Engineer

App

et program folk bruker for å få gjort en oppgave

Hva det er

En app er et program laget for at en person skal åpne det og arbeide direkte i det: det har et ikon som starter det, en egen skjerm eller et eget vindu, og en jobb brukeren kjenner igjen, som å sende en melding, sjekke en rutetabell eller levere en reiseregning. «App» er en forkortelse av applikasjon, og de to betyr det samme. Den lange formen lever videre i formell tekst og i sammensetninger som nettapplikasjon og skrivebordsapplikasjon, og utviklere sier ofte klient om en app som er den synlige enden av en tjeneste som kjører et annet sted.

En app er applikasjonsprogramvare, i motsetning til systemprogramvaren som holder selve maskinen i gang. NIST samler flere definisjoner i ordlisten sin og påpeker at application og software application ofte brukes om hverandre; én av definisjonene regner også lønns- og regnskapssystemer som applikasjoner. Et bibliotek ligger på den andre siden av streken: kode som andre programmer kaller, og som ingen person noen gang åpner.

Apper finnes i tre brede familier:

- **Native apper** skrives for én plattform i plattformens eget språk og verktøysett, f.eks. Swift og SwiftUI på en Mac, og installeres fra en appbutikk eller som en signert nedlasting.
- **Nettapper** kjører i en nettleser og nås via en lenke. Et web app-manifest, en liten JSON-fil som gir nettleseren appens navn, ikoner, start-URL og foretrukne visningsmodus, lar en nettapp installeres på en hjemskjerm eller i en dock og åpnes i et eget vindu. En service worker, et skript nettleseren kjører mellom siden og nettverket, er det som lar den fortsette å virke uten nett. De to sammen kalles som regel en progressiv nettapp, eller PWA.
- **Plattformuavhengige skall** pakker én kodebase inn for hvert operativsystem den skal kjøre på, ofte ved å legge en nettlesermotor inn i appen.

Bruk begrepet i en beslutning

Bruk «Hva er en app?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Et nettsted som er raskt, lett å finne og kan redigeres av dine egne ansatte, uten en regning hver gang en formulering endres.

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. Utviklet 100 webapplikasjoner med HTML/HTML5 — 100 webapplikasjoner bygd

Utviklet 100 webapplikasjoner med HTML/HTML5 Ta kontakt →

Hvorfor det er viktig

Spørsmålet bak de fleste app-prosjekter er hvilken familie som skal bygges, og det handler mest om brukere og distribusjon, ikke om kode.

En nettapp koster én kodebase og når alle som har en nettleser, via en lenke. En oppdatering går ut når serveren gjør det: neste innlasting henter den, og ingen gjennomgang i en appbutikk står mellom en rettelse og folkene som venter på den. Prisen er dybde. En nettleser gir tilgang til mindre av enheten enn operativsystemet gjør, og hver nettleser presenterer installasjon på sin egen måte. På iOS dukker installasjon aldri opp som et forslag, så et team som vil ha nettappen sin på hjemskjermene, må fortelle folk hvordan.

En native app får plattformens egne kontroller, tastaturoppførsel, støtte for universell utforming og ytelse, og tilgang til alt systemet tilbyr. Den koster én kodebase per plattform, folk som kan hver av dem, og en distribusjonsprosess: gjennomgang i appbutikken, eller utenfor en butikk kodesignering og, på macOS, notarisering, som Apple krever for programvare distribuert med en Developer ID siden macOS 10.15. Noen må også drifte oppdateringskanalen så lenge appen finnes.

Et plattformuavhengig skall ligger mellom de to: én kodebase og en app som kan installeres, betalt med et medfølgende kjøretidsmiljø og et grensesnitt som ligner plattformen uten helt å oppføre seg som den.

Det som oftest avgjør, er hvor brukerne befinner seg når de åpner appen. Den som sitter ved en pult med kablet nett, tilgir mye. Den som står med en telefon og svak dekning, tilgir svært lite, og for dem er oppførsel uten nett et krav og ikke finpuss. Det andre spørsmålet som avgjør, er hvem som skal levere oppdateringer i det tredje året, fordi den billigste appen å bygge kan bli den dyreste å holde i live.

Slik ser det ut i praksis

På én plattform er mange av brukerne sjøfolk og feltpersonell som jobber fra mobilen, ofte helt uten dekning. Svaret der var en nettapp som kan installeres: full støtte for Progressive Web App gir den ikoner på hjemskjermen i størrelsene plattformene ber om, en oppstartsskjerm og temafarger. Service workeren settes opp gjennom programtillegget next-pwa, cachingen under kjøring i Workbox serverer fonter, bilder, lyd, video og CDN-ressurser fra cachen først (cache-first), sammen med HTTP-hoder for cache-control, og installasjons- og offline-oppførselen ble testet over HTTPS på ekte enheter. Folk kan installere appen, fortsette å bruke den når nettet er borte, og få gjentatte besøk servert fra cachen.

