Engineer

Gemini

ett lättviktigt internetprotokoll för att leverera hypertext

Vad det är

Gemini är ett applikationsprotokoll för att hämta dokument, där hela utbytet ryms i en enda krypterad anslutning: klienten skickar en adress, servern skickar ett svar och anslutningen stängs. Projektets egna dokument kallar det Gemini-protokollet, adresserna börjar med `gemini://` , en webbplats som levereras över det kallas en kapsel och nätverket av kapslar kallas Geminispace.

Projektet startade i juni 2019 under sin grundare Solderpunk, som fortfarande är dess ”Benevolent Dictator”. Specifikationen, version 0.24.1, är släppt som allmän egendom under CC0 och beskriver protokollet som en stegvis förbättring av Gopher snarare än en avskalad variant av HTTP. Projektets FAQ gör jämförelsen från andra hållet: protokollet påminner om något mellan HTTP 0.9 och HTTP 1.0, med en enda sorts begäran, ungefär en `GET` .

Klienten ansluter till TCP-port 1965, ett nummer som syftar på den första bemannade flygningen i Gemini-programmet, i mars 1965, och genomför en TLS-handskakning; båda sidor måste stödja TLS 1.2 eller senare. Den skickar den absoluta URL:en följd av vagnretur och radmatning, och URL:en får vara högst 1 024 byte lång. Servern svarar med en enda huvudrad: en tvåsiffrig statuskod, ett mellanslag och en kort text vars betydelse beror på koden. Koderna faller i sex grupper: 10–19 ber om inmatning, 20–29 betyder att begäran lyckades, 30–39 omdirigerar, 40–49 är tillfälliga fel, 50–59 är permanenta och 60–69 gäller klientcertifikat. När begäran lyckas är texten en MIME-typ, och innehållet följer som råa byte, utan komprimering och utan uppdelning i block. Servern stänger sedan anslutningen med TLS-meddelandet close_notify, och det är så klienten vet att den har fått allt.

Protokollet kan bära vilken sorts fil som helst, men det utformades tillsammans med ett eget dokumentformat, gemtext, som levereras som `text/gemini` .

Använd begreppet i ett beslut

Använd ”Vad är Gemini?” 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

Det verkliga beslutet gäller sällan om en webbplats ska flytta till Gemini. Enligt FAQ:n är protokollet inte tänkt att ersätta webben eller Gopher, och projektet har ett namn för att publicera samma innehåll på två eller tre av dem: bihosting eller trihosting. Frågan för den som äger webbplatsen är om en andra ytterdörr är värd att hålla öppen.

Det protokollet erbjuder är integritet som sitter i själva konstruktionen. FAQ:n slår fast att Gemini inte har och aldrig kommer att få någon motsvarighet till huvudena Cookie, Referer eller User-Agent. En begäran är en URL och ingenting annat, så en server har mycket lite att registrera även om operatören skulle vilja, och i läsarens klient körs inget skript som kan rapportera tillbaka. FAQ:n räknar ”defensive non-extensibility” bland prioriteringarna bakom designen: formaten för begäran och svar lämnar inget utrymme för ett nytt huvudfält, så en senare version har ingenstans att föra in spårning igen. För den som publicerar ren text är det hela poängen.

Priset är mycket av det som webben har fått sedan HTTP 1.0. I stället för kakor ber en kapsel som måste känna igen en återkommande läsare om ett klientcertifikat, som läsaren själv måste skapa och ta vara på. Förtroendet vilar som standard inte på någon certifikatutfärdare: specifikationen rekommenderar starkt Trust on First Use, eller TOFU, där klienten godtar det certifikat som en server visar första gången och sedan låser fast det. Därmed blir serverns certifikat ett långsiktigt löfte. Krypteringen döljer sidan, men serverns namn färdas i TLS-tillägget Server Name Indication, och FAQ:n medger att värdnamnet fortfarande läcker. Läsarna behöver en särskild Gemini-klient i stället för en webbläsare, så publiken är liten. Stora filer passar också dåligt, eftersom ett svar varken anger storlek eller kontrollsumma.

Ett team som behöver interaktiva formulär, konton med lösenord, webbanalys eller en rik layout hör hemma på webben. Ett team som publicerar dokument och vill ha en kanal utan kakor eller skript på klientsidan kan lägga till Gemini för en liten kostnad, så länge innehållet redan finns som filer.

