

Hypertext Transfer Protocol (HTTP)

Jeszenszky Péter
Debreceni Egyetem, Informatikai Kar
jeszenszky.peter@inf.unideb.hu

Verzió: 201502.1
Utolsó módosítás: 2015. február 27.

Tartalom

- Bevezetés, alapfogalmak
- Üzenetek
- Metódusok
- Állapotkódok
- Tartalomgyegetetés
- Részleges GET kérések
- Feltételes kérések
- Gyorsítótárazás
- Űrlapok
- Proxy szerver használata
- Hitelesítés
- Sütik
- Kapcsolatkezelés
- Webszerver szoftverek

Hypertext Transfer Protocol

- Alkalmazásszintű protokoll elosztott, kollaboratív hipermédia rendszerekhez
- Fejlesztése az IETF és a W3C közötti együttműködés keretében történt

Jellemzők

- A kliens-szerver modellen alapuló kérés-válasz protokoll
- Állapotnélküliség: az egymást követő kérések egymástól függetlenül kezeltek
- Kiterjeszthetőség
 - Például metódusok, állapotkódok, fejlécmezők
- Általános célú
 - Kliensek és webszerverek közötti kommunikációhoz használják elsősorban, de elvileg tetszőlegesen egyéb célra felhasználható

Történet (1)

- Az első dokumentált verzió:
 - HTTP 0.9 (Tim Berners-Lee)
<http://www.w3.org/Protocols/HTTP/AsImplemented.html>
 - Nagyon egyszerű, mindössze olyan GET kérések végrehajtását támogatja, melyre válaszként ASCII karakterekből álló HTML dokumentumok kerülnek visszaküldésre
- HTTP/1.0:
 - Tim Berners-Lee, Roy T. Fielding, Henrik Frystyk Nielsen, *Hypertext Transfer Protocol—HTTP/1.0*, RFC 1945, May 1996.
<http://tools.ietf.org/html/rfc1945>
 - MIME-szerű üzenetek átvitele, melyek az átvitt adatokról is tartalmaznak metaadatokat
 - Már nem csupán HTML dokumentumok, hanem tetszőleges média típusú tartalmak átvitelét támogatja
 - Többféle metódus támogatása (GET, HEAD, POST, PUT, DELETE, LINK, ULINK)
 - Hitelesítés (*basic authentication*)
 - ...

Történet (2)

- HTTP/1.1:
 - Roy T. Fielding, James Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, Tim Berners-Lee, *Hypertext Transfer Protocol—HTTP/1.1*, RFC 2068, January 1997.
<http://tools.ietf.org/html/rfc2068>
 - Újdonságok: perzisztens kapcsolatok, tartalomegyeztetés, kifinomultabb gyorsítótárzás, Transfer-Encoding fejlécmező, részleges GET,...
 - Roy T. Fielding, James Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach, Tim Berners-Lee, *Hypertext Transfer Protocol—HTTP/1.1*, RFC 2616, June 1999.
<http://tools.ietf.org/html/rfc2616>
 - Az RFC 2068 javított és korszerűsített változata

A jelenleg aktuális szabvány

- Roy T. Fielding (ed.), Julian F. Reschke (ed.), *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*, RFC 7230, June 2014.
<http://tools.ietf.org/html/rfc7230>
- Roy T. Fielding (ed.), Julian F. Reschke (ed.), *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*, RFC 7231, June 2014.
<http://tools.ietf.org/html/rfc7231>
- Roy T. Fielding (ed.), Julian F. Reschke (ed.), *Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests*, RFC 7232, June 2014.
<http://tools.ietf.org/html/rfc7232>
- Roy T. Fielding (ed.), Yves Lafon (ed.), Julian F. Reschke (ed.), *Hypertext Transfer Protocol (HTTP/1.1): Range Requests*, RFC 7233, June 2014.
<http://tools.ietf.org/html/rfc7233>
- Roy T. Fielding (ed.), Mark Nottingham (ed.), Julian F. Reschke (ed.), *Hypertext Transfer Protocol (HTTP/1.1): Caching*, RFC 7234, June 2014.
<http://tools.ietf.org/html/rfc7234>
- Roy T. Fielding (ed.), Julian F. Reschke (ed.), *Hypertext Transfer Protocol (HTTP/1.1): Authentication*, RFC 7235, June 2014. <http://tools.ietf.org/html/rfc7235>

HTTP kiterjesztések

- Lisa Dusseault (ed.), *HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)*, RFC 4918, June 2007. <http://tools.ietf.org/html/rfc4918>
 - Lehetővé teszi kliensek számára webes tartalomszerkesztési műveletek végrehajtását
- Lisa Dusseault, James M. Snell, *PATCH Method for HTTP*, RFC 5789, March 2010. <http://tools.ietf.org/html/rfc5789>
 - Új metódus (PATCH) definiálása erőforrás részleges módosításához

Biztonságos HTTP

- Eric Rescorla, *HTTP Over TLS*, RFC 2818, May 2000. <http://tools.ietf.org/html/rfc2818>
 - Eredetileg ez a specifikáció írta le a `https` URI-sémát, melyet jelenleg az RFC 7230 definiál
- Rohit Khare, Scott Lawrence, *Upgrading to TLS Within HTTP/1.1*, RFC 2817, May 2000. <http://tools.ietf.org/html/rfc2817>
- Tim Dierks, Eric Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, RFC 5246, August 2008. <https://tools.ietf.org/html/rfc5246>

HTTP/2.0

- Jelenleg fejlesztés alatt (Internet-Draft specifikációk): <https://http2.github.io/>
 - Mike Belshe, Roberto Peon, Martin Thomson (ed.), *Hypertext Transfer Protocol version 2*, draft-ietf-httpbis-http2-16, February 11, 2015.
<https://tools.ietf.org/html/draft-ietf-httpbis-http2-17>
 - Roberto Peon, Herve Ruellan, HPACK – Header Compression for HTTP/2, draft-ietf-httpbis-header-compression-12, February 17, 2015.
<https://tools.ietf.org/html/draft-ietf-httpbis-header-compression-12>

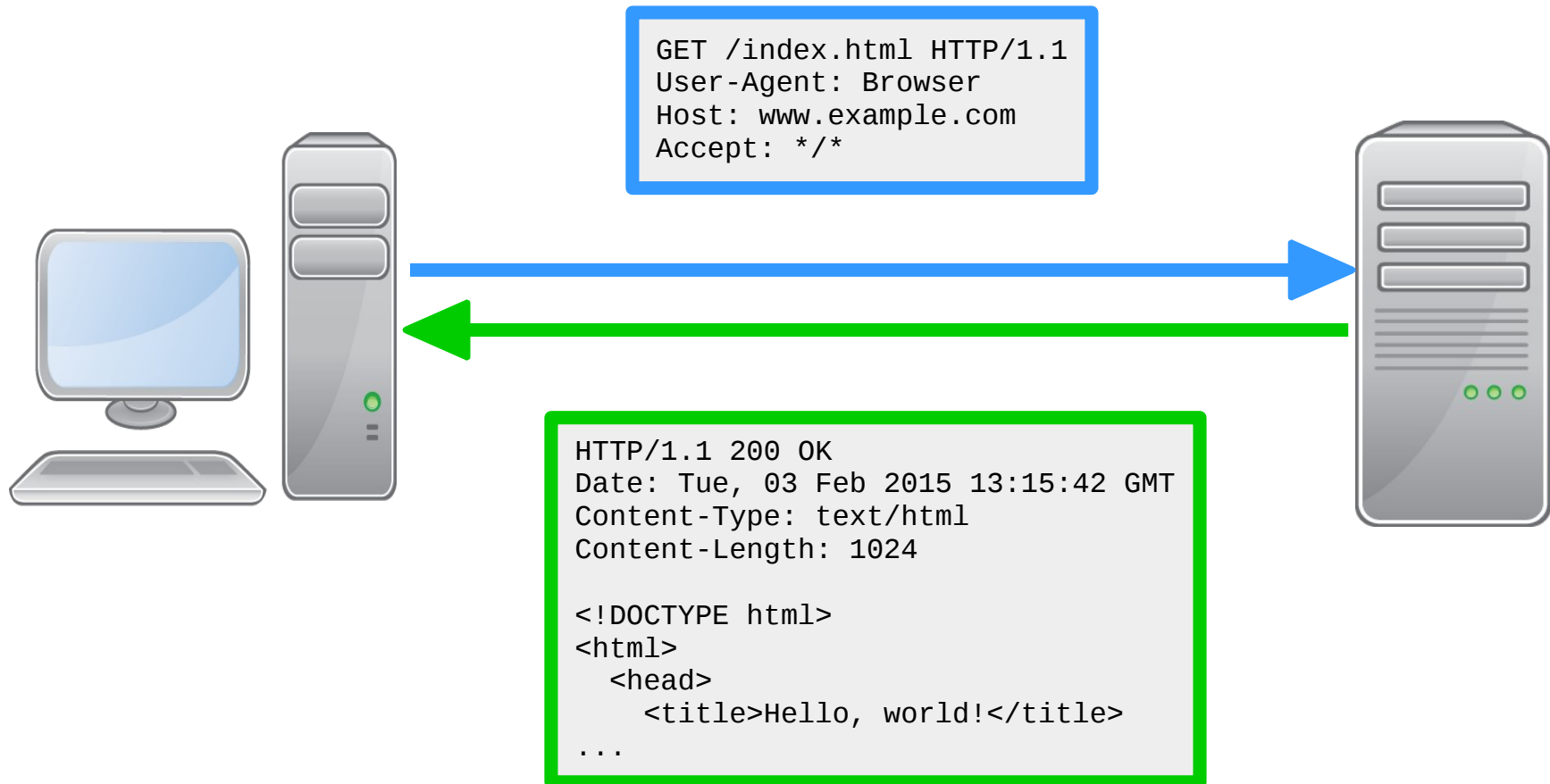
Munkamenetek

- **Munkamenetnek (*session*)** nevezzük kérések és válaszok egy kliens és egy szerver közötti sorozatát
- A HTTP protokoll természeténél fogva állapot nélküli, és nem ad támogatást a munkamenetek nyomkövetéséhez
- **Sütik (*cookies*)** segítségével valósítható meg munkamenetek kezelése

Hasznos dokumentáció

- Mozilla Developer Network – HTTP
<https://developer.mozilla.org/docs/Web/HTTP>
- Apache HTTP Server Documentation
<http://httpd.apache.org/docs/>

Működés



cURL

- Többféle protokollt támogató adatátviteli könyvtár (`libcurl`) és parancssori eszköz (`curl`) <http://curl.haxx.se/>
 - Programozási nyelv: C
 - Platform: Linux, Mac OS X, Windows, ...
 - Licenc: X11 License
- Támogatott protokollok: FTP, HTTP, HTTPS, SCP, SFTP, ...

A cURL használata

- `curl http://www.w3.org/Consortium/`
- `curl http://www.w3.org/Consortium/ \`
`-o About_W3C.html`
- `curl -v http://www.w3.org/Consortium/ \`
`-o About_W3C.html`
- `curl http://www.w3.org/Consortium/ \`
`-o About_W3C.html --dump-header header.txt`
- `curl http://www.w3.org/Consortium/ \`
`-o About_W3C.html --trace-ascii trace.txt`
- `curl --head http://www.w3.org/Consortium/`
- `curl -v --request DELETE \`
`http://www.w3.org/Consortium/`

Online szolgáltatások

- Online szolgáltatások HTTP kérések végrehajtásához:
 - API Kitchen <http://apikitchen.com/>
 - hurl.eu <http://hurl.eu/>
 - Hurl.it <http://www.hurl.it/>
 - Online Curl <http://onlinecurl.com/>
 - ...

Webfejlesztő eszköztár

- **Firefox (Gecko):**

- Firefox Developer Tools <https://developer.mozilla.org/docs/Tools>
- Firebug <http://getfirebug.com/>
- HttpFox <https://addons.mozilla.org/en-US/firefox/addon/httpfox/>
- HTTP logging
https://developer.mozilla.org/en-US/docs/Mozilla/Debugging/HTTP_logging

- **Chromium (Blink), Opera (Blink):**

- Chrome Developer Tools (DevTools) <https://developer.chrome.com/devtools>

- **Internet Explorer (Trident):**

- F12 Developer Tools
<https://msdn.microsoft.com/en-us/library/ie/bg182326%28v=vs.85%29.aspx>

Fogalmak (1)

- **Erőforrás (*resource*):** egy HTTP kérés célja, melyet egy URI azonosít
- **Reprezentáció (*representation*):**
 - Olyan információ, mely egy adott erőforrás múltbeli, jelenlegi vagy kívánt állapotát hivatott tükrözni
 - A protokollon átvihető formában van
 - Reprezentáció metaadatokból és reprezentáció adatokból áll
- **Tartalomegyeztetés (*content negotiation*):**
 - Egy eredet szerver számára egy erőforráshoz több, annak jelenlegi állapotát tükröző reprezentáció állhat rendelkezésre, vagy képes lehet több reprezentáció előállítására
 - A tartalomegyeztetés egy olyan mechanizmus, mely révén kiválasztható egy adott kéréshez legmegfelelőbb reprezentáció
 - Ezt a legmegfelelőbb reprezentációt **kiválasztott reprezentációnak (*selected representation*)** nevezzük

Fogalmak (2)

- **Üzenet (*message*):** a kommunikáció alapegysége
- **Payload:**
 - Jelentése „hasznos teher”
 - Az üzenetben továbbított reprezentációt értjük alatta

Fogalmak (3)

- A kliens és a szerver olyan szerepek, melyeket egy adott kapcsolatnál töltenek be programok. Ugyanaz a program lehet akár kliens és szerver is.
 - **Kliens:** egy szerverrel egy vagy több HTTP kérés küldése céljából kapcsolatot létrehozó program
 - **Szerver:** egy olyan program, mely kapcsolatokat fogad el abból abból a célból, hogy HTTP kéréseket szolgáljon ki HTTP válaszok küldésével

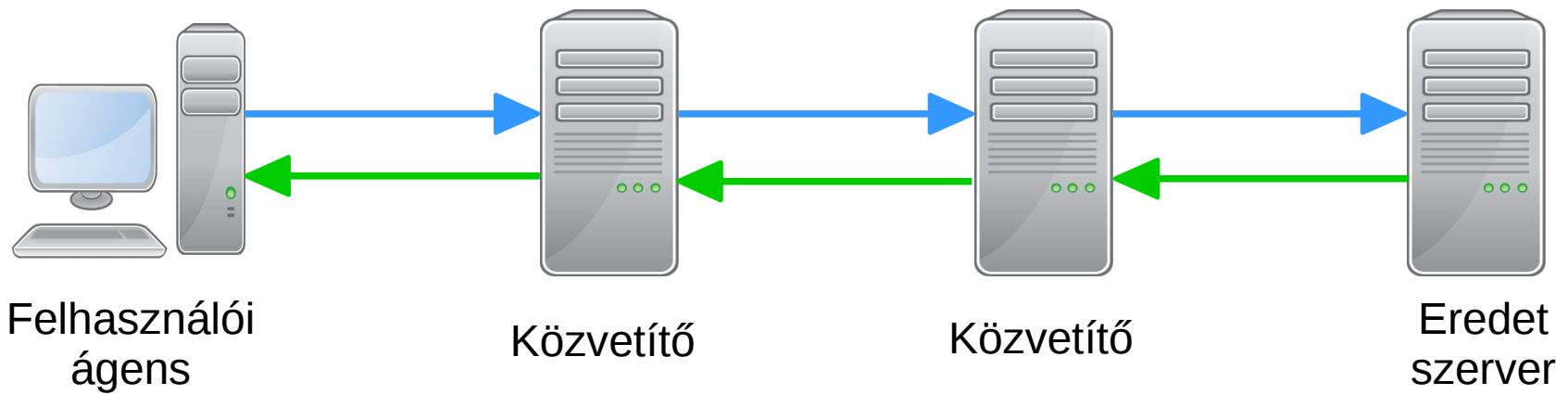
Fogalmak (4)

- **Felhasználói ágens (*user agent*):** egy HTTP kérést kezdeményező program
 - Például böngésző, keresőrobot, parancssoros eszköz (`curl`, `wget`), egyedi alkalmazás, ...
- **Eredet szerver (*origin server*):** azt a programot jelenti, mely hiteles válaszokat tud létrehozni egy adott cél erőforráshoz
- **Küldő (*sender*)/fogadó (*recipient*):** egy adott üzenet elküldő/fogadó program

Fogalmak (5)

- **Közvetítő (*intermediary*):** lehetővé teszi kérések kapcsolatok egy láncán keresztül történő kiszolgálását
 - 3 fajta: *proxy*, átjáró, alagút
 - Bizonyos esetekben egy közvetítő eredet szerver, *proxy*, átjáró és alagút szerepét is betöltheti, a kérések fajtája alapján változtatva a viselkedését

Közvetítők



Fogalmak (6)

- **Proxy:** Üzenettovábbító ágens, mely bizonyos fajta abszolút URI-kra vonatkozó kéréseket fogad
 - A kliens választja ki, általában lokális beállításokon keresztül
 - A proxy lefordítja a kéréseket, mely akár teljesen eltérő alkalmazás-szintű protokollok közötti fordítást igényelhet
- **Átjáró (*gateway*):** eredet szerverként viselkedő közvetítő, mely lefordítja a kéréseket és egy vagy több szerverhez továbbítja őket
 - Úgy fogadja a kéréseket, mintha ő lenne a kért erőforrás eredet szervere, a kliens nincs is tudatában annak, hogy egy átjáróval kommunikál
 - **Fordított proxynak (*reverse proxy*)** is nevezik
- **Alagút (*tunnel*):** két kapcsolat közötti reléként szolgál és az üzeneteket módosítás nélkül továbbítja

Az implementációk sokfélesége

- A felhasználói ágensek és az eredet szerverek is sokfélék lehetnek
 - Felhasználói ágensek: általános célú böngészők, háztartási gépek, szórakoztató elektronikai eszközök, parancssoros programok, mobilalkalmazások, ...
 - Eredet szerverek: webszerverek, konfigurálható hálózati komponensek, irodagépek, autonóm robotok, közlekedési kamerák, ...

