Engineer

Service worker

et skript som svarer på forespørsler når nettet ikke kan

Hva det er

En service worker er et skript et nettsted registrerer hos nettleseren, som kjører det atskilt fra alle sider og sender nettverksforespørslene fra sidene det styrer, gjennom det. Navnet brukes på engelsk også på norsk. Service workers er en type web worker: hver av dem kjører i sin egen tråd, uten dokument og uten vindu. Nettleseren starter en når det er noe for den å håndtere, og kan stoppe den så snart det er tomt for oppgaver, så alt den holder i en variabel, kan være borte ved neste hendelse.

En side installerer en med ett enkelt kall, og spesifikasjonen setter tre betingelser. Siden må være en sikker kontekst, som i praksis betyr HTTPS, med `localhost` og loopback-adressene tillatt under utvikling. Skriptet må komme fra sidens egen opprinnelse, så det kan ikke ligge på et CDN, selv om det kan importere kode derfra. Og registreringen gjelder et omfang: som standard mappen skriptet ligger i, og ikke noe videre med mindre serveren sender et `Service-Worker-Allowed` -hode sammen med skriptet.

```
// on the page
navigator.serviceWorker.register("/sw.js");

// in /sw.js
async function answer(request) {
    const copy = await caches.match(request);
    return copy ?? fetch(request);
}

self.addEventListener("fetch", (event) => {
    event.respondWith(answer(event.request));
});
```

Denne workeren svarer på hver forespørsel fra sin egen kopi når den har en, og fra nettet ellers. Å legge kopiene dit er livssyklusens jobb. Når skriptet er hentet, utløser nettleseren `install`, øyeblikket for å laste ned og lagre det workeren vil trenge. Avvises et løfte som er gitt til `event.waitUntil()` der, feiler installasjonen. Deretter kommer `activate`, øyeblikket for å rydde bort det en eldre versjon etterlot, og først etter det får workeren `fetch`-hendelser. En side kobles til workeren sin idet den lastes, så siden som registrerte den første workeren, fortsetter uten den til den lastes på nytt, med mindre workeren kaller `clients.claim()`.

Kopiene ligger i Cache Storage, som den samme spesifikasjonen definerer: navngitte cacher med par av forespørsel og svar, som workeren selv fyller, leser og sletter. De er atskilt fra nettlesercachen, nettleserens vanlige HTTP-cache. En oppføring utløper aldri, fornyes aldri av seg selv og overlever en oppdatering av workerskriptet, så den blir liggende til workeren sletter den eller nettleseren tømmer lagringen for nettstedet.

Bruk begrepet i en beslutning

Bruk «Hva er en service worker?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Grensesnitt som laster raskt, fungerer på mobilen og oppfører seg likt i alle nettlesere kundene dine måtte bruke.

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. Styrte fullstack-utviklingen av GIS-kart — fullstack-kartutvikling, fra database til nettleser

Styrte fullstack-utviklingen av GIS-kart Ta kontakt →

Hvorfor det er viktig

Service workere gir et nettsted eksplisitt kontroll over hvordan det fortsetter å virke helt uten nett. En service worker avgjør forespørsel for forespørsel, i kode nettstedet selv har skrevet, om svaret skal komme fra nettet, fra de egne kopiene eller fra begge, og det er det som gjør en installert webapp brukbar på et skip eller i en kjeller. Push-meldinger når den også, selv når ingen side fra nettstedet er åpen.

De fleste workere koker ned til å velge en strategi for hver type forespørsel, og Workbox, et bibliotek for å skrive dem, navngir fem. Cache first svarer fra kopien og går til nettet bare når kopien mangler, og det passer for filer som aldri endrer

seg under samme navn. Network first spør nettet og faller tilbake på kopien, og det passer for sider. Stale-while-revalidate svarer fra kopien og fornyer den i bakgrunnen. Network only og cache only er de to ytterpunktene.

Det er også kode som står mellom hver side i omfanget og nettet, på leserens enhet, og den lever lenger enn besøket: en feil i den blir værende til et rettet skript erstatter den. Spesifikasjonen er bygd rundt den risikoen. Hver navigasjon får nettleseren til å se etter et nytt skript, og andre hendelser gjør det når det har gått et døgn siden forrige sjekk. Som standard revalideres hovedskriptet mot serveren i stedet for å hentes fra HTTP-cachen, så en lang `max-age` på workerfilen kan aldri holde en ødelagt versjon på plass. Importerte skript får samme behandling når det har gått et døgn, eller alltid når `updateViaCache` er `none`. Nettleseren sammenligner det hentede skriptet med det installerte byte for byte, og bare en forskjell starter en oppdatering. En omlasting med Shift hopper over workeren for den ene navigasjonen, og derfor kan en feil se rettet ut på en utviklers maskin.

