Svetovni Splet - World Wide Web

Izrazoslovje

- Spletni pregledovalnik / brskalnik = browser
- Splet = Web (samostalnik)
- spletni = web (pridevnik)
- Svetovni splet = World Wide Web

WWW - World Wide Web

HTTP ODJEMALEC (SPLETNI PREGLEDOVALNIK)

HTTP STREŽNIK (SPLETNI STREŽNIK)

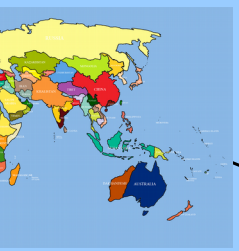
HTTP Request

Spošt. strežnik, pošlji mi, prosim, datoteko X

HTTP Response

Spošt. odjemalec, tu sledi datoteka X,

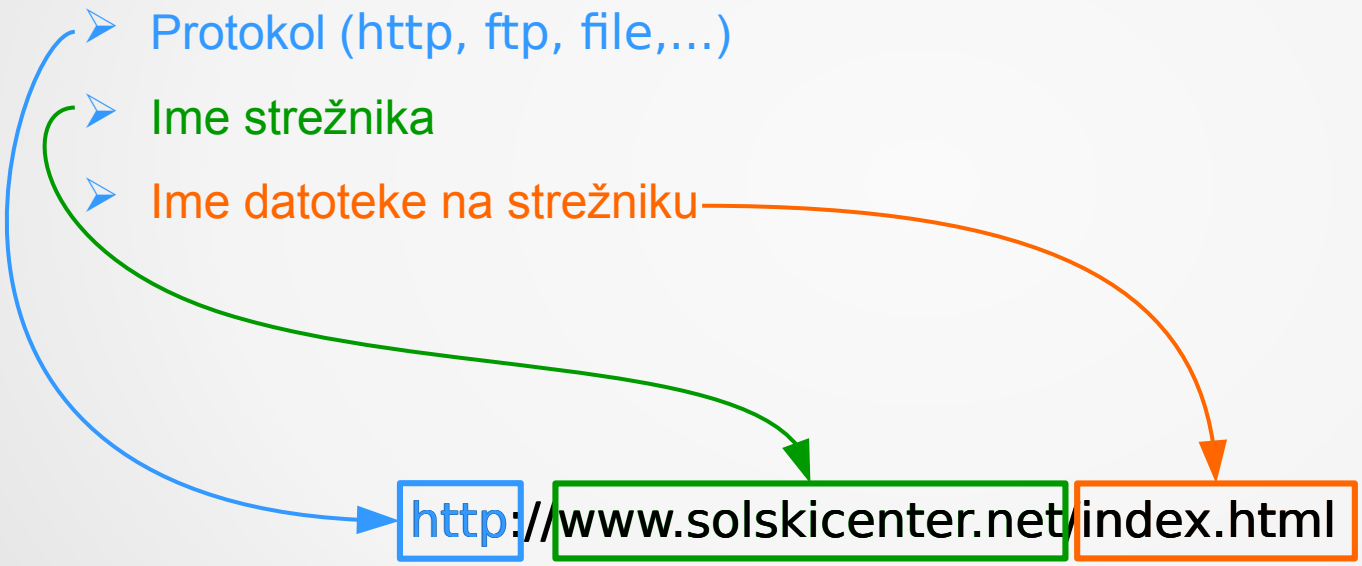
za katero si me zaprosil



URI

Uniform Resource Identifier (URI) - enolični identifikator vira:

Enolično določa (identificira) vir (naslov, besedilo, datoteko, storitev, naslov elektronske pošte, itd.) v svetovnem spletu:



URI, URL in URN (I)

- **Uniform Resource Identifier (URI) – enolični identifikator vira**
  - Niz znakov, ki enolično določajo (identificirajo) **vir** v svetovnem spletu. Ta vir je lahko naslov, besedilo, datoteka, storitev, naslov elektronske pošte, itd.
- **Uniform Resource locator (URL) – enolični krajevnik vira**
  - Niz znakov, ki enolično določajo (identificirajo) **naslov** vira (besedila, slike, video posnetka) v svetovnem spletu.
- **Uniform Resource Name (URN) – enolično ime vira**
  - Niz znakov, ki enolično določajo (identificirajo) **ime** vira znotraj istega imenskega prostora, tj. znotraj neke množice imen, ki so si med seboj različna.

