Engineer

Leaflet

et lite bibliotek for kart i en nettleser

Hva det er

Leaflet er et JavaScript-bibliotek med åpen kildekode som gjør ett element på en nettside om til et kart en person kan panorere, zoome og klikke i. Prosjektet oppgir størrelsen til rundt 42 KB JavaScript og lister ingen eksterne avhengigheter. Det ble laget av Volodymyr Agafonkin, utvikles nå av et fellesskap av bidragsytere og er utgitt under lisensen BSD 2-Clause.

Prosjektets egen oversikt sier at det ikke prøver å gjøre alt for alle, og at det konsentrerer seg om å få det grunnleggende til å fungere godt. `L.map` tar et element og et sett med innstillinger og blir selve kartet. Et fliselag henter kvadratiske bilder fra en URL-mal med `{z}`, `{x}` og `{y}` i seg og legger dem ut som et rutenett av bildeelementer, og en WMS-variant gjør det samme mot en Web Map Service. Polylinjer, polygoner, sirkler og rektangler tegnes som SVG som standard, eller på ett enkelt canvas når kartet opprettes med `preferCanvas`. Markører er ikoner, enten et bilde eller en vanlig `div`, med popuper og verktøytips knyttet til seg. Et GeoJSON-lag gjør hver feature om til et av disse lagene, et punkt til en markør med mindre noe annet er angitt. Zoomknapper, kildehenvisning, en lagvelger og en målestokk utgjør kontrollene. Standard koordinatreferansesystem er EPSG:3857, som referansen kaller det vanligste for kart på nett.

Alt utover den listen kommer fra programtillegg, plugins. Prosjektets database over programtillegg beskriver Leaflet som ment å være så lett som mulig, og viser til hundrevis av programtillegg fra tredjeparter, sortert i grupper som vektorfliser, klynging, varmekart, redigering av geometri, ruteplanlegging og geokoding.

To utgivelseslinjer finnes i skrivende stund. Den stabile er 1.9.4, utgitt 18. mai 2023, og API-referansen på prosjektets nettsted gjenspeiler den. Forhåndsutgivelsen, 2.0.0-alpha.1 fra 16. august 2025, dropper støtten for Internet Explorer og leveres som en ES-modul. Den beholder den globale `L` bare i en samlet `leaflet-global.js`, erstatter fabrikkfunksjoner med konstruktører, slik at `L.marker(latlng)` blir `new Marker(latlng)`, og tar imot inndata gjennom pointer-hendelser.

Bruk begrepet i en beslutning

Bruk «Hva er Leaflet?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Alt som skjer på et sted — fartøy, kjøretøy, tomter, sensorer — lagt på et kart du kan søke i, måle på og dele.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Gjorde en geodatapipeline 50 ganger raskere — 50x raskere pipelines for geografiske data

Gjorde en geodatapipeline 50 ganger raskere Ta kontakt →

Hvorfor det er viktig

Å sette et kart på en side er to avgjørelser som er lette å forveksle med én: hvilken kode som tegner kartet i nettleseren, og hvor kartbildene kommer fra. Leaflet holder dem adskilt, og valget om å bruke det er for det meste et valg om å holde dem adskilt.

Leaflet tegner. Flisene kommer fra der URL-malen peker, så bakgrunnskartet er en egen avgjørelse med sin egen pris, sine egne vilkår og sin egen eier. Det kan være flisserveren til OpenStreetMap mens en funksjon bygges, en kommersiell leverandør i produksjon, en flisserver prosjektet driver selv, eller en kommunes WMS gjennom `TileLayer.WMS`. Å flytte mellom dem endrer en streng og en linje med kildehenvisning, og applikasjonskoden rundt kartet blir som den er.

Størrelsen og lisensen avgjør hvor Leaflet passer uten å skave. Rundt 42 KB uten avhengigheter passer en side der kartet er én del blant mange: et skjema der noen setter en nål, eller en administrasjonsskjerm som viser hvor postene befinner seg. BSD 2-Clause-lisensen setter ingen konto, nøkkel eller forbruksmåling mellom biblioteket og siden.

Prisen for en liten kjerne er at tyngre funksjoner kommer som egne pakker, hver med sin egen vedlikeholder og sitt eget utgivelsestempo, og et kart som lener seg på seks programtillegg, arver seks oppgraderingsløp. Alfaversjonen av 2.0

viser hva det betyr: migreringsnotatene peker brukere av eldre programtillegg til en V1-polyfill, fordi de programtilleggene forventer den globale `L` og fabrikkfunksjonene. Et kart som trenger vektorfliser stilsatt i nettleseren på hvert zoomnivå, passer bedre i et bibliotek som er bygd rundt dem, en avgjørelse som helst tas før seks programtillegg holder kartet oppe.

Slik ser det ut i praksis

Leaflet dukker opp på to sider av dette nettstedet, begge fra rollen som teamleder for utvikling av GIS-programvare mellom 2015 og 2022, og på begge deler det frontenden med Mapbox.

