Engineer

Next.js

N et React-framework til at bygge webapps

Hvad det er

Next.js er et framework, der pakker React ind – biblioteket til at bygge brugerflader af komponenter – og leverer de dele, React overlader til hver app: routing, rendering på serveren, hentning af data og et build, der er sat op på forhånd. Pakken og dens kommando hedder begge `next`, som i `next dev` og `next build`.

Det udkom første gang i 2016, ifølge forfatterne for at give en nem måde at server-rendere React-applikationer på. Det tidlige design hvilede på nogle få principper: ingen opsætning, filsystemet som API, så en fil i en `pages/`-mappe blev til en route, automatisk server-rendering og kodeopdeling, og hentning af data overladt til udvikleren. I 2023 krediterede udgivelsesnoterne over 2.600 udviklere sammen med et kerneteam hos Vercel.

Next.js 13 tilføjede den 25. oktober 2022 en `app`-mappe i beta, bygget på React Server Components, med indlejrede layouts og streaming. Med Next.js 13.4 den 4. maj 2023 blev denne App Router erklæret stabil, og dokumentationen har beskrevet to routere lige siden. Pages Router omtales som den oprindelige router, der stadig understøttes og bliver forbedret. De kører ikke engang den samme React: App Router har canary-udgaver af React bygget ind, mens Pages Router bruger den version, projektets `package.json` angiver. I App Router renderes sider og layouts på serveren, medmindre en fil vælger browseren med direktivet `"use client"`.

Next.js 16 kom den 21. oktober 2025. Det gjorde Turbopack til standard-bundler, omdøbte filen `middleware.ts` til `proxy.ts` og indførte Cache Components, en model, hvor direktivet `"use cache"` markerer, hvad der må caches, og alt andet

renderes ved hver request. Next.js 16.4 fra den 6. oktober 2026 anbefaler Cache Components til alle apps, slår dem til i nye projekter lavet med `create-next-app` og siger, at de bliver standard i Next.js 17.

Brug begrebet i en beslutning

Brug »Hvad er Next.js?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Brugerflader, der indlæses hurtigt, virker på en telefon og opfører sig ens i alle de browsere, jeres kunder nu bruger.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har styret full-stack-udviklingen af GIS-kort — full stack-kortløsning, fra database til browser

Har styret full-stack-udviklingen af GIS-kort Kontakt os →

Hvorfor det betyder noget

Det valg, Next.js besvarer, er, om et framework skal eje routing, rendering og server i en React-applikation, eller om man selv samler delene og beholder valgene.

Det, frameworket tilbyder, er konvention i stedet for konfiguration. Routes kommer fra filer, bundlere og compilere sættes op automatisk, og hver route kan bygges på forhånd, renderes på serveren ved hver request eller renderes i browseren. Server Components holder dataadgang og hemmeligheder på serveren og sender mindre JavaScript til browseren. For en applikation, der blander offentlige sider med et område bag login og sin egen backend, er valget route for route den største attraktion.

Den første pris er en server, der kører Node.js 20.9 eller nyere til Next.js 16. Med `output: 'export'` skrives én HTML-fil pr. route til en hvilken som helst statisk webserver, og vejledningen præsenterer det som en måde at starte statisk og tilføje serverfunktioner senere. Dens liste over det, der kræver en Node.js-server, dækker dog meget af det, der adskiller Next.js fra andre: Proxy, Server Actions, cookies, rewrites, redirects, headers, Incremental Static Regeneration og billedoptimering med standard-loaderen.

Den anden pris er tempoet. Fire funktionsudgivelser fulgte efter 16.0 inden for et år, og cachemodellen er på vej til at skifte standard igen. Ifølge 16.4-noterne afløser Cache Components den implicitte caching i de tidligere versioner af App Router, så kode skrevet til Next.js 13 til 15 cacher efter andre regler. Vejledninger på nettet er også delt mellem de to routere.

Den tredje pris er, at opgraderinger er sikkerhedsarbejde. Den 3. december 2025 udsendte projektet en advarsel om CVE-2025-66478, en kritisk sårbarhed i protokollen for React Server Components, og bad brugerne opgradere til rettede versioner med det samme, og bloggen bragte tre sikkerhedsindlæg mere mellem den 30. september og den 14. oktober 2026. Et team, der vælger Next.js, forpligter sig til at følge udgivelserne tæt. Et team, der udgiver dokumenter, som ser ens ud for alle besøgende, betaler for en servermodel, det næsten ikke bruger.

Sådan ser det ud i praksis

Next.js 16-frontenden i grundlæggerens arbejde hos NextMariner vælger rendering for hver del, og det er valget route for route sat i arbejde. Marketingsider og offentlige sider genereres statisk, de reelt interaktive dele renderes fortsat på klienten, og på det autentificerede dashboard og de store oversigtslister henter en factory, `createPrefetchedServerPage`, sidens data på serveren og giver dem videre til klienten, som dermed allerede har sine data. Sider flyttes over på factoryen én ad gangen, og siden om arbejdet er ærlig om, at resten af den autentificerede app stadig renderes på klienten og venter på sin tur bag den: »en migrering med en fungerende mekanisme og en oplagt næste side, ikke en afsluttet gennemgang«. App Router er bygget til netop den slags flytning; udgivelsesnoterne til 13.4 beskriver, hvordan den tages i brug gradvist.