A http és a https URI-séma (1)

- Erőforrások azonosítása adott számú TCP porton figyelő eredet szervereken
 - A `https` séma esetén TLS-titkosított kapcsolaton keresztül történik a kommunikáció
- Szintaxis:
 - `'http://' host [':' port] [útvonal] ['?' lekérdezés]`
 - Ha nincs megadva port, akkor az alapértelmezés a 80-as számú TCP port
 - `'https://' host [':' port] [útvonal] ['?' lekérdezés]`
 - Ha nincs megadva port, akkor az alapértelmezés a 443-as számú TCP port
- Az útvonal `'/'` karakterrel kell, hogy kezdődjön, vagy üres

A http és a https URI-séma (2)

- Egy URI eredet szervert a *host* és a *port* komponense azonosítja
 - Az útvonal és az opcionális lekérdezés egy potenciális erőforrást azonosít az eredet szervert névterében
- Ha adott egy `http` vagy `https` URI, abból nem következik, hogy mindig egy HTTP szervert figyel az adott hoston és porton

A http és https URI-séma (3)

- URI-k összehasonlítása:
 - Az üres útvonal ekvivalens a '/' útvonallal
 - A séma és a *host* komponensek összehasonlítása kisbetű-nagybetű érzéketlen módon történik, a többi kisbetű-nagybetű érzékeny módon
 - Minden egyes nem fenntartott karakter ekvivalens a százalékos kódolásával kapott karakterlánccal
- Ekvivalensek például a következő URI-k:
 - `http://www.inf.unideb.hu/`, `http://www.inf.unideb.hu:80/`,
`http://www.inf.unideb.hu`, `http://www.inf.unideb.hu:80`
 - `http://www.inf.unideb.hu/~jeszy/`,
`http://www.inf.unideb.hu/%7Ejeszy/`,
`HTTP://www.INF.UNIDEB.hu/~jeszy/`

A http és https URI-séma (4)

- Teljesen különbözőnek tekintendők a http és a https URI-sémákon keresztül elérhető erőforrások

Üzenetek

- Kétféle üzenet:
 - **Kérés (request)**
 - **Válasz (*response*)**
- Feldolgozásuk oktettek sorozataként kell, hogy történjen

Üzenetek felépítése

- A kérések és válaszok felépítésüket tekintve csupán az első sorokban térnek el
 - Az első sort **kezdősornak** (***start line***) nevezik
- A kezdősor után tetszőleges számú **fejlécmező** (***header field***) következik, melyek mindegyikét CRLF követ
- A fejlécmezők után egy üres sor következik (CRLF)
- Opcionálisan szerepelhet az üzenet végén egy **üzenettörzs** (***message body***)

Kérések (1)

- Egy kérés kezdősora az alábbi felépítésű:
metódus kérés-cél HTTP-verzió CRLF
 - Egyetlen szóköz karakterrel kötelező elválasztani a sorban a komponenseket
- A *kérés-cél* a cél erőforrást azonosítja, melyre a kérés vonatkozik

Kérések (2)

- Példa:

```
> GET /licenses/ HTTP/1.1
> Host: www.gnu.org
> User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:35.0)
  Gecko/20100101 Firefox/35.0
> Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
> Accept-Encoding: gzip, deflate
> Accept-Language: hu-hu,hu;q=0.8,en-US;q=0.5,en;q=0.3
> Connection: keep-alive
> Cache-Control: max-age=0
>
```

Kérések (3)

- A *kérés-cél* négyféle módon jelenhet meg, a metódustól és attól függően, hogy a kérés proxyhoz van-e intézve:

(1) A CONNECT és OPTIONS metódusok kivételével közvetlenül egy eredet szerverhez intézett kéréseknél az alábbi forma kötelező:

útvonal ['?' *lekérdezés*]

- Ha a cél URI nem tartalmaz útvonalat, akkor a fenti alakban '/' kötelező a helyén
- A cél URI *host* és *port* komponense a Host fejlécmezőben kerül továbbításra
- Példa:
 - GET /copyleft/gpl.html HTTP/1.1
 - Host: www.gnu.org

Kérések (4)

- A *kérés-cél* négyféle módon jelenhet meg, a metódustól és attól függően, hogy a kérés proxyhoz van-e intézve:

(2) A CONNECT és szerver-szintű OPTIONS

metódusok kivételével egy proxyhoz intézett kéréseknél az alábbi forma kötelező:

séma ' : ' *hierarchikus-rész* [' ? ' *lekérdezés*]

- Példa:

- GET http://www.grumpycats.com/ HTTP/1.1

Kérések (5)

- A *kérés-cél* négyféle módon jelenhet meg, a metódustól és attól függően, hogy a kérés proxyhoz van-e intézve:

(3) Ha a CONNECT metódussal kerül létrehozásra alagút egy vagy több proxyn keresztül, az alábbi forma kötelező:

host [' : ' *port*]

- Példa:

– CONNECT cache.lib.unideb.hu:5128 HTTP/1.1

Kérések (6)

- A *kérés-cél* négyféle módon jelenhet meg, a metódustól és attól függően, hogy a kérés proxyhoz van-e intézve:

(4) Kizárólag OPTIONS metódus esetén megadható
' * ' (lásd az OPTIONS metódusnál leírtakat)

- Példa:

– OPTIONS * HTTP/1.1

Válaszok (1)

- Az üzenet első sorát **állapotsornak** (*status line*) nevezzük, felépítése az alábbi:
HTTP-verzió állapotkód indok_frázis CRLF
 - Egyetlen szóköz karakterrel kötelező elválasztani a sorban a komponenseket
- Az állapotkód egy háromjegyű decimális egész szám, melyet az állapotkód tömör szöveges leírása követ
- Példák állapotsorra:
 - HTTP/1.1 200 OK
 - HTTP/1.1 404 Not Found

Válaszok (2)

- Példa:

```
< HTTP/1.1 200 OK
< Date: Fri, 23 Jan 2015 21:04:54 GMT
< Server: Apache/2.2.14
< Content-Location: licenses.html
< Vary: negotiate,accept-language,accept-encoding
< TCN: choice
< Accept-Ranges: bytes
< Cache-Control: max-age=0
< Expires: Fri, 23 Jan 2015 21:04:54 GMT
< Content-Encoding: gzip
< Content-Length: 9349
< Connection: close
< Content-Type: text/html
< Content-Language: en
<
< <gzip tömörített adatok>
```

Fejlécmezők (1)

- Az alábbi felépítésűek:

mezőnév ' : ' érték

- Mezőnév:

- Legalább egy karakterből áll, bizonyos US-ASCII karakterek megengedettek benne (betű és számjegy karakterek, '!', '#', '\$', ...)
- Kezelésük kisbetű-nagybetű érzéketlen módon történik

- Érték:

- Nyomtatható US-ASCII karakterek, szóköz és vízszintes tabulátor karakterek alkotják
- Előtte egyetlen szóköz karakter elhelyezése ajánlott
- Az érték előtti és utáni szóköz és vízszintes tabulátor karakterek figyelmen kívül hagyása

Fejlécmezők (2)

- Sorrendjük nem lényeges
- Egy üzenetben egynél több fejlécmezőnél is előfordulhat ugyanaz a mezőnév, de csak akkor, ha a fejlécmező értéke definíció szerint egy olyan lista, melyben a vessző karakter az elválasztó
 - Ilyenkor az azonos nevű mezők értékeit egy listává fűzheti össze a fogadó
 - Egy további speciális kivétel a Set - Cookie fejlécmező, mely a gyakorlatban gyakran többször is megjelenik a válaszban

Fejlécmezők (3)

- Fajtaik:
 - Reprezentáció fejléc mezők
 - *Payload* fejlécmezők
 - Kérés fejlécmezők
 - Válasz fejlécmezők
- A Cache-Control fejlécmező kérésekben és válaszokban is előfordulhat

Fejlécmezők (4)

- **Reprezentáció fejléc mezők:**
 - A reprezentációról szolgáltatnak metaadatokat
 - Ha az üzenet tartalmaz *payload*-törzset, akkor azt írják le, hogy hogyan kell értelmezni a benne mellékelte reprezentációt
 - HEAD metódus esetén azt a reprezentációt írják le, melyet egy GET kérés esetén tartalmazna a *payload*-törzs
 - A következő fejlécmezők tartoznak ide:
 - Content-Type
 - Content-Encoding
 - Content-Language
 - Content-Location

Fejlécmezők (5)

- ***Payload* fejlécmezők:** a *payload*-ot írják le
 - Content-Length
 - Content-Range
 - Trailer
 - Transfer-Encoding

Fejlécmezők (6)

- **Kérés fejlécmezők:**

- A kliensek által kérésekben elküldött fejlécmezők, melyek a kérés környezetéről szolgáltatnak információkat, feltételelessé teszik a kérést, preferált formátumokat javasolnak a válaszhoz, hitelesítő adatokat szolgáltatnak vagy a kérés feldolgozását módosítják
- A következő fejlécmezők tartoznak ide: Accept, Accept - Charset, Accept - Encoding, Accept - Language, Authorization, Cache - Control, Expect, From, Host, If - Match, If - None - Match, If - Modified - Since, If - Unmodified - Since, If - Range, Max - Forwards, Pragma, Proxy - Authorization, Range, Referer, TE, User - Agent

Fejlécmezők (7)

- **Válasz fejlécmezők:**

- Az állapotsorban megjelenőknél túl további információkat adhat vissza általuk a szerver a válaszról
 - Például magáról a szerverről, a cél erőforrás eléréséről, kapcsolódó erőforrásokról, ...
- A következő fejlécmezők tartoznak ide: Accept - Ranges, Age, Allow, Cache-Control, Date, ETag, Last-Modified, Location, Proxy-Authenticate, Retry-After, Server, Vary, Warning, WWW-Authenticate

Fejlécmezők (8)

- A specifikációban definiáltakon túl sok más specifikáció definiál fejlécmezőket
- A fejlécmezők regisztrálását az IANA végzi
 - Lásd: *Message Headers*
<http://www.iana.org/assignments/message-headers/message-headers.xhtml>

A User-Agent fejlécmező (1)

- A kérést kezdeményező felhasználói ágensről tartalmaz információkat
- Felhasználható tartalomegyeztetéshez és a böngészőre vagy operációs rendszerre vonatkozó elemzésekhez
- A felhasználói ágens számára ajánlott minden kérésben elküldeni
- Értéke egy vagy több termékazonosító, melyek mindegyikét 0 vagy több megjegyzés követheti
 - Minden egyes termékazonosító egy névből és egy opcionális verziószámból áll
 - Megjegyzések megadása ' (' és ') ' határolók között

A User-Agent fejlécmező (2)

- **Curl:** curl/7.35.0
- **Firefox:** Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:35.0) Gecko/20100101 Firefox/35.0
- **Chromium:** Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/40.0.2214.94 Chrome/40.0.2214.94 Safari/537.36
- **Opera:** Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.94 Safari/537.36 OPR/27.0.1689.66
- **Internet Explorer:** Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko

A User-Agent fejlécmező (3)

- További hasznos címek:
 - UserAgentString.com
<http://www.useragentstring.com/>
 - Gecko user agent string reference
https://developer.mozilla.org/en-US/docs/Web/HTTP/Gecko_user_agent_string_reference

A q paraméter

- Több kérés fejlécmező (Accept, Accept - Charset, Accept - Encoding, Accept - Language) használ egy q nevű paramétert, melynek értéke egy relatív súly, és amely az adott fajtájú reprezentáció elfogadhatóságának mértékét fejezi ki
- Értéke egy 0 és 1 közötti valós szám, ahol a decimális pont karakter jobb oldalán legfeljebb 3 számjegy megengedett
 - 0 jelzi a nem elfogadható tartalmat
 - 0.001 jelzi a legkevésbé preferált tartalmat
 - 1 jelzi a leginkább preferált tartalmat
- 1 az alapértelmezett érték

Az Accept fejlécmező (1)

- A felhasználói ágens számára a válaszban elfogadható média típusok jelzésére szolgál
- Értéke média tartományok egy listája, melyben minden média tartományt 0 vagy több média típus paraméter (például charset) követhet, valamint egy opcionális q paraméter
- Média tartomány:
 - *típus/altípus*: a megfelelő média típust jelenti
 - *típus/**: a típus összes altípusát jelenti
 - **/**: az összes média típust jelenti
- Accept fejlécmező nélküli kérés azt jelenti, hogy a felhasználói ágens tetszőleges média típust elfogad a válaszban

Az Accept fejlécmező (2)

- Firefox (Gecko):
 - Accept: text/html,
application/xhtml+xml,
application/xml;q=0.9, */*;q=0.8

Média tartomány	q
text/html	1
application/xhtml+xml	1
application/xml	0.9
/	0.8

Az Accept fejlécmező (3)

- Chromium (Blink), Opera (Blink):
 - Accept: text/html,
application/xhtml+xml,
application/xml;q=0.9, image/webp,
/;q=0.8

Média tartomány	q
text/html	1
application/xhtml+xml	1
image/webp	1
application/xml	0.9
/	0.8

Az Accept fejlécmező (4)

- Internet Explorer (Trident):
 - Accept: text/html,
application/xhtml+xml, */*

Média tartomány	q
text/html	1
application/xhtml+xml	1
/	1

Üzenettörzs (1)

- A kéréshez vagy válaszhoz tartozó *payload-törzs*et hordozza
 - A *payload-törzs* vagy a Transfer-Encoding fejlécmezőben jelzett módon kódolt *payload-törzs* alkotja
 - Tetszőleges oktettek alkotják

Üzenettörzs (2)

- Jelenlétét a Content - Length vagy a Transfer - Encoding fejlécmező jelzi kérésekben
 - GET, HEAD, DELETE, CONNECT és OPTIONS kérésekben az üzenettörzsnek nincs a specifikáció által meghatározott jelentése
 - TRACE kérésben tilos üzenettörzs küldése
- Válaszban a kérésben használt metódustól és az állapotkódtól függ, hogy megjelenik-e
 - HEAD kérésekre adott válaszok soha sem tartalmazznak üzenettörzset, akkor sem, ha a fejlécmezők erre utalnak
 - CONNECT kérésre adott 2xx állapotkódú válaszok nem tartalmazznak üzenettörzset
 - 1xx, 204 (No Content) és 304 (Not Modified) állapotkód esetén nincs üzenettörzs
 - Az összes többi válasz tartalmaz üzenettörzset, bár az lehet 0 hosszúságú

Átviteli kódolás (1)

- Az **átviteli kódolás** (*transfer coding*) a *payload*-törzsre alkalmazott olyan kódolás, mely biztosítja annak biztonságos hálózati átvitelét
- Az alkalmazott átviteli kódolás az üzenet tulajdonsága, nem pedig az átvitt reprezentációé

Átviteli kódolás (2)

- A HTTP/1.1 specifikáció az alábbi átviteli kódolásokat definiálja:
 - **chunked**: adatok továbbítása darabokban
 - **compress**: a Unix rendszerekben rendelkezésre álló compress tömörítőprogram formátuma
 - **deflate**: DEFLATE tömörítésű adatok zlib formátumban
 - **gzip**: GZIP tömörítés
 - **x-compress**: elavult, a compress szinonimája
 - **x-gzip**: elavult, a gzip szinonimája
- A specifikáció aktuális kiadása már nem támogatja az alábbi átviteli kódolást:
 - **identity**: azt jelzi, hogy nem kerül alkalmazásra átviteli kódolás

Átviteli kódolás (3)

- Kérésben a TE fejlécmező szolgál a kliens által a chunked mellett elfogadott további átviteli kódolások jelzésére
 - Tilos a chunked átviteli kódolás megadása, mivel az minden HTTP/1.1 üzenetfogadó számára elfogadható kell, hogy legyen
 - Példa:
TE: deflate
- Válaszban a Transfer-Encoding fejlécmező szolgál annak jelzésére, hogy az üzenettörzs létrehozása során milyen átviteli kódolások kerültek alkalmazásra a *payload*-törzsre
 - Értéke alapján végezhető el a dekódolás
 - Példa:
Transfer-Encoding: gzip, chunked

Átviteli kódolás (4)

- A TE és Transfer-Encoding fejlécmezőkben használható kódolásokat az IANA adminisztrálja
 - A kódolások azonosítóinak kezelése kisbetű-nagybetű érzéketlen módon történik
 - Lásd: *HTTP Transfer Coding Registry*
<http://www.iana.org/assignments/http-parameters/http-parameters.xhtml#transfer-coding>

Átviteli kódolás (5)

- A chunked átviteli kódolás:
 - Lehetővé teszi dinamikusan előállított tartalom átvitelét, melynél nem ismert előre a teljes méret
 - Minden HTTP/1.1 üzenetfogadó képes kell, hogy legyen feldolgozására
 - A küldő számára tilos egynél többszöri alkalmazása az üzenettörzsre
 - Ha a chunked átviteli kódolástól eltérő bármely átviteli kódolás kerül alkalmazásra kérés *payload*-törzsére, akkor utolsó kódolásként a chunked átviteli kódolást kell alkalmazni

Átviteli kódolás (6)

- A chunked átviteli kódolás:
 - Minden egyes darab elején egy külön sorban hexadecimálisan kell megadni a méretét, azaz az oktettek számát, a sor végét CRLF jelzi
 - A darabot alkotó adatok végét CRLF jelzi,
 - A kódolt adatok végét egy 0 méretű darab jelzi
 - Ez követően megadhatóak fejlécmezők, mivel bizonyos mezők értéke csak a tartalom előállítás után határozható meg
 - Végül egy üres sor (CRLF) következik

Átviteli kódolás (7)

- Példa a chunked átviteli kódolás használatára:

```
< HTTP/1.1 200 OK
< Date: Sat, 24 Jan 2015 15:32:04 GMT
< Server: Apache/2.4.7 (Ubuntu)
< Content-Type: text/plain
< Transfer-Encoding: chunked
<
< 1a
< abcdefghijklmnopqrstuvwxyz
< a
< 0123456789
< 0
<
```

Üzenettörzs hossza

- Ha egy üzenetben jelen van a Transfer-Encoding fejlécmező és a chunked az utolsó átviteli kódolás, akkor az üzenettörzs hossza a feldarabolt adatok beolvasásával és dekódolásával határozható meg
- Ha egy üzenetben jelen van a Content-Length fejlécmező és nincs Transfer-Encoding fejlécmező, akkor decimális értéke oktettekben adja meg az üzenettörzs várható hosszát
- Ha egy üzenet Transfer-Encoding és Content-Length fejlécmezőt is tartalmaz, akkor az utóbbit figyelmen kívül kell hagyni

Payload szemantika (1)

- A *payload* célját kérésekben a metódus szemantikája határozza meg
 - Például POST metódusnál a cél erőforrás által feldolgozandó adatokat ábrázol
- A *payload* célját válaszokban a metódus és az állapot kód határozza meg
 - Például hiba állapot kód esetén a *payload* általában a hibát leírását szolgáltatja

Payload szemantika (2)

- Egy HTTP üzenethez tartozó reprezentáció adatokat az üzenet *payload* törzse tartalmazza
- A reprezentáció adatok formátumát és kódolását a reprezentáció metaadat fejlécmezők határozzák meg
 - A `Content-Type` fejlécmező jelzi a reprezentáció média típusát
 - Például:
 - `Content-Type: text/html; charset=utf-8`
 - A `Content-Encoding` fejlécmező jelzi a reprezentációra alkalmazott tartalomkódolás(oka)t
 - Dekódolás révén kaphatók meg a `Content-Type` fejlécmezőben jelzett média típusú adatok

