Engineer

Leaflet

ett litet bibliotek för kartor i en webbläsare

Vad det är

Leaflet är ett JavaScript-bibliotek med öppen källkod som gör ett element på en webbsida till en karta som går att panorera, zooma och klicka i. Projektet anger storleken till ungefär 42 KB JavaScript och listar inga externa beroenden. Det skapades av Volodymyr Agafonkin, utvecklas i dag av en gemenskap av bidragsgivare och ges ut under licensen BSD 2-Clause.

Projektets egen översikt säger att det inte försöker göra allt för alla, utan koncentrerar sig på att få grunderna att fungera väl. `L.map` tar ett element och en uppsättning alternativ och blir kartan. Ett lager med kartplattor (tiles) hämtar kvadratiska bilder från en URL-mall med `{z}` , `{x}` och `{y}` i sig och lägger ut dem som ett rutnät av bildelement, och en WMS-variant gör samma sak mot en Web Map Service. Linjer, polygoner, cirklar och rektanglar ritas som SVG som standard, eller på en enda canvas när kartan skapas med `preferCanvas` . Markörer är ikoner, antingen en bild eller en enkel `div` , med popup-rutor och verktygstips kopplade till sig. Ett GeoJSON-lager gör varje objekt till ett av de lagren, en punkt till en markör om inget annat anges. Zoomknappar, källhänvisning, en lagerväljare och en skalstock utgör kontrollerna. Standardsystemet för koordinater är EPSG:3857, som referensdokumentationen kallar det vanligaste för kartor på nätet.

Allt utöver den listan kommer från tillägg. Projektets databas över tillägg beskriver Leaflet som tänkt att vara så lätt som möjligt och pekar på hundratals tillägg från tredje part, sorterade i grupper som vektorplattor, klustring, värmekartor, redigering av geometri, ruttplanering och geokodning.

Två utgivningslinjer finns när detta skrivs. Den stabila är 1.9.4, utgiven den 18 maj 2023, och API-referensen på projektets webbplats motsvarar den. Förhandsversionen, 2.0.0-alpha.1 från den 16 augusti 2025, tar bort stödet för

Internet Explorer och levereras som en ES-modul. Den behåller det globala `L` bara i en paketerad `leaflet-global.js`, ersätter fabriksfunktioner med konstruktorer, så att `L.marker(latlng)` blir `new Marker(latlng)`, och tar emot inmatning via pekarhändelser (pointer events).

Använd begreppet i ett beslut

Använd ”Vad är Leaflet?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Allt som händer på en plats — fartyg, fordon, tomter, sensorer — på en karta du kan söka i, mäta på och dela.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Gjorde pipelinen för geodata 50 gånger snabbare — 50x snabbare datapipelines för geodata

Gjorde pipelinen för geodata 50 gånger snabbare Hör av er →

Varför det spelar roll

Att sätta en karta på en sida är två beslut som är lätta att ta för ett: vilken kod som ritar kartan i webbläsaren, och varifrån kartbilderna kommer. Leaflet håller isär dem, och valet att använda det är till stor del ett val att hålla isär dem.

Leaflet ritar. Kartplattorna kommer från dit URL-mallen pekar, så bakgrundskartan är ett separat beslut med eget pris, egna villkor och egen ägare. Den kan vara OpenStreetMaps server för kartplattor medan en funktion byggs, en kommersiell leverantör i produktion, en server som projektet driver själv eller en kommuns WMS via `TileLayer.WMS`. Att byta mellan dem ändrar en sträng och en rad med källhänvisning, och applikationskoden runt kartan förblir som den är.

Storleken och licensen avgör var Leaflet passar bra. Ungefär 42 KB och inga beroenden passar en sida där kartan är en del bland många: ett formulär där någon sätter ut en nål, eller en administrationsvy som visar var poster finns. Licensen BSD 2-Clause lägger inget konto, ingen nyckel och ingen mätning av användningen mellan biblioteket och sidan.

Priset för en liten kärna är att tyngre funktioner kommer som separata paket, vart och ett med egen underhållare och egen utgivningstakt, och en karta som lutar sig mot sex tillägg ärver sex uppgraderingsvägar. Alfan av 2.0 visar vad det betyder: dess migreringsanvisningar hänvisar den som använder äldre tillägg till en polyfill för V1, eftersom de tilläggen förväntar sig det globala `L` och

fabriksfunktionerna. En karta som behöver vektorplattor som stilsätts i webbläsaren på varje zoomnivå passar bättre i ett bibliotek som är byggt kring dem, och det beslutet fattas helst innan sex tillägg håller uppe kartan.

Så ser det ut i praktiken

Leaflet förekommer på två sidor på den här webbplatsen, båda från rollen som teamledare för utveckling av GIS-mjukvara mellan 2015 och 2022, och på båda delar det frontenden med Mapbox.