En meldingsprotokoll med en skrivebordsklient for macOS som var et plattformuavhengig skall, gikk motsatt vei, med en native macOS-klient i Swift 6 og SwiftUI på 11 141 linjer fordelt på 53 filer, med protokollens Rust-kjerne lenket inn som et statisk arkiv bygd fra en fastlåst revisjon. Den starter som en Mac-app, bruker plattformens egne kontroller og har ingen innebygd nettleser med seg. Siden sier også hvor den stopper: dette er ikke en levert utgivelse, det finnes verken en notarisert versjon eller en oppdateringskanal, og grensesnittet snakker ett språk.

Dette nettstedet kan også installeres som en app, og installasjonssiden gjennomgår det nettleser for nettleser: ingen appbutikk, ingen konto, ingen separat app-pakke å laste ned, bare et ikon og et eget vindu. Safari på en iPhone tilbyr det under Del. Firefox på Windows kan fra versjon 143 legge til nettsteder som åpnes i egne nettappvinduer. Hvis Firefox er installert fra Microsoft Store, er funksjonen tilgjengelig fra versjon 150. Mozilla tilbyr ikke denne funksjonen på macOS eller Linux. Siden sier også hvor grensen går: bruk uten nett krever en service worker, nettstedet leverer ingen kjørbar JavaScript og registrerer ingen service worker, og derfor trenger den installerte appen en tilkobling akkurat som en nettleserfane.

Hvor det går galt

Skjermen blir hvit når dekningen forsvinner

Feltbrukere melder om et tomt vindu eller nettleserens egen side for manglende nett. Appen ble utformet og testet på den raske forbindelsen inne i bygget, og hver skjerm henter det den trenger idet den åpnes. Bestem hvilke skjermer som

må virke uten nett, legg dem og ressursene deres i cache med en service worker, og test på ekte enheter med nettet slått av, ikke i simuleringen i en nettleser på datamaskinen.

Brukerne installerer en nettapp og venter seg en frakoblet kopi

Noen legger nettstedet til på hjemskjermen, åpner det på et tog og møter den samme feilen en nettleserfane ville vist. Et manifest gjør et nettsted installerbart; oppførsel uten nett kommer bare fra en service worker, og de to er separate beslutninger. Lever enten workeren, eller si rett ut i installasjonsveiledningen at den installerte appen trenger en tilkobling.

En ny versjon går ut, og brukerne beholder den gamle

Etter en produksjonssetting kjører noen fortsatt forrige ukes kode i flere dager. Service workeren serverer appens sider fra cachen først uten at noe markerer en ny versjon, eller en ny worker blir stående og vente til hver fane den gamle styrer, er lukket. Behold cache-first for ressurser som ikke endrer seg fra forespørsel til forespørsel, som fonter, medier og filer med en versjon i navnet, hent dokumenter fra nettverket først, og vis en melding om å laste inn på nytt når en ny versjon er klar.

Skallet føles fremmed på skrivebordet

Mac-brukere klager over menyer, hurtigtaster og tekstmarkering, og appen bruker minne som en nettleser fordi den inneholder en. Et medfølgende kjøretidsmiljø etterligner plattformen i stedet for å bruke den. Der skrivebordsopplevelsen er selve produktet, bør du sammenligne en native klient med skallet, og holde den delte logikken i et kjernebibliotek som hver klient kaller gjennom et stabilt grensesnitt.

Distribusjonen blir skjøvet til slutt

Appen virker på hver utviklers maskin, og den første kunden på macOS møter en advarsel fra Gatekeeper i stedet for appen. En rettelse går ut, og halvparten av brukerne beholder den ødelagte versjonen. Signering, notarisering og oppdateringer ble behandlet som pakkearbeid for den siste uka. Bygg signering og notarisering inn i byggingen fra første utgivelse, og bestem hvordan oppdateringer når fram til folk før noen installerer versjon én.

Les mer

- NIST CSRC Glossary: application — definisjonene NIST-publikasjonene bruker, fra ethvert system bygd for å oppfylle et sett krav til lønns- og regnskapssystemer.
- W3C Web Application Manifest — JSON-filen som navngir en nettapp, lister opp ikonene og start-URL-en, og setter visningsmodusen den åpnes i.
- W3C Service Workers — skriptet som ligger mellom en nettapp og nettverket, med hendelsene for cache og fetch som gjør bruk uten nett mulig.
- Apple Developer: Notarizing macOS software before distribution — hva notartjenesten sjekker, og når programvare distribuert utenfor Mac App Store må gjennom den.

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 app?» 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/app/