

# A HTTP működése

A gyakorlat két részből áll:

- 1) HTTP vizsgálata parancssorból **curl** segítségével ([httpbin.org](http://httpbin.org))
- 2) HTTP vizsgálata böngészővel, a beépített **Developer Tools** használatával.

Cél: a HTTP metódusok, headerek, státuszkódok, cache-elés és content negotiation valós működésének megértése.

## I. rész – HTTP vizsgálata curl segítségével

A **curl** parancs Windows verziója letölthető innen: <https://curl.se/windows/>

Nyissük meg a parancssort (command prompt) és lépjünk be a “bin” könyvtárba, ahol a **curl** program telepítve lett.

### 1. HTTP GET kérés alapjai

Parancs:

**curl.exe https://httpbin.org/get**

Megfigyelési szempontok:

- Milyen adatokat kapsz vissza?
- Mely HTTP metódus lett használva?

Magyarázat:

Ez egy klasszikus HTTP GET kérés. A szerver JSON formátumban visszaadja a kliens által küldött headereket és paramétereket.

### 2. HTTP POST metódus

Parancs:

**curl.exe -X POST -d "name=John&age=30" https://httpbin.org/post**

Megfigyelési szempontok:

- Hol jelenik meg az elküldött adat?
- Mi a különbség a GET és POST között?

Magyarázat:

A POST metódust adatküldésre használjuk. Az adatok a kérés törzsében (request body) helyezkednek el.

### **3. HTTP headerek vizsgálata**

Parancs:

**curl.exe -H "X-Test-Header: Oktatas" https://httpbin.org/headers**

Megfigyelési szempontok:

- Megjelenik-e az egyedi header?

Magyarázat:

A HTTP headerek metaadatokat hordoznak a kérésről és válaszról. Egyedi headerekkel a kliens és szerver extra információkat cserélhet.

### **4. HTTP státuszkódok**

Parancsok:

**curl.exe -i https://httpbin.org/status/200**

**curl.exe -i https://httpbin.org/status/404**

**curl.exe -i https://httpbin.org/status/500**

Megfigyelési szempontok:

- Mit jelentenek a státuszkódok?

Magyarázat:

A státuszkódok jelzik a kérés eredményét. 2xx = sikeres, 4xx = kliens hiba, 5xx = szerver hiba.

### **5. Content negotiation (Accept header)**

Parancsok:

**curl.exe -H "Accept: application/json" https://httpbin.org/anything**

**curl.exe -H "Accept: text/html" https://httpbin.org/anything**

Magyarázat:

A kliens az Accept headerrel jelzi, milyen formátumú választ szeretne. Ez a content negotiation alapja.

### **6. Cache-elés alapjai**

Parancs:

**curl.exe -i https://httpbin.org/cache/30**

Megfigyelési szempontok:

- Cache-Control header

Magyarázat:

A cache-elés csökkenti a hálózati terhelést és gyorsítja az alkalmazásokat.

## II. rész – HTTP vizsgálata böngészővel (Developer Tools)

### 7. Network fül megismerése

Feladat:

- Nyisd meg a böngésző Developer Tools-t (F12)
- Network fül
- Nyisd meg: <https://httpbin.org/get>

Magyarázat:

A Network fül valós időben mutatja a HTTP kéréseket és válaszokat.

### 8. HTTP metódusok böngészőben

Feladat:

- Vizsgáld meg egy **GET** kérés metódusát
- Küldj **POST** kérést egy HTML form segítségével (<https://httpbin.org/forms/post>)
- Vizsgáld meg a többi HTTP metódust is

Magyarázat:

A böngésző GET metódust használ a lekérésekhez. POST metódust használnak az adatok (űrlapok) elküldésére.

### 9. Request és Response headerek

Feladat:

- Kattints egy kérésre
- Vizsgáld meg a Request Headers és Response Headers részét

Magyarázat:

Itt látható, milyen információkat küld a böngésző a szervernek, és mit válaszol a szerver.

### 10. HTTP státuskódok valós példákon

Feladat:

- <https://httpbin.org/status/404>
- <https://httpbin.org/status/403>

Magyarázat:

A böngésző vizuálisan is jelzi a hibás válaszokat.

## **11. Cache működés vizsgálata**

Feladat:

- Egy weboldal újratöltése (reload) normál módban
- Most újratöltés (reload) "Disable cache" bekapcsolásával

Magyarázat:

A cache hatással van az oldal betöltési idejére és a hálózati forgalomra.

## **12. Content negotiation böngészőben**

Feladat:

- Vizsgáld meg az Accept és Accept-Language headereket

Magyarázat:

A böngésző a felhasználó nyelvi és formátum preferenciáit is elküldi.