Rollen hade ansvar för mer än trettio GIS-projekt genom utveckling och driftsättning. Det projekten visade utåt var webbkartor byggda på Leaflet och Mapbox GL, ibland kopplade mot ArcGIS- eller HERE-API:erna, och koden runt kartorna skrevs i JavaScript, Python, PHP och SQL. Under låg PostgreSQL med PostGIS, och GDAL/OGR flyttade data mellan shapefiler, GeoJSON, GeoTIFF, KML och vektorplattor. Ansvaret sträckte sig längre än till leveransen, ända in i supporten i produktion.

Samma roll fanns också med i designprocessen för projektens användargränssnitt. Det var frontends i React, Angular och Vue ovanpå kartor i Mapbox och Leaflet, och sidan gör en poäng som är värd att ha bredvid vilket kartbibliotek som helst: i en webbkartprodukt är kartan den enkla delen, och det är i kontrollerna, filtreringen och flödet runt den som folk går vilse. Användbarheten satt i detaljer som hur en datamängd filtrerades och hur man rörde sig mellan våningsplan på en inomhuskarta, och sidan tillskriver det mesta av skillnaden att de frågorna ställdes under designen och inte efter lanseringen.

Var det går fel

Leaflet är så litet att de flesta av dess fel kommer från sidan runt det, eller från de tjänster det pekas mot.

Kartan visas som en grå ruta eller en remsa av lösa kartplattor. Leaflet tar sin storlek från sin behållare. Snabbstartsguiden listar två saker att göra innan någon kartkod körs: ladda Leaflets CSS-fil före JavaScript-koden, och ge behållaren en bestämd höjd. En `div` utan höjd ger en karta utan höjd. Ett lurigare fall är en karta som skapas inuti en flik eller en dialog som är dold just då: den mäter sig själv till noll och behåller den storleken när den visas. Leaflet 1.x följer när webbläsarfönstret ändrar storlek, via `trackResize`, så en behållare som ändrar

storlek av sig själv passerar obemärkt tills koden anropar `invalidateSize()`, som referensen säger ska anropas när kartans storlek har ändrats. Alfan av 2.0 använder `ResizeObserver` i stället för fönstrets resize-händelse.

Punkter hamnar tusentals kilometer bort. GeoJSON, så som RFC 7946 definierar det, skriver en position med longituden först. Leaflets `LatLng` och de arrayer som det tar emot i stället har latituden först. `L.geoJSON` översätter mellan de två på egen hand, så buggen dyker upp när koordinater plockas ut ur GeoJSON för hand och skickas direkt till en markör eller en linje. En punkt i Stockholm hamnar, omkastad, i Arabiska havet. Packa upp paret med namn vid gränsen:

```
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
```

Panoreringen hackar när datan växer. Med SVG-renderaren är varje linje eller polygon ett eget element i dokumentet, och varje standardmarkör en egen bild. Ett lager med många tusen objekt betyder många tusen element som webbläsaren måste flytta vid varje panorering och zoomning. Alternativet `preferCanvas: true` ritar banor på en enda canvas, men gäller bara banor, så punkter måste bli `L.circleMarker`, en cirkel med fast radie i pixlar som själv är en bana, via GeoJSON-lagrets alternativ `pointToLayer`. Bortom den gränsen är svaret att skicka mindre: ett tillägg för klustring, eller ett API som bara returnerar det som får plats i den aktuella vyn och zoomnivån.

Kartplattorna slutar laddas efter lanseringen. Snabbstartsguiden bygger sin första karta på OpenStreetMaps server för kartplattor och säger rakt ut att de plattorna går bra under programmering, men att användningsvillkoren, Tile Usage Policy, ska läsas före produktion. Villkoren ber om synlig källhänvisning, en User-Agent som identifierar applikationen och en giltig Referer från webbsidor, cachning enligt HTTP-huvudena och att ingen massnedladdning eller förhämtning görs. De erbjuder inget servicenivåavtal, och åtkomsten kan spärras utan förvarning. En välbesökt karta i produktion hör hemma hos en leverantör vars villkor passar trafiken, eller på en egen server för kartplattor, och med Leaflet består flytten av URL-mallen och källhänvisningen.

Läs mer

- Leaflet — projektets översikt, funktionslista och en första karta på några få rader.
- Leaflet API reference — varje klass, alternativ och metod i 1.9.4, bl.a. `preferCanvas` , `invalidateSize` och alternativen för GeoJSON.
- Leaflet Quick Start Guide — CSS-filen, behållarens höjd och ett första lager med kartplattor, steg för steg.
- Leaflet 2.0 Alpha released — vad som ändras i 2.0 och hur kod och tillägg migreras.
- Tile Usage Policy — villkoren för att använda kartplattorna från openstreetmap.org.
- RFC 7946: The GeoJSON Format — GeoJSON-standarden, inklusive ordningen med longituden först i positioner.

Ordlista

Välj utifrån era faktiska förutsättningar

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är Leaflet?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/leaflet/