Tartalomkódolás (1)

- A **tartalomkódolás** (*content coding*) egy reprezentáció veszteségmentes tömörítésére vagy egyéb átalakítására szolgáló kódolás
 - A reprezentáció gyakran eleve tömörített formában van tárolva, így közvetlenül átvihető
- Az alkalmazott tartalomkódolás(ok) a reprezentáció tulajdonsága(i)
 - A reprezentációt leíró metaadatok is a kódolt formájára vonatkoznak

Tartalomkódolás (2)

- A HTTP/1.1 specifikáció az alábbi átviteli kódolásokat definiálja:
 - **compress**: a Unix rendszerekben rendelkezésre álló compress tömörítőprogram formátuma
 - **deflate**: DEFLATE tömörítésű adatok `zlib` formátumban
 - **gzip**: GZIP tömörítés
 - **x-compress**: elavult, a `compress` szinonimája
 - **x-gzip**: elavult, a `gzip` szinonimája

Tartalomkódolás (3)

- Kérésekben az `Accept-Encoding` fejlécmező szolgál a felhasználói ágens által a válaszokban elfogadott tartalomkódolások jelzésére
 - Példa:
 - `Accept-Encoding: compress, gzip`
 - `Accept-Encoding: *`
- Válaszokban a `Content-Encoding` fejlécmező szolgál a reprezentációra alkalmazott tartalomkódolás(ok) jelzésére
 - Értéke alapján végezhető el a dekódolás
 - Példa:
 - `Content-Encoding: gzip`

Tartalomkódolás (4)

- Az Accept-Encoding és Content-Encoding fejlécmezőkben használható kódolásokat az IANA adminisztrálja
 - A kódolások azonosítóinak kezelése kisbetű-nagybetű érzéketlen módon történik
 - Lásd: *HTTP Content Coding Registry*
<http://www.iana.org/assignments/http-parameters/http-parameters.xhtml#content-coding>

Biztonságos metódusok

- **Biztonságosnak** nevezzük az információk lekérdezésére szolgáló metódusokat, melyeknek elvileg nincs szerveroldali mellékhatása, azaz nem változtatják meg a szerver állapotát
 - Biztonságosak a GET, HEAD, OPTIONS és TRACE metódusok
 - Nem biztonságosak a CONNECT, DELETE, POST és PUT metódusok
- A gyakorlatban akár például a GET metódusnak is lehet szerveroldali mellékhatása
 - Ha van is, nem tehető érte felelőssé a felhasználó
 - Lehetnek ártalmatlan mellékhatások, mint például egy számláló értékének növelése

Idempotens metódusok

- **Idempotens**nek nevezzük azokat a metódusokat, melyek egymás után többszöri végrehajtása ugyanazt a kívánt hatást eredményezi, mint egyetlen végrehajtás
 - A biztonságos metódusok definíció szerint idempotensek
 - Idempotensek a DELETE és PUT metódusok is
 - A CONNECT és a POST metódus nem idempotens
- A hangsúly azon van, hogy a szerver állapota ugyanaz marad a többszöri végrehajtás esetén
 - Az egyes kérések esetén eltérő lehet az állapot kód!
- Ha kommunikációs hiba lép fel egy idempotens metódus esetén azt megelőzően, hogy a kliens be tudná olvasni a szerver válaszát, akkor a kérés automatikusan megismételhető

Gyorsítótárazható metódusok

- **Gyorsítótárazható**nak nevezünk azokat a metódusokat, melyekre adott válaszok eltárolhatóak jövőbeli újrafelhasználás céljából
 - Gyorsítótárazhatóak a GET, HEAD és POST metódusok
 - Nem gyorsítótárazhatóak a CONNECT, DELETE, OPTIONS, PUT és TRACE metódusok

HTTP metódusok (1)

- A HTTP/1.1 specifikáció által definiált metódusok:
 - GET
 - HEAD
 - POST
 - PUT
 - DELETE
 - CONNECT
 - OPTIONS
 - TRACE
- Egy metódus jelentését tovább finomíthatja bizonyos fejlécmezők jelenléte a kérésben
- Általános célú szerverek a GET és a HEAD metódusokat kell, hogy támogassák, a többi metódus opcionális

HTTP metódusok (2)

Metódus	Biztonságos	Idempotens
CONNECT		
DELETE		X
GET	X	X
HEAD	X	X
OPTIONS	X	X
POST		
PUT		X
TRACE	X	X

HTTP metódusok (3)

- A metódusok neve kisbetű-nagybetű érzékeny
- A specifikáción kívül további metódusok is szabványosításra kerültek
 - Lásd: *Hypertext Transfer Protocol (HTTP) Method Registry*
<http://www.iana.org/assignments/http-methods/http-methods.xhtml>

HTTP metódusok (4)

- Az 501 (Not Implemented) állapotkód szolgál annak jelzésére a válaszban, hogy egy eredet szerver nem ismer fel vagy nem implementál egy metódust
- A 405 (Method Not Allowed) állapotkód szolgál annak jelzésére a válaszban, hogy egy eredet szerver ismeri ugyan a metódust, de nem engedélyezi az erőforráshoz

GET

- Az adott URI által azonosított erőforrás egy aktuális kiválasztott reprezentációjának lekérdezésére szolgál
 - A Range fejlécmező révén részleges lekérdezés is történhet (*partial* GET)
 - A válasz gyorsítótárazható
 - Egy gyorsítótár felhasználhatja későbbi GET és HEAD kérésekhez, kivéve, ha a Cache-Control fejlécmező másként rendelkezik

HEAD

- Megegyezik a GET metódussal, azonban a válasz nem tartalmazhat üzenettörzset
 - Úgy kérdezhetőek le általa metaadatok a kiválasztott reprezentációról, hogy nem történik meg a reprezentáció adatok átvitele
 - A válasz gyorsítótárazható
 - Egy gyorsítótár felhasználhatja későbbi HEAD kérésekhez, kivéve, ha a `Cache-Control` fejlécmező másként rendelkezik

POST (1)

- Annak kérésére szolgál, hogy a cél erőforrás dolgozza fel a kérésben mellékelte reprezentációt a saját szemantikájának megfelelően
- Felhasználási lehetőségek:
 - Egy létező erőforrás annotálása
 - Adatok (például űrlap adatok) továbbítása egy adatfeldolgozó folyamat számára
 - Egy üzenet postázása egy hírcsoportba, levelezési listára vagy blogra
 - Egy új erőforrás létrehozása
 - Adatok hozzáfűzése egy erőforrás létező reprezentációhoz
 - ...

POST (2)

- A válasz szinte bármely, a szemantikának megfelelő állapotkódot tartalmazhatja
 - A 206 (Partial Content), 304 (Not Modified) és 416 (Range Not Satisfiable) állapotkódok használata nem engedélyezett
- Ha a szerveren egy vagy több erőforrás jött létre a kérés sikeres végrehajtásának eredményeként, akkor állapotkódként a 201 (Created) ajánlott
 - Ilyenkor a válasznak tartalmaznia ajánlott egy, a kérés állapotát leíró és az új erőforrásra hivatkozó reprezentációt, valamint egy Location fejlécmezőt
- A válasz csak akkor gyorsítótárazható, ha explicit frissességi információt tartalmaz

PUT (1)

- Annak kérésére szolgál, hogy a szerver hozza létre vagy helyettesítse a cél erőforrás állapotát a mellékelt reprezentáció által definiált állapottal
- Sikeres végrehajtása egy adott reprezentációra arra enged következtetni, hogy egy következő GET ugyanarra a cél erőforrásra egy ekvivalens reprezentációt eredményez egy 200 (OK) állapotkódú válaszban
- Válasz állapotkód:
 - Ha a cél erőforrásnak nincs aktuális reprezentációja és a PUT sikeresen hoz létre egyet, akkor az állapotkód 201 (Created) kell, hogy legyen
 - Egy aktuális reprezentáció sikeres módosítás esetén 200 (OK) vagy 204 (No Content) kell, hogy legyen az állapotkód

PUT (2)

- Alapvető eltérés a POST és a PUT metódus között a mellékelt reprezentáció céljában van:
 - POST kérésben a mellékelt reprezentációt a cél erőforrás kezeli
 - PUT kérésben a mellékelt reprezentáció a cél erőforrás állapotát helyettesíti
- A válasz nem gyorsítótárazható

DELETE

- Annak kérésére szolgál, hogy az eredet szerver törölje a cél erőforrás és aktuális funkcionalitása közötti kapcsolatot
 - Ha a cél erőforrásnak egy vagy több aktuális reprezentációja van, akkor ezeket az eredet szerver vagy megsemmisíti, vagy nem, a kapcsolódó tárterület vagy felszabadításra kerül, vagy nem, ...
 - Az erőforrás az eredet szerver általi megvalósításától függ, hogy mi történik
- Sikeres végrehajtás esetén a válaszban állapotkódként az alábbiak ajánlottak:
 - 200 (OK), ha a művelet elrendelésre került és a válasz tartalmaz egy, az állapotot leíró reprezentációt
 - 202 (Accepted), ha a művelet valószínűleg sikeres lesz, de még nem került elrendelésre
 - 204 (No Content), ha a művelet elrendelésre került, de a szerver nem ad vissza további információt
- A válasz nem gyorsítótárazható

CONNECT

- Annak kérésére szolgál, hogy a fogadó hozzon létre egy alagutat a kérésben célként azonosított eredet szerverhez
 - Az alagút feladata a lezárásáig a csomagok továbbítása mindkét irányban
- Kizárólag proxyhoz intézett kérésekben használható
- A válasz nem gyorsítótárazható

OPTIONS (1)

- A cél erőforrás kommunikációs opcióiról (az eredet szerveren vagy egy közbülső közvetítőnél) rendelkezésre álló információk lekérdezésére szolgál
- Cél erőforrásként megadható ' * ', mely általában a szerveret jelenti, nem pedig egy adott erőforrást
- Sikeres válaszban a szerver el kell, hogy küldjön minden olyan fejlécmezőt, mely a szerver által implementált és a cél erőforrásra alkalmazható opcionális lehetőségeket jelezhet
 - Például az Allow fejlécmező a cél erőforrás által támogatott metódusok jelzésére szolgál
- A válaszok nem gyorsítótárazhatóak

OPTIONS (2)

- Példa:

```
> OPTIONS /foundation/contact.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: apache.org
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Thu, 22 Jan 2015 12:42:08 GMT
< Server: Apache/2.4.7 (Ubuntu)
< Allow: GET,HEAD,POST,OPTIONS,TRACE
< Cache-Control: max-age=3600
< Expires: Thu, 22 Jan 2015 13:42:08 GMT
< Content-Length: 0
< Content-Type: text/html
<
```

TRACE (1)

- A kérés visszaküldését eredményezi
 - A végső fogadó kell, hogy visszaküldje a kapott üzenetet egy 200-as állapotkódú üzenet törzsében
- A kliens egy 200-as állapotkódú üzenetben kapja vissza a kérést, melyet az üzenettörzs tartalmaz
 - A Content - Type fejlécmező értéke message/http
- A válasz nem gyorsítótárazható

TRACE (2)

- Példa:

```
> TRACE / HTTP/1.1
> User-Agent: curl/7.35.0
> Host: www.apache.org
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Thu, 22 Jan 2015 13:00:25 GMT
< Server: Apache/2.4.11 (Unix) OpenSSL/1.0.11
< Transfer-Encoding: chunked
< Content-Type: message/http
<
< 50
< TRACE / HTTP/1.1
< User-Agent: curl/7.35.0
< Host: www.apache.org
< Accept: */*
<
<
< 0
<
```

HTTP állapotkódok (1)

- Háromjegyű decimális egész számok
- Az első számjegy a válasz fajtáját határozza meg
- A klienseknek nem szükséges megérteniük minden regisztrált állapotkód jelentését
 - Kötelező azonban az állapotkód fajtájának megértése az első számjegy alapján
 - Nem felismert állapotkódot az $x00$ állapotkóddal ekvivalensként kell kezelni, ahol x a nem felismert állapotkód első számjegye
 - Ilyenkor azonban tilos a válasz gyorsítótárazása

HTTP állapotkódok (2)

- Az állapotkódok kiterjeszthetők
 - A specifikációban definiáltakon túl továbbiak is definiálhatók
- Az állapotkódok regisztrálását az IANA végzi
 - Lásd: *Hypertext Transfer Protocol (HTTP) Status Code Registry*
<http://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml>

HTTP állapotkódok fajtái

- **1xx: Informational** (tájékoztató)
 - A végső választ megelőző ideiglenes választ jelez
 - A kérés csupán az állapotsorból és opcionális fejlécekből áll, a végét egy üres sor jelzi
- **2xx: Success** (siker)
 - A szerver megkapta, megértette és elfogadta a kérést
- **3xx: Redirection** (átirányítás)
 - A kérés kiszolgálásához a felhasználói ágens további művelet kell, hogy végrehajtson, ezt automatikusan elvégezheti
- **4xx: Client Error** (kliens hiba)
- **5xx: Server Error** (szerver hiba)

Kliens és szerver hibák

- A HEAD kérésekre adott válaszok kivételével a 4xx és 5xx állapotkódú válaszoknak tartalmazniuk kell egy reprezentációt, mely a hiba magyarázatát tartalmazza, valamint azt, hogy a hibát okozó körülmények átmenetiek vagy állandóak

Fontosabb HTTP állapotkódok (1)

Állapotkód	Indok	Leírás
100	Continue (Folytatás)	<p>A kliens több részletben küldheti el a kérést a szervernek, melyre a végső választ megelőzően több 100 állapotkódú választ kaphat.</p> <p>Az állapotkód azt jelzi, hogy a szerver megkapta a kérés első részét és hogy a kliensnek folytatni kell a kérést. Ha a kliens befejezte a kérést, akkor ezt a választ figyelmen kívül hagyhatja.</p>

Fontosabb HTTP állapotkódok (2)

Állapotkód	Indok	Leírás
200	OK	A kérés feldolgozása sikeres volt. A válaszban visszaadott <i>payload</i> a kérésben használt metódustól függ. Például GET kérés esetén a válasz a kért erőforrás egy reprezentációját tartalmazza.
201	Created (Létrehozva)	A kérés feldolgozása sikeres volt, eredményeként egy vagy több új erőforrás került létrehozásra.
202	Accepted (Elfogadva)	A kérés elfogadásra került, de a feldolgozása nem fejeződött be.
204	No Content (Nincs tartalom)	A kérés feldolgozása sikeres volt, de a válasz nem tartalmaz üzenettörzset.
206	Partial Content (Részleges tartalom)	Sikeres volt a részleges GET kérés feldolgozása.

Fontosabb HTTP állapotkódok (3)

Állapotkód	Indok	Leírás
300	Multiple Choices (Több alternatíva)	Az erőforrásnak több reprezentációja van. HEAD kérés kivételével a válasz <i>payload</i> -ként kell, hogy tartalmazzon egy olyan listát, amelyből a kliens kiválaszthatja a számára megfelelőt.
301	Moved Permanently (Véglegesen áthelyezve)	Az erőforrás URI-ja véglegesen megváltozott, további kérésekben az új URI-t kellene használni.
302	Found (Találat)	Az erőforrás URI-ja ideiglenesen megváltozott.
303	See Other (Lásd máshol)	A válasz egy másik URI alatt található, a GET metódussal kapható meg.
304	Not Modified (Nem módosult)	Feltételes GET metódus esetén azt jelzi, hogy az erőforrás nem módosult.
307	Temporary Redirect (Ideiglenes átirányítás)	Az erőforrás URI-ja ideiglenesen megváltozott. Az új kérésben tilos az eredeti kérésétől eltérő metódust használni.

Fontosabb HTTP állapotkódok (4)

Állapotkód	Indok	Leírás
400	Bad Request (Hibás kérés)	A szerver nem tudja vagy nem fogja feldolgozni a kérést valamilyen kliens hiba miatt (például a kérés szintaktikailag hibás).
401	Unauthorized (Jogosulatlan)	A kérés hitelesítést igényel.
403	Forbidden (Tiltva)	A szerver visszautasítja a kérés kiszolgálását. Ha a kérésben hitelesítő adatok vannak, akkor azok nem megfelelőek a jogosultsághoz.
404	Not Found (Nem található)	A szerver nem találja a cél erőforrás aktuális reprezentációját vagy nem kívánja azt nyilvánosságra hozni.
405	Method Not Allowed (Nem engedélyezett metódus)	A cél erőforráshoz nem engedélyezett a metódus.
406	Not Acceptable (Nem elfogadható)	Nem állítható elő a kívánt jellemzőkkel rendelkező reprezentáció és a szerver nem hajlandó egy alapértelmezett reprezentációt szolgáltatni (tartalomjegyzétek).

Fontosabb HTTP állapotkódok (5)

Állapotkód	Indok	Leírás
500	Internal Server Error (Belső serverhiba)	Olyan váratlan esemény történt a szerveren, amely miatt nem szolgálható ki a kérés.
501	Not Implemented (Nincs megvalósítva)	A szerver nem támogatja a kérés kiszolgálásához szükséges funkcionalitást (metódust).
503	Service Unavailable (A szolgáltatás nem érhető el)	A szerver jelenleg nem képes a kérés kiszolgálására (például ideiglenes túlterhelés vagy karbantartás miatt).