En lagdelt testsuite giver en Next.js-frontend sit eget lag: 543 Vitest- og Playwright-specs, heraf 48 end-to-end-filer og 30 browser-specs, ved siden af separate lag for Go-API'et og databasen. Hvert lag kontrolleres dér, hvor dets adfærd hører hjemme, i stedet for at alt kontrolleres udefra gennem en browser, og det betyder ekstra meget i et framework, hvor ét komponenttræ kører dels på serveren, dels i browseren.

Da grundlæggeren i sit arbejde hos NextMariner hærdede en applikation, kom de beskyttelser, der vender mod browseren, i `proxy.ts`, den omdøbte middleware. Den sætter en Content-Security-Policy med en ny nonce for hver request og `strict-dynamic`, plus HSTS og SameSite-cookies. Rettighederne ligger et andet sted: tier, rolle, organisation og afgrænsning til skibe kontrolleres igen på serveren ved hver request, og kontrollerne i frontenden betragtes kun som UX. Den deling svarer til referencen for Proxy, som kalder funktionen en sidste udvej og beder udviklere kontrollere autorisation inde i hver Server Function i stedet for kun at stole på Proxy.

Hvor det går galt

Efter en omdirigering til login er tilføjet, indlæses loginsiden uden styles. Uden en `matcher` kører Proxy på hver request, også for byggede filer under `_next/static`, optimerede billeder under `_next/image` og alt i `public/`, så en omdirigering, der var tænkt til sider, også fanger den CSS, det JavaScript og de billeder, loginsiden skal bruge. Giv Proxy en matcher, der springer de stier over:

```
// proxy.ts
export const config = {
    matcher: ["/((?!_next/static|_next/image|favicon.ico).*)"],
};
```

Filer fra `public/` har intet fælles præfiks, så tilføj også deres stier.

En formularhandling virker for en bruger, der er logget ud, på en route, der ser beskyttet ud. Server Functions deler route med den side, de bruges på: de kommer som POST-requests dertil, så en matcher, der springer en sti over, også springer funktionerne dér over. Referencen for Proxy advarer om, at en ændret matcher eller en refaktorering, der flytter en funktion til en anden route, kan fjerne Proxy-dækningen i stilhed. Kontrollér sessionen og brugerens rettigheder inde i hver Server Function, og behandl Proxy som et ydre lag og aldrig som det eneste.

En bruger gemmer en ændring, og siden viser stadig den gamle værdi. I Next.js 16 giver `revalidateTag(tag, 'max')` taggede poster stale-while-revalidate-adfærd: den cachede kopi bliver ved med at blive serveret, mens en frisk hentes, hvilket passer til indhold, der kan tåle, at data først bliver konsistente efter en forsinkelse, og overrasker den, der lige har trykket Gem. Kald `updateTag(tag)` fra den Server Action, der lavede ændringen; den lader posten udløbe og læser friske data i samme request. Den gamle form med ét argument, `revalidateTag(tag)`, er forældet og giver nu en TypeScript-fejl.

Opgraderingens codemod bliver færdig, og buildet fejler stadig på `params` eller `cookies()`. Next.js 15 gjorde request-API'erne asynkrone med en midlertidig synkron kompatibilitet, og Next.js 16 fjerner den. Codemoden `upgrade` kører ikke alle omskrivningerne, så kør også `npx @next/codemod@canary next-async-request-api .` og derefter `npx next typegen` for at generere

hjælpetyperne `PageProps`, `LayoutProps` og `RouteContext`. Samme opgradering fjerner `next lint` og får `next build` til at holde op med at linte, så tilføj ESLint eller Biome som et selvstændigt trin.

Læs mere

- Next.js Docs — den officielle dokumentation, med et skift mellem App Router- og Pages Router-udgaven af hver side.
- Next.js 16 — udgivelsen, der bragte Cache Components, `proxy.ts` og Turbopack som standard-bundler, med dens brud på bagudkompatibilitet.
- How to upgrade to version 16 — codemods, versionskrav og alle fjernede eller omdøbte API'er.
- proxy.js — matchere, hvor Proxy ligger i rækkefølgen for en request, og hvorfor middleware blev omdøbt.
- Next.js 16.4 — argumenterne for Cache Components og planen om at gøre dem til standard i Next.js 17.
- How to create a static export — hvilke funktioner der kræver en server, og hvad `output: 'export'` skriver.

Ordliste

Vælg ud fra jeres faktiske vilkår

Beskriv arbejdsgangen, de eksisterende grænseflader og de brugere, der har brug for ændringen. Vi kan definere en testbar leverance og en overdragelse, der gør teamet i stand til at vedligeholde løsningen.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er Next.js?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

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

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/nextjs/