Venteregelen er den andre halvdelen av den utformingen. En ny versjon installeres ved siden av den gamle og venter så, ubrukt, til ingen åpen side lenger er avhengig av den gamle, slik at to versjoner aldri deler på én side. Et kall til `skipWaiting()` lar den ta over med en gang, mot at ny kode kjører under sider den gamle lastet.

Datalagring er den siste kostnaden. Cache Storage teller mot plassen nettstedet har på leserens enhet, og etter Storage Standard er den plassen som standard best-effort: nettleseren kan tømme den når enheten får dårlig plass, med mindre nettstedet har bedt om varig lagring og fått det innvilget.

Slik ser det ut i praksis

Grunnen til å ha en er tydeligst der nettet er svakest. Progressive Web App-en bygd for sjøfolk og feltpersonell betjener folk som ofte er frakoblet eller sitter på en dårlig forbindelse. Service workeren settes opp gjennom programtillegget next-pwa, og cachingen under kjøring i Workbox bruker cache first for det som er likt fra forespørsel til forespørsel, fonter, bilder, lyd, video og ressurser fra CDN, sammen med HTTP-hoder for cache-control. Fordi en service worker bare registreres i en sikker kontekst, ble frakoblet bruk og installasjon testet over HTTPS på ekte enheter, og gjentatte innlastinger kommer fra cachen i stedet for over nettet.

Dette nettstedet gjør det motsatte valget og sier det. Siden om informasjonskapsler opplyser at nettstedet verken har en service worker, IndexedDB eller Cache Storage: ingenting fra det kjører i nettleserens bakgrunn før leseren åpner det eller etter at det lukkes. Sidene leverer ingen kjørbar JavaScript. Søkekoden er bevart, men slått av, og registrerer ingen worker.

Installasjonssiden trekker konsekvensen for alle som legger nettstedet til på en hjemskjerm: installasjonen gir nettstedet et eget vindu uten å opprette en administrert frakoblet kopi. Dette nettstedet leverer verken en service worker eller skriptstyrt lagring for frakoblet bruk, så den installerte appen trenger en tilkobling ved vanlig bruk.

En worker registreres fra et kjørbart skript, og nettstedets gjeldende Content-Security-Policy forbyr det med `script-src 'none'` på hver side. JSON-LD er data og kjøres ikke. En tidligere kommentar i hodemalen beskrev et bygg der 550 av 793 dokumenter oppga `'none'` og 243 oppga `'self'`, tilsvarende sidene som bar listefilteret den gangen. Tallene beskriver det tidligere bygget med skript, ikke dagens policy.

Hvor det går galt

Den gamle versjonen som svarer videre etter en produksjonssetting. Den nye workeren installeres og venter så lenge en åpen fane fortsatt bruker den gamle, så en leser som aldri lukker den fanen, fortsetter å kjøre den gamle koden. Kall `skipWaiting()` når den nye versjonen er trygg under sider den gamle lastet, eller vis et varsel om oppdatering og la siden sende en melding til den ventende workeren om å kalle det.

Sider som blir stående på en gammel versjon i ukevis. En cache first-strategi brukt på HTML svarer på hver navigasjon fra Cache Storage, der oppføringene aldri utløper og ikke bryr seg om cachehodene fra serveren. Behold cache first for filer som får nytt navn når innholdet endres, og bruk network first for sider.

Én adresse i installasjonslisten som avgjør hele oppdateringen. En av adressene i listen for installasjonssteget svarer med en feil, så `cache.addAll()` avvises, løftet som er gitt til `waitUntil()`, avvises med det, og den nye versjonen forkastes mens den gamle blir. Bygg den listen av filer som finnes, og hent valgfrie filer senere i stedet for under installasjonen.

Et omfang som stopper ved mappen skriptet ligger i. Et skript servert fra `/js/sw.js` har `/js/` som både standardomfang og videste mulige omfang, så sider utenfor den mappen lastes uten workeren. Server skriptet fra roten av nettstedet, eller send `Service-Worker-Allowed: /` sammen med det.

Lagring på enheten som vokser med hver utgivelse. Cachene overlever en oppdatering av workerskriptet, så hver versjon som åpner en cache under et nytt navn, etterlater den forrige. Legg et versjonsnummer i hvert cachenavn, og slett i `activate`-håndtereren hver cache den gjeldende versjonen ikke bruker.

Les mer

- Service Workers — registrering, omfang, oppdateringssjekken, livssyklusen og Cache Storage.
- Secure Contexts — hvilke opprinnelser som regnes som potensielt pålitelige, `localhost` og loopback-adressene medregnet.
- Fetch Standard — objektene `Request` og `Response` som en `fetch`-håndterer får og gir tilbake, og hvordan en forespørsel gis videre til en service worker.
- Storage Standard — lagringsbøttene Cache Storage ligger i, anslag over kvoten og hvordan et nettsted ber om varig lagring.
- Workbox: workbox-strategies — de fem cachingstrategiene og hvilken type forespørsel hver av dem passer for.

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 en service worker?» 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/service-worker/