Tartalomegyeztetés (1)

- Egy eredet szerver a több, a cél erőforrás aktuális állapotát tükröző reprezentációval rendelkezhet vagy előállítására lehet képes
 - Például a reprezentációk formátuma, nyelve vagy kódolása lehet eltérő
 - A felhasználók és a felhasználói ágensek képességei, jellemzői vagy igényei is eltérőek
- A **tartalomegyeztetés** (***content negotiation***) egy válaszhoz legalkalmasabb reprezentáció kiválasztásának mechanizmusát jelenti

Tartalomegyeztetés (2)

- A specifikáció a tartalomegyeztetés két formáját definiálja:
 - **Proaktív (*proactive*)**: a szerver választja ki a reprezentációt a felhasználói ágens kifejezett preferenciái alapján
 - Ezt hívják **szerver-vezérelt (*server-driven*)** tartalomegyeztetésnek is
 - **Reaktív (*reactive*)**: a szerver választásra kínálja fel a felhasználói ágensnek a reprezentációk listáját
 - Ezt hívják **ágens-vezérelt (*agent-driven*)** tartalomegyeztetésnek is
- Továbbiak is léteznek, a különböző formák nem zárják ki egymást kölcsönösen

Proaktív tartalomegyeztetés (1)

- A válaszhoz legjobb reprezentáció kiválasztása a szerver oldalon történik egy algoritmussal
- A választás a rendelkezésre álló reprezentációk és a kérdésben adott információk – bizonyos fejlécmezők és implicit jellemzők, mint például a kliens hálózati címe – alapján történik
- Az alábbi fejlécmezők kerülhetnek felhasználásra:
 - Accept, Accept -Charset, Accept -Encoding, Accept -Language, User -Agent
- Válaszban a Vary fejlécmező szolgál annak jelzésére, hogy az eredet szerveren mely fejlécmezők befolyásolhatják a reprezentáció kiválasztását

Proaktív tartalomegyeztetés (2)

- Előnyös:
 - Ha bonyolult módon írható le a felhasználói ágensnek a rendelkezésre álló reprezentációk közül választás algoritmus
 - Ha a szerver az első válaszában el akarja küldeni a felhasználói ágensnek az általa legjobbnak becsült reprezentációt, elkerülendő egy további kérés kiszolgálását

Proaktív tartalomegyeztetés (3)

- Hátrányok:
 - A szerver nem ismerheti a felhasználói ágens minden képességét és a válasz tervezett felhasználási módját
 - Nem hatékony minden kérdésben megadni a felhasználói ágens képességeit, ráadásul ez a felhasználó magánszférája is lehetséges kockázatot jelent
 - Bonyolítja az eredet szerver és egy kéreésre a választ generáló algoritmus megvalósítását
 - Osztott gyorsítótárzásnál korlátozza a válaszok újrafelhasználhatóságát

Proaktív tartalomegyeztetés (4)

- Példa:

- `curl -v http://www.gnu.org/ -H "Accept-Language: fr"`

```
> GET / HTTP/1.1
> User-Agent: curl/7.35.0
> Host: www.gnu.org
> Accept: */*
> Accept-Language: fr
>
< HTTP/1.1 200 OK
< Date: Wed, 28 Jan 2015 12:51:02 GMT
< Server: Apache/2.2.14
< Content-Location: home.fr.html
< Vary: negotiate,accept-language,Accept-Encoding
< TCN: choice
< Accept-Ranges: bytes
< Cache-Control: max-age=0
< Expires: Wed, 28 Jan 2015 12:51:02 GMT
< Transfer-Encoding: chunked
< Content-Type: text/html
< Content-Language: fr
<
< ...
```

Reaktív tartalomegyeztetés (1)

- A válaszhoz legjobb reprezentáció kiválasztását a felhasználói ágens végzi el, miután egy olyan választ kap az eredet szervertől, mely a választható reprezentációkhoz sorol fel erőforrásokat egy listában
 - A kiválasztást elvégezheti automatikusan a felhasználói ágens vagy kézzel a felhasználó

Reaktív tartalomegyeztetés (2)

- Akkor előnyös, ha a válasz több általánosan használt dimenzió (például típus, nyelv, kódolás) mentén változik és az eredet szerver nem képes a felhasználói ágens képességeit a kérésből megállapítani
- Hátránya:
 - A választható reprezentációk listájának továbbítása után egy további kérés végrehajtása szükséges
 - A specifikáció nem biztosít mechanizmust az automatikus kiválasztáshoz

Átirányítás (1)

- Példa:

- `curl -v http://www.oracle.com/`

```
> GET / HTTP/1.1
> User-Agent: curl/7.35.0
> Host: www.oracle.com
> Accept: */*
>
< HTTP/1.1 301 Moved Permanently
< Location: http://www.oracle.com/index.html
< Server: BigIP
< Content-Length: 0
< Date: Tue, 20 Jan 2015 13:31:42 GMT
< Connection: keep-alive
<
```

Átirányítás (2)

- Példa:

- `curl -v -L http://www.oracle.com/`

```
> GET / HTTP/1.1
> User-Agent: curl/7.35.0
> Host: www.oracle.com
> Accept: */*
>
< HTTP/1.1 301 Moved Permanently
< Location: http://www.oracle.com/index.html
< ...
<
> GET /index.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: www.oracle.com
> Accept: */*
>
```

Átirányítás (3)

- Példa (folytatás):

```
< HTTP/1.1 200 OK
< Access-Control-Allow-Origin: *
< X-Powered-By: Servlet/2.5 JSP/2.1
< X-Frame-Options: SAMEORIGIN
< Content-Type: text/html; charset=utf-8
< Content-Language: en
klisted
< Server: Oracle-Application-Server-11g Oracle-Web-Cache-11g/11.1.1.6.0
  (H;max-age=300+0;age=3;ecid=912517215829426,0:1)
< X-Akamai-Transformed: 9 - 0 pmb=mRUM,1
< Date: Tue, 20 Jan 2015 13:31:42 GMT
< Content-Length: 38963
< Connection: keep-alive
<
< <!DOCTYPE html>
<
< <html lang="en-US" class="no-js">
< ...
```

Átirányítás (4)

- Példa:

- `curl -v http://dbpedia.org/resource/Hungary`

```
> GET /resource/Hungary HTTP/1.1
> User-Agent: curl/7.35.0
> Host: dbpedia.org
> Accept: */*
>
< HTTP/1.1 303 See Other
< Date: Tue, 20 Jan 2015 13:43:20 GMT
< Content-Type: text/html; charset=UTF-8
< Content-Length: 0
< Connection: keep-alive
< Server: Virtuoso/07.10.3211 (Linux) x86_64-redhat-linux-gnu VDB
< Location: http://dbpedia.org/page/Hungary
< Expires: Tue, 27 Jan 2015 13:43:20 GMT
< Cache-Control: max-age=604800
<
```

Tartalomjegyzet (1)

- Példa:

- `curl -v --raw -L -H "Accept-Language: fr" \`
`http://www.opera.com/`

```
> GET / HTTP/1.1
> User-Agent: curl/7.35.0
> Host: www.opera.com
> Accept: */*
> Accept-Language: fr
>
< HTTP/1.1 302 Moved Temporarily
< Accept-Ranges: bytes
< Cache-Control: max-age=0
< Content-Type: text/html
< Date: Tue, 20 Jan 2015 14:16:55 GMT
< Location: http://www.opera.com/fr
< Vary: Accept-Encoding
< X-Cache: MISS
< X-Via: 2-ww,3-13238
< Content-Length: 154
< Connection: keep-alive
<
```

```
< <html>
< <head><title>302 Found</title></head>
< <body bgcolor="white">
< <center><h1>302 Found</h1></center>
< <hr><center>nginx</center>
< </body>
< </html>
```

Tartalomjegyzetítés (2)

- Példa (folytatás):

```
> GET /fr HTTP/1.1
> User-Agent: curl/7.35.0
> Host: www.opera.com
> Accept: */*
> Accept-Language: fr
>
< HTTP/1.1 200 OK
< Cache-Control: max-age=3600
< Content-language: fr-FR
< Content-Type: text/html; charset=utf-8
< Date: Tue, 20 Jan 2015 14:16:55 GMT
< Last-Modified: Tue, 20 Jan 2015 14:16:22 GMT
< Served-by: www.opera.com
< Vary: Accept-Encoding
< X-Cache: HIT:1
< X-Via: 1-141130,2-ww,3-13238
< transfer-encoding: chunked
< Connection: keep-alive
<
< 3731
< <!DOCTYPE html>
< <html lang="fr">
< ...
```

Tartalomegyeztetés (3)

- Példa:

- `curl -v -L -H "Accept: application/json" \`
`http://dbpedia.org/resource/Hungary`

```
> GET /resource/Hungary HTTP/1.1
> User-Agent: curl/7.35.0
> Host: dbpedia.org
> Accept: application/json
>
< HTTP/1.1 303 See Other
< Date: Tue, 20 Jan 2015 14:44:57 GMT
< Content-Type: application/json; qs=0.6
< Content-Length: 0
< Connection: keep-alive
< Server: Virtuoso/07.10.3211 (Linux) x86_64-redhat-linux-gnu VDB
< TCN: choice
< Vary: negotiate,accept
< Alternates: {"/data/Hungary.atom" 0.500000 {type application/atom+xml}},
  {"/data/Hungary.jrdf" 0.600000 {type application/rdf+json}},
  {"/data/Hungary.jsod" 0.500000 {type application/odata+json}},
  {"/data/Hungary.json" 0.600000 {type application/json}}, ...
< Link: <http://mementoarchive.lanl.gov/dbpedia/timegate/http://dbpedia.org/resource/Hungary>;
  rel="timegate"
< Location: http://dbpedia.org/data/Hungary.json
< Expires: Tue, 27 Jan 2015 14:44:57 GMT
< Cache-Control: max-age=604800
<
```

Tartalomegyeztetés (4)

- Példa (folytatás):

```
> GET /data/Hungary.json HTTP/1.1
> User-Agent: curl/7.35.0
> Host: dbpedia.org
> Accept: application/json
>
< HTTP/1.1 200 OK
< Date: Tue, 20 Jan 2015 15:04:50 GMT
< Content-Type: application/json
< Content-Length: 1525038
< Connection: keep-alive
< Vary: Accept-Encoding
< Server: Virtuoso/07.10.3211 (Linux) x86_64-redhat-linux-gnu VDB
< Expires: Tue, 27 Jan 2015 15:04:50 GMT
< Link: <http://dbpedia.org/data/Hungary.xml>; rel="alternate";
  type="application/rdf+xml"; title="Structured Descriptor Document (RDF/XML format)",
  <http://dbpedia.org/data/Hungary.n3>; rel="alternate"; type="text/n3";
  title="Structured Descriptor Document (N3/Turtle format)", ...
< X-SPARQL-default-graph: http://dbpedia.org
< Cache-Control: max-age=604800
< Accept-Ranges: bytes
<
< ...
```

Tartalomegyeztetés (5)

- Folytatás:

- `curl -v -L -H "Accept: application/rdf+xml" \`
`http://dbpedia.org/resource/Hungary \`
`-o Hungary.rdf`
- `curl -v -L -H "Accept: text/turtle" \`
`http://dbpedia.org/resource/Hungary \`
`-o Hungary.ttl`
- `curl -v -L -H "Accept: text/html" \`
`http://dbpedia.org/resource/Hungary \`
`-o Hungary.html`

Részleges GET kérés (1)

- Az első 256 bájt:
 - `curl -v -r 0-255 \`
`http://www.gnu.org/licenses/gpl-3.0.txt`

```
> GET /licenses/gpl-3.0.txt HTTP/1.1
> Range: bytes=0-255
> User-Agent: curl/7.35.0
> Host: www.gnu.org
> Accept: */*
>
```

Részleges GET kérés (2)

- Az első 256 bájt (válasz):

```
< HTTP/1.1 206 Partial Content
< Date: Tue, 20 Jan 2015 15:18:32 GMT
< Server: Apache/2.2.14
< Last-Modified: Sun, 01 Jul 2007 22:55:35 GMT
< ETag: "e2ca7-894b-4343b9c30c7c0"
< Accept-Ranges: bytes
< Content-Length: 256
< Vary: Accept-Encoding
< Content-Range: bytes 0-255/35147
< Content-Type: text/plain
< Content-Language: non-html
<
<
<          GNU GENERAL PUBLIC LICENSE
<          Version 3, 29 June 2007
<
< Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
< Everyone is permitted to copy and distribute verbatim copies
< of this license document, but
```

Részleges GET kérés (3)

- Az utolsó 256 bájt:
 - `curl -v -r -256 \`
`http://www.gnu.org/licenses/gpl-3.0.txt`

```
> GET /licenses/gpl-3.0.txt HTTP/1.1
> Range: bytes=-256
> User-Agent: curl/7.35.0
> Host: www.gnu.org
> Accept: */*
>
```

Részleges GET kérés (4)

- Az utolsó 256 bájt (válasz):

```
< HTTP/1.1 206 Partial Content
< Date: Sun, 01 Feb 2015 09:15:23 GMT
< Server: Apache/2.2.14
< Last-Modified: Sun, 01 Jul 2007 22:55:35 GMT
< ETag: "e2ca7-894b-4343b9c30c7c0"
< Accept-Ranges: bytes
< Content-Length: 256
< Vary: Accept-Encoding
< Content-Range: bytes 34891-35146/35147
< Content-Type: text/plain
< Content-Language: non-html
<
< ider it more useful to permit linking proprietary applications with
< the library.  If this is what you want to do, use the GNU Lesser General
< Public License instead of this License.  But first, please read
< <http://www.gnu.org/philosophy/why-not-lgpl.html>.
```

Részleges GET kérés (5)

- Az első és utolsó 50 bájt:
 - `curl -v -r 0-49, -50 \`
`http://www.gnu.org/licenses/gpl-3.0.txt`

```
> GET /licenses/gpl-3.0.txt HTTP/1.1
> Range: bytes=0-49, -50
> User-Agent: curl/7.35.0
> Host: www.gnu.org
> Accept: */*
>
```

Részleges GET kérés (6)

- Az első és utolsó 50 bájt (válasz):

```
< HTTP/1.1 206 Partial Content
< Date: Tue, 20 Jan 2015 15:30:28 GMT
< Server: Apache/2.2.14
< Last-Modified: Sun, 01 Jul 2007 22:55:35 GMT
< ETag: "e2ca7-894b-4343b9c30c7c0"
< Accept-Ranges: bytes
< Content-Length: 297
< Vary: Accept-Encoding
< Content-Type: multipart/byteranges; boundary=50d171e663760287
< Content-Language: non-html
<
<
< --50d171e663760287
< Content-type: text/plain
< Content-range: bytes 0-49/35147
<
<
< GNU GENERAL PUBLIC LICENSE
<
< --50d171e663760287
< Content-type: text/plain
< Content-range: bytes 35097-35146/35147
<
< http://www.gnu.org/philosophy/why-not-lgpl.html>.
<
< --50d171e663760287--
```

Feltételes kérések (1)

- Egy vagy több olyan fejlécmezőt tartalmazó kérések, melyek egy olyan előfeltételt szolgáltatnak, melyet ellenőrizni kell a metódus a cél erőforrásra történő alkalmazása előtt

Feltételes kérések (2)

- Gyakorlati felhasználás:
 - **Gyorsítótárazás:** a feltételes GET kérések jelentik a gyorsítótár frissítés leghatékonyabb módját
 - **Az elveszett módosítás (*lost update*) problémájának megelőzése:** előfeltételek alkalmazhatóak az erőforrások állapotát megváltoztató metódusokra is (PUT, DELETE), megelőzhető általuk, hogy egy kliens véletlenül felülírja egy vele párhuzamosan cselekvő másik kliens munkáját
 - Optimista konkurenciavezérlés (*optimistic concurrency control*) megvalósítása

Feltételes kérések (3)

- **Érvényesítő (*validator*)**: egy erőforrás állapotát tükröző olyan metaadat, mely révén észlelhető az állapot megváltozása
- A specifikáció az alábbi két érvényesítőt definiálja:
 - **módosítási dátum** (lásd a `Last-Modified` fejlécmezőt)
 - **entitás címke (*entity tag*)** (lásd az `ETag` fejlécmezőt)

Feltételes kérések (4)

- Az érvényesítők két fajtája:
 - **Erős érvényesítő (*strong validator*)**: olyan reprezentáció metaadat, melynek megváltozik az értéke, valahányszor olyan változás történik a reprezentáció adatokban, mely megfigyelhető lenne egy GET kérésre adott 200 (OK) állapotkodú válasz *payload* törzsében
 - Csak akkor áll az eredet szerver érdekében megváltoztatni az értékét, amikor érvényteleníteni kell a távoli gyorsítótárak által tárolt válaszokat
 - **Gyenge érvényesítő (*weak validator*)**: olyan reprezentáció metaadat, mely nem minden esetben változik meg a reprezentáció adatok változásakor
 - Az eredet szerver számára ajánlott az értékének megváltoztatása, ha a korábbi reprezentációkat elfogadhatatlannak tekinti az aktuális reprezentáció helyettesítőjeként

Feltételes kérések (5)

- Egy adott erőforrás reprezentációhoz tartozó erős érvényesítők kell, hogy egyediek legyenek
- Ez nem jelenti azt, hogy az egyediség követelmény különböző erőforrások reprezentációi között
 - Tehát ugyanaz az erős érvényesítő egyidejűleg több erőforrás reprezentációjához használható, az egyezés nem jelent a reprezentációk közötti ekvivalenciát

Feltételes kérések (6)

- Erős érvényesítők:
 - Például verziókezelésen alapuló verzióazonosító, a reprezentáció adatokra alkalmazott ütközésmentes hasítófüggvény értéke
 - Nagyon nehéz lehet a hatékony előállításuk

Feltételes kérések (7)

- Gyenge érvényesítők:
 - A gyengeség lehetséges okai:
 - Az érték kiszámítása során felmerülő korlátok (például időmérésnél a felbontás)
 - Nem lehet biztosítani az egyediséget az erőforrás minden lehetséges reprezentációjához
 - Reprezentációk egyenértékűként tekintése az erőforrás tulajdonosa által
 - Gyenge érvényesítő például egy reprezentáció másodperc pontosságú módosítási ideje, ha a reprezentáció egy másodpercen belül kétszer is megváltozhat és lekérdezhető a módosítások között
 - Könnyen előállíthatóak

Feltételes kérések (8)

- **Last-Modified:** válasz fejlécmező, melynek értéke a kiválasztott reprezentáció utolsó módosításának dátumát és időpontját jelző időbélyeg
 - Pontosság: másodperc
 - Időzóna: egyezményes koordinált világidő (UTC)
 - Példa:
 - Last-Modified: Sun, 01 Jul 2007 22:55:35 GMT

Feltételes kérések (9)

- **ETag**: válasz fejlécmező, melynek értéke a kiválasztott reprezentáció aktuális entitás címkéje
 - Az entitás címke egy átlátszatlan érvényesítő ugyanazon erőforrás több különböző reprezentációjának megkülönböztetésére
 - Több reprezentáció létezésének okai: (1) az erőforrás állapota idővel változik, (2) az erőforrás tartalomegyeztetésnek van alávetve
 - Formailag egy ' " ' határolók között megadott karakterlánc, melyet megelőzhet a gyengeséget jelző W/ előtag
 - Példa:
 - ETag: "101b85-67bf-4a50dc474f600"
 - ETag: "785b937e834cb8ad8a997c38e9aacbf3"
 - ETag: "1423137102365|#public|0|en|||0"
 - ETag: W/"884da3e-4b59-50dba044d35c1"

Feltételes kérések (10)

- Entitás címkék összehasonlítása:
 - **Erős összehasonlítás:** két entitás címke ekvivalens, ha mindkettő erős és karakterről-karakterre megegyeznek
 - **Gyenge összehasonlítás:** két entitás címke ekvivalens, ha az esetleges előtagtól eltekintve karakterre-karakterre megegyeznek

Címke ₁	Címke ₂	Erős összehasonlítás	Gyenge összehasonlítás
W/"xyz"	W/"xyz"	nincs egyezés	egyezés
W/"xyz"	W/"abc"	nincs egyezés	nincs egyezés
W/"xyz"	"xyz"	nincs egyezés	egyezés
"xyz"	"xyz"	egyezés	egyezés