Rollen bar leveransen av over tretti GIS-prosjekter, med driftsstøtte også etter produksjonssettingen. Det brukerne så av prosjektene, var webkart i nettleseren, bygd på Mapbox GL og Leaflet, og i noen av prosjektene gikk kartet mot API-ene til ArcGIS eller HERE. Bak kartene sto et applikasjonslag i JavaScript, Python, PHP og SQL, de romlige dataene lå i PostGIS, og GDAL/OGR flyttet dem mellom formater som shapefiler, GeoJSON, GeoTIFF, KML og vektorfliser. Ansvaret strakk seg forbi leveransen og inn i driftsstøtten i produksjon.

Den samme rollen satt også i designprosessen for prosjektenes brukergrensesnitt. Dette var frontender i React, Angular og Vue oppå kart i Mapbox og Leaflet, og siden poengterer noe som er verdt å ha ved siden av ethvert kartbibliotek: i et webkartprodukt er kartet den enkle delen, og det er kontrollene, filtreringen og flyten rundt det som får folk til å gå seg vill. Brukervennligheten lå i detaljer som hvordan et datasett ble filtrert, og hvordan noen flyttet seg mellom etasjer på et innendørskart, og siden tilskriver det meste av forskjellen at de spørsmålene ble stilt under designarbeidet og ikke etter lansering.

Hvor det går galt

Leaflet er lite nok til at de fleste feilene kommer fra siden rundt det, eller fra tjenestene det peker mot.

Kartet vises som en grå boks eller en stripe med løse fliser. Leaflet henter størrelsen sin fra beholderen. Hurtigstarten lister to ting som må gjøres før noen kartkode kjører: last inn CSS-filen til Leaflet før JavaScript-koden, og gi beholderen en bestemt høyde. En `div` uten høyde gir et kart uten høyde. Et mer subtilt tilfelle er et kart som opprettes i en fane eller dialog som er skjult i

øyeblikket: det måler seg selv til null og beholder den størrelsen når det kommer til syne. Leaflet 1.x følger med når nettleservinduet endrer størrelse, gjennom `trackResize`, så en beholder som endrer størrelse av seg selv, går upåaktet hen til koden kaller `invalidateSize()`, som referansen sier skal kalles etter at kartets størrelse er endret. Alfaversjonen av 2.0 bruker `ResizeObserver` i stedet for vinduets resize-hendelse.

Punkter havner tusenvis av kilometer unna. GeoJSON, slik RFC 7946 definerer det, skriver en posisjon med lengdegraden først. `LatLng` i Leaflet og matrisene det godtar i stedet, har breddegraden først. `L.geoJSON` gjør om mellom de to av seg selv, så feilen oppstår når koordinater hentes ut av GeoJSON for hånd og sendes rett til en markør eller en polylinje. Et punkt i København havner, med koordinatene byttet om, i Arabiahavet. Pakk ut paret etter navn ved grensen:

```
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 hakker når dataene vokser. Med SVG-tegneren er hver linje eller polygon sitt eget element i dokumentet, og hver standardmarkør er sitt eget bilde. Et lag med mange tusen features betyr mange tusen elementer nettleseren må flytte ved hver panorering og zooming. `preferCanvas: true` tegner baner på ett canvas, men det gjelder bare baner, så punkter må bli `L.circleMarker`, en sirkel med fast radius i piksler som selv er en bane, gjennom innstillingen `pointToLayer` på GeoJSON-laget. Forbi det punktet er svaret å sende mindre: et programtillegg for klynging, eller et API som bare returnerer det som passer det gjeldende utsnittet og zoomnivået.

Flisene slutter å laste etter lansering. Hurtigstarten bygger det første kartet sitt på flisserveren til OpenStreetMap og sier rett ut at de flisene er greie for programmering, men at retningslinjene for bruk av fliser, Tile Usage Policy, skal leses før produksjon. Retningslinjene krever synlig kildehenvisning, en User-Agent som identifiserer applikasjonen og en gyldig Referer fra nettsider, caching i tråd med HTTP-headerne, og ingen massenedlasting eller forhåndshenting. De tilbyr ingen tjenestenivåavtale og kan blokkere tilgangen uten varsel. Et travelt kart i produksjon hører hjemme hos en leverandør med vilkår som passer trafikken, eller på en egen flisserver, og med Leaflet er den flyttingen URL-malen og teksten i kildehenvisningen.

Les mer

- Leaflet — prosjektets oversikt, funksjonsliste og et første kart på noen få linjer.
- Leaflet API reference — hver klasse, innstilling og metode i 1.9.4, også `preferCanvas`, `invalidateSize` og innstillingene for GeoJSON.
- Leaflet Quick Start Guide — CSS-filen, høyden på beholderen og et første fliselag, steg for steg.
- Leaflet 2.0 Alpha released — hva som endres i 2.0, og hvordan kode og programtillegg migreres.
- Tile Usage Policy — vilkårene for å bruke flisserverne på openstreetmap.org.
- RFC 7946: The GeoJSON Format — GeoJSON-standarden, også rekkefølgen med lengdegraden først i posisjoner.

Ordliste

Velg ut fra deres faktiske rammer

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er Leaflet?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/leaflet/