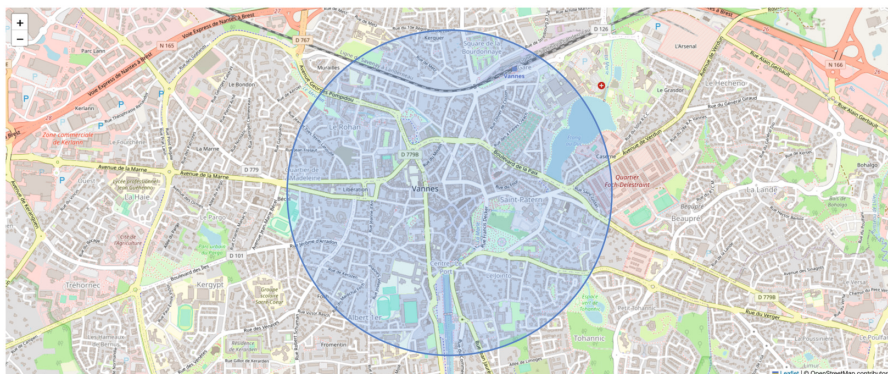
Tracer un cercle sur une carte est chose assez aisée, et également pratique pour indiquer une distance autour d'un point. Reprenons notre base habituelle d'une carte avec un centre, et rendons-nous à Vannes (ou Gwened si nous conservons les tuiles bretonnes).

```
let center = [47.65845, -2.75791];
let map = L.map("map").setView(center, 13);
let layer = L.tileLayer(
  "https://tile.osm.org/{z}/{x}/{y}.png",
  { attribution: `&copy; OpenStreetMap contributors` }
).addTo(map);
```

L'ajout du cercle va se faire en utilisant la bien nommée fonction `circle`, qui doit impérativement prendre un paramètre, le rayon, exprimé en mètres.

```
L.circle(center, { radius: 1000 }).addTo(map);
```

Figure 10.1 : Cercle bleu par défaut



16

Cas pratique : autour d'un point

16.1. Cahier des charges

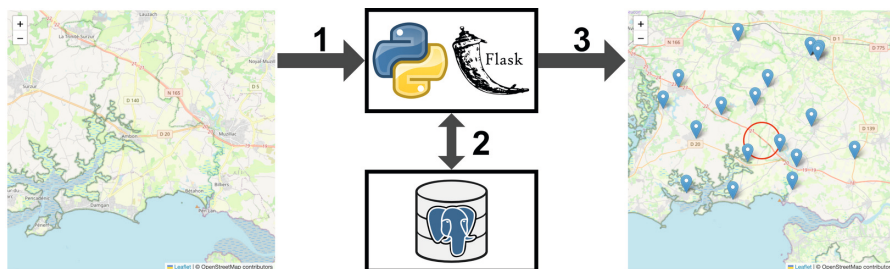
Pour ce cas pratique, nous allons construire une application un peu élaborée, mais qui est fort utile. À savoir communiquer entre le client (Leaflet) et le serveur (Python dans ce cas, avec Flask) de façon à récupérer des points d'intérêts.

Nous verrons en fait deux méthodes :

- les points les plus proches de nous (que l'on peut limiter, de façon arbitraire, à 20) ;
- tous les points compris dans un rayon donné.

Les deux méthodes fonctionnent peu ou prou de la même façon.

Figure 16.1 : Plusieurs phases, entre client, serveur et base de données



À savoir :

- nous commençons par tracer une carte, selon nos méthodes habituelles, centrée sur un point quelconque (il le faut...) ;
- nous récupérons les coordonnées de l'utilisateur ;

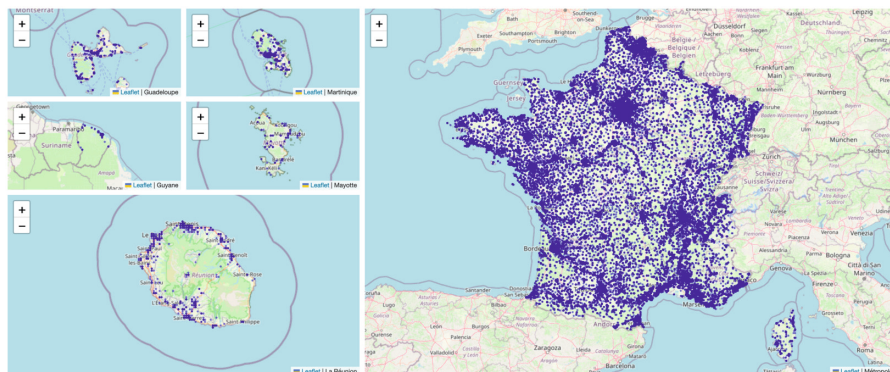
25

Cas pratique : désert médical

25.1. Multicartes

Comme nous l'avons vu, et utilisé pour notre carte narrative, il est possible de placer plusieurs cartes sur la même page. Nous allons le refaire, mais cette fois en présentant des données. L'idée est de montrer la répartition des professionnels de santé sur le territoire français, en affichant simultanément la métropole et les départements et régions d'outre-mer (Guadeloupe, Martinique, Guyane, La Réunion, et Mayotte), soit six cartes sur la même page.

Figure 25.1 : Professionnels de santé, en France métropolitaine et dans les Outre-mer



Pour la petite histoire j'attendais chez le médecin, ce qui est toujours un peu long (car un patient se doit de l'être) lorsque je planchais sur ce chapitre et cherchais une idée de données pour l'illustrer.