Feltételes kérések (11)

- Egy eredet szerver számára az a javasolt viselkedés, hogy GET és HEAD kérésekre adott 200 (OK) állapotkodú válaszokban ETag és Last-Modified fejlécmezőt is küldjön
 - Lehetőleg erős entitás címkét küldjön, ha van rá mód

Feltételes kérések (13)

- Előfeltétel fejléc mezők:
 - If-Match
 - If-None-Match
 - If-Modified-Since
 - If-Unmodified-Since
 - If-Range

Feltételes kérések (14)

- Tilos a feltételes fejlécmezők kiértékelése olyan szerver számára, mely nem a cél erőforrás eredet szervere és nem tud gyorsítótárként működni a cél erőforrásra vonatkozó kérésekhez
- A szerver figyelmen kívül kell, hogy hagyja a feltételes fejlécmezőket olyan kéréseknél, melyek metódusa nem jár reprezentáció kiválasztásával vagy módosításával (CONNECT, OPTIONS, TRACE)

Feltételes kérések (15)

- A fogadó figyelmen kívül kell, hogy hagyja az `If-Modified-Since` fejlécmezőt, ha a kérés `If-None-Match` fejlécmezőt is tartalmaz
- A fogadó figyelmen kívül kell, hogy hagyja az `If-Unmodified-Since` fejlécmezőt, ha a kérés `If-Match` fejlécmezőt is tartalmaz

Entitás címkék előállítása

- Apache HTTP Server:
 - *Apache HTTP Server Version 2.4 Documentation*
 - *FileETag Directive*
 - <http://httpd.apache.org/docs/current/mod/core.html#fileetag>
- Internet Information Services (IIS):
 - *Configuring ETag and Max-Age in IIS*
 - <https://technet.microsoft.com/en-us/library/ee619764%28v=ws.10%29.aspx>

Feltételes kérések: If - Match (1)

- If - Match fejlécmezőt fogadó eredet szerver a metódus végrehajtása előtt ki kell, hogy értékelje az előfeltételt
 - Ha a mező értéke ' * ':
 - A feltétel hamis, ha az eredet szervernek nincs a cél erőforráshoz aktuális reprezentációja, egyébként igaz
 - Ha a mező értéke entitás címkék egy listája:
 - A feltétel hamis, ha egyik entitás címke sem egyezik meg a kiválasztott reprezentáció entitás címkéjével, egyébként igaz
- Az eredet szerver erős összehasonlítást kell, hogy használjon az entitás címkék egyezőségének vizsgálatához

Feltételes kérések: If-Match (2)

- Ha az előfeltétel hamis, akkor tilos a metódus végrehajtása, helyette 412 (Precondition Failed) vagy 2xx (Successful) állapotkóddal kell válaszolni
 - Az utóbbival akkor, ha az eredet szerver megállapította, hogy egy állapot módosítását kérelmezték és már a végső állapotot tükrözi a cél erőforrás aktuális állapota
 - Azaz már sikerült a kért a módosítás, de a felhasználói ágensnek erről nincs tudomása, mert a korábbi válasz elveszett, vagy mert egy másik felhasználói ágens egy kompatibilis módosítást végzett

Feltételes kérések: If - Match (3)

- A leggyakrabban állapotmódosító metódusokhoz (POST, PUT, DELETE) használják az elveszett módosítás problémájának megelőzéséhez
- Gyorsítótárak figyelmen kívül hagyhatják a fejlécmezőt, mivel nem vonatkozik tárolt válaszokra

Feltételes kérések: If - None - Match (1)

- If - None - Match fejlécmezőt fogadó eredet szerver a metódus végrehajtása előtt ki kell, hogy értékelje az előfeltételt
 - Ha a mező értéke ' * ':
 - A feltétel hamis, ha az eredet szervernek van a cél erőforráshoz aktuális reprezentációja, egyébként igaz
 - Ha a mező értéke entitás címkék egy listája:
 - A feltétel hamis, ha az entitás címkék valamelyike megegyezik a kiválasztott reprezentáció entitás címkéjével, egyébként igaz
- Az eredet szerver gyenge összehasonlítást kell, hogy használjon az entitás címkék egyezőségének vizsgálatához

Feltételes kérések: If - None - Match (2)

- Ha az előfeltétel hamis, akkor tilos a metódus végrehajtása, helyette
 - 304 (Not Modified) állapotkóddal kell válaszolni, ha a metódus GET vagy HEAD
 - 412 (Precondition Failed) állapotkóddal kell válaszolni az összes többi metódus esetén

Feltételes kérések: If - None - Match (3)

- Elsősorban GET kérésekben használják gyorsítótárazott reprezentációk hatékony frissítéséhez
- Ha ' * ' az értéke, akkor nem biztonságos metódusok (például PUT) esetén megelőzhető általa a cél erőforrás egy létező reprezentációjának nem szándékos módosítása
 - Amikor a kliens azt feltételezi, hogy nincs a cél erőforrásnak aktuális reprezentációja

Feltételes kérések: If-None-Match (4)

- Példa:

```
- curl -v \  
-H "If-None-Match: \"e2ca7-894b-  
4343b9c30c7c0\"" http://httpd.apache.org/
```

```
> GET /licenses/gpl-3.0.txt HTTP/1.1  
> User-Agent: curl/7.35.0  
> Host: www.gnu.org  
> Accept: */*  
> If-None-Match: "e2ca7-894b-4343b9c30c7c0"  
>  
< HTTP/1.1 304 Not Modified  
< Date: Sun, 01 Feb 2015 11:01:55 GMT  
< Server: Apache/2.2.14  
< ETag: "e2ca7-894b-4343b9c30c7c0"  
< Vary: Accept-Encoding  
<
```

Feltételes kérések: If-Modified-Since (1)

- Értéke egy időbélyeg
- Az eredet szerver figyelmen kívül kell, hogy hagyja az If-Modified-Since fejlécmezőt, ha a metódus nem GET vagy HEAD
- If-Modified-Since fejlécmezőt fogadó eredet szerver számára ajánlott az előfeltétel kiértékelése a metódus végrehajtása előtt
- Ha a kiválasztott reprezentáció utolsó módosítási ideje korábbi vagy ugyanaz, mint a fejlécmező értékeként adott dátum, akkor a szerver számára nem ajánlott a metódus végrehajtása, helyette egy 304 (Not Modified) állapotkódú válasz küldése ajánlott

Feltételes kérések: If - Modified - Since (2)

- Két eltérő cél:
 - Lehetővé teszi egy olyan gyorsítótárazott reprezentáció hatékony frissítését, melynek nincs entitás címkéje
 - Lehetővé teszi az információ lekérdezés olyan erőforrásokra való korlátozását, melyek nemrég módosultak

Feltételes kérések: If-Modified-Since (3)

- Példa:

- `curl -v -z "20 Jan 2015 12:00:00 CET" \`
`http://www.gnu.org/licenses/gpl-3.0.txt`

```
> GET /licenses/gpl-3.0.txt HTTP/1.1
> User-Agent: curl/7.35.0
> Host: www.gnu.org
> Accept: */*
> If-Modified-Since: Tue, 20 Jan 2015 11:00:00 GMT
>
< HTTP/1.1 304 Not Modified
< Date: Sun, 01 Feb 2015 10:56:14 GMT
< Server: Apache/2.2.14
< ETag: "e2ca7-894b-4343b9c30c7c0"
< Vary: Accept-Encoding
<
```

Feltételes kérések:

If - Unmodified - Since (1)

- Értéke egy időbélyeg
- If - Unmodified - Since fejlécmezőt fogadó eredet szerver a metódus végrehajtása előtt ki kell, hogy értékelje az előfeltételt
- Ha a kiválasztott reprezentáció utolsó módosítási ideje későbbi, mint a fejlécmező értékeként adott dátum, akkor tilos a metódus végrehajtása, helyette 412 (Precondition Failed) vagy 2xx (Successful) állapotkóddal kell válaszolni
 - Lásd az If - Match fejlécmezőnél leírtakat

Feltételes kérések:

If - Unmodified - Since (2)

- A leggyakrabban állapotmódosító metódusokhoz (POST, PUT, DELETE) használják az elveszett módosítás problémájának megelőzéséhez
- Gyorsítótárak figyelmen kívül hagyhatják a fejlécmezőt, mivel nem vonatkozik tárolt válaszokra

Feltételes kérések: If-Unmodified-Since (3)

- Példa:

```
- curl -v -z -" `date --date=yesterday -R` " \  
http://www.w3.org/
```

```
> GET / HTTP/1.1  
> User-Agent: curl/7.35.0  
> Host: www.w3.org  
> Accept: */*  
> If-Unmodified-Since: Wed, 04 Feb 2015 14:09:47 GMT  
>
```

Feltételes kérések:

If-Unmodified-Since (4)

- Példa (folytatás):

```
< HTTP/1.1 412 Precondition Failed
< Date: Thu, 05 Feb 2015 14:09:47 GMT
< Server: Apache/2
< Content-Location: Home.html
< Vary: negotiate,accept
< TCN: choice
< Last-Modified: Thu, 05 Feb 2015 10:16:09 GMT
< ETag: "83d7-50e5497b7cc40;89-3f26bd17a2f00"
< Accept-Ranges: bytes
< Content-Length: 0
< Cache-Control: max-age=600
< Expires: Thu, 05 Feb 2015 14:19:47 GMT
< P3P: policyref="http://www.w3.org/2014/08/p3p.xml"
< Content-Type: text/html; charset=utf-8
<
```

Feltételes kérések: If - Range (1)

- Értéke egy entitás címke vagy egy időbélyeg
- A szerver figyelmen kívül kell, hogy hagyja, ha a kérés nem tartalmaz Range fejlécmezőt
- Az If - Match és If - Unmodified - Since fejlécmezőkhöz hasonló feltételes mechanizmus, de az érvényesítők nem egyezése esetén 412 (Precondition Failed) állapotkodú válasz helyett a fogadó figyelmen kívül kell, hogy hagyja a Range fejlécmezőt, mely az új kiválasztott reprezentáció átvitelét eredményezi
- Az eredet szerver erős összehasonlítást kell, hogy használjon az entitás címkék egyezőségének vizsgálatához

Feltételes kérések: If-Range (2)

- Példa:
 - `curl -v -r 64-127 \`
`-H "If-Range: \"e2ca7-894b-4343b9c30c7c0\"" \`
`http://www.gnu.org/licenses/gpl-3.0.txt`

Feltételes kérések: If-Range (3)

```
> GET /licenses/gpl-3.0.txt HTTP/1.1
> Range: bytes=64-127
> User-Agent: curl/7.35.0
> Host: www.gnu.org
> Accept: */*
> If-Range: "e2ca7-894b-4343b9c30c7c0"
>
< HTTP/1.1 206 Partial Content
< Date: Tue, 20 Jan 2015 20:12:02 GMT
< Server: Apache/2.2.14
< Last-Modified: Sun, 01 Jul 2007 22:55:35 GMT
< ETag: "e2ca7-894b-4343b9c30c7c0"
< Accept-Ranges: bytes
< Content-Length: 64
< Vary: Accept-Encoding
< Content-Range: bytes 64-127/35147
< Content-Type: text/plain
< Content-Language: non-html
<
<
<           Version 3, 29 June 2007
<
```

Gyorsítótárazás (1)

- **Gyorsítótár (*cache*):** HTTP válaszok lokális tárolására szolgáló alrendszer, melynek célja gyorsítótárazható válaszok tárolása a válaszidő és a hálózati forgalom csökkentése céljából
 - Minden kliens és szerver is használhat gyorsítótárat, csak alagútként működő szerver nem
- **Gyorsítótárazható (*cacheable*):** egy válasz gyorsítótárazható, ha a gyorsítótár eltárolhatja egy másolatát későbbi kérésekhez való felhasználáshoz

Gyorsítótárak (2)

- A felhasználók számának szempontjából a gyorsítótárak két fajtája:
 - **Saját gyorsítótár (*private cache*):** csupán egyetlen felhasználó számára hozzáférhető
 - Gyakran a felhasználói ágens komponenseként van telepítve
 - **Megosztott gyorsítótár (*shared cache*):** több felhasználó számára is hozzáférhető
 - Általában, de nem mindig, egy közvetítő részeként van telepítve

Gyorsítótárzás (3)

- **Gyorsítótár bejegyzés (*cache entry*):** egy gyorsítótár kulcsból és vagy több, ugyanezt a kulcsot használó kérésnek megfelelő válaszból áll
 - A bejegyzések leggyakrabban egy GET kérésre adott 200 (OK) állapotkódú válaszokat tartalmaznak
 - A kulcsot a kérés metódusa és a cél erőforrás URI-ja alkotja
 - Ha a cél erőforrás tartalommegyeztetésnek van alávetve, akkor a bejegyzés több tárolt válaszból állhat, melyeket az eredeti kérés kiválasztó fejlécmezőinek értékei különböztetnek meg
 - Kiválasztó fejlécmező a reprezentáció kiválasztásban szerepet játszó fejlécmező (például Accept, Accept - Language)

Gyorsítótárazás (4)

- Alapul szolgáló mechanizmusok:
 - **Frissesség (*freshness*)**: sok esetben szükségtelessé teszi kérések végrehajtását
 - **Érvényesítés (*validation*)**: sok esetben szükségtelessé teszi teljes válaszok elküldését

Gyorsítótárazás (5)

- **Kor (*age*):** egy válasz kora az eredet szerveren történő létrehozása vagy sikeres érvényesítése óta eltelt idő
- **Lejárati idő (*expiration time*):** az az időpont, melyet követően a gyorsítótár nem használhat fel egy választ további érvényesítés nélkül
 - Az eredet szerver explicit módon az Expires válasz fejlécmezőben illetve a max-age vagy s-maxage válasz direktívával jelezheti
 - Ha nincs megadva egy válaszban explicit módon, akkor a gyorsítótár heurisztikus becslést használhat meghatározásához
- **Frissesség élettartama (*freshness lifetime*):** a válasz az eredet szerveren történő létrehozása és lejárat ideje között eltelt idő

Gyorsítótárzás (6)

- A frissesség élettartamának meghatározása:
 1. Ha a gyorsítótár megosztott és meg van adva az `s-maxage` válasz direktíva, akkor annak értéke szolgáltatja
 2. Ha meg van adva a `max-age` válasz direktíva, akkor annak értéke szolgáltatja
 3. Ha meg van adva az `Expires` válasz fejlécmező, akkor annak értékének és a `Date` válasz fejlécmező értékének különbsége szolgáltatja
 4. Egyébként nincs explicit lejáratási idő a válaszban, heurisztikus becslést lehet használni

Gyorsítótárak (7)

- **Friss válasz (*fresh response*):** olyan válasz, melynek kora nem haladja meg a frissességi élettartamot
- **Elévült válasz (*stale response*):** olyan válasz, amely nem friss
- **Érvényesítés (*validation*):** folyamat, melynek során ellenőrzésre kerül az eredet szerveren, hogy a gyorsítótár egy elévült tárolt válasza továbbra is felhasználható-e egy kérés kiszolgálásához
 - Az érvényesítéshez használható feltételes kérés, GET kérésekre adott válaszok esetén a HEAD metódus

Gyorsítótárzás (8)

- Példa:
 - `curl -v http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_day.csv -o all_day.csv`
 - A fenti címen az elmúlt nap földrengéseinek adatai érhetőek el (frissítés 5 percenként)
 - A `max-age` válasz direktíva elsőbbséget élvez az `Expires` válasz fejlécmezővel szemben, ha egy válasz mindkettőt tartalmazza, akkor a frissesség élettartamát a direktíva határozza meg

Gyorsítótárzás (9)

```
> GET /earthquakes/feed/v1.0/summary/all_day.csv HTTP/1.1
> User-Agent: curl/7.35.0
> Host: earthquake.usgs.gov
> Accept: */*
>
< HTTP/1.1 200 OK
< X-Powered-By: PHP/5.5.20
< Last-Modified: Sat, 07 Feb 2015 15:31:52 GMT
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Methods: *
< Access-Control-Allow-Headers: accept,origin,authorization,
content-type
< Content-Type: text/csv
< Cache-Control: public, max-age=204
< Expires: Sat, 07 Feb 2015 15:36:23 GMT
< Date: Sat, 07 Feb 2015 15:32:59 GMT
< Content-Length: 20344
< Connection: keep-alive
<
< ...
```

Gyorsítótárazás (10)

- **Cache-Control**: kérésekben és válaszokban is megengedett, a gyorsítótárazást vezérlő direktívák megadására szolgáló fejlécmező
 - A direktívákat kisbetű-nagybetű érzéketlen tokenek azonosítják, egy opcionális argumentumuk van
 - Példa:
 - `Cache-Control: no-cache, must-revalidate, max-age=180`
 - A direktívák regisztrálását az IANA végzi
 - Lásd: *Hypertext Transfer Protocol (HTTP) Cache Directive Registry*
<http://www.iana.org/assignments/http-cache-directives/http-cache-directives.xhtml>

Gyorsítótárazás (11)

- Fontosabb Cache-Control válasz direktívák:

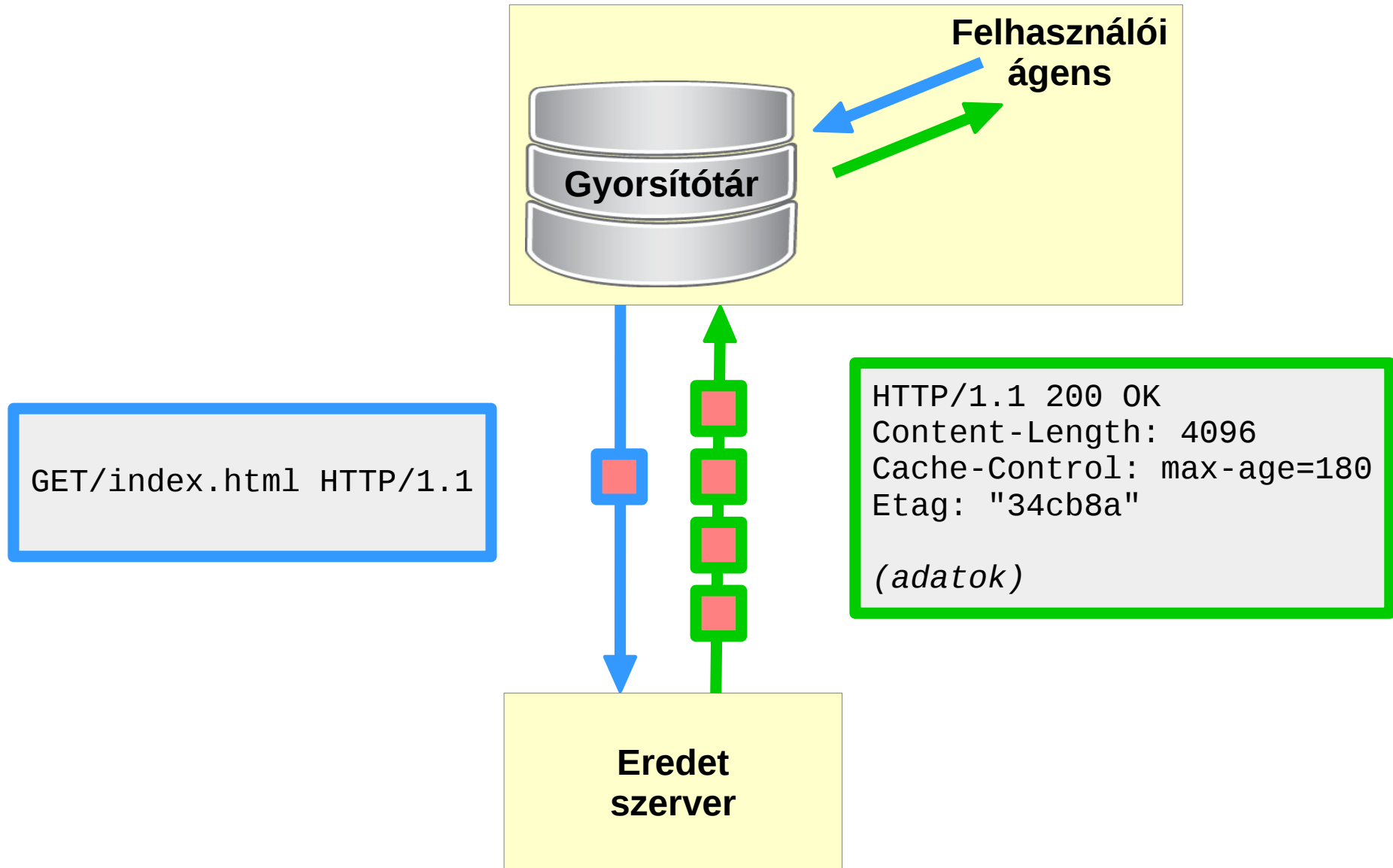
Direktíva	Leírás
<code>max-age=nnn</code>	A válasz frissességi élettartamát szolgáltatja másodpercben
<code>must-revalidate</code>	Ha a válasz elévül, tilos az eredet szerverrel való sikeres érvényesítés nélkül újabb kérések kielégítéséhez használni
<code>no-cache</code>	Tilos a választ az eredet szerverrel való sikeres érvényesítés nélkül újabb kérések kielégítéséhez használni
<code>no-store</code>	Tilos a válasz eltárolása gyorsítótárazás céljából
<code>private</code>	Megosztott gyorsítótár számára tilos a válasz tárolása
<code>public</code>	Bármely gyorsítótár eltárolhatja a választ (akkor is, ha normál esetben nem gyorsítótárazható)

Gyorsítótárzás (12)

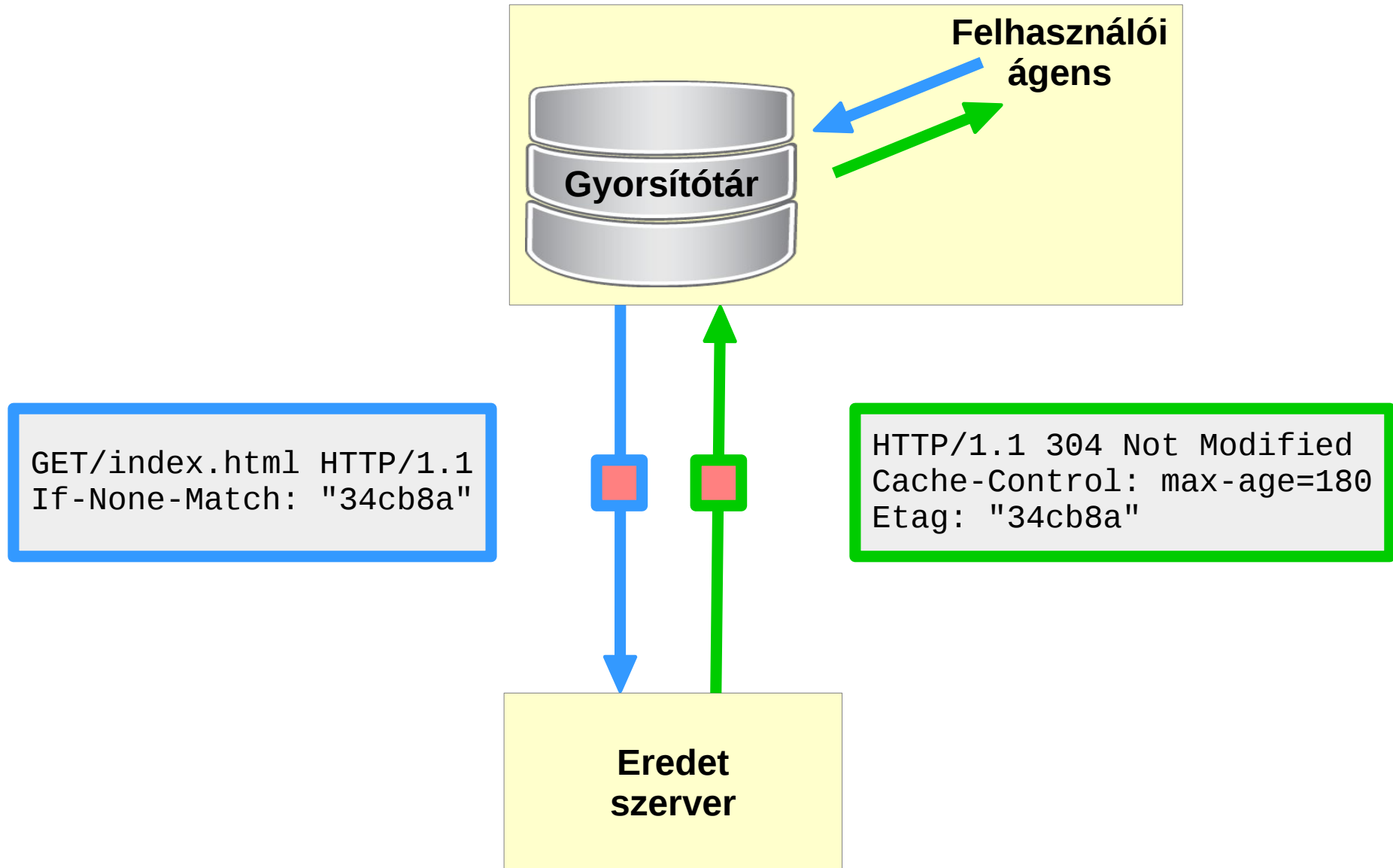
- Példa gyorsítótárzás tiltására:

```
> GET / HTTP/1.1
> User-Agent: curl/7.35.0
> Host: www.bookdepository.com
> Accept: */*
>
< HTTP/1.1 200 OK
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
  pre-check=0
< Content-Type: text/html; charset=UTF-8
< Date: Sat, 07 Feb 2015 15:45:23 GMT
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
< Pragma: no-cache
< Server: nginx
< Set-Cookie: ENTITY_SESS_ID_UK=a1bdc254b4083f54464703d542bd91b3;
  path=/; domain=bookdepository.com; HttpOnly
< transfer-encoding: chunked
< Connection: keep-alive
<
< ...
```

Gyorsítótárazás (13)



Gyorsítótárazás (14)



Gyorsítótárazás (15)

- Egy választ akkor tárolhat el egy gyorsítótár, ha teljesülnek az alábbiak:
 - A gyorsítótár „megéri” a metódust, mely gyorsítótárazhatóként definiált
 - A gyorsítótár „megéri” az állapotkódot
 - Sem a kérésben, sem a válaszban nem fordul a no-cache direktíva
 - A válaszban nem fordul elő a private válasz direktíva, ha a gyorsítótár megosztott
 - Az Authorization fejlécmező nem fordul elő a kérésben, ha a gyorsítótár megosztott
 - Az alábbiak valamelyike teljesül a válaszra:
 - Tartalmaz Expires fejlécmezőt
 - Tartalmaz max-age válasz direktívát
 - Tartalmaz s-maxage válasz direktívát és a gyorsítótár megosztott
 - Olyan állapotkódú, mely alapértelmezés szerint gyorsítótárazható (például 200)
- Metódus/állapotkód „megértése”: a gyorsítótár felismeri, valamint megvalósítja a gyorsítótárazáshoz a specifikáció által előírt valamennyi funkcionalitást

Gyorsítótárzás (16)

- További információk:
 - HTTP caching
<https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/http-caching>

Kódolások űrlap adatok továbbításához

- `application/x-www-form-urlencoded`
 - Lásd:
 - <http://www.w3.org/TR/html401/interact/forms.html#h-17.13.4.1>
 - <https://html.spec.whatwg.org/multipage/forms.html#url-encoded-form-data>
- `multipart/form-data`
 - Larry Masinter, *Returning Values from Forms: multipart/form-data*, RFC 2388, August 1998.
<http://tools.ietf.org/html/rfc2388>

Űrlapok (1)

- Példa:

- curl -v

- ```
"http://validator.w3.org/check?\nuri=http%3A%2F%2Fwww.w3.org&\ncharset=(detect+automatically) \n&doctype=Inline&group=0"
```

```
<form method="get" action="check">\n <p class="instructions">\n Validate a document online:\n </p>\n <p>\n <label title="Address of page to Validate" for="uri">Address:</label>\n <input type="text" name="uri" id="uri" size="45"/>\n </p>\n ... \n</form>
```

# Úrlapok (2)

```
> GET /check?uri=http%3A%2F%2Fwww.w3.org&charset=(detect+automatically)
 &doctype=Inline&group=0 HTTP/1.1
> User-Agent: curl/7.35.0
> Host: validator.w3.org
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Tue, 20 Jan 2015 15:47:13 GMT
< Server: Apache/2.2.16 (Debian)
< Content-Language: en
< X-W3C-Validator-Recursion: 1
< X-W3C-Validator-Status: Valid
< X-W3C-Validator-Errors: 0
< X-W3C-Validator-Warnings: 0
< Vary: Accept-Encoding
< Content-Type: text/html; charset=UTF-8
< Connection: close
< Transfer-Encoding: chunked
<
< 42d
< <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
< <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
< ...
```

# Űrlapok (3)

- Példa:

```
- curl -v \
https://www.otpbank.hu/portal/hu/Arfolyamok/OTP \
-d "arf_otp_valto_form=arf_otp_valto_form\
fromValue=150\
valtas=keszpenz\
fromDiso=EUR\
toDiso=HUF"
```

```
<form id="arf0tpValtoCommand" name="form" action="#"#arf_valto_box" method="post">
 <input type="hidden" name="arf_otp_valto_form" value="arf_otp_valto_form"/>
 <div class="form-header">
 <div class="row">
 <label for="sum">Összeg</label>
 <input id="sum" name="fromValue" class="textfield" onfocus="this.select();" type="text" value="150" maxlength="10"/>
 </div>
 ...
```

# Ürlapok (4)

```
> POST /portal/hu/Arfolyamok/OTP HTTP/1.1
> User-Agent: curl/7.35.0
> Host: www.otpbank.hu
> Accept: */*
> Content-Length: 91
> Content-Type: application/x-www-form-urlencoded
>
> arf_otp_valto_form=arf_otp_valto_form&fromValue=150&valtas=keszpenz&fromDiso=EUR
 &toDiso=HUF
< HTTP/1.1 200 OK
< Content-Type: text/html; charset=utf-8
< Server:
< Set-Cookie: JSESSIONID=482FFBD79EB58C25080A73C538F8362A.jvm2; Path=/; Secure;
 HttpOnly
< X-Powered-By:
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1
< X-Frame-Options: SAMEORIGIN
< Date: Sun, 01 Feb 2015 19:10:20 GMT
< Content-Length: 71393
<
< <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<
< <html xmlns="http://www.w3.org/1999/xhtml" lang="hu" xml:lang="hu">
< <head>
< <title>OTP Bank - OTP valuta-deviza árfolyamok</title>
< ...
```

# Űrlapok (5)

- Példa:

- `curl http://validator.w3.org/check \`  
-F `uploaded_file=@index.html \`  
-F `doctype=Inline \`  
-F `ss=1 --trace trace.txt -o validator.html`

```
<form method="post" enctype="multipart/form-data" action="check">
 <p class="instructions">Upload a document for validation:</p>
 <p>
 <label title="Choose a Local File to Upload and Validate"
 for="uploaded_file">File:</label>
 <input type="file" id="uploaded_file" name="uploaded_file"
 size="30" />
 </p>
 ...
```

# Úrlapok (6)

```
> POST /check HTTP/1.1
> User-Agent: curl/7.35.0
> Host: validator.w3.org
> Accept: */*
> Content-Length: 531
> Expect: 100-continue
> Content-Type: multipart/form-data; boundary=-----8aa5fe2b7b5695dc
>
> -----8aa5fe2b7b5695dc
> Content-Disposition: form-data; name="uploaded_file"; filename="index.html"
> Content-Type: text/html
>
> <!DOCTYPE html>
> <html>
> <head>
> <meta charset="UTF-8"/><title>Hello, world!</title>
> </head>
> <body><p>Hello, world!</p></body>
> </html>
>
> -----8aa5fe2b7b5695dc
> Content-Disposition: form-data; name="doctype"
>
> Inline
> -----8aa5fe2b7b5695dc
> Content-Disposition: form-data; name="ss"
>
> 1
> -----8aa5fe2b7b5695dc--
```

# Úrlapok (7)

```
< HTTP/1.1 200 OK
< Date: Sun, 01 Feb 2015 18:51:02 GMT
< Server: Apache/2.2.16 (Debian)
< Content-Language: en
< X-W3C-Validator-Recursion: 1
< X-W3C-Validator-Status: Valid
< X-W3C-Validator-Errors: 0
< X-W3C-Validator-Warnings: 1
< Content-Type: text/html; charset=UTF-8
< Vary: Accept-Encoding
< Connection: close
< Transfer-Encoding: chunked
<
< 444
< <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
< "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
< <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
< <head>
< <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
< <title>
< [Valid]
< Markup Validation of index.html - W3C Markup Validator</title>
< ...
```

# Proxy szerver használata (1)

- Példa:

- `curl --proxy 66.35.68.145:3127 \`  
`--head http://www.grumpycats.com/ -v`

```
Connected to 66.35.68.145 (66.35.68.145) port 3127 (#0)
> HEAD http://www.grumpycats.com/ HTTP/1.1
> User-Agent: curl/7.35.0
> Host: www.grumpycats.com
> Accept: */*
> Proxy-Connection: Keep-Alive
>
```

# Proxy szerver használata (2)

```
< HTTP/1.1 200 OK
< Date: Tue, 17 Feb 2015 16:05:23 GMT
< Server: LiteSpeed
< Accept-Ranges: bytes
< ETag: "f275-54e353f3-95d956462c419b5d"
< Last-Modified: Tue, 17 Feb 2015 14:45:07 GMT
< Content-Type: text/html
< Content-Length: 62069
< Age: 144
< X-Cache: HIT from CacheServer
< X-Cache-Lookup: HIT from CacheServer:3127
< Via: 1.1 CacheServer (squid/3.2.13)
< Connection: keep-alive
<
```

# Proxy szerver használata (3)

- Via fejlécmező:
  - Kéréseknél a felhasználói ágens és a szerver, válaszoknál az eredet szerver és a kliens közötti közvetítők jelenlétének jelzésére szolgáló fejlécmező
  - Értéke egy lista, melynek minden egyes eleme egy az üzenetet továbbító közvetítőt képvisel
  - Minden egyes közvetítő információkat ad hozzá magáról a mezőhöz

# Proxy szerver használata (4)

- Információvesztés történik, amikor egy proxy egy kérést továbbít
  - Elvész például az IP-cím, ahonnan a kérés eredetileg érkezett
- Az információvesztés nem minden esetben kívánatos
- Kívánatos akkor, ha a cél anonimitás biztosítása a kliens számára
  - Kapcsolódó fogalom: *anonymizer*, *anonymous proxy*

# Proxy szerver használata: X-Forwarded-For

- A gyakorlatban elterjedten használt nem szabványos fejlécmező a kérést kezdeményező kliens IP-címének jelzéséhez
  - Az üzenet továbbítás során minden egyes közvetítő hozzáadhat egy IP-címet a fejlécmező értékéhez
  - Példa:
    - X-Forwarded-For : 172.22.197.80
    - X-Forwarded-For : 172.22.197.80, 193.6.136.39
  - Lásd:
    - *Squid Web Proxy Wiki – X-Forwarded-For*  
[http://wiki.squid-cache.org/SquidFaq/ConfiguringSquid#What\\_is\\_.22HTTP\\_X\\_FORWARDED\\_FOR.22.3F\\_\\_Why\\_does\\_squid\\_provide\\_it\\_to\\_WWW\\_servers.2C\\_and\\_how\\_can\\_I\\_stop\\_it.3F](http://wiki.squid-cache.org/SquidFaq/ConfiguringSquid#What_is_.22HTTP_X_FORWARDED_FOR.22.3F__Why_does_squid_provide_it_to_WWW_servers.2C_and_how_can_I_stop_it.3F)
    - *Apache Module mod\_proxy*  
[http://httpd.apache.org/docs/current/mod/mod\\_proxy.html#x-headers](http://httpd.apache.org/docs/current/mod/mod_proxy.html#x-headers)

# Proxy szerver használata: Forwarded (1)

- Szabványos fejlécmező definiálása, mely lehetővé teszi proxyk számára az üzenettovábbítás során elvesző információk közlését
  - Andreas Petersson, Martin Nilsson, *Forwarded HTTP Extension*, RFC 7239, June 2014.  
<http://tools.ietf.org/html/rfc7239>

# Proxy szerver használata: Forwarded (2)

- Csak kérésekben használható
- Értéke paraméter-azonosító párok egy listája
  - Olyan információkat tartalmaz, melyek a proxy az üzenettovábbításban való közreműködése során elvesztek vagy módosultak
- Példa:
  - Forwarded: for=193.6.135.21
  - Forwarded: for=193.6.135.21,  
for=193.6.136.39:5128
- Ha a kérés több proxyn is keresztülmegy, mindegyik hozzáadhat paramétereket a fejlécmező értékéhez

# Proxy szerver használata: Forwarded (3)

- Ha a proxy az IP-címet titokban akarja tartani, akkor küldhet egy generált azonosítót (ez az ajánlott alapértelmezett viselkedés)
  - Minden kéréshez egy új véletlen azonosító ajánlott
  - Véletlenszerűen generált azonosítók aláhúzójellel kell, hogy kezdődjenek
    - Példa:
      - Forwarded: for=\_hidden, for=\_EB46F7

# Hitelesítés (1)

- Roy T. Fielding (ed.), Julian F. Reschke (ed.), *Hypertext Transfer Protocol (HTTP/1.1): Authentication*, RFC 7235, June 2014. <http://tools.ietf.org/html/rfc7235>
  - Általános keretrendszer a hozzáférés-vezérléshez és hitelesítéshez: **hitelesítési sémák** (*authentication schemes*)
- Az alkalmazások más megoldásokat is használhatnak a hitelesítéshez
  - Például az átviteli rétegben történő hitelesítés, további fejlécmezők használata, ...

