Engineer

Mapbox

🗺 en kommersiell kilde til kartfliser og kartstiler

Hva det er

Mapbox er både et selskap og utviklerplattformen det selger: delene i et interaktivt kart levert som tjenester, altså grunndataene, utseendet, koden som tegner kartet, og API-ene rundt. I kode dukker det ofte opp som Mapbox GL, etter tegnebiblioteket. Selskapets egen kom-i-gang-veiledning kaller det en lokasjonsplattform som gir utviklere byggeklosser for kart, navigasjon og søk.

Kartdataene ligger i tilesets, som regel delt opp i vektorfliser, slik at en klient bare henter det området og det zoomnivået som vises på skjermen. Mapbox vedlikeholder globale datasett med grunndata, og et teams egne data kommer inn i et tileset via Mapbox Tiling Service (MTS) eller Uploads API. Hvordan kartet ser ut, bestemmer en stil. Det er et JSON-dokument skrevet etter Mapbox Style Specification som navngir kildene sine og lister opp lag; hvert lag plukker ut objekter fra én kilde og sier hvordan de skal tegnes. Stiler adresseres med URL, f.eks. `mapbox://styles/mapbox/streets-v12`, og lages i kontoens konsoll med designverktøyet Mapbox Studio.

Mapbox GL JS tegner resultatet i en nettleser, og Maps SDK-ene for iOS og Android gjør det samme i mobilapper. GL JS-veiledningen, som sto på versjon 3.33.0 da dette ble skrevet, sier at biblioteket gjengir 2D- og 3D-kart med OpenGL ved å kombinere vektorfliser med stilregler i nettleseren og ikke på en server. Det er derfor et kart kan bytte stil eller data mens noen bruker det. Static Images API returnerer et bilde av et kart, og Static Tiles API leverer rasterfliser laget fra Studio-stiler til biblioteker som Leaflet. Hvert kart opprettes med et tilgangstoken, og tokenet knytter kartet til en Mapbox-konto.

Selve flisformatet, Mapbox Vector Tile-spesifikasjonen, er publisert åpent på GitHub og koder hver flis som en Protocol Buffers-melding.

Bruk begrepet i en beslutning

Bruk «Hva er Mapbox?» 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

Et webkart trenger kode som tegner det, og data å tegne, enten fra én leverandør eller fra flere. Mapbox er svaret med én leverandør: data, stil, rendering og hosting under én konto og på én regning. Å velge Mapbox er dermed å bestemme hvor mye av kartstakken dere vil leie.

Gevinstene er reelle. Grunndataene er det en annen som må holde oppdatert. Kartets utseende bor i et stildokument med sitt eget designverktøy, adskilt fra applikasjonskoden. Fordi tegningen skjer i nettleseren, kan ett tileset vises i flere stiler, et lag kan filtreres eller fargelegges på nytt ved et klikk, og det samme kartet kan vise terreng i 3D. For et produkt der hovedskjermen er kartet, er det mye arbeid som blir liggende hos leverandøren.

Kostnadene følger med den samme kontoen. GL JS faktureres per kartinnlasting (map load), og prisguiden teller én hver gang et `Map` -objekt initialiseres på en side. Panorering, zooming, av- og påslag av lag og bytte av stil inne i det kartet legger ingenting til, og én kartinnlasting varer i høyst 12 timer. Tokenet som kartet faktureres mot, ligger i nettleseren hos hver eneste besøkende. Og siden versjon 2.0.0, utgitt i desember 2020, har selve renderen vært underlagt kommersielle vilkår: endringsloggen for den utgivelsen sier at GL JS ikke lenger er under 3-Clause BSD-lisensen, og at en oppgradering betyr å godta Mapbox' tjenestevilkår.

Flyttbarhet bør derfor avklares tidlig. Flisformatet er åpent. MapLibre GL JS begynte som en fork med åpen kildekode av mapbox-gl-js, tatt før lisensendringen. 1.x-linjen var ment som en direkte erstatning for åpen kildekode-utgaven av Mapbox 1.x, og README-en sier at prosjektet har utviklet seg mye siden. Det går an å forlate Mapbox, men det er en migrering, og den er billigere å vurdere før et produkt bygges rundt én leverandørs stiler og tilesets.

Slik ser det ut i praksis

Som teamleder for GIS-programvareutvikling fra 2022 til 2024 var oppgaven å styre fullstack-utviklingen av et GIS-kart som hadde vokst frem bit for bit: romlige data i PostgreSQL, ReactJS i frontenden, NodeJS bak og Mapbox til å tegne kartet. Siden oppsummerer Mapbox-delen i én linje: Mapbox tok seg av tegningen, og jobben var å mate det med de riktige dataene på de riktige zoomnivåene i stedet for alt på én gang. Resultatet ble én integrert kartapplikasjon som ble en kjernedel av plattformen og tok imot nye lag uten å knekke sammen.