URI, URL in URN (II)

- **Razlikovanje: PAZI!!!!**
- URN je URI, ki identificira vir, vendar, z razliko od URL-ja, ne dovoljuje identifikacije naslova samega vira. Na primer ISBN koda, ki enolično identificira knjigo, a nam ničesar pove, kje se ta knjiga nahaja.
-
- URL je URI. Tudi URN je URI.-
- URL je URI, ki ga imenujemo *spletni naslov*.
-
- Če uporabimo prispodobo, je URN ime osebe (njeno istovetnost, identiteto), URL pa je naslov hiše, kjer se tista oseba nahaja, oz. način, kako osebo najti.
-
- **V pogovornem jeziku se namesto URI pogosto uporablja besedo URL (Uniform Resource Locator), ki naj bi bila sopomenka za URI. Tehnično pa to ni pravilno.**

WWW – Statične in dinamične strani

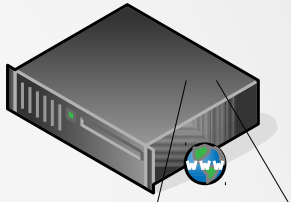
- Statična spletna mesta: strani so shranjene na strežniku v taki obliki, kot si jih bo uporabnik ogledal
- Dinamična spletna mesta: informacije se pogosto posodabljajo in spreminjajo vsakič, ko je neka spletna stran zahtevana

WWW – Statične strani

HTTP ODJEMALEC (SPLETNI PREGLEDOVALNIK)

HTTP STREŽNIK (SPLETNI STREŽNIK)

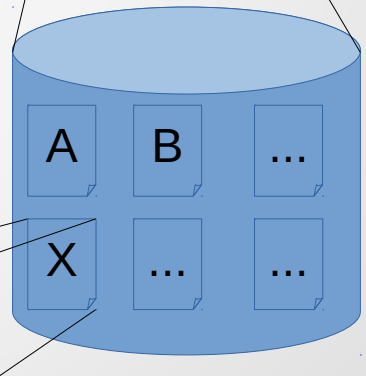
1) **HTTP Request**
Spošt. strežnik, pošlji mi, prosim, datoteko X