# Hitelesítés (2)

- **Felszólítás-válasz (*challenge-response*)**

hitelesítés:

- A sémák révén egy szerver „kihívást” intézhet egy klienshez, a kliens pedig hitelesítő információkat szolgáltat

# Hitelesítés (3)

- Szintaxis a kihívás és a hitelesítő adatok ábrázolására:
  - A hitelesítési sémát egy kisbetű-nagybetű érzéketlen karakterlánc azonosítja, melyet a sémán keresztül történő hitelesítéshez szükséges további információk követnek
    - Az utóbbi paraméterek egy vesszővel elválasztott listája vagy egyetlen Base-64/Base32/Base16 (hex) kódolt karakterlánc
    - A paraméterek *név=érték* párok, ahol a név kisbetű-nagybetű érzéketlen, minden név csak egyszer fordulhat elő kihívásonként
  - Példa:
    - `Basic realm="Authentication Required"`
    - `Basic amVzenk6c2VjcmV0`

# Hitelesítés (4)

- Egy eredet szerver egy 401 (Unauthorized) állapotkódú válaszban szólítja fel hitelesítésre a felhasználói ágenst
  - A válaszban kötelezően megjelenik a WWW-Authenticate fejlécmező, mely legalább egy, a kért erőforrásra vonatkozó kihívást tartalmaz
  - A fejlécmező értéke kihívások egy listája, melyben minden egyes kihívás egy hitelesítő sémából és opcionális paramétereiből áll

# Hitelesítés (5)

- A felhasználói ágens az `Authorization` fejlécmezővel hitelesítheti magát az eredet szerver felé
  - Ez általában, de nem szükségszerűen egy `401` (`Unauthorized`) állapotkódú válasz után történik
  - A fejlécmező értéke a kért erőforrás tartományára vonatkozó hitelesítő adatokból áll

# Hitelesítés (6)

- Kényelmi okokból a HTTP kliensek és felhasználói ágensek tipikusan megjegyzik és korlátlan ideig tárolják a hitelesítő adatokat

# Hitelesítés (7)

- **Védelmi terület (tartomány):**
  - A tartományok lehetővé teszik egy szerver védett erőforrásainak részekre való felosztását, melyek mindegyikéhez saját hitelesítési séma és hozzáférési adatbázis tartozhat
  - Egy védelmi tartományt egy karakterlánc azonosít, melynek megadására a `realm` hitelesítési paraméter van fenntartva
  - Ha egy kérés engedélyezésre került, akkor a felhasználói ágens a hitelesítési séma, a paraméterek és a felhasználói beállítások által meghatározott időtartamon belül újra felhasználhatja a hitelesítő adatokat a védelmi terület erőforrásaira vonatkozó kérésekhez

# Hitelesítés (8)

- Példa:

- `curl -v https://www.w3.org/WAI/PF/Group/`

```
> GET /WAI/PF/Group/ HTTP/1.1
> User-Agent: curl/7.35.0
> Host: www.w3.org
> Accept: */*
>
< HTTP/1.1 401 Authorization Required
< Date: Sun, 01 Feb 2015 10:46:50 GMT
< Server: Apache/2
< WWW-Authenticate: Basic realm="W3CACL"
< Last-Modified: Wed, 14 May 2014 14:52:00 GMT
< ETag: "5fa-4f95d529ae400"
< Accept-Ranges: bytes
< Content-Length: 1530
< P3P: policyref="http://www.w3.org/2014/08/p3p.xml"
< Content-Type: text/html; charset=iso-8859-1
<
< ...
```

# Hitelesítés (9)

- Példa:
  - `curl -v https://www.googleapis.com/urlshortener/v1/url/history`

```
> GET /urlshortener/v1/url/history HTTP/1.1
> User-Agent: curl/7.35.0
> Host: www.googleapis.com
> Accept: */*
>
```

# Hitelesítés (10)

- Példa (folytatás):

```
< HTTP/1.1 401 Unauthorized
< Vary: X-Origin
< WWW-Authenticate: Bearer
 realm="https://accounts.google.com/AuthSubRequest"
< Content-Type: application/json; charset=UTF-8
< Date: Sun, 01 Feb 2015 10:31:54 GMT
< Expires: Sun, 01 Feb 2015 10:31:54 GMT
< Cache-Control: private, max-age=0
< X-Content-Type-Options: nosniff
< X-Frame-Options: SAMEORIGIN
< X-XSS-Protection: 1; mode=block
< Server: GSE
< Alternate-Protocol: 443:quic,p=0.02
< Accept-Ranges: none
< Vary: Origin,Accept-Encoding
< Transfer-Encoding: chunked
<
< ...
```

# Hitelesítés (11)

- A hitelesítési sémák nevei kisbetű-nagybetű érzéketlenek
- A hitelesítési sémákat az IANA regisztrálja
  - Lásd: *Hypertext Transfer Protocol (HTTP) Authentication Scheme Registry*  
<http://www.iana.org/assignments/http-authschemes/http-authschemes.xhtml>

# Hitelesítés (12)

- John Franks, Phillip M. Hallam-Baker, Jeffery L. Hostetler, Scott D. Lawrence, Paul J. Leach, Ari Luotonen, Lawrence C. Stewart, *HTTP Authentication: Basic and Digest Access Authentication*, RFC 2617, June 1999.

<http://tools.ietf.org/html/rfc2617>

- **Basic authentication:** a felhasználói név és a jelszó átvitele titkosítás nélkül történik
- **Digest access authentication:** a felhasználói név és a jelszó helyett azokból és további értékekből (metódusnév, tartomány, URI, és egy, a szerver által generált egyedi karakterlánc) egy kriptográfiai hasítófüggvénnyel (MD5) számított ellenőrzőösszeg átvitele

# Basic authentication (1)

- Példa a `WWW-Authenticate` fejlécmezőben történő használatára:
  - `WWW-Authenticate: Basic realm="Authentication Required"`
- A felhasználói ágens az `Authorization` fejlécmezőben a Base64-kódolt felhasználói nevet és jelszót küldi vissza
  - *A felhasználói\_név ' : ' jelszó* karakterlánc kerül kódolásra
  - Példa (felhasználói név: `jeszy`, jelszó: `secret`):
    - `Authorization: Basic amVzenk6c2VjcmV0`

# Hitelesítés: Basic Authentication (2)

- Minden egyes olyan kérdésben meg kell adni a felhasználói nevet és jelszót, melynél hitelesítés szükséges
  - Kényelmi okokból a böngészők megjegyzik a felhasználói nevet és jelszót, a kérésekbe automatikusan beszúrnak az `Authorization` fejlécmezőt
- Nem biztonságos hitelesítési megoldás, mivel a hitelesítő adatok átvitele titkosítás nélkül történik, melyek így lehallgathatóak
  - Az átvitel közben történő lehallgatás veszélye kiküszöbölhető, ha HTTPS-t használjuk a HTTP helyett

# Basic authentication (3)

- Példa:

- `curl -v http://localhost/protected/`

```
> GET /protected/ HTTP/1.1
> User-Agent: curl/7.35.0
> Host: localhost
> Accept: */*
>
< HTTP/1.1 401 Unauthorized
< Date: Sun, 01 Feb 2015 09:56:57 GMT
< Server: Apache/2.4.7 (Ubuntu)
< WWW-Authenticate: Basic realm="Authentication Required"
< Content-Length: 455
< Content-Type: text/html; charset=iso-8859-1
<
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>401 Unauthorized</title>
</head><body>
< ...
```

# Basic authentication (4)

- Példa:

- `curl -v http://localhost/protected/ -u jeszy:secret`

```
> GET /protected/ HTTP/1.1
> Authorization: Basic amVzenk6c2VjcmV0
> User-Agent: curl/7.35.0
> Host: localhost
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Sun, 01 Feb 2015 10:09:15 GMT
< Server: Apache/2.4.7 (Ubuntu)
< Last-Modified: Mon, 07 Apr 2014 05:07:12 GMT
< ETag: "128f-4f66cd711a000"
< Accept-Ranges: bytes
< Content-Length: 4751
< Vary: Accept-Encoding
< Content-Type: text/html
<
< ...
```

# Hitelesítés (13)

- Proxy esetén a Proxy-Authenticate és Proxy-Authorization fejlécmezők szolgálnak a hitelesítésre

# Hitelesítés (14)

- Példa: DEENK elektronikus szolgáltatás elérése egyetemen kívülről
  - `curl -v \`
    - `--proxy cache.lib.unideb.hu:5128 \`
    - `--proxy-user <olvasójegy_száma>:<születési_nap_(hhnn)> \`
    - `-d "mod_disp=keres/keresdisp&\`
    - `keres_nyelv=2&keres_mod=1&\`
    - `keres_1keresoszo=proxy" \`
    - `http://eisz.szotar.org/`
    - `-o output.html`
  - Lásd: *Elektronikus szolgáltatások elérése otthonról*  
[http://hu.wiki.lib.unideb.hu/index.php/Elektronikus\\_szolg%C3%A1ltat%C3%A1sok\\_el%C3%A9r%C3%A9se\\_otthonr%C3%B3l](http://hu.wiki.lib.unideb.hu/index.php/Elektronikus_szolg%C3%A1ltat%C3%A1sok_el%C3%A9r%C3%A9se_otthonr%C3%B3l)

# Hitelesítés (15)

- Példa (folytatás):

```
> POST http://eisz.szotar.org/ HTTP/1.1
> Proxy-Authorization: Basic S0cxMjM0NTY6MDYyMg==
> User-Agent: curl/7.35.0
> Host: eisz.szotar.org
> Accept: */*
> Proxy-Connection: Keep-Alive
> Content-Length: 72
> Content-Type: application/x-www-form-urlencoded
>
> mod_disp=keres/keresdisp&keres_nyelv=2&keres_mod=1&keres_1keresozo=proxy
```

# Süti (1)

- **Süti (*cookie*):**
  - Egy név-érték pár és kapcsolódó metaadatok (attribútumok), melyeket egy eredet szerver a válaszok Set - Cookie fejlécmezőjében küld a felhasználói ágenseknek
    - Az attribútumok révén az eredet szervert egy hatáskört határozhat meg
  - A felhasználói ágensek a további kérésekben a név-érték párt a Cookie fejlécmezőben küldik vissza az eredet szervernek
- A jelenleg aktuális specifikáció:
  - Adam Barth, *HTTP State Management Mechanism*, RFC 6265, April 2011. <http://tools.ietf.org/html/rfc6265>
    - A Cookie és a Set - Cookie és fejlécmezők definiálása

# Sütik (2)

- Felhasználás:
  - Munkamenet kezelés
  - Felhasználó követés (lásd a Referer fejlécmezőt)

# Sütik (3)

- Példa:

- `curl --head http://www.bbc.co.uk/news/`

```
< HTTP/1.1 200 OK
< Server: Apache
< Expires: Fri, 20 Feb 2015 08:26:20 GMT
< Content-Language: en-GB
< X-IGOR: True
< Content-Type: text/html
< Content-Length: 168463
< Date: Fri, 20 Feb 2015 08:26:20 GMT
< Connection: keep-alive
< Set-Cookie: BBC-UID=15446e<...>4f5c40curl/7.35.0;
 expires=Tue, 19-Feb-19 08:26:20 GMT; path=/; domain=.bbc.co.uk
< Cache-Control: private, max-age=60
< X-Cache-Action: MISS
< X-Cache-Age: 0
< X-LB-NoCache: true
< Vary: X-CDN
<
```

# Sütik (4)

- Egy eredet szerver akár több sütit is küldhet egy válaszban:
  - `curl --head https://www.youtube.com/`

```
< HTTP/1.1 200 OK
< Date: Fri, 20 Feb 2015 08:43:01 GMT
< Server: gwiseguy/2.0
< X-XSS-Protection: 1; mode=block;
 report=https://www.google.com/appserve/security-bugs/log/youtube
< X-Frame-Options: SAMEORIGIN
< P3P: CP="This is not a P3P policy! See
 http://support.google.com/accounts/answer/151657?hl=hu for more info."
< X-Content-Type-Options: nosniff
< Cache-Control: no-cache
< Expires: Tue, 27 Apr 1971 19:44:06 EST
< Content-Type: text/html; charset=utf-8
< Set-Cookie: YSC=3XFpxFdHbAM; path=/; domain=.youtube.com; HttpOnly
< Set-Cookie: PREF=f1=50000000; expires=Wed, 21-Oct-2015 20:36:01 GMT;
 path=/; domain=.youtube.com
< Set-Cookie: VISITOR_INFO1_LIVE=bUNLGBfOpVs; expires=Wed, 21-Oct-2015 20:36:01 GMT;
 path=/; domain=.youtube.com; HttpOnly
< Alternate-Protocol: 443:quic,p=0.08
< Transfer-Encoding: chunked
< Accept-Ranges: none
< Vary: Accept-Encoding
<
```

# Süтик (5)

- Amikor a felhasználói ágens egy Set - Cookie fejlécmezőt kap, eltárolja az attribútumaival együtt
- A továbbiakban amikor a felhasználói ágens egy HTTP kérést hajt végre, a Cookie fejlécmezőbe illeszti az alkalmazható, nem lejárt sütiket
  - Csak a név-érték párokat, az attribútumokat nem
- Ha a felhasználói ágens egy olyan sütit kap, melynek neve, valamint Domain és Path attribútuma megegyezik egy tárolt sütiével, akkor az új sütivel felülírja a korábbi

# Süti attribútumok (1)

- A specifikáció az alábbi attribútumokat definiálja:
  - Expires
  - Max-Age
  - Domain
  - Path
  - Secure

# Süti attribútumok (2)

- A süti maximális élettartamát jelző attribútumok:
  - **Expires**: a süti maximális élettartamát jelző dátum és idő
  - **Max-Age**: a süti maximális élettartama másodpercben
- Ha egy sütinek Expires és Max-Age attribútuma is van, akkor a Max-Age élvez elsőbbséget
  - **Perzisztens sütiknek** nevezzük az Expires vagy Max-Age attribútummal rendelkező sütiket, mert ezeket a felhasználói ágens több munkameneten keresztül tárolhatja
- Ha egy sütinek nincs Expires és Max-Age attribútuma sem, akkor élettartama az aktuális munkamenet végéig tart

# Süti attribútumok (3)

- **Domain:**

- Meghatározza, hogy a süti mely szervereknek lesz elküldve
- Ha nincs megadva, akkor a felhasználói ágens csak az eredet szervernek küldi vissza a sütit
- A felhasználói ágens elutasít minden olyan sütit, melynél a Domain attribútum által meghatározott hatáskör nem tartalmazza az eredet szervert
- Biztonsági okokból sok felhasználói ágens úgy van beállítva, hogy elutasítson minden olyan sütit, melynek Domain attribútuma egy nyilvános regisztrátor ellenőrzése alatt áll (például com, co.uk, blogspot.com, ...)

- **Path:**

- A süti hatáskörét az adott útvonalra korlátozza
- Ha nincs megadva, akkor alapértelmezett értéke a kérésben adott URI útvonalának „könyvtára”

# Süti attribútumok (4)

- **Secure:**

- A süti hatáskörének biztonságos csatornákra korlátozása
- Jelenléte esetén csak olyan kérdésben küldi el a felhasználói ágens a sütit, melynek átvitele biztonságos csatornán keresztül történik
  - Mivel a süti érzékeny információt tartalmazhat és kockázatot rejt a sima szöveggént történő átvitele

- **HttpOnly:**

- HTTP kérésekre korlátozza a süti hatáskörét
- Jelenléte esetén a süti nem érhető el API-kon keresztül, például kliens oldali szkriptekből

# Sütik kezelése a kliens oldalon (1)

- A felhasználói ágensnek törölnie kell a lejárt sütiket
  - A lejárat időpontja előtt is törölhető egy süti
- Az aktuális munkamenet végén a felhasználói ágensnek törölnie kell az összes nem perzisztens sütit
- A felhasználói ágensek számára ajánlott a felhasználók számára lehetővé tenni a tárolt sütik kezelését
  - Például egy adott időszakban fogadott vagy egy adott tartományhoz kapcsolódó sütik törlését
- Ajánlott továbbá annak lehetővé tétele, hogy a felhasználók letilthassák a sütik használatát

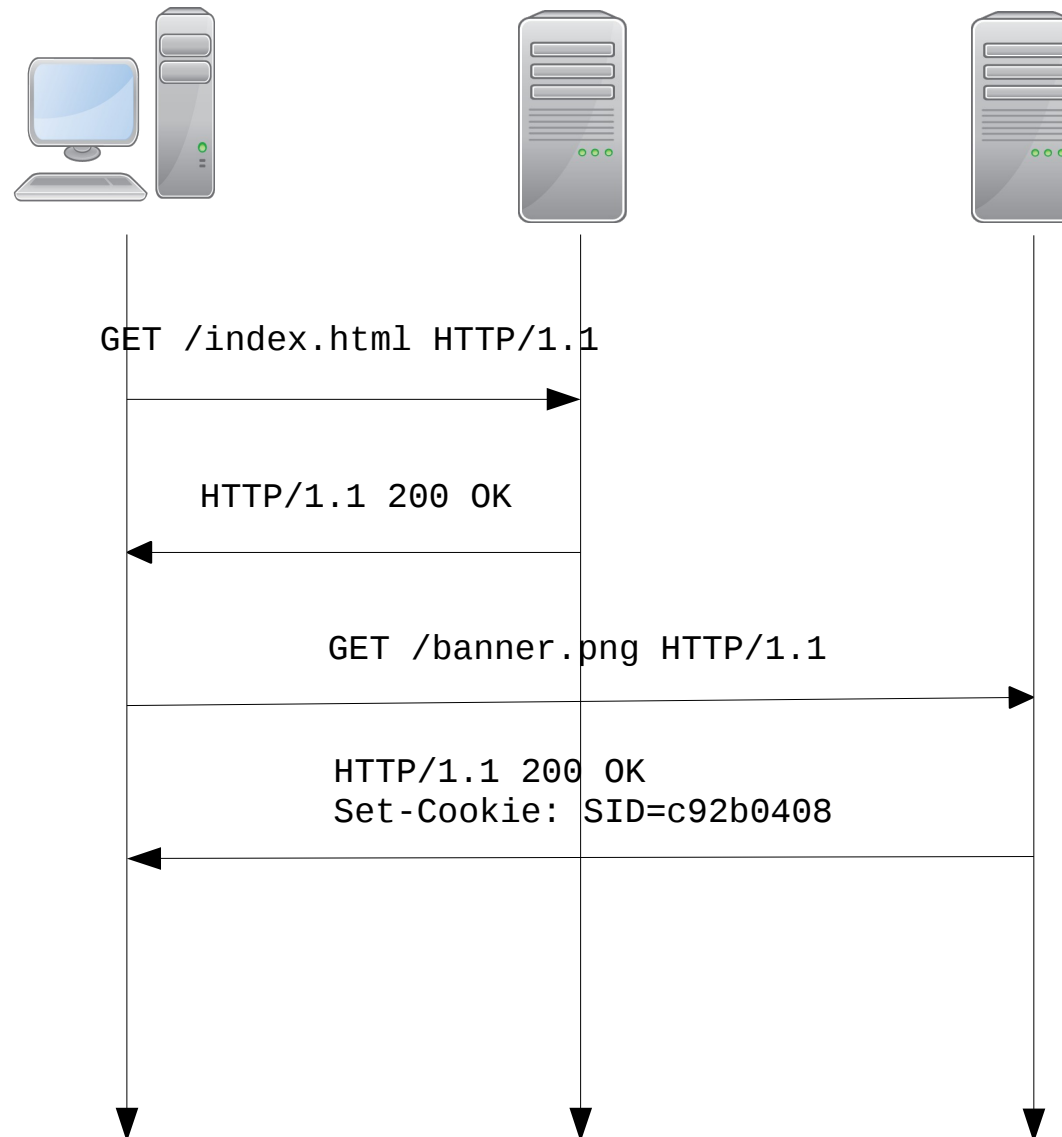
# Sütik kezelése a kliens oldalon (2)