Med samme tittel, fra 2015 til 2022, satt rollen i designprosessen for prosjektenes brukergrensesnitt, på React-, Angular- og Vue-frontender oppå Mapbox- og Leaflet-kart. Poenget på siden er at kartet er den enkle delen: folk går seg vill i kontrollene, filtreringen og flyten rundt det, i detaljer som det å bytte etasje i et innendørskart.

Som dataingeniør fra 2022 til 2024 var jobben å løse over 1 000 problemer i geografiske data og tidsseriedata med GDAL, QGIS, ArcGIS, PostGIS og Bash. De rensede dataene gikk deretter inn i interaktive visualiseringer bygget med Mapbox, noe som gjorde dem lettere å nå for sluttbrukerne. Siden krediterer arbeidet som helhet med 35 % kortere kjøretid for romlige spørringer.

Hvor det går galt

De fleste problemene med Mapbox begynner når et kart møter ekte trafikk, ekte data eller en større oppgradering.

Regningen vokser med sidevisningene, ikke med bruken av kartet. En kartinnlasting telles ved initialisering, så kostnaden følger hvor ofte en side oppretter et kart. En enkeltsideapp (SPA) som lager `Map` på nytt ved hvert rutebytte, betaler hver gang, og det gjør også et kart langt nede på en lang side som de fleste besøkende aldri ruller ned til. Mapbox' veiledning om å holde kostnadene for webkart nede foreslår å opprette kartet først når beholderen ruller inn i visningsområdet, å legge det bak en Load Map-knapp og å bruke Static Images API der ingen trenger å panorere eller zoome. Én kartinstans som holdes i live på tvers av ruter, løser tilfellet med enkeltsideappen.

Et token som kopieres ut av siden, blir brukt andre steder. Token-veiledningen slår fast at ethvert token på en offentlig nettside eller i kildekode er synlig for alle som leter etter det. Tokens med hemmelige scopes, `sk` -typen, hører hjemme på en server. Et offentlig `pk` -token på en side bør være laget for akkurat den applikasjonen, med URL-begrensninger, og ikke være kontoens standard offentlige token, som mangler slike begrensninger. Veiledningen kaller URL-begrensninger et tiltak etter beste evne som er avhengig av Referer-headeren nettleseren sender. Etter en lekkasje roterer du: lag et nytt token, bytt det inn i koden, produksjonssett på nytt, og slett deretter det gamle.

Kartet blir tregere etter hvert som datasettet vokser. Klienten gjør en GeoJSON-kilde om til vektorfliser underveis, så en stor fil betyr at nettleseren til hver besøkende gjør flisingen. Veiledningen om store GeoJSON-kilder lister opp rådene: last filen fra en URL i stedet for fra et JavaScript-objekt, senk kildens `maxzoom` under standardverdien 18 (12 passer for de fleste punktdata), hev lagets `minzoom` over 0, samle tette punkter i klynger, og fjern ubrukte egenskaper og desimaler etter den sjette. Det siste steget krymper eksempelfilen fra 19,3 MB til 3,3 MB. Over 500 000 punkter foreslår veiledningen å dele opp kilden eller å lage flisene på serveren med MTS eller Uploads API.

```
map.addSource("sites", {
    type: "geojson",
    data: "/data/sites.geojson", // en URL, ikke et objekt i minnet
    maxzoom: 12, // standardverdien er 18
    cluster: true, // samle tette punkter ved lav zoom
});
map.addLayer({
    id: "sites",
    type: "circle",
    source: "sites",
    minzoom: 6, // skjult under zoom 6
});
```

En større oppgradering endrer både lisensen og regningen. Overgangen fra GL JS 1.x til 2.0.0 flyttet biblioteket fra BSD-lisensen til Mapbox' tjenestevilkår og endret regelen for fakturering. Under 1.x krevde en kartinnlasting både en initialisering og en forespørsel etter en flis som Mapbox hostet; fra 2.0.0 teller initialiseringen alene, så et kart som bare tegnet egenhostede fliser, begynner å generere kartinnlastinger. Endringsloggen ber deg lese prisdokumentasjonen før du oppgraderer, og MapLibre GL JS er veien å gå for et prosjekt som vil bli værende på vilkår med åpen kildekode.

Les mer

- Getting Started — selskapets oversikt over produktene for kart, søk og navigasjon.
- Mapbox GL JS pricing — hva som teller som en kartinnlasting, økten på 12 timer og de eldre reglene.
- Token management — offentlige og hemmelige tokens, scopes, URL-begrensninger og rotasjon.
- Working with large GeoJSON sources in Mapbox GL JS — zoomgrenser, klynger, opprydding og fliser laget på serveren.
- Mapbox Vector Tile specification — det åpne formatet bak flisene.
- MapLibre GL JS — renderen med åpen kildekode som startet som en fork av mapbox-gl-js.

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 Mapbox?» 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/mapbox/