HTTP protokoll

A HTTP *(**H**yper**T**ext **T**ransfer **P**rotocol)* egy TCP feletti, alkalmazási rétegbeli protokoll, melyet széleskörben használnak webalkalmazások integrációja során.

Működése

- A HTTP egy **kliens-szerver** protokoll, amely kapcsolatot teremt egy kliens (például egy webböngésző) és egy szerver (például egy webszerver) között.
- A HTTP **kérés-válasz modellt** használ. A kliens küld egy kérést a szervernek, majd a szerver választ küld a kliensnek.
    - A HTTP **kérések** egy metódusból, egy URL-ből, opcionális fejlécekből és törzsből állnak. A leggyakoribb HTTP metódusok a GET, POST, PUT és DELETE.
    - A HTTP **válaszok** egy állapotkódból, valamint opcionális fejlécekből és választörzsből állnak. Az állapotkód jelzi, hogy a kérés sikeres volt-e, és a válasz törzse tartalmazza a kért adatokat (vagy adott esetben a hiba okát).
- A HTTP egy **állapotmentes** protokoll, ami azt jelenti, hogy nem emlékszik a korábbi kommunikációra a kliens és a szerver között. Minden kérés/válasz csere független az előző adatcseréktől.

Kérés

Egy példa kérés felépítése a következő:

```
GET /template/web/img/logo-uni-miskolc.png HTTP/1.1
User-Agent: curl/7.35.0
Host: www.uni-miskolc.hu
Accept: */*
```

A kérés első sora jelöli meg az elérni kívánt erőforrást. Ezt a kéréshez tartozó, tetszőleges számú **fejléc** (header) követi, `Fejléc: érték` alakban.

Ezt követi az opcionális **törzs** (body) rész, melyben a kérés teljesítéséhez szükséges adatokat helyezhetünk el tetszőleges formátumban (pl. egy új felhasználó adatait JSON formátumban).

Metódusok

A leggyakrabban alkalmazott metódusok és szimbolikus jelentésük:

- GET: Adatok lekérdezése
- POST: Új adat hozzáadása
- PUT: Meglévő adat módosítása

- DELETE: Meglévő adat törlése

Query paraméterek

A GET kérés speciális, mert törzs részt nem tartalmazhat. Helyette - amennyiben szükséges - a kéréshez tartozó paramétereket ún. query paraméterekként adhatjuk meg.

Példa: `/template/web/img/logo-uni-miskolc.png?time=2020-05-01&thumb=true`

A paraméterek `név=érték` formában adhatóak meg. A paramétereket az útvonaltól `?`, egymástól `&` jel választja el.

Path paraméterek

A HTTP szabványnak ugyan nem része, de a szerverek gyakran támogatják a paraméterek átadását a megjelölt útvonal részeként:

Példa: GET `/api/users/`**`Bela123`**`/repos/`**`my-awesome-project`**

A szerver ebben az esetben a kérésben szereplő útvonalból olvas ki paramétereket. A fenti példában ilyen paraméter a felhasználó (`Bela123`) és a keresett repository elnevezése (`my-awesome-project`). Az útvonal ezen részei a keresett repository-nak megfelelően módosíthatók tetszőleges értékekre.

Kérés törzsében szereplő paraméterek

POST, PUT, DELETE kéréseknél opcionálisan a törzs rész is kitölthető.

Például a következő kérés egy felhasználó létrehozására szolgál:

```
POST /users HTTP/1.1
User-Agent: curl/7.35.0
Host: api.example.com
Accept: */*
Content-Type: application/json
Content-Length: 28

{ "name": "feri", "gender": "male", "age": 15 }
```

A fenti üzenet tartalmazza a megfelelő HTTP metódust (`POST`) és útvonalat (`/users`). Ezek együttesen azonosítják a szerver számára, hogy a kliens milyen műveletet szeretne végrehajtani (új felhasználó létrehozása). Emellett láthatunk még néhány szükséges fejlécet, és a kérés törzsében az újonnan létrehozni kívánt felhasználó adatait, JSON formátumban.

Válasz

A kliens kéréseire a szerver válaszokat ad. Egy kéréshez legfeljebb egy válasz tartozhat.

A HTTP válasz a következőképpen épül fel:

```
HTTP/1.1 200 OK
Date: Thu, 18 Mar 2021 13:02:06 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 29
Connection: keep-alive
X-Powered-By: Express
X-Ratelimit-Limit: 1000
X-Ratelimit-Remaining: 999
X-Ratelimit-Reset: 1616072536
Cache-Control: no-cache
Pragma: no-cache

{ "id": 101, "name": "feri", "gender": "male", "age": 15 }
```

A válasz egyebek mellett tartalmaz egy **státuszkódot** (a fenti példában 200), ami a művelet sikerességéről ad információt.

Ezt követik a válaszhoz tartozó fejlécek.

A törzs rész opcionális: a fenti válaszban a szerver visszaküldte az adatbázishoz hozzáadott felhasználó adatait, köztük egy azonosítóval, amire a későbbiekben a kliens hivatkozhat, ha szeretné elérni a létrehozott felhasználót.

HTTP státuszkódok

Az egyes HTTP státuszkódok jelentését a HTTP specifikációja tartalmazza, általánosságban a következők írhatóak le róluk:

- 1xx: Informatív üzenet (pl. 101 Switching Protocols)
- 2xx: A kérést a szerver sikeresen megkapta, értelmezte, teljesítette (pl. 200 OK, 201 Created)
- 3xx: Átirányítás (pl. 301 Moved Permanently)
- 4xx: Kliens hiba (pl. 400 Bad Request, 404 Not found)
- 5xx: Szerver hiba (pl. 503 Service Unavailable)

(Elérhető státuszkódok)

HTTP kérések feldolgozása fejlesztői eszközökkel

HTTP kérések küldésére és válaszok fogadására számos módszer áll rendelkezésre, amelyek az

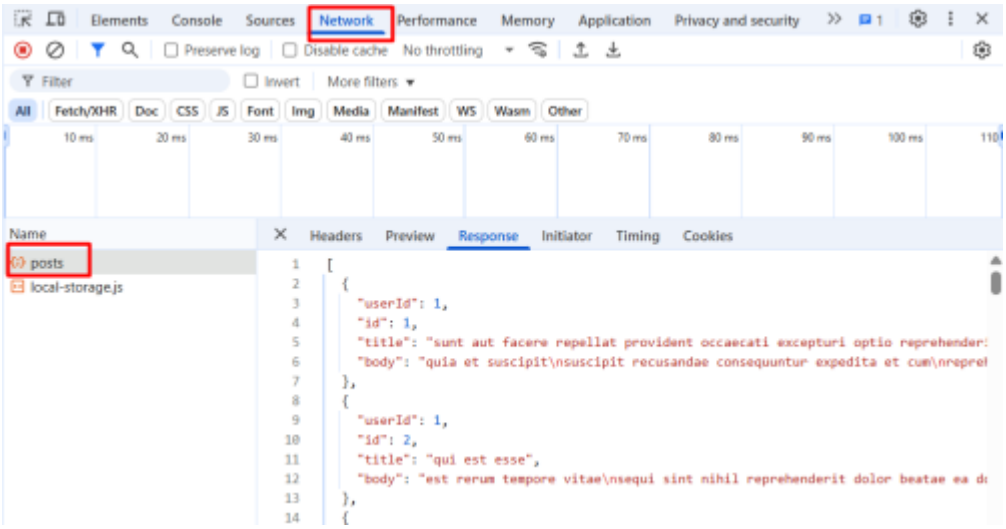
egyszerű parancssori alkalmazásoktól (pl. curl, wget) kezdve a grafikus felülettel rendelkező eszközökön (pl. webböngészők, Postman) át a programozói könyvtárakig (pl. Fetch API, Axios, OkHttp) terjednek. Míg a programkönyvtárakat általában webalkalmazások fejlesztésére használjuk, addig a parancssori eszközök és a GUI-val rendelkező alkalmazások elsősorban tesztelési célokat szolgálnak, lehetővé téve a HTTP kérések gyors ellenőrzését és elemzését.

Google Chrome

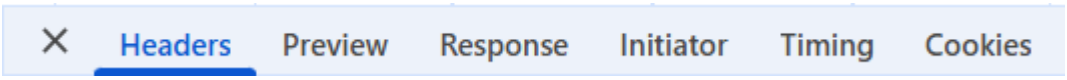
A Chrome fejlesztői eszközeivel (DevTools) könnyen nyomon követhetők a HTTP kérések:

Gyors teszteléshez nyiss egy új lapot, nyomd le az F12 gombot, majd töltsd be a https://jsonplaceholder.typicode.com/posts URL-t!

A DevTools-ban válts át a Network fülre, majd keresd meg az indított kérést:



Az itt megjelenő eszköztár segítségével elemezhető a HTTP forgalom:



A legfontosabb funkciók:

- **Headers** fül: Kérés és válasz fejléceinek, valamint a válasz státuszkódjának a megtekintése.
- **Preview** fül: A válasz törzsének megtekintése, elemzése.
- **Response** fül: A válasz törzsének megtekintése nyers szövegként.

A DevTools segítségével a webalkalmazásunk által kezdeményezett HTTP kommunikáció is megtekinthető. Fontos azonban, hogy POST, PUT, és DELETE kéréseket közvetlenül csak programkódból tudunk küldeni. Ahhoz, hogy ilyen kéréseket gyorsan, manuálisan tudjunk előállítani, szükség lehet valamilyen külső alkalmazás (pl. Postman) használatára.

Postman

HTTP kérések feldolgozása JavaScript segítségével

1. Fetch API