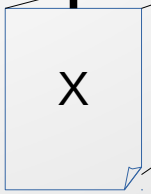


HTTP Response
Spošt. odjemalec, tu sledi datoteka X, za katero si me zaprosil

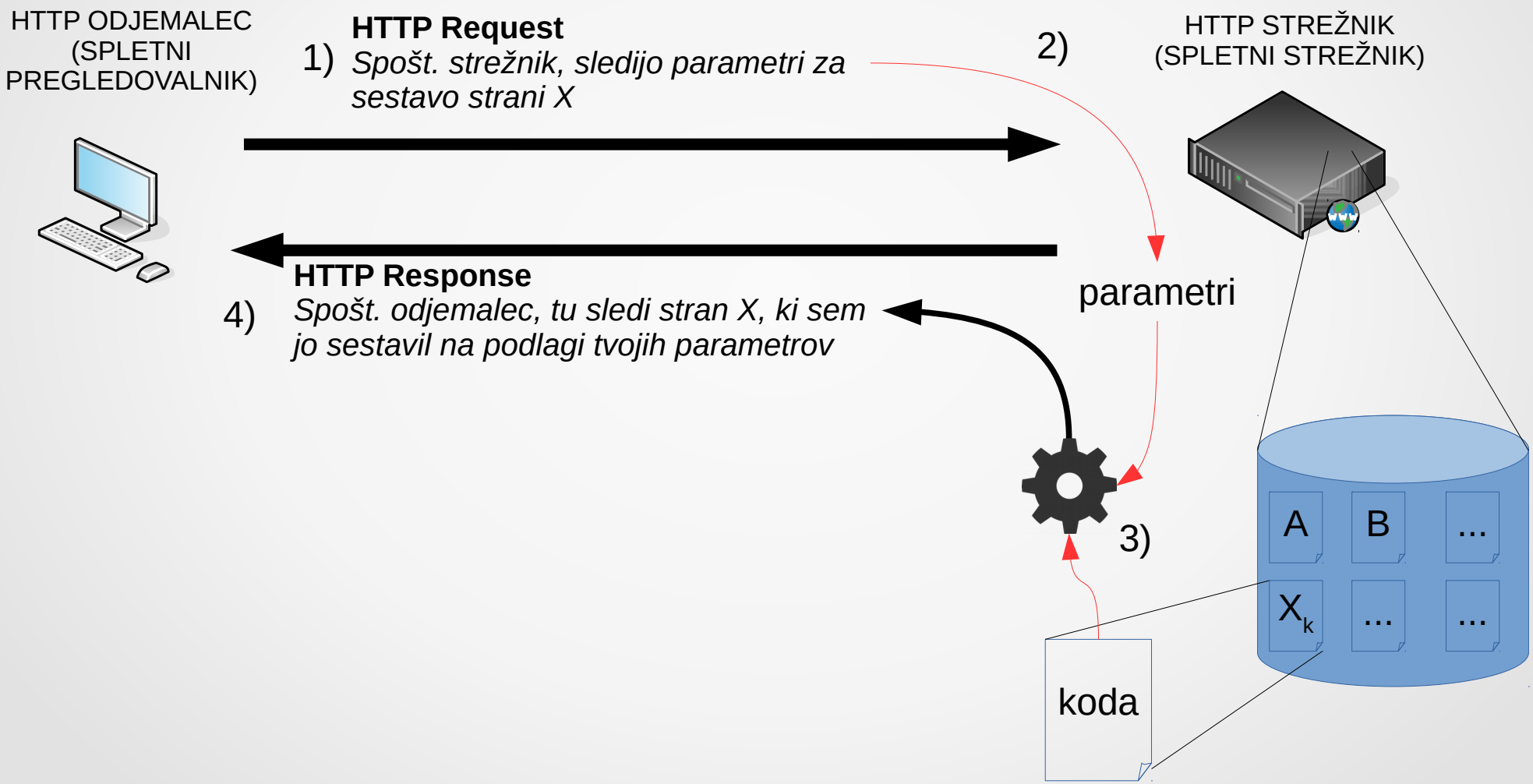
3)

2)





WWW – Dinamične strani



HTTP - Uvod

- Skoraj vse, kar vidimo v brskalniku, se prenaša po spletu preko HTTP protokola. Npr., ko odpremo določeno spletno stran, brskalnik pošlje tudi nad 40 HTTP zahtevki in sprejme HTTP odgovore na vsakega od teh.
- HTTP (**H**yper**T**ext **T**ransfer **P**rotocol - Protokol za prenos hiperbesedil) je mrežni protokol, ki se uporablja za pošiljanje vseh virov po spletu in je prvotno namenjen objavljanju in prejemanju HTML strani, z značilno arhitekturo odjemalec – strežnik (client – server).
- Viri na spletu so vse datoteke in drugi podatki kot so HTM datoteke, slike, rezultati poizvedb ali drugo.
- Razvoj HTTP protokola upravlja WWW Consortium - W3C (konzorcij za svetovni splet http://www.w3.org/)

HTTP – Delovanje (I)

- HTTP temelji na mehanizmu zahtevek/odgovor (client/server – odjemalec/strežnik): odjemalec pošlje strežniku zahtevek in strežnik mu odgovori. Navadno soupada odjemalec z *brskalnikom* (ki mu pravimo tudi *preglednik*), spletni strežnik pa s spletnim mestom (točneje, z aplikacijo, ki se izvaja na strežniku-računalniku). Potemtakem, obstajata dve vrsti HTTP sporočil: zahtevki in odgovori.
- HTTP se od drugih protokolov, kot je npr. FTP razlikuje po tem, da *se povezava zaključi, ko se z odgovorom zadosti zahtevku* (ali zaporedju zahtevkov, ki so nanj vezani). Zaradi takšnega delovanja, je HTTP protokol idealen za Svetovni Splet, saj večkrat spletne strani vsebujejo povezave (linke) na strani, ki so na drugih strežnikih in s tem mehanizmov se zmanjša število aktivnih povezav: dejansko je število omejeno na tiste povezave, ki so nujno potrebne in se tako poveča učinkovitost (manjša obremenitev) tako odjemalca kot strežnika.

HTTP – Delovanje (II)

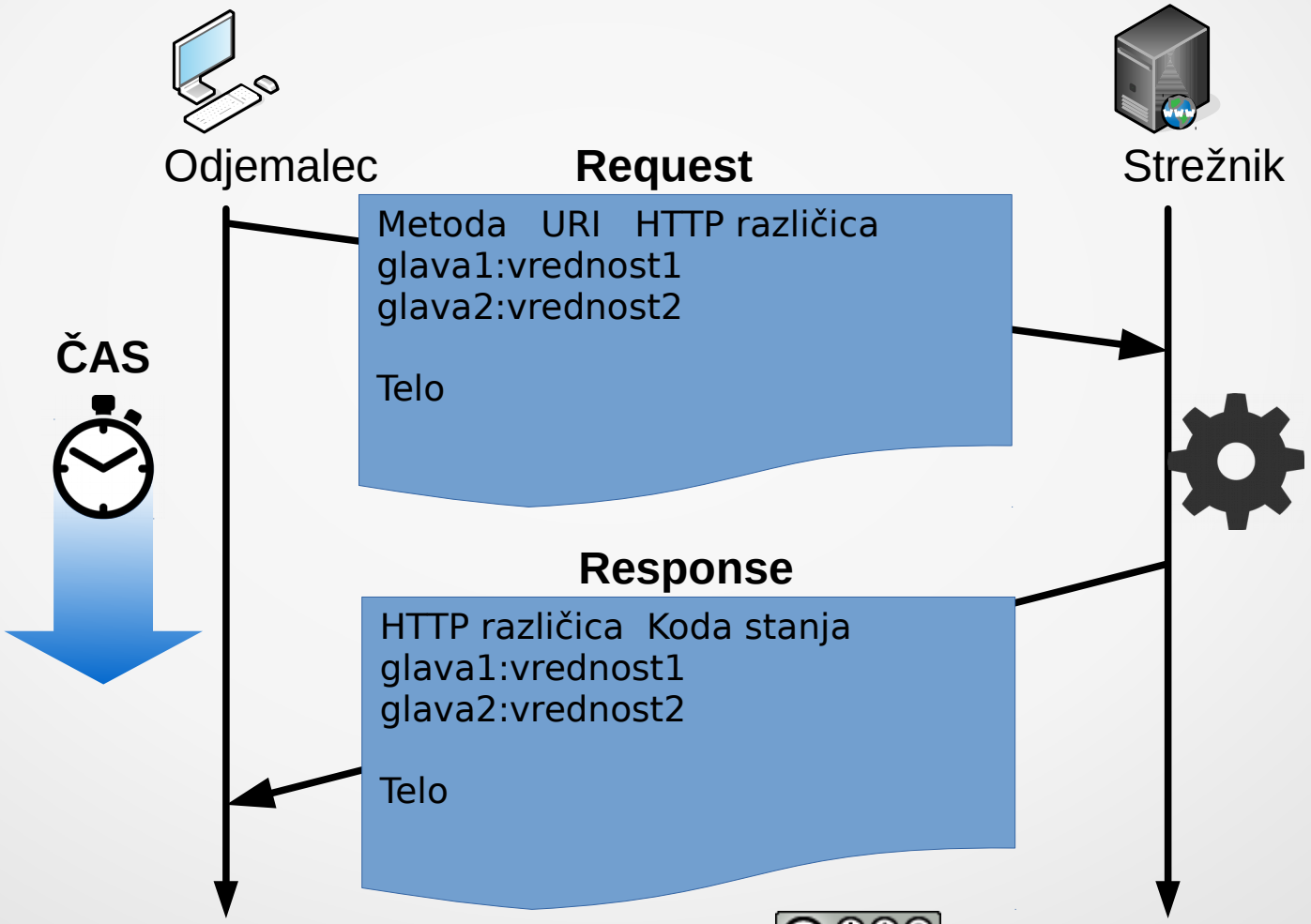
- Navadno odjemalec začne zahtevek tako, da vspostavi TCP povezavo na izbranih vratih (ali priklopu) na oddaljenem gostitelju (privzeta je številka 80). HTTP strežnik, ki na tem priklopu pričakuje, da bo odjemalec poslal svoj zahtevek na primer

```
GET / HTTP/1.1
```

- (ta zahtevek prosi za privzeto spletno stran na strežniku), čemur sledi sporočilo, ki vsebuje vrsto obveznih in neobveznih podatkov za informiranje strežnika (podatek o gostitelju *Host* je na primer obvezen), čemur lahko sledi neobvezno polje poljubnih podatkov.
- Strežnik se nato odzove s sporočilom odgovora (HTTP Response), ki običajno vsebuje vir (vsebino), za katero je zahteval odjemalec.

HTTP – Delovanje (IIa)

- Vse HTTP povezave potekajo v obliki, kot je prikazano v sledeči sliki:



HTTP – Delovanje (III)

- HTTP je protokol brez stanja (*stateless*), v katerem odjemalec pošilja zahtevke, strežnik na zahtevek odgovori in se povezava zaključi. Izraz *stateless* pomeni prav to, da **nobeden od udeleženih računalnikov ne ohranja informacij o določeni povezavi**: ko strežnik izpolni vse zahtevke odjemalca, se povezava zaključi. Odjemalec začne povezavo kot sledi:
- [Še preden odjemalec vzpostavi HTTP povezavo, mora poznati strežnikov IP naslov, ki ga pridobi s pomočjo DNS-a. Šele nato vzpostavi TCP povezavo in:]
- 1) Najprej odjemalec pošlje strežniku *sporočilo zahtevka* na primernih vratih (priklopu): za protokol HTTP je privzeto število vrat **80.**

Omrežna Vrata

- (*Omrežna*) *Vrata* (*port*) v računalništvu pomenijo vmesnik, skozi katerega pošiljamo in prejemamo podatke.
- Vrata ima vsak računalnik v omrežju, ki lahko uporablja za komunikacijo protokola UDP in TCP. Razpon števila vrat je med 0 in 65535, kjer so vrata 0 rezervirana za lokalni računalnik, razpon vrat od 1 do 1023 pa imenujemo priviligirana vrata, ki jih v GNU/Linux sistemih in v vseh sistemih UNIX in podobnih sistemih lahko uporablja samo superuporabnik, navadni uporabniki, pa lahko uporabljajo razpon vrat od 1024 do 65535. To območje vrat pa imenujemo nepriviligirano območje vrat. Območja je predpisala IANA (Internet Assigned Numbers Authority - http://www.iana.org/). Na priviligiranem območju tako navadno poslušajo (pričakujejo vzpostavitev povezave) strežniki, kot so: FTP, TELNET, SSH, POP3, SMTP, HTTP, ...
- Tako na primer odjemalec na našem računalniku lahko uporabi za povezavo vrata iz nepriviligiranega področja in se poveže na strežnik na oddaljenem gostitelju, ki uporablja vrata iz priviligiranega območja. To je precej pomembno pri gradnji požarnega zidu, saj lahko bolj natančno omejimo posamezne povezave.
- Ker je TCP povezovalni protokol, se najprej vzpostavi povezava med odjemalcem in strežnikom. Pri povezavi je določen odjemalčev naslov IP in vrata, ter strežnikov naslov IP in vrata na katerih posluša strežnik. Naslov IP povezan z določenimi vrati tvorita *vtičnico* (*socket*) in par odjemalčeve in strežnikove vtičnice tvorita povezavo TCP, ki je edinstveno določena. Glava (header) paketa TCP vsebuje izvorni naslov IP in vrata, ciljni naslov IP in vrata, zaporedna številka paketa, številka potrditve in kontrolne zastavice. Npr.:
-

HTTP - Sporočilo zahtevka

- Sporočilo zahtevka je sestavljeno iz:
  - *vrstice zahtevka* (*request line*);
  - *glav* (*headers*): dodatne informacije
  - *telesa* (*body*): vsebina sporočila: lahko je datoteka, spletna stran ali drug vir
- Telo od ostalih dveh delov ločuje prazna vrstica, kot v sledeči shemi:

<vrstica zahtevka>

Glava1: vrednost1

Glava2: vrednost2

[Druge glave...]

<----[prazna vrstica]

<morebitno telo sporočila je tukaj: lahko gre za parametre; lahko je dolgo več vrstic, ali pa več bytov (binarnih) podatkov>

HTTP - Sporočilo zahtevka - Vrstica

- Prvi del zahtevka je vrstica zahtevka (request line), oz. del sporočila, v katerem imamo nekaj osnovnih informacij o zahtevku:
  - ukaz (ali metoda), kateremu sledita
  - identifikator vira (URI) in
  - razčličica HTTP protokola.
- Npr.: GET /index.html HTTP/1.0
- V zgornjem primeru smo uporabili metodo GET, da bi zahtevali za datoteko /index.html in smo izjavili, da uporabljamo različico 1.0 HTTP protokola. Čeprav se v HTTP zahtevkih najpogosteje uporablja metoda GET, HTTP podpira še druge metode, ki - dejansko - točneje opišejo namen povezave. Mi bomo obravnavali izključno metodi GET in POST, o katerih bomo podali nekaj primerov in obrazložili njihovo uporabo.

HTTP - Vrstica zahtevka

- Prvi del zahtevka je vrstica zahtevka (request line), oz. del sporočila, v katerem imamo nekaj osnovnih informacij o zahtevku:
  - ukaz (ali metoda), kateremu sledita
  - identifikator vira (URI) in
  - razčličica HTTP protokola.

GET /tutorials/other/top-20-mysql-best-practices/ HTTP/1.1

HTTP - Vrstica zahtevka - Metode

- Pri različici 1.1 imamo sledeče metode:
  - **GET – najpogosteje uporabljena metoda za pridobitev vsebine datoteke.**
  - **POST - podobno kot GET, samo, da vsebuje telo zahtevka podatke, ki jih želimo sporočiti strežniku.**
  - HEAD - enako kot GET, le da na to strežnik odgovori le z glavo datoteke. Uporabno za pridobivanje meta-oznak.
  - PUT - se uporablja za pošiljanje datotek na ciljni spletni strežnik.
  - DELETE - redko uporabljen; zbriše dotično datoteko z strežnika.
  - TRACE - vrne nazaj zahtevek, zato, da lahko odjemalec ugotovi, ali so vmes kaki dodatni strežniki, ki spreminjajo zahtevke.
  - OPTIONS - vrne seznam HTTP zahtevkov, ki jih strežnik podpira. Zahtevek se uporablja takrat, ko želimo ugotoviti, ali spletni strežnik deluje.
  - CONNECT - se uporablja pri spletnih posrednikih za vzpostavitev SSL tunela (povezave).

HTTP - Vrstica zahtevka - Različica

- Prva različica protokola **HTTP/0.9**, sega proti konuc letih 80 prejšnjega stoletja in je, skupaj z jezikom HTML in z URL naslovi, jedro *World Wide Web (WWW) global information initiative*, ki jo je razvil Tim Berners-Lee na CERN-u v Ženevi za porazdelitev informacij med različnimi skupnostmi fizikov visokih energij. HTTP/0.9 Podpira samo GET ukaz in se ni nikoli uveljavil.
- Prvo dejansko uporabno različico protokola, **HTTP/1.0**, je uresničil sam Berners-Lee leta 1991 in jo predlagal kot RFC 1945 organizaciji IETF leta 1996. Ta različica je še vedno uporabljena, posebno s strani spletnih posrednikov.
- Z razširitvijo grafičnega brskalnika NCSA Mosaic, se je WWW močno razvil in kmalu so prišle na dan šibkosti različice HTTP/1.0, zlasti nezmožnost gostovanja številnih spletnih mest na istem strežniku (virtual host) in nezadostna zaščita varnostnih mehanizmov. Protokol so zaradi tega razširili z različico **HTTP/1.1**, ki so jo predložili kot *RFC* 2068 leta 1997 in jo nadalje posodobili leta 1999. Poleg tega, omogoča HTTP/1.1 *trajne povezave* (znane tudi kot keep-alive povezave) z več kot enim zahtevkom z odgovorom na povezavo, za povečanje odzivnosti in učinkovitosti.
- **Internet Engineering Task Force (IETF)** je organizacija, ki standardizira Internetne protokole: ocenjuje kandidate za potencialne standarde in vodi proces potrditve standardov.
- **Request For Comment (RFC)** je zbirka dokumentov, katerih glavni namen je, da priskrbijo informacije ali opišejo delo v teku v zvezi z določenim standardom. Kdorkoli lahko pošlje predlog standarda (draft) RFC v oceno spletni skupnosti.

HTTP - Vrstica zahtevka – Glava (I)

- Po vrstici zahtevka, imamo v sporočilu neobvezne dele, ki jim pravimo glave (*headers*): to so vrstice, ki vsebujejo dodatne informacije o konfiguracji odjemalca in o želenih lastnostih vira. Vsaka informacija je podana v svoji vrstici, in sicer v obliki:
- ime glave:vrednost
- Npr. odjemalec (brskalnik) lahko pošlje svoje ime in kodo različice ali želene lastnosti datoteke:

```
GET /tutorials/other/top-20-mysql-best-practices/ HTTP/1.1
Host: net.tutsplus.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.5)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: PHPSESSID=r2t5uvjq435r4q7ib3vtdjq120
Cache-Control: no-cache
```

HTTP - Vrstica zahtevka – Glava (II)

- Najpogostejši glavi sta:
  - **Host (Gostitelj):** ime strežnika, na katerega se nanaša URL. Je obvezna v zahtevkih, ki so v skladu z različico HTTP/1.1, ker omogoča uporabo virtualnih gostiteljev (virtual host), ki temeljijo na imenih.
  - **User-Agent:** identificira odjemalca: pregledovalnik, proizvajalec, različica, itd.

```
GET /tutorials/other/top-20-mysql-best-practices/ HTTP/1.1
Host: net.tutsplus.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.5)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: PHPSESSID=r2t5uvjq435r4q7ib3vtdjq120
Cache-Control: no-cache
```

HTTP - Sporočilo odgovora

- Sporočilo odgovora je sestavljeno iz:
  - *vrstice stanja* (*status line*);
  - *glave* (*header*);
  - *prazne vrstice*;
  - telesa(*body*) - vsebina odgovora: lahko je datoteka, spletna stran ali drug vir
- Podobno, kot pri sporočilu zahtevka, telo od ostalih dveh delov ločuje prazna vrstica, kot v sledeči shemi:

<vrstica stanja>

Glava1: vrednost1

Glava2: vrednost2

[Druge glave...]

<----[prazna vrstica]

<morebitno telo sporočila je tukaj: gre za vsebovano datoteko ali za podatke poizvedbe; lahko je dolgo več vrstic, ali pa več bytov (binarnih) podatkov>

HTTP - Sporočilo zahtevka - Vrstica Stanja

- Prvi del odgovora je vrstica stanja (status line), oz. del sporočila, v katerem imamo nekaj osnovnih informacij o odgovoru:
  - razčličica HTTP protokola in
  - *koda stanja* (*status code*).

Npr.: HTTP/1.1 200 OK

- V zgornjem primeru odgovori strežnik odjemalcu, da je zahtevek bil uspešen
- Vrstica stanja vsebuje kodo, ki predstavlja stanje vira, za katerega je zaprosil odjemalec. Prva številka kode predstvalja enega od petih razredov (vrst) odgovora (**xx** pomeni, da nadomestimo z dvema številkama):
  - **1xx** Informational - Informativne kode
  - **2xx** Success - Uspeh
  - **3xx** Redirection - Preusmeritev
  - **4xx** Client Error - Napaka odjemalca
  - **5xx** Server Error - Napaka strežnika

HTTP - Vrstica Stanja 1xx in 2xx

- **1xx Informational - Informativne kode:** Zahtevek prejet, nadaljuj obdelavo.
- **100 Continue**: Strežnik je prejel glavo (header) zahtevka in odