Engineer

Gopher

en menyprotokoll for å hente dokumenter over internett

Hva det er

Gopher er en protokoll for å hente dokumenter og menyer over internett. En klient åpner en TCP-forbindelse, sender én linje som sier hva den vil ha, leser svaret og er ferdig. Det formelle navnet er Internet Gopher-protokollen, adressene begynner med `gopher://`, og et nettsted som publiseres over den, kalles som regel et Gopher-hull. Protokollen er fra 1991, og RFC 1436 beskrev den i mars 1993.

Utvekslingen er så kort at du kan skrive den for hånd. Klienten kobler seg til port 70 og sender en selektor, en streng serveren har delt ut tidligere og som godt kan være tom, etterfulgt av vognretur og linjeskift. Serveren svarer og lukker forbindelsen. Hver forespørsel står for seg selv, uten økt, uten informasjonskapsel og uten headere. En tom selektor ber om serverens øverste meny.

Menyen er selve kjernen. Hver linje i en meny er ett element, satt sammen av ett enkelt typetegn, en visningstekst og deretter en selektor, en vert og en port, skilt fra hverandre med tabulatorer. En linje som bare inneholder et punktum, avslutter menyen. Her er en liten meny, med hver tabulator skrevet som `<TAB>`:

```
0About this server<TAB>/about.txt<TAB>example.org<TAB>70
1Reports<TAB>/reports<TAB>example.org<TAB>70
7Search the reports<TAB>/search<TAB>example.org<TAB>70
.
```

Typetegnet forteller klienten hva den får, før den spør. RFC 1436 definerer `0` for en tekstfil, `1` for en annen meny, `7` for et søk som tar imot en spørring, `9` for en binærfil, `g` for et GIF-bilde og en håndfull til. To typer som er i bred

bruk i dag, kommer fra praksis og ikke fra RFC-en: `i` for en linje ren tekst som vises inne i en meny, og `h` for et HTML-dokument, som klienter også bruker til å gi en webadresse videre til en nettleser.

Utformingen er bevisst liten. RFC 1436 har en del med overskriften «Simplicity is intentional», og den har statusen Informational, ikke statusen som internettstandard. Adresseformatet `gopher://` kom senere, i RFC 4266, der det første tegnet i stien er elementtypen og resten er selektoren.

Bruk begrepet i en beslutning

Bruk «Hva er Gopher?» 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

Gopher betyr mindre i dag som en vei til et stort publikum enn som et klart bilde av hva et dokument trenger, og hva det klarer seg fint uten. En meny er navigasjon skrelt ned til beinet: hver vei ut av en side står oppført, med type og på sin egen linje, og alt ligger åpent i stedet for bak en hover-effekt eller et skript. Å publisere der tvinger fram spørsmål som HTML lar en forfatter skyve foran seg. Hva heter denne siden? Hvilke av lenkene er de virkelige? Gir teksten fortsatt mening når lenkene er tatt ut av den?

For en organisasjon er et Gopher-hull en ekstra inngangsdør, aldri en erstatning for weben. Leserne trenger en egen klient, og de er få. Hva døren koster, avhenger helt av hvordan den er bygd. Holdes den ved like for hånd som en andre kopi av nettstedet, kommer den ut av takt første gang den ene kopien blir redigert og den andre glemt. Genereres den fra den samme kilden av det samme bygget, koster den en mal og litt diskplass, og da tjener den mest som en stresstest av skrivingen.

Det andre som er verdt å forstå, er sikkerheten. Gopher har ikke noe krypteringslag. Hver selektor, og dermed hver side en leser ber om, krysser nettet i klartekst, og det samme gjør hvert svar. RFC 1436 har ingenting for

innlogging, og spørringen en leser skriver inn i et søkeelement, går også i klartekst. Det er greit for offentlige dokumenter og galt for alt annet, og det betyr at operatørens egne logger fortjener like mye omtanke som innholdet.

Protokollen er også billig å feilsøke. Siden den verken har TLS, headere eller tilstand, viser én enkelt kommandolinje nøyaktig de bytene en klient mottar, og det gjør en Gopher-kopi til en av de enkleste utgavene av et nettsted å kontrollere.

Slik ser det ut i praksis

Dette nettstedet er et Gopher-hull like mye som et vanlig nettsted. Siden Engineer i en terminal oppgir adressen, `gopher://engineer.company/`, beskriver Gopher som en menyprotokoll fra 1991 med ren TCP, uten kryptering og uten skripting, og sier at nettstedet der er et sett med menyer, én per mappe. Den nevner Lynx, Bombadillo og gopher-cli som klienter og viser en forespørsel gjort for hånd med `printf` og `nc` mot port 70, hele protokollen på én linje.

Bygget bak er beskrevet i gjennomgangen av speilingen av nettstedet til Gemini og Gopher. Det samme bygget som skriver HTML-en, skrev også 777 Gopher-menyer på det dokumenterte implementeringstidspunktet. De ble satt i drift i det samme treet og servert av en liten daemon på samme vert. En menylinje er et sett tabulatorseparerte felt, slik at en lenke ikke kan stå inne i en setning, og prosaen får faste linjeskift ved sekstiåtte kolonner mens nettstedet bygges. Å arbeide innenfor den rammen skjerpet skrivingen.

Malene viser hvordan hver side blir en meny. Lenker inne i teksten beholder ordene sine og mister målene, overskrifter mister #-tegnene sine, punktlister blir innrykkede stjerner, og prosaen brytes ved 68 kolonner. Kode er unntaket og går gjennom ubrutt, fordi en kodelinje som brytes ved kolonne 68, er et annet program. Kategorier og tjenester blir elementer av type `1` som peker på sine egne menyer, og nederst på hver side står et `h` -element der en `URL:` -selektor fører til webversjonen av den samme siden.

Å drifte et Gopher-hull lærte også nettstedet noe om logger. Siden Personvern forklarer at nettstedet siden september 2026 bare teller hvor mange forespørsler Gemini- og Gopher-kopiene besvarte hver dag, uten adresse, side eller besøkende. Den dagen tellingen begynte, viste det seg at operativsystemet

starter Gopher-kopien på nytt for hver besøkende og gir det øyeblikket navn etter adressen den besøkende kom fra, i en egen logg. Den loggen utløper nå etter tretti dager, som alt annet nettstedet tar vare på.

Hvor det går galt

En side stopper halvveis og lister opp mappen

En leser åpner en side og får den første halvdelen, deretter en liste over filene i mappen. Servere som Gophernicus leser en gophermap før de sender den, og en linje uten tabulatorer der første tegn er et av direktivene deres, blir behandlet som en instruksjon: `#` skjuler linjen som en kommentar, `=` henter inn en annen fil, og `*` eller `.` stopper menyen. Nettstedet støtte på dette da en CSS-linje i den ukrainske utgaven av notatet om understrekinger i Safari begynte med stjernen som avslutter en CSS-kommentar, og serveren i drift kuttet siden der og skrev ut index.html, index.md og index.gmi i stedet. Skriv hver slik linje som et eksplisitt `i` -element, og la bygget feile på enhver naken linje som fortsatt begynner med et direktivtegn.

Oppmerking og entiteter kommer fram som tekst

Et avsnitt dukker opp som én svært lang linje full av vinkelparenteser, eller med et og-tegn som fortsatt er stavet ut som HTML-entiteten sin. Konverteringsverktøyet kopierte HTML inn i et format som ikke vet hva HTML er. Fjern taggene og behold lenketeksten, dekod entitetene, og la innebygd kode være i fred, slik at symbolene i den overlever.

Lange linjer renner ut over høyre kant

Prosaen flyter forbi skjermkanten, eller visningstekster blir kuttet midt i et ord. Gopher-klienter har ingen plikt til å bryte om teksten, og RFC 1436 ber om visningstekster på under 70 tegn. Bryt prosaen under bygget, kort ned menyetiketter med vilje, og hold koden utenfor linjebryteren.

En lenke henter noe annet enn den skulle

En `gopher://` -adresse åpner en feilmelding eller et element som ikke har noe med saken å gjøre. RFC 4266 leser det første tegnet i stien som elementtypen, så `gopher://example.org/reports` ber om type `r` og selektoren `eports` . Sett typen foran selektoren, som i `gopher://example.org/1/reports` , og kopier adresser fra en meny som virker, i stedet for å skrive dem inn selv.

Leselisten går over nettet i klartekst

Hver side en leser åpner, er synlig for alle langs nettverksveien, og kanskje også i logger ingen hadde tenkt å ta vare på. Publiser bare det som ville vært greit å sende på et postkort, les loggene for hele maskinen og ikke bare daemonens, og gi hver logg en utløpsdato.

Les mer

- RFC 1436: The Internet Gopher Protocol — den opprinnelige beskrivelsen: syklusen med forespørsel og svar, menylinjer, elementtyper og grunnene til å holde det enkelt.
- RFC 4266: The gopher URI Scheme — hvordan en `gopher://` -adresse koder vert, port, elementtype og selektor.
- IANA Service Name and Transport Protocol Port Number Registry — den offisielle registreringen av port 70 som Gopher-porten.
- Gophernicus — kildekode og dokumentasjon for en Gopher-server som er i bruk i dag, også gophermap-formatet den leser.

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 Gopher?» 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/gopher/