- Az implementációk korlátozzák a tárolható sütik számát és méretét
- A specifikáció az alábbi alsó korlátokat ajánlja:
  - A sütik mérete legalább 4096 bájt
  - Tartományonként legalább 50 süti tárolása
  - Összesen legalább 3000 süti tárolása
- Lásd:
  - *Browser Cookie Limits*  
<http://browsercookielimits.squawky.net/>

# Sütik: adatvédelmi és biztonsági kérdések (1)

- A sütiket gyakran bírálják azért, mert lehetővé teszik a szerverek számára a felhasználók követését
- Különösen sok bírálat éri a **harmadik féltől származó sütiket** (*third-party cookies*)
  - Egy HTML oldal megjelenítésénél egy felhasználói ágens gyakran kér le erőforrásokat más szerverekről
  - Ezek a kívülálló szerverek sütiket használhatnak a felhasználó követésére még akkor is, ha a felhasználó közvetlenül soha nem látogatja meg őket

# Sütik: adatvédelmi és biztonsági kérdések (2)



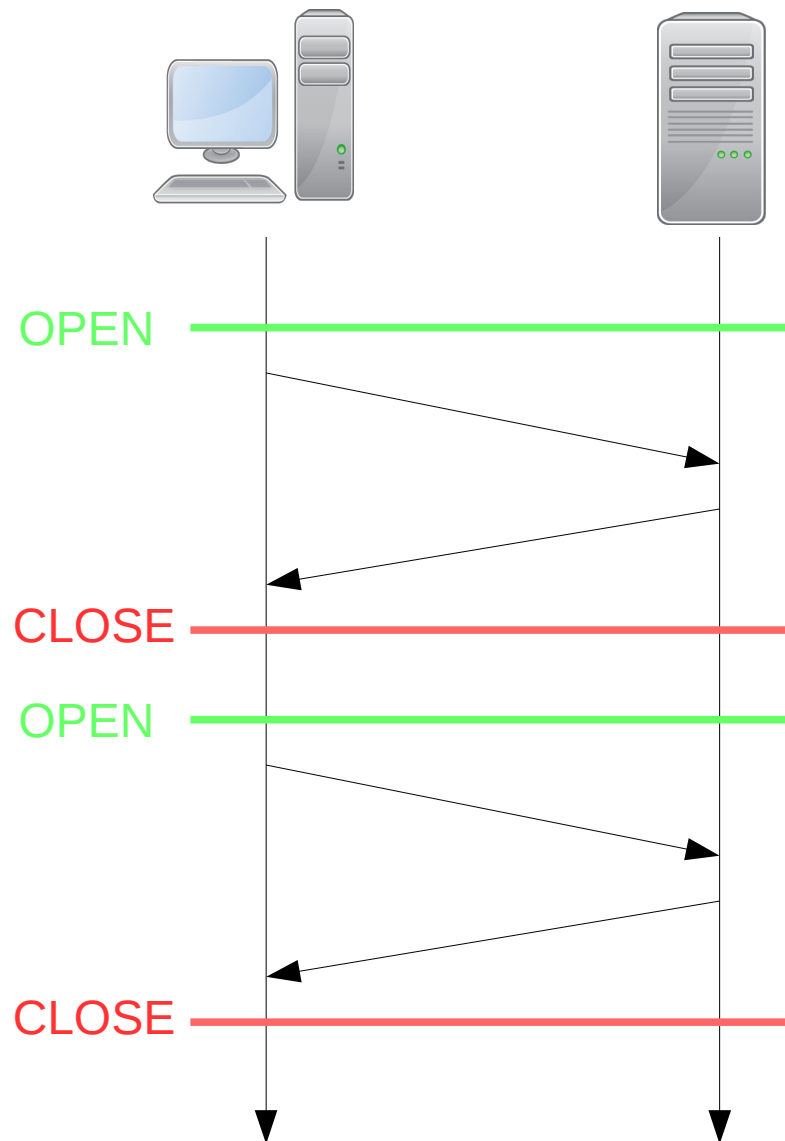
# Sütik: adatvédelmi és biztonsági kérdések (3)

- A jelenleg érvényes EU-s szabályozás:
  - **2002/58/EK**: *Elektronikus hírközlési adatvédelmi irányelv* (2002. július 12.)  
<http://eur-lex.europa.eu/legal-content/HU/TXT/?uri=CELEX:32002L0058>
  - **2009/136/EK**: *Az Európai Parlament és a Tanács 2009/136/EK irányelve* (2009. november 25.)  
<http://eur-lex.europa.eu/legal-content/HU/TXT/?uri=CELEX:32009L0136>
    - Lásd az 2002/58/EK irányelv 5. cikkének (3) bekezdésének módosítását
    - Ez azt jelenti, hogy süti csak akkor helyezhető el a felhasználó számítógépén, ha ehhez a felhasználó tájékoztatás alapján előzetes hozzájárulását adja
  - **WP 194**: *2012/4. számú vélemény a sütikhez való hozzájárulás alóli mentességről* (2012. június 7.)  
[http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2012/wp194\\_hu.pdf](http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2012/wp194_hu.pdf)

# Süтик: adatvédelmi és biztonsági kérdések (4)

- Hacsak nem biztonságos csatornán (például TLS) keresztül történik a kommunikáció, a Cookie és a Set - Cookie fejlécekben nyílt szöveggként kerülnek továbbításra az információk
- A fejlécekben átvitt érzékeny információ lehallgatható, rosszindulatú közvetítő módosíthatja
- Ajánlott a szerverek számára a sütik tartalmának titkosítása és aláírása (titkosított csatorna esetén is)
- Biztonságos csatorna esetén a szerver számára ajánlott a Secure attribútum beállítása minden sütihez

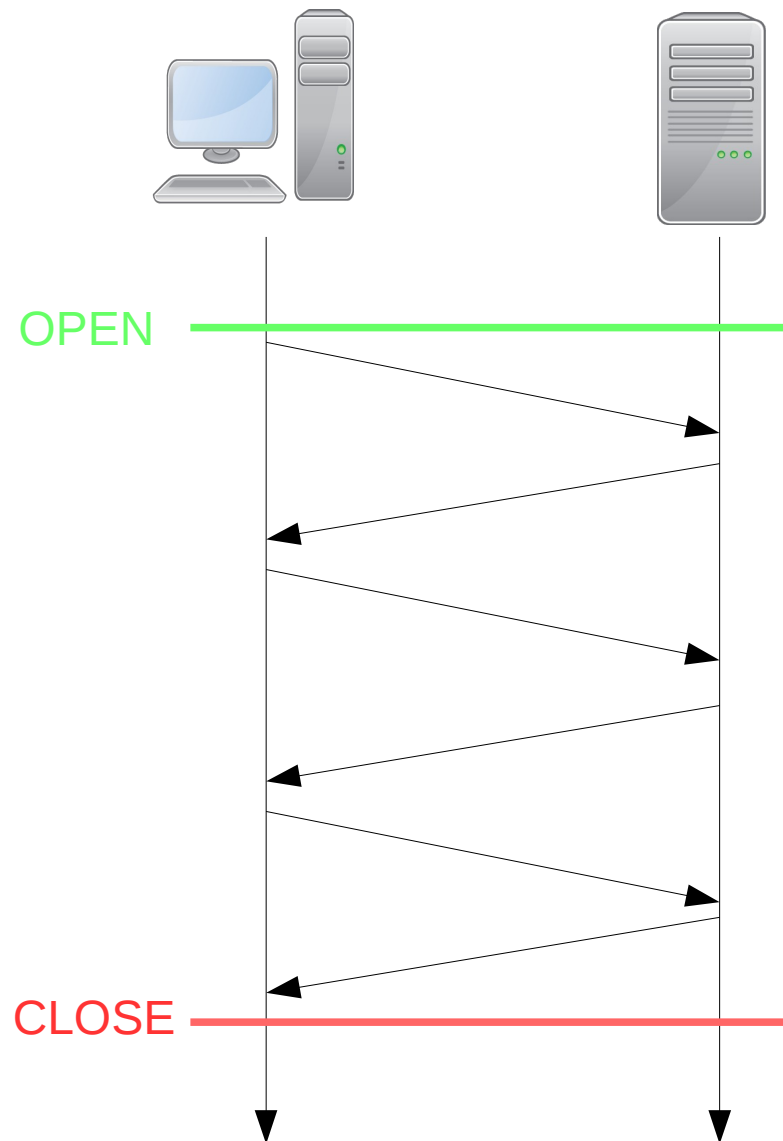
# HTTP/1.0 kapcsolatkezelés



# Perzisztens kapcsolatok (1)

- A HTTP/1.1 vezette be, lehetővé teszi több kérés végrehajtását ugyanazon a TCP kapcsolaton keresztül
- Perzisztens kapcsolatok használata alapértelmezésben
- Korábban minden kéréshez egy új kapcsolat létrehozása volt szükséges

# Perzisztens kapcsolatok (2)



# Kapcsolatkezelés: beállítások

- A `Connection` fejlécmezőben jelezheti az üzenetküldő az aktuális kapcsolatra vonatkozó beállításait
  - A fejlécmező értéke kisbetű-nagybetű érzékeny opciók listája

# Kapcsolatkezelés

- A HTTP üzenetküldés független az alapul szolgáló szállítási réteg vagy viszony réteg protokolltól, csupán kérések és a válaszok megfelelő sorrendű biztonságos továbbítását feltételezi
- Egy HTTP interakcióhoz használandó kommunikációs protokollt a kliens beállításai és az URI határozzák meg
  - Például a `http` URI séma TCP/IP kapcsolatot jelez, alapértelmezésben a 80-as számú TCP porton
    - A kliens beállítható úgy, hogy egy proxyt használjon más porton vagy protokollon keresztül

# Kapcsolatkezelés: kérések újrapróbálása

- Egy kapcsolat bármikor lezáródhat, szándékosan vagy szándékolatlanul (hiba)
- Ha a kapcsolat idő előtt lezárásra kerül, akkor a kliens létrehozhat egy új kapcsolatot és automatikusan újraküldheti kérések egy megszakított sorozatát, ha minden kérés metódusa idempotens
- Nem idempotens metódus esetén csak akkor küldhető újra egy kérés
  - ha a felhasználói ágens valamilyen módon tudja, hogy a kérés szemantikája ténylegesen idempotens a metódustól függetlenül,
  - vagy ha valamilyen módon érzékelni képes, hogy az eredeti kérés soha nem került végrehajtásra

# Kapcsolatkezelés: pipelining (1)

- Perzisztens kapcsolatokat támogató kliens egymás után több kérést is elküldhet anélkül, hogy megvárná a válaszokat
- A szerver a kéréseket párhuzamosan is feldolgozhatja, ha minden egyes kérés metódusa biztonságos, de a válaszokat a kérések sorrendjében kell, hogy elküldje



# Kapcsolatkezelés: pipelining (3)

- Ha a kapcsolat még azelőtt lezárul, hogy kéréseket csővezetékező kliens az összes választ megkapná, akkor megkísérelheti újra elküldeni a megválaszolatlan kéréseket.
- Az idempotens metódusok csővezetékezhetőek, mivel ezek automatikusan újraküldetők a kapcsolat megszakadását követően
- Nem ajánlott azonban nem idempotens metódus után kéréseket csővezetékezni addig, amíg a felhasználói ágens meg nem kapja a nem idempotens metódusra a végső választ

# Kapcsolatkezelés: pipelining (4)

- Nagymértékben csökkentheti a TCP/IP csomagok számát, mivel egy TCP/IP csomagba több HTTP kérés is becsomagolható
  - Lásd: *HTTP Pipelining FAQ*  
[https://developer.mozilla.org/en-US/docs/Web/HTTP/Pipelining\\_FAQ](https://developer.mozilla.org/en-US/docs/Web/HTTP/Pipelining_FAQ)

# Kapcsolatkezelés: pipelining (5)

- Böngésző támogatás:
  - **Firefox (Gecko)**: támogatott, de alapértelmezésben ki van kapcsolva
    - Lásd a `network.http.pipelining` opciót (`about:config`)
  - **Chromium (Blink)**: eltávolították az engedélyezésre szolgáló opciót
    - Lásd:  
<http://www.chromium.org/developers/design-documents/network-stack/http-pipelining>
  - **Opera (Blink)**: támogatott és alapértelmezésben be van kapcsolva
    - Lásd: *Opera Browser Wiki – The HTTP/1.1 Protocol*  
<http://operawiki.info/httpprotocol>
  - **Internet Explorer (Trident)**: nem támogatott

# Kapcsolatkezelés: kapcsolat bontása

- Explicit kapcsolatbontás:
  - A `Connection` fejlécmező biztosít egy `close` opciót, mellyel az üzenetküldő jelezheti, hogy az aktuális kérés/válasz után le kívánja zárni a kapcsolatot
  - A kliens és a szerver is kezdeményezheti általa a kapcsolat lezárását
  - Használat:
    - `Connection: close`
- Időtúllépés:
  - A szerverek általában egy bizonyos várakozási idő letelte után lezárják az inaktív kapcsolatokat

# Kapcsolatkezelés: konkurens kapcsolatok (1)

- A szerverek akár több ezer kapcsolatot képesek egyidejűleg kezelni
- A legtöbb kliens több kapcsolatot kezel párhuzamosan, egy szerverhez akár többet is
  - Jellemzően a **sor eleji blokkolás** (*head-of-line blocking*) problémájának elkerüléséhez használnak több kapcsolatot

# Kapcsolatkezelés: konkurens kapcsolatok (2)

- A HTTP korábbi kiadásai felső korlátot határoztak meg az egy adott szerverhez egyidejűleg megnyitható kapcsolatok számára
  - RFC 2616: a felső korlát 2
- Ez sok alkalmazás számára célszerűtlennek bizonyult, ezért az aktuális specifikáció nem ír elő korlátot, de azt ajánlja a kliensek számára, hogy legyenek óvatosak több kapcsolat megnyitásakor
  - Mivel minden egyes kapcsolat a szerver erőforrásait fogyasztja, a szerver megtagadhatja a kiszolgálást, ha egy kliens túl sok kapcsolatot nyit meg

# Kapcsolatkezelés: konkurens kapcsolatok (3)

- Korlátok a kapcsolatok számára:
  - **Firefox (Gecko)**: lásd az alábbi opciókat (`about:config`):
    - `network.http.max-connections` (alapértelmezett érték: 256)
    - `network.http.max-persistent-connections-per-proxy` (alapértelmezett érték: 32)
    - `network.http.max-persistent-connections-per-server` (alapértelmezett érték: 6)
  - **Chromium (Blink)**: a Firefox fenti opciónál adott alapértelmezett értékek használata rögzített módon
    - Lásd:  
<http://www.chromium.org/developers/design-documents/network-stack#TOC-Connection-Management>
  - **Internet Explorer (Trident)**:
    - Lásd: *maxConnectionsPerServer* property  
<https://msdn.microsoft.com/en-us/library/ie/cc197013%28v=vs.85%29.aspx>
- Lásd még: <http://www.browserscope.org/?category=network>

# Webszerver szoftverek (1)

- Webszerverek szoftverek piaci részesedése:
  - *Netcraft Web Server Survey*  
<http://news.netcraft.com/archives/category/web-server-survey/>

# Webszerver szoftverek (2)

- A legelterjedtebben használt webszerver szoftverek:

Szoftver	Platform	Licenc	Megjegyzés
Apache HTTP Server <a href="http://httpd.apache.org/">http://httpd.apache.org/</a>	platformfüggetlen	Apache License 2.0	
nginx <a href="http://nginx.org/">http://nginx.org/</a>	platformfüggetlen	Simplified BSD License	<ul style="list-style-type: none"><li>• Kiejtés: <i>engine x</i></li><li>• Webszerver és fordított proxy</li><li>• Felhasználók: Dropbox, GitHub, Instagram, Netflix, Pinterest, SoundCloud, SourceForge, ...</li></ul>
Internet Information Services (IIS) <a href="http://www.iis.net/">http://www.iis.net/</a>	Windows	nem szabad szoftver	
Google Web Server (GWS)	Linux?	nem szabad szoftver	A Google által használt saját fejlesztésű webszerver

# Webszerver szoftverek (3)

- Néhány további webszerver szoftver:

Szoftver	Platform	Licenc	Megjegyzés
Apache Traffic Server <a href="http://trafficserver.apache.org/">http://trafficserver.apache.org/</a>	Unix	Apache License 2.0	Eredetileg a Yahoo! által fejlesztett gyorsítótárazó fordított proxy szerver
Jetty <a href="http://eclipse.org/jetty/">http://eclipse.org/jetty/</a>	Java	Apache License 2.0/Eclipse Public License 1.0	
Squid <a href="http://www.squid-cache.org/">http://www.squid-cache.org/</a>	platformfüggetlen	GNU GPL v2	Gyorsítótárazó proxy szerver
Varnish <a href="https://www.varnish-cache.org/">https://www.varnish-cache.org/</a>	Unix	Simplified BSD License	Gyorsítótárazó fordított proxy szerver