Engineer

Leaflet

kleine Bibliothek für Karten im Browser

Was es ist

Leaflet ist eine quelloffene JavaScript-Bibliothek, die ein Element einer Webseite in eine Karte verwandelt, die sich verschieben, zoomen und anklicken lässt. Das Projekt gibt seine Größe mit etwa 42 KB JavaScript an und nennt keine externen Abhängigkeiten. Volodymyr Agafonkin hat es entwickelt; heute wird es von einer Gemeinschaft von Mitwirkenden weiterentwickelt und unter der BSD-2-Clause-Lizenz veröffentlicht.

Die eigene Übersicht erklärt, dass Leaflet nicht alles für jeden leisten will, sondern sich darauf konzentriert, die grundlegenden Dinge gut zu erledigen. `L.map` übernimmt ein Element und eine Reihe von Optionen und wird zur Karte. Eine Kachelebene ruft quadratische Bilder aus einer URL-Vorlage mit `{z}`, `{x}` und `{y}` ab und ordnet sie als Raster aus Bildelementen an. Eine WMS-Variante tut dasselbe mit einem Web Map Service. Linienzüge, Polygone, Kreise und Rechtecke werden standardmäßig als SVG gezeichnet oder auf einer einzigen Canvas-Fläche, wenn die Karte mit `preferCanvas` erstellt wird. Marker sind Symbole, entweder ein Bild oder ein einfaches `div`, mit angehängten Popups und Tooltips. Eine GeoJSON-Ebene verwandelt jedes Geoobjekt in eine dieser Ebenen, einen Punkt in einen Marker, sofern nichts anderes vorgegeben ist. Zoomschaltflächen, Quellenangaben, ein Ebenenumschalter und ein Maßstabsbalken bilden die Bedienelemente. Das standardmäßige Koordinatenreferenzsystem ist EPSG:3857, laut Referenz das häufigste für Onlinekarten.

Alles über diese Liste hinaus kommt aus Plugins. Die Plugin-Datenbank des Projekts beschreibt Leaflet als möglichst leichtgewichtig und verweist auf Hunderte von Drittanbieter-Plugins, sortiert in Gruppen wie Vektorkacheln, Clustering, Heatmaps, Geometriebearbeitung, Routenplanung und Geokodierung.

Zum Zeitpunkt der Erstellung gibt es zwei Entwicklungslinien. Die stabile Version ist 1.9.4, veröffentlicht am 18. Mai 2023, und die API-Referenz auf der Projektwebsite entspricht ihr. Die Vorabversion 2.0.0-alpha.1 vom 16. August 2025 stellt die Unterstützung für Internet Explorer ein und wird als ES-Modul ausgeliefert. Die globale Variable `L` bleibt nur in der gebündelten Datei `leaflet-global.js` erhalten. Factory-Funktionen werden durch Konstruktoren ersetzt, sodass aus `L.marker(latlng)` ein `new Marker(latlng)` wird, und Eingaben werden über Pointer Events verarbeitet.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist Leaflet?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Alles, was an einem Ort geschieht — Schiffe, Fahrzeuge, Grundstücke, Sensoren — auf einer Karte, die Sie durchsuchen, vermessen und teilen können.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird.
Geografische Datenpipeline 50-mal schneller — 50x schnellere Geodaten-Pipelines

Geografische Datenpipeline 50-mal schneller Kontakt aufnehmen →

Warum es wichtig ist

Eine Karte in eine Seite einzubauen umfasst zwei Entscheidungen, die leicht für eine gehalten werden: welcher Code die Karte im Browser zeichnet und woher das Kartenbild stammt. Leaflet hält beides auseinander. Sich dafür zu entscheiden bedeutet vor allem, diese Trennung beizubehalten.

Leaflet zeichnet. Die Kacheln kommen von dort, wohin die URL-Vorlage zeigt. Die Grundkarte ist deshalb eine eigene Entscheidung mit eigenen Kosten, Bedingungen und einem eigenen Eigentümer. Während eine Funktion entsteht, kann sie vom OpenStreetMap-Kachelserver kommen, im Produktionsbetrieb von einem kommerziellen Anbieter, von einem selbst betriebenen Kachelserver des Projekts oder über `TileLayer.WMS` vom WMS einer Gemeinde. Ein Wechsel zwischen diesen Quellen ändert eine Zeichenkette und eine Zeile mit Quellenangaben; der Anwendungscode rund um die Karte bleibt unverändert.

