Engineer

Leaflet

Petite bibliothèque de cartes pour le navigateur

Ce que c’est

Leaflet est une bibliothèque JavaScript open source qui transforme un élément de page web en carte sur laquelle on peut se déplacer, zoomer et cliquer. Le projet annonce environ 42 Ko de JavaScript et aucune dépendance externe. La bibliothèque a été créée par Volodymyr Agafonkin ; elle est désormais développée par une communauté de contributeurs et distribuée sous licence BSD 2-Clause.

Sa présentation indique qu’il ne cherche pas à tout faire pour tout le monde et se concentre sur le bon fonctionnement des fonctions de base. `L.map` reçoit un élément et des options, et devient la carte. Une couche de tuiles récupère des images carrées depuis un modèle d’URL contenant `{z}` , `{x}` et `{y}` , puis les dispose en grille d’éléments image ; une variante WMS fait de même auprès d’un Web Map Service. Les polylignes, polygones, cercles et rectangles sont dessinés en SVG sans autre option, ou sur un seul canevas si la carte est créée avec `preferCanvas` . Les marqueurs sont des icônes, images ou simples `div` , auxquels sont attachées des fenêtres contextuelles et des infobulles. Une couche GeoJSON transforme chaque entité en l’une de ces couches, un point devenant un marqueur sauf indication contraire. Les contrôles regroupent des boutons de zoom, l’attribution, un sélecteur de couches et une barre d’échelle. Le système de référence de coordonnées prédéfini est EPSG:3857, que la documentation de référence décrit comme le plus courant pour les cartes en ligne.

Tout ce qui dépasse cette liste vient de plugins. Le catalogue du projet décrit Leaflet comme conçu pour être aussi léger que possible et renvoie à des centaines de plugins tiers, classés par groupes : tuiles vectorielles, regroupement, cartes de chaleur, édition géométrique, calcul d’itinéraires et géocodage, par exemple.

Deux branches de versions existent au moment de la rédaction. La stable est la 1.9.4, publiée le 18 mai 2023, et la documentation d'API du site du projet la décrit. La préversion 2.0.0-alpha.1 du 16 août 2025 abandonne Internet Explorer et est distribuée comme module ES. Elle ne conserve la variable globale `L` que dans le paquet `leaflet-global.js` et remplace les fonctions de fabrique par des constructeurs : `L.marker(latlng)` devient `new Marker(latlng)`. Elle reçoit les entrées par des événements de pointeur.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce que Leaflet ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Tout ce qui se situe quelque part — navires, véhicules, parcelles, capteurs — sur une carte où chercher, mesurer et partager.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Pipeline géographique accéléré de 50 fois — 50x d'accélération des pipelines de données géographiques

Pipeline géographique accéléré de 50 fois Contactez-nous →

Pourquoi c'est important

Mettre une carte sur une page implique deux décisions faciles à confondre : quel code dessine la carte dans le navigateur et d'où vient son fond cartographique. Leaflet les sépare, et le choisir consiste surtout à préserver cette séparation.

Leaflet dessine. Les tuiles viennent de l'endroit désigné par le modèle d'URL : le fond de carte est donc un choix séparé avec son prix, ses conditions et son propriétaire. Ce peut être le serveur de tuiles OpenStreetMap pendant le développement d'une fonctionnalité, un fournisseur commercial en production, un serveur exploité par le projet lui-même ou le WMS d'une commune via `TileLayer.WMS`. Passer de l'un à l'autre change une chaîne et une ligne d'attribution, sans modifier le code applicatif autour de la carte.

La taille et la licence déterminent les contextes où Leaflet convient bien. Environ 42 Ko et aucune dépendance conviennent à une page dont la carte est un élément parmi d'autres : un formulaire où placer un marqueur ou un écran d'administration indiquant l'emplacement des enregistrements. La licence BSD 2-Clause n'impose ni compte, ni clé, ni facturation à l'usage entre la bibliothèque et la page.

Le prix d'un petit cœur est que les fonctions plus lourdes arrivent dans des paquets séparés, chacun avec son responsable et son rythme de publication : une carte reposant sur six plugins hérite de six parcours de mise à jour. L'alpha 2.0 en montre les conséquences : ses notes de migration orientent les utilisateurs d'anciens plugins vers un polyfill V1, car ces plugins attendent la variable globale `L` et les fonctions de fabrique. Pour une carte nécessitant des tuiles vectorielles mises en forme dans le navigateur à chaque niveau de zoom, une bibliothèque conçue autour de ces tuiles est plus adaptée : mieux vaut le décider avant que la carte ne repose sur six plugins.

À quoi cela ressemble en pratique

Leaflet apparaît sur deux pages de ce site, toutes deux liées au rôle de responsable d'équipe de développement logiciel SIG entre 2015 et 2022. Dans les deux cas, il partage le frontend avec Mapbox.

Dans ce rôle, le travail a consisté à piloter plus de trente projets SIG, de la construction au déploiement. Leur interface reposait sur des cartes web Mapbox GL et Leaflet, parfois connectées aux API ArcGIS ou HERE, avec une couche applicative en JavaScript, Python, PHP et SQL. En dessous, PostgreSQL utilisait PostGIS, et GDAL/OGR déplaçait les données entre fichiers shapefile, GeoJSON, GeoTIFF, KML et tuiles vectorielles. La responsabilité allait au-delà de la livraison, jusqu'au support en production.

