Engineer

App

ett program som människor använder för att utföra en uppgift

Vad det är

En app är ett program som är gjort för att en människa ska öppna det och arbeta i det direkt: det har en ikon som startar det, en egen skärm eller ett eget fönster och en uppgift som användaren känner igen, som att skicka ett meddelande, kolla en tidtabell eller lämna in ett utlägg. ”App” är en förkortning av applikation, och orden betyder samma sak. Den långa formen lever kvar i formell text och i sammansättningar som webbapplikation och skrivbordsapplikation, och utvecklare säger ofta klient om en app som är den synliga änden av en tjänst som körs någon annanstans.

En app är applikationsmjukvara, till skillnad från den systemmjukvara som håller själva maskinen igång. NIST:s ordlista samlar flera definitioner och påpekar att application och software application ofta används synonymt, och en av definitionerna räknar även löne- och bokföringssystem som applikationer. Ett bibliotek står på andra sidan gränsen: kod som andra program anropar och som ingen människa någonsin öppnar.

Appar delas grovt in i tre familjer:

- **Nativa appar** skrivs för en plattform i plattformens eget språk och verktyg, t.ex. Swift och SwiftUI på en Mac, och installeras från en appbutik eller från en signerad nedladdning.
- **Webbappar** körs i en webbläsare och nås via en länk. Ett webbappsmanifest, en liten JSON-fil som ger webbläsaren appens namn, ikoner, start-URL och önskade visningsläge, gör att en sådan app kan installeras på en hemskärm eller i Dock och öppnas i ett eget fönster. En service worker, ett skript som webbläsaren kör mellan sidan och nätverket, är det som låter den fortsätta fungera offline. Tillsammans brukar paret kallas en progressive web app, eller PWA.

- **Plattformsoberoende skal** paketerar en och samma kodbas för flera operativsystem, ofta genom att bygga in en webbläsarmotor i appen.

Använd begreppet i ett beslut

Använd ”Vad är en app?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. En webbplats som är snabb, lätt att hitta och går att redigera av din egen personal, utan en faktura för varje ändrad formulering.

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. Utvecklade 100 webbapplikationer med HTML/HTML5 — 100 byggda webbapplikationer

Utvecklade 100 webbapplikationer med HTML/HTML5 Hör av er →

Varför det spelar roll

Frågan bakom de flesta projekt som ska ge användare en app är vilken familj som ska byggas, och den handlar mer om användare och distribution än om kod.

En webbapp kostar en kodbas och når alla som har en webbläsare via en länk. En uppdatering går ut när servern gör det: nästa omladdning hämtar den, och ingen granskning i en appbutik står mellan en rättning och de människor som väntar på den. Priset är djupet. En webbläsare exponerar mindre av enheten än operativsystemet gör, och varje webbläsare presenterar installationen på sitt eget sätt. På iOS föreslås installationen aldrig av systemet, så ett team som vill ha sin webbapp på hemskärmar måste själv tala om för människor hur de gör.

En nativ app får plattformens egna kontroller, tangentbordsbeteende, stöd för skärmläsare och andra hjälpmedel och plattformens prestanda, och tillgång till allt som systemet erbjuder. Den kostar en kodbas per plattform, människor som kan varje plattform och en distributionsprocess: granskning i appbutiken eller, utanför en butik, kodsignering och på macOS dessutom notarisering, som Apple sedan macOS 10.15 kräver för mjukvara som distribueras med Developer ID. Någon måste också sköta uppdateringskanalen så länge appen finns.

Ett plattformsoberoende skal hamnar mellan de två: en kodbas och en installerbar app, betalda med en medföljande körmiljö och ett gränssnitt som liknar plattformen utan att riktigt bete sig som den.

Det som oftast avgör är var användarna befinner sig när de öppnar appen. Den som sitter vid ett skrivbord med fast uppkoppling förlåter mycket. Den som har en telefon och svag täckning förlåter väldigt lite, och för den användaren är beteendet offline ett krav snarare än finputs. Den andra avgörande frågan är vem som ska leverera uppdateringar det tredje året, eftersom den app som är billigast att bygga kan bli den dyraste att hålla vid liv.

Så ser det ut i praktiken

På en plattform är många av användarna sjöfolk och fältpersonal som arbetar från mobilen, ofta offline eller på en dålig uppkoppling. Där blev svaret en webbapp som går att installera: fullt stöd för Progressive Web App ger den ikoner på hemskärmen i de storlekar som plattformarna begär, en startskärm och temafärger. Dess service worker sätts upp via pluginet next-pwa, och Workbox cachar vid körning med strategin CacheFirst för typsnitt, bilder, ljud, video och CDN-resurser, tillsammans med HTTP-huvuden för cachning. Beteendet vid installation och offline testades över HTTPS på riktiga enheter. Användarna kan installera appen, fortsätta använda den när nätet är borta och få upprepade laddningar serverade från cachen.