Größe und Lizenz entscheiden, wo Leaflet gut hineinpasst. Etwa 42 KB und keine Abhängigkeiten passen zu einer Seite, auf der die Karte nur ein Bestandteil unter vielen ist: ein Formular, in dem jemand eine Markierung setzt, oder eine

Verwaltungsansicht, die die Orte von Datensätzen zeigt. Die BSD-2-Clause-Lizenz verlangt zwischen Bibliothek und Seite weder ein Konto noch einen Schlüssel oder eine Verbrauchsmessung.

Der Preis eines kleinen Kerns ist, dass aufwendigere Funktionen als separate Pakete hinzukommen, jedes mit eigenen Betreuern und einem eigenen Veröffentlichungstakt. Eine Karte, die auf sechs Plugins baut, übernimmt sechs Aktualisierungspfade. Die Alpha-Version 2.0 zeigt, was das bedeutet: Ihre Migrationshinweise verweisen Nutzer älterer Plugins auf ein V1-Polyfill, weil diese Plugins die globale Variable `L` und die Factory-Funktionen erwarten. Eine Karte, die bei jeder Zoomstufe im Browser gestaltete Vektorkacheln braucht, passt besser zu einer darauf aufgebauten Bibliothek. Diese Entscheidung sollte fallen, bevor sechs Plugins die Karte tragen.

Wie es in der Praxis aussieht

Leaflet kommt auf zwei Seiten dieser Website vor. Beide stammen aus der Rolle als Teamleitung für GIS-Softwareentwicklung zwischen 2015 und 2022; auf beiden teilt es sich das Frontend mit Mapbox.

In dieser Rolle wurden mehr als dreißig GIS-Projekte von der Entwicklung bis zur anschließenden Bereitstellung geleitet. Im Frontend dieser Projekte standen Webkarten auf Basis von Mapbox GL und Leaflet, teils mit Zugriff auf die APIs von ArcGIS oder HERE, mit der Anwendungsschicht in JavaScript, Python, PHP und SQL. Darunter lag PostgreSQL mit PostGIS, und GDAL/OGR übertrug Daten zwischen Shapefiles, GeoJSON, GeoTIFF, KML und Vektorkacheln. Die Verantwortung reichte über die Auslieferung hinaus bis zur Betreuung im Produktionsbetrieb.

Zu derselben Rolle gehörte auch die Mitwirkung am Entwurfsprozess für die Benutzeroberflächen ihrer Projekte. Es waren Frontends in React, Angular und Vue über Karten von Mapbox und Leaflet. Die Seite hebt einen Gedanken hervor, der neben jeder Kartenbibliothek stehen sollte: Bei einem Webkartenprodukt ist die Karte der einfache Teil; die Bedienelemente, die Filterung und die Abläufe darum herum sind die Stellen, an denen Menschen sich verirren. Die Bedienbarkeit steckte in Details wie der Filterung eines Datensatzes und dem Wechsel zwischen Stockwerken auf einer Innenraumkarte. Laut Seite entstand der größte Unterschied dadurch, dass diese Fragen schon beim Entwurf und nicht erst nach der Veröffentlichung gestellt wurden.

Wo es schiefgeht

Leaflet ist klein genug, dass die meisten Fehler von der umgebenden Seite oder von den Diensten ausgehen, auf die es zugreift.

