

HTTP

Forståelse af HTTP-protokollen

HTTP (Hypertext Transfer Protocol) er en kommunikationsprotokol, der styrer dataudveksling mellem klienter (som browsere eller scripts) og servere på internettet. Når man arbejder med software, der kommunikerer med hjemmesider eller interagerer med webapplikationer, er det vigtigt at forstå de forskellige HTTP-metoder og statuskoder for effektiv fejlfinding og integration.

HTTP Metoder:

- **GET:** Bruges til at anmode om data fra serveren uden at ændre serverens tilstand. GET-anmodninger bruges typisk til at hente filer eller hjemmesider. Data sendes som parametre i URL'en, hvilket gør GET mindre egnet til sensitive oplysninger, da de er synlige i URL'en.
- **POST:** Bruges til at sende data til serveren, som typisk vil ændre serverens tilstand. POST er velegnet til at uploade filer, indsende formularer eller foretage handlinger, der kræver ændringer på serveren. Data sendes som en del af anmodningens brødtekst og er ikke synlig i URL'en.
- **PUT:** Anvendes til at uploade eller opdatere en eksisterende ressource på serveren. I modsætning til POST er PUT idempotent, hvilket betyder, at det samme kald kan gentages uden at ændre resultatet.
- **DELETE:** Bruges til at slette en ressource på serveren. Som PUT er DELETE også idempotent, hvilket betyder, at gentagne anmodninger om at slette en ressource ikke fører til fejl, hvis ressourcen allerede er slettet.
- **HEAD:** Minder om GET, men uden at returnere selve brødteksten i svaret. Bruges ofte til at kontrollere, om en ressource findes, eller til at få metadata som længde og type uden at hente hele indholdet.

HTTP Statuskoder:

Når en HTTP-anmodning sendes, returnerer serveren en statuskode, der angiver, hvordan anmodningen blev behandlet. Her er nogle af de vigtigste kategorier af statuskoder:

- **2xx (Succesfulde svar):** Disse koder indikerer, at anmodningen blev behandlet med succes.
 - **200 OK:** Anmodningen blev gennemført, og svaret indeholder de ønskede data.

- **201 Created:** Bruges ved succesfuld oprettelse af en ressource, typisk ved POST eller PUT.
- **3xx (Omdirigeringer):** Bruges til at fortælle klienten, at den skal foretage en anden anmodning til en ny URL.
 - **301 Moved Permanently:** Ressourcen er permanent flyttet til en ny adresse.
 - **302 Found:** Ressourcen er midlertidigt tilgængelig på en anden adresse.
- **4xx (Klientfejl):** Indikerer, at noget gik galt med klientens anmodning.
 - **400 Bad Request:** Serveren forstod ikke anmodningen pga. syntaksfejl.
 - **401 Unauthorized:** Klienten mangler autorisation til at få adgang til ressourcen.
 - **404 Not Found:** Den ønskede ressource kunne ikke findes på serveren.
- **5xx (Serverfejl):** Indikerer problemer på serverens side.
 - **500 Internal Server Error:** En generel fejl opstod på serveren.
 - **503 Service Unavailable:** Serveren er midlertidigt ikke tilgængelig, typisk på grund af overbelastning eller vedligeholdelse.

requests-modulet i Python

requests-modulet er et kraftfuldt og simpelt bibliotek til at sende HTTP-anmodninger i Python. Dette er ikke en del af python standard biblioteket og skal derfor installeres gennem pip. Det understøtter alle standard HTTP-metoder og gør det nemt at håndtere svar fra serveren, inklusive håndtering af fejl.

Et eksempel på en simpel GET-anmodning for at printe en hjemmeside kunne se sådan ud:

```
import requests

url = "https://example.com"
response = requests.get(url)

if response.status_code == 200:
    print("Succeed in getting resource:", response.status_code)
    print(response.content)
else:
    print("Failed get resource:", response.status_code)
```

Her bruges requests.get() til at sende en GET-anmodning til en hjemmeside, og hvis anmodningen lykkes (statuskode 200), printes indholdet.

Kilder og Yderligere Information

For en information kan følgende ressourcer benyttes:

1. **Requests Bibliotek**

<https://requests.readthedocs.io/en/latest/>

2. **HTTP Statuskoder**

https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#

3. **HTTP Metoder**

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>