Ce rôle comprenait aussi une participation au processus de conception des interfaces utilisateur des projets. Il s'agissait de frontends React, Angular et Vue au-dessus de cartes Mapbox et Leaflet. La page souligne un point à garder à l'esprit avec toute bibliothèque cartographique : dans un produit de cartographie web, la carte est la partie facile ; ce sont les contrôles, le filtrage et le parcours autour qui désorientent les utilisateurs. L'ergonomie tenait à des détails comme le filtrage d'un jeu de données ou le passage entre étages sur une carte intérieure, et la page attribue l'essentiel de la différence à ces questions posées pendant la conception plutôt qu'après la sortie.

Où les problèmes surviennent

Leaflet est assez léger pour que la plupart de ses défaillances viennent de la page qui l'entoure ou des services vers lesquels il pointe.

La carte apparaît comme un rectangle gris ou une bande de tuiles éparses. Leaflet tire sa taille de son conteneur. Le guide de démarrage indique deux préalables à l'exécution du code de la carte : inclure le CSS de Leaflet avant le JavaScript et donner une hauteur définie au conteneur. Un `div` sans hauteur donne une carte sans hauteur. Un cas plus subtil est la carte créée dans un onglet ou une boîte de dialogue alors masqués : elle est mesurée à zéro et conserve cette taille lorsqu'elle apparaît. Leaflet 1.x suit le redimensionnement de la fenêtre du navigateur via `trackResize` : un conteneur qui change seul de taille passe donc inaperçu jusqu'à l'appel à `invalidateSize()` , que la documentation de référence prescrit après un changement de taille de la carte. L'alpha 2.0 utilise `ResizeObserver` à la place de l'événement de redimensionnement de la fenêtre.

Les points apparaissent à des milliers de kilomètres de leur emplacement. GeoJSON, tel que défini par la RFC 7946, écrit la longitude en premier dans une position. Le `LatLng` de Leaflet et les tableaux qu'il accepte à sa place mettent la latitude en premier. `L.geoJSON` convertit automatiquement entre les deux : le défaut apparaît donc lorsque les coordonnées sont extraites à la main de GeoJSON et transmises directement à un marqueur ou une polyligne. Un point à Copenhague dont les coordonnées sont inversées aboutit en mer d'Arabie. Lors du passage d'un format à l'autre, décomposez la paire en nommant chaque coordonnée :

```
const res = await fetch("/api/sites/42");
const site = await res.json();
const [lon, lat] = site.geometry.coordinates; // GeoJSON: lon, lat
L.marker([lat, lon]).addTo(map); // Leaflet: lat, lng
```

Le déplacement de la carte saccade quand les données augmentent. Avec le moteur SVG, chaque ligne ou polygone est un élément distinct du document, et chaque marqueur par défaut est une image distincte. Une couche de plusieurs milliers d'entités impose au navigateur de déplacer plusieurs milliers d'éléments à chaque mouvement et zoom. `preferCanvas: true` dessine les tracés sur un seul canevas, mais ne s'applique qu'aux tracés : les points doivent donc devenir des `L.circleMarker` , cercles de rayon fixe en pixels qui sont eux-mêmes des tracés, par l'option `pointToLayer` de la couche GeoJSON. Au-delà, la solution est d'envoyer moins : un plugin de regroupement ou une API qui ne renvoie que ce qui correspond à la vue et au zoom courants.

Les tuiles cessent de charger après le lancement. Le guide de démarrage construit sa première carte avec le serveur de tuiles OpenStreetMap et précise que ces tuiles conviennent à la programmation, mais qu'il faut lire la politique d'utilisation avant la production. Elle demande une attribution visible, un User-Agent identifiant l'application et un Referer valide depuis les pages web, une mise en cache conforme aux en-têtes HTTP, et interdit les téléchargements en masse ou la récupération anticipée. Elle n'offre aucun accord de niveau de service et peut bloquer l'accès sans préavis. Une carte de production très fréquentée doit utiliser un fournisseur dont les conditions conviennent à son trafic ou son propre serveur de tuiles ; avec Leaflet, changer consiste à modifier le modèle d'URL et le texte d'attribution.

Pour aller plus loin

- Leaflet — la présentation du projet, ses fonctions et une première carte en quelques lignes.
- Documentation de l'API Leaflet — chaque classe, option et méthode de la version 1.9.4, dont `preferCanvas` , `invalidateSize` et les options GeoJSON.
- Guide de démarrage de Leaflet — le CSS, la hauteur du conteneur et une première couche de tuiles, étape par étape.
- Publication de Leaflet 2.0 alpha — les changements de la version 2.0 et la migration du code et des plugins.
- Politique d'utilisation des tuiles — les conditions d'utilisation des serveurs de tuiles openstreetmap.org.
- RFC 7946 : format GeoJSON — le standard GeoJSON, dont l'ordre longitude d'abord des positions.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce que Leaflet ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/leaflet/