Så ser det ut i praktiken

Den här webbplatsen är själv en kapsel på `gemini://engineer.company` . Kolofonen nämner Agate för Gemini, Gophernicus för Gopher och Tor för onion-tjänsten, vid sidan av Caddy för webben. Alla fyra läser samma mapp med filer, så en driftsättning uppdaterar alla fyra och ingen av dem kan glida isär från de andra. Det är FAQ:ns bihosting utsträckt till fyra kanaler och gjort på det billigaste sätt som finns: kapseln är ytterligare en läsare av en mapp som varje driftsättning ändå fyller, inte en andra webbplats som måste hållas i takt.

Samma sida beskriver begränsningen ovan: Gemini och Gopher behöver var sin klient snarare än en webbläsare, och Gopher skickar sidförfrågan i klartext. Gemini krypterar den, men nätverket du läser från ser ändå vilken serveradress du bad om. Det är serverns namn som läcker enligt FAQ:n, sett ur läsarens perspektiv i stället för konstruktörens.

Integritetssidan skiljer mellan applikationernas åtkomstlogg och operativsystemets loggar. Gemini- och Gopher-applikationerna för ingen åtkomstlogg; sedan september 2026 registrerar deras dagliga räknare bara det sammanlagda antalet förfrågningar. Gopher har också en anslutningslogg i operativsystemet som innehåller källadressen; den loggen raderas efter 30 dagar.

Var det går fel

Den första raden i ett svar avslöjar det mesta som går snett med Gemini, och den raden är lätt att läsa för hand med enbart OpenSSL:

```
printf 'gemini://engineer.company/\r\n' |
  openssl s_client -quiet -connect engineer.company:1965 \
    -servername engineer.company
```

Den första raden som kommer tillbaka är huvudraden, `20 text/gemini;lang=en` för den här kapseln, och sidan följer efter den.

Återkommande läsare varnas för att kapseln kan vara utsatt för en attack. Klienter som följer specifikationen sparar fingeravtrycket och utgångsdatumet för det första certifikat de ser. Ett annat certifikat innan det datumet har passerat räknas som ett möjligt tecken på en mellanhandsattack, man-in-the-middle. En operatör som skapar ett nytt certifikat vid varje driftsättning, eller flyttar kapseln till en ny maskin utan dess nyckel, utlöser den varningen hos varje trogen läsare. Säkerhetskopiera nyckeln och certifikatet tillsammans med resten av servern, utfärda certifikatet för en lång period och byt det först efter det utgångsdatum som klienterna redan har sparat.

En sökruta svarar med status 59. Ett svar med status 10 ber läsaren om en rad inmatning, och klienten skickar tillbaka den som frågedelen av samma URL, procentkodad. URL:en är begränsad till 1 024 byte, och en server måste avvisa allt längre som en felaktig begäran. Ett långt svar når gränsen fort, och text utanför ASCII når den ännu fortare: en kyrillisk bokstav är två byte i UTF-8 och sex när den har kodats. Be om korta svar och använd status 11 för känslig inmatning som en lösenfras, så att klienterna inte visar den på skärmen.

Läsare loggas ut mellan sidorna. Protokollet saknar kakor, så sessionstillstånd som hålls på webbens vis överlever inte från en begäran till nästa. En kapsel som behöver veta vem som frågar svarar med status 60, läsarens klient skapar ett certifikat och servern känner igen det vid senare förfrågningar. Specifikationen begränsar certifikatet till den värd och port som bad om det och till den begärda sökvägen med allt under den, så lägg alla sidor som behöver identiteten under en och samma sökväg. Skriv på sidan vad certifikatet ska användas till, eftersom läsaren måste godkänna det.

En stor nedladdning kommer fram avkortad utan att någon märker det. Ett svar anger varken längd eller kontrollsumma, så en överföring som stannar halvvägs kan se ut nästan som en som blev klar, bortsett från det uteblivna close_notify. Guiden om bästa praxis föreslår att filer större än 100 MiB levereras på något annat sätt. Lägg stora filer på HTTPS och länka till dem från kapseln; en Gemini-sida får länka till vilket URL-schema som helst.

Läs mer

- Gemini network protocol specification — förfrågningar, varje statuskod, hur TLS och fastlåsta certifikat används, i ett kort dokument.
- Project Gemini FAQ — projektets mål, dess jämförelse med HTTP och Gopher och dess eget svar på kritiken att TOFU är svagt.

- Best practices for Gemini implementations — filnamn, stora filer, gränser för omdirigeringar och inställningar för TLS.
- History of Project Gemini — hur en idé från Gopherspace 2019 växte till ett eget nätverk, berättat av projektet.

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 Gemini?” 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/gemini/