Die Karte erscheint als grauer Kasten oder als Streifen versprengter Kacheln. Leaflet übernimmt seine Größe vom Container. Der Schnelleinstieg nennt zwei Dinge, die vor jedem Kartencode zu erledigen sind: die CSS-Datei von Leaflet vor dem JavaScript einbinden und dem Container eine definierte Höhe geben. Ein `div` ohne Höhe ergibt eine Karte ohne Höhe. Weniger offensichtlich ist der Fall einer Karte, die in einem zu diesem Zeitpunkt verborgenen Tab oder Dialog erstellt wird: Sie misst sich mit null und behält diese Größe beim Erscheinen bei. Leaflet 1.x folgt über `trackResize` Größenänderungen des Browserfensters. Ändert der Container allein seine Größe, bleibt das unbemerkt, bis der Code `invalidateSize()` aufruft. Laut Referenz ist dieser Aufruf nach einer Größenänderung der Karte nötig. Die Alpha-Version 2.0 verwendet `ResizeObserver` statt des Resize-Ereignisses des Fensters.

Punkte erscheinen Tausende Kilometer entfernt. GeoJSON schreibt eine Position gemäß RFC 7946 mit dem Längengrad zuerst. Leaflets `LatLng` und die Arrays, die es an dessen Stelle akzeptiert, beginnen mit dem Breitengrad. `L.geoJSON` wandelt selbstständig zwischen beiden um. Der Fehler tritt deshalb auf, wenn Koordinaten von Hand aus GeoJSON herausgenommen und unmittelbar an einen Marker oder einen Linienzug übergeben werden. Ein Punkt in Kopenhagen landet bei vertauschten Werten im Arabischen Meer. Entpacken Sie das Paar an der Schnittstelle nach Namen:

```
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
```

Das Verschieben ruckelt, sobald die Datenmenge wächst. Mit dem SVG-Renderer ist jede Linie und jedes Polygon ein eigenes Element im Dokument und jeder Standardmarker ein eigenes Bild. Eine Ebene mit vielen Tausend Geoobjekten bedeutet viele Tausend Elemente, die der Browser bei jedem Verschieben und Zoomen bewegen muss. `preferCanvas: true` zeichnet Pfade auf einer Canvas-Fläche, gilt aber nur für Pfade. Punkte müssen deshalb zu `L.circleMarker` werden, also zu Kreisen mit festem Pixelradius, die selbst Pfade

sind. Dafür dient die Option `pointToLayer` der GeoJSON-Ebene. Darüber hinaus lautet die Antwort, weniger zu senden: ein Clustering-Plugin oder eine API, die nur liefert, was zum aktuellen Ausschnitt und zur Zoomstufe passt.

Kacheln laden nach dem Start des Produktivbetriebs nicht mehr. Der Schnelleinstieg baut seine erste Karte auf dem Kachelserver von OpenStreetMap auf und erklärt ausdrücklich, dass diese Kacheln für die Programmierung geeignet sind, die Tile Usage Policy aber vor dem Produktionsbetrieb gelesen werden muss. Die Richtlinie verlangt sichtbare Quellenangaben, einen User-Agent, der die Anwendung identifiziert, und von Webseiten einen gültigen Referer, außerdem Caching gemäß den HTTP-Headern sowie den Verzicht auf Massendownloads und Vorabrufe. Sie bietet kein Service-Level-Agreement und kann den Zugriff ohne Ankündigung sperren. Eine stark genutzte Produktionskarte gehört zu einem Anbieter, dessen Bedingungen zu ihrem Datenverkehr passen, oder auf einen eigenen Kachelserver. Bei Leaflet besteht dieser Wechsel aus der URL-Vorlage und dem Text der Quellenangabe.

Weiterführende Lektüre

- Leaflet — die Übersicht des Projekts, seine Funktionsliste und eine erste Karte in wenigen Zeilen.
- Leaflet-API-Referenz — jede Klasse, Option und Methode in 1.9.4, einschließlich `preferCanvas`, `invalidateSize` und der GeoJSON-Optionen.
- Leaflet-Schnelleinstieg — CSS, Containerhöhe und eine erste Kachelebene, Schritt für Schritt.
- Leaflet 2.0 Alpha veröffentlicht — was sich in 2.0 ändert und wie Code und Plugins migriert werden.
- Tile Usage Policy — die Bedingungen für die Nutzung der Kachelserver von openstreetmap.org.
- RFC 7946: Das GeoJSON-Format — der GeoJSON-Standard, einschließlich der Reihenfolge mit dem Längengrad zuerst.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist Leaflet?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/leaflet/