Ett meddelandeprotokoll vars skrivbordsklient för macOS var ett plattformsoberoende skal tog den motsatta vägen, med en nativ macOS-klient i Swift 6 och SwiftUI som omfattar 11 141 rader i 53 filer, med protokollets Rust-kärna inlänkad som ett statiskt arkiv byggt från en fastlåst revision. Den startar som en Mac-app, använder plattformens egna kontroller och bär inte på någon inbäddad webbläsare. Sidan säger också rakt ut var den slutar: det är ingen levererad release, eftersom både en notariserad distribution och en uppdateringskanal saknas, och gränssnittet talar ett enda språk.

Den här webbplatsen kan också installeras som en app, och installationssidan går igenom det webbläsare för webbläsare: ingen appbutik, inget konto, inget separat app-paket att ladda ned, bara en ikon och ett eget fönster. Safari på en iPhone erbjuder det under Dela. Firefox på Windows kan från version 143 lägga till webbplatser som öppnas i egna webbappfönster. Om Firefox har installerats från Microsoft Store är funktionen tillgänglig från version 150. Mozilla erbjuder inte den här funktionen på macOS eller Linux. Sidan anger också gränsen: offlinestöd kräver en service worker, webbplatsen levererar ingen körbar JavaScript och registrerar ingen service worker, och därför behöver den installerade appen en anslutning precis som en flik i webbläsaren.

Var det går fel

Skärmen blir vit när signalen försvinner

Fältanvändare rapporterar ett tomt fönster eller webbläsarens egen offlinesida. Appen designades och testades på kontorets egen snabba uppkoppling, och varje skärm hämtar det den behöver när den öppnas. Bestäm vilka skärmar som måste fungera offline, cacha dem och deras resurser med en service worker, och testa på riktiga enheter med nätet avstängt i stället för i en skrivbordswebbläsares simulering.

Användare installerar en webbapp och väntar sig en offlinekopia

Någon lägger till webbplatsen på hemskärmen, öppnar den på ett tåg och möter samma felmeddelande som en flik i webbläsaren skulle visa. Ett webbappsmanifest gör en webbplats installerbar, medan beteendet offline enbart kommer från en service worker, och de två är separata beslut. Leverera antingen workern, eller skriv tydligt i installationsanvisningarna att den installerade appen behöver en anslutning.

En release går ut och användarna behåller den gamla

Efter en driftsättning kör en del människor förra veckans kod i flera dagar. Service workern serverar appens sidor cache-first utan något som markerar en ny version, eller så väntar en ny worker tills varje flik som styrs av den gamla har stängts. Behåll cache-first för resurser som inte ändras per anrop, som typsnitt, media och filer med en version i namnet, hämta dokument från nätverket först och visa en uppmaning att ladda om när en ny version är klar.

Skalet känns främmande på skrivbordet

Mac-användare klagar på menyer, kortkommandon och textmarkering, och appen drar minne som en webbläsare eftersom den innehåller en. En medföljande körmiljö imiterar plattformen i stället för att använda den. Där skrivbordsupplevelsen är själva produkten bör en nativ klient jämföras med skalet, och den delade logiken bör ligga i ett kärnbibliotek som varje klient anropar genom ett stabilt gränssnitt.

Distributionen lämnas till slutet

Appen fungerar på varje utvecklares maskin, och den första kunden på macOS möter en varning från Gatekeeper i stället för appen. En rättning levereras, och hälften av användarna behåller den trasiga versionen. Signering, notarisering och

uppdateringar behandlades som paketeringssysslor för den sista veckan. Bygg in signering och notarisering från den första releasen, och bestäm hur uppdateringar når människor innan någon installerar version ett.

Läs mer

- NIST CSRC Glossary: application — de definitioner som NIST:s publikationer använder, från vilket system som helst som byggts för att uppfylla en uppsättning krav till löne- och bokföringssystem.
- W3C Web Application Manifest — JSON-filen som namnger en webbapp, listar dess ikoner och start-URL och anger det visningsläge den öppnas i.
- W3C Service Workers — skriptet som sitter mellan en webbapp och nätverket, med de cache- och fetch-händelser som gör användning offline möjlig.
- Apple Developer: Notarizing macOS software before distribution — vad notarietjänsten kontrollerar och när mjukvara som distribueras utanför Mac App Store måste klara den.

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 en app?” 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/app/