Engineer

URL

adressen, der siger, hvor en ting er

Hvad det er

En URL, forkortelsen for Uniform Resource Locator og i daglig tale bare en webadresse eller et link, er en tekststreng, der både navngiver en ressource og fortæller software, hvordan den hentes: hvilken protokol der skal tales, hvilken maskine der skal spørges, og hvad maskinen skal spørges om. Sæt den ind i en browser, en terminal eller en konfigurationsfil, og alle tre kan handle på den uden yderligere hjælp.

Opbygningen ligger fast i den generiske syntaks, som ethvert URL-skema deler:

```
https://shop.example.com:8443/orders/42?view=full#items

skema         https
vært          shop.example.com
port          8443
sti           /orders/42
forespørgsel  view=full
fragment      items
```

Skemaet siger, hvordan tingen nås, her HTTP over TLS. Værten og den valgfri port siger hvor, stien siger hvilken ressource på værten, og forespørgselsdelen bærer parametre, som regel som par af `navn=værdi` adskilt af `&`. Fragmentet efter `#` peger ind i resultatet, på en overskrift eller en position. Det tilhører klienten: browseren udelader det fra HTTP-forespørgslens request-target. Det gør det ikke hemmeligt for serveren, for kode på klienten kan læse det og udtrykkeligt sende det i en anden forespørgsel.

Tegn som `/`, `?`, `#`, `&` og `=` har hver deres opgave i den struktur. Data, der ville kollidere med en afgrænser i den komponent, der bygges, skal procentkodes, altså skrives som `%` efterfulgt af to hexadecimale cifre. Hvilke tegn det gælder, afhænger af komponenten og dens syntaks. Et mellemrum bliver til `%20`, et og-tegn inde i en forespørgselsparameters værdi til `%26` og procenttegnet selv til `%25`. Bogstaver uden for ASCII rejser på samme måde som deres UTF-8-bytes, og det er sådan, æ, ø og å kommer igennem i en sti.

Er en URL det samme som en URI? Næsten. En URI, Uniform Resource Identifier, er enhver streng i den syntaks, der identificerer noget. En URL er den slags, der også siger, hvor tingen kan hentes, mens `urn:isbn:9780000000002` navngiver en bog uden at sige, hvor den findes. RFC 3986 beder nye specifikationer om at skrive URI. WHATWG's URL Standard, som browserne følger, gik den modsatte vej, valgte URL som det ene ord og beskriver, hvad virkelige implementeringer gør, med det mål at afløse RFC'en. De to er enige om almindelige adresser og uenige om kanttilfælde, og det er værd at huske, når en browser og et serverbibliotek fortolker den samme mærkelige streng.

Brug begrebet i en beslutning

Brug »Hvad er en URL?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Et site, der er hurtigt, kan findes og kan redigeres af jeres egne folk, uden en regning for hver ændret sætning.

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 udviklet 100 webapplikationer med HTML/HTML5 — 100 webapplikationer bygget

Har udviklet 100 webapplikationer med HTML/HTML5 Kontakt os →

Hvorfor det betyder noget

En offentliggjort URL er et løfte. Den havner i bogmærker, søgeindeks, mails, trykte brochurer, chatlogs og på andres sider, og ingen af de kopier opdaterer sig selv. Skifter adressen senere, knækker de alle sammen, medmindre den gamle adresse bliver ved med at svare med en omdirigering til den nye.

Derfor er den vigtige beslutning en tidlig en: hvad adressen bygges af. Alt, der sandsynligvis ændrer sig, er et dårligt fundament, hvad enten det er en sidetitel, et kategoritræ, et produktnavn eller en filendelse, der afslører det nuværende framework. En kort identifikator, eller en slug gemt som data og kun ændret med vilje, holder. Det samme gælder en plan for, hvad der sker, når en adresse alligevel må flytte: den gamle bliver en permanent omdirigering frem for en 404.

Den anden beslutning handler om, hvad der holdes ude. RFC 9110 siger det ligeud: adresser er lavet til at blive delt, ikke til at blive beskyttet. De skrives i logfiler på servere og proxyer, gemmes i browserens historik, kopieres ind i analyseværktøjer og sendes, afhængigt af politikken, videre i `Referer` -feltet. Sessions-id'er, nulstillingstokens, API-nøgler og persondata hører hjemme i headerfelter eller i forespørgslens krop.

Længden er et mindre spørgsmål. HTTP-specifikationen anbefaler, at software accepterer adresser på mindst 8.000 oktetter. Det er rummeligt, men en forespørgselsstreng med en hel formular eller et serialiseret filter rammer som regel en kortere grænse et sted i kæden, før den når så langt.

Sådan ser det ud i praksis

Arbejdet på dette site behandler adresser som data, der lever længere end teksten på siden. Da 495 achievement-sider på tre sprog blev genereret fra en skrivebeskyttet SQLite-eksport på det dokumenterede implementeringstrin i august 2026, handlede den vigtigste beslutning om adressen. Sitet havde udledt en sides URL fra de første ord i det engelske udsagn, så en rettet sætning flyttede siden i stilhed og ødelagde hvert eneste link til den. Adressen blev derefter skrevet som data, én række pr. adresse pr. achievement: den første række var den aktuelle, og hver senere række var en pensioneret adresse, som sitet udsendte som en omdirigering. Det dokumenterede resultat var, at alle adresser, der var offentliggjort frem til det trin, stadig svarede.

En note om mobile wallets viser, hvad der sker, når én URL rejser inde i en anden. Phantom, Solflare og Backpack åbner en dApp gennem universelle links, der bærer måladressen i deres egen sti og den forespørgende oprindelse i en `ref` -parameter, begge sendt gennem `encodeURIComponent` . Et ukodet `?` eller `&` i målet ødelægger fortolkningen, og wallet'en åbner på sin startskærm i stedet for browserfanen. Med et gættet eget skema som `backpack://` går det værre: appen har ikke registreret nogen rute til det skema, og brugeren lander på installationssiden.

Adresser tjener også som identitet. Da 963 blokke med strukturerede data blev erstattet af én sammenkædet graf, blev identifikatorerne for organisationen, sitet og personen dannet ud fra sitets oprindelse, og oversigtssider udgiver deres elementer alene som URL'er. På det dokumenterede implementeringstrin i juli 2026 blev det valg målt: at navngive elementerne på porteføljeoversigten ville

have betydet 107 fulde udsagn og en komprimeret side, der var 52 procent større. Tjekket, der vogter grafen, efterprøver, at hver URL, den nævner, faktisk blev bygget.

Hvor det går galt

Fejl i URL'er er sjældent højlydte. Adressen ser rigtig ud; den peger bare et andet sted hen eller afslører mere end tilsigtet.

Kodningen sker nul gange eller to. En returadresse sat råt ind i en forespørgsel, som i `?next=/cart?item=4&size=m`, deles ved det ukodede `&`, og serveren ser `size=m` som sin egen parameter. Kodes en allerede kodet streng igen, bliver `%20` til `%2520`, og siden modtager et bogstaveligt `%20`. Kod hver del præcis én gang, i det øjeblik den sættes ind i adressen, med den funktion, der er beregnet til dele. I JavaScript er det `encodeURIComponent`; `encodeURI` lader `?`, `&` og `#` være, fordi den forventer en hel adresse.

En hemmelighed rejser med i forespørgselsstrengen. Et nulstillingstoken eller en API-nøgle i URL'en kopieres til adgangslogs, browserhistorik og analyseværktøjer, og med en løs referrer-politik sendes den i `Referer` til hver tredjepart, siden henter indhold fra. Læg legitimationsoplysninger i et headerfelt. Skal et token rejse i et link, som i en mail om ny adgangskode, så lad det kun virke én gang og udløbe hurtigt, og sæt en `Referrer-Policy`, der holder det hjemme.

Én side svarer på fem adresser. `http` og `https`, med og uden `www`, med og uden skråstreg til sidst, `/About` og `/about`. Skema og vært skelner per definition ikke mellem store og små bogstaver; stien gør, medmindre serveren bestemmer andet. Søgemaskiner deler sidens placering mellem kopierne, caches gemmer hver for sig, og statistikken tæller dem hver for sig. Vælg én form, omdirigér alle andre permanent til den, og angiv den som kanonisk.

Et relativt link løses mod den forkerte base. En reference som `img/logo.png` erstatter alt efter den sidste skråstreg i den aktuelle sti. På `/docs/guide` bliver det til `/docs/img/logo.png`, på `/docs/guide/` til `/docs/guide/img/logo.png`. Én skråstreg til sidst flytter hvert relativt link på siden. Links, der starter med `/`, går uden om problemet.

Logikken afhænger af fragmentet. Omdirigeringer på serveren, adgangslogs og statistik, der læser HTTP-forespørgslens request-target, modtager hverken # eller fragmentet. En klient kan sende det særskilt, når programmet har brug for det. Det, der skal nå serveren som en del af den første forespørgsel, hører hjemme i stien eller forespørgselsdelen.

Læs mere

- RFC 3986: Uniform Resource Identifier (URI): Generic Syntax — grammatikken, procentkodning, opløsning af relative referencer og normalisering.
- WHATWG URL Standard — den fortolkningsalgoritme, browserne implementerer, trin for trin.
- RFC 3987: Internationalized Resource Identifiers — hvordan adresser skrevet i alle skriftsystemer bliver til procentkodede bytes.
- RFC 8820: URI Design and Ownership — hvorfor ejeren af en vært, og ingen andre, bestemmer, hvordan dens stier ser ud.
- RFC 9110, afsnit 17.9 — sikkerhedsnoten om følsomme oplysninger i adresser.

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 en URL?« 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/url/