

HTTP Protokoll

A **H**yper**T**ext **T**ransfer **P**rotocol (HTTP) egy alkalmazásréteg protokoll az interneten, melyet weboldalak lekérdezésére és átvitelére használnak. A HTTP lehetővé teszi a *hiperhivatkozások* által összekapcsolt dokumentumok elérését a World Wide Web-en.

Rövid történet

1. **HTTP/0.9** – Az első, egyszerű verzió, ami csak egyetlen GET kérést támogatott (HTML fájlok letöltése).
2. **HTTP/1.0** (1996) – Ezzel a verzióval bevezetésre kerültek a fejlécek (headers), amik lehetővé tették a média típusok kezelését és a státuszkódokat.
3. **HTTP/1.1** (1997) – Jelentős frissítés, amely támogatta a tartalom átvitel kódolását, a kapcsolat újrafelhasználását és több egyéb javítást.
4. **HTTP/2** (2015) – A protokoll teljesítményének növelése érdekében bevezetett változások, mint a szerver push, a fejléc tömörítés, és az adatfolyamok párhuzamos kezelése.
5. **HTTP/3** (2023) – Az újabb verzió, amely a QUIC protokollra épül, javítva az adatátvitel sebességét és megbízhatóságát.

A protokoll lényege

A HTTP protokoll állapotmentes protokoll, ami azt jelenti, hogy minden egyes kérés független, és nem tárol információt a korábbi kérésekről. A kommunikáció egy kliens (általában webböngésző) és egy szerver között zajlik, ahol a kliens kéréseket küld a szerverre (pl. weboldal lekérése), a szerver pedig válaszokat küld vissza (pl. HTML dokumentumok).

A kérések és válaszok fejlécekkel (headers) és üzenettörzzsel (body) rendelkeznek, amelyek információkat tartalmaznak az átvitelről, az adatok típusáról, és egyéb metaadatokról.

A HTTP biztonságos változata, a HTTPS (Hypertext Transfer Protocol Secure), titkosítást használ az adatok védelmére, ami kulcsfontosságú a bizalmas információk online kezelése során.

HTTP kérés-válasz példa

Egy tipikus HTTP tranzakció két fő részből áll: egy **kérésből** (request) a kientől és egy **válaszból** (response) a szervertől.

HTTP Kérés

A kérés tartalmazza a kérés metódusát (pl. GET, POST), az elérni kívánt erőforrás URI-ját (Uniform Resource Identifier), a HTTP verzióját, valamint különböző fejléceket, amik információt adnak a kérésről.

Például egy egyszerű GET kérés egy weboldal lekérésére:

```
HTTP/1.1 GET /index.html
Host: www.pelda.hu
```

Ebben a példában a kliens a “www.pelda.hu” szerveren található “index.html” nevű erőforrást kéri a HTTP 1.1 protokoll használatával, és megadja a szerver host nevét.

HTTP Válasz

A válasz tartalmazza a HTTP verziót, a státuszkódot és szöveget (pl. 200 OK), amely jelzi a kérés sikerességét, valamint fejléceket és esetlegesen egy üzenettörzset, amely magát az erőforrást (például egy HTML dokumentumot) tartalmazza.

Példa egy egyszerű HTTP válaszra: (a fejléc és az törzs elválasztása egy üres sor)

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2022 22:38:34 GMT
Content-Type: text/html
Content-Length: 85

<html>
<head>
<title>Példa Oldal</title>
</head>
<body>
<h1>Üdvözlünk a Példa Oldalon!</h1>
<p>Ez egy példa szöveg.</p>
</body>
</html>
```

HTTP kérések

A HTTP protokoll négy alapvető kérésmódot (más néven “metódust”) definiál, amelyek különböző típusú műveletek végrehajtására szolgálnak a kliens és a szerver között. Ezek: **GET**, **POST**, **PUT**, és **DELETE**.

GET

A **GET** metódus egy erőforrás lekérésére használatos. Ez a leggyakrabban használt HTTP kérés, amely alkalmas arra, hogy információt kérjünk a szerverről.

POST

A **POST** metódus új erőforrások létrehozására vagy adatok beküldésére használatos a szerveren. Gyakran használják űrlapok adatainak elküldésére szerverre. A POST kérések törzset (body)

tartalmaznak, amely az elküldendő adatokat tartalmazza.

Példa:

```
POST /api/users HTTP/1.1
Host: www.pelda.hu
Content-Type: application/json

{
  "name": "Példa János",
  "email": "janos@pelda.hu"
}
```

PUT

A **PUT** metódus egy meglévő erőforrás frissítésére vagy, ha az adott URI-n még nem létezik erőforrás, annak létrehozására használatos. A PUT kérések törzset tartalmaznak, amely a frissítendő vagy létrehozandó erőforrás teljes új tartalmát tartalmazza.

Példa:

```
PUT /api/users/123 HTTP/1.1
Host: www.pelda.hu
Content-Type: application/json

{
  "name": "Új Név",
  "email": "ujnev@pelda.hu"
}
```

DELETE

A **DELETE** metódus egy meglévő erőforrás eltávolítására használatos a szerverről. A DELETE kérések általában nem tartalmaznak törzset, és az erőforrás az URI alapján azonosított.

Példa:

```
DELETE /api/users/123 HTTP/1.1
Host: www.pelda.hu
```

Webes Sütik Szabályai

A webes sütik kisméretű adatok, amelyeket a weboldalak helyeznek el a felhasználók számítógépén vagy mobil eszközén, amikor azokat meglátogatják. Céljuk, hogy emlékezzenek a felhasználó preferenciáira, segítsék a felhasználói munkamenetek kezelését, és gyűjtsenek adatokat a weboldal használatával kapcsolatban.

Főbb Szabályok és Előírások

A sütik használatára vonatkozó szabályok országonként és régióként eltérőek lehetnek, de számos általános irányelv és jó gyakorlat létezik, amelyeket a weboldalaknak be kell tartaniuk, különösen az Európai Unióban:

- **Tájékoztatás és beleegyezés:** Az EU-s e-Privacy Irányelv és a GDPR előírja, hogy a weboldalaknak tájékoztatniuk kell a látogatókat a sütik használatáról, és kifejezett beleegyezést kell kérniük a nem szükséges sütik használatához.
- **Adatvédelmi tájékoztató:** A weboldalaknak részletes adatvédelmi tájékoztatót kell biztosítaniuk, amely magyarázatot ad arra, hogy milyen típusú sütiket használnak, milyen adatokat gyűjtenek, hogyan használják ezeket az adatokat, és hogyan lehet a sütiket kezelni vagy letiltani.
- **Süтик kezelése:** A felhasználóknak lehetőséget kell biztosítani arra, hogy beállíthassák, mely sütiket engedélyezik, és bármikor módosíthassák a beállításait.
- **Biztonság:** A sütik használatakor biztosítani kell az adatok biztonságát, különösen az olyan sütik esetében, amelyek érzékeny információkat tárolnak, mint például belépési adatok vagy személyes azonosítók.

From:
<https://edu.iit.uni-miskolc.hu/> - Institute of Information Science - University of Miskolc

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:muszaki_informatika:http_protokol?rev=1710888701

Last update: 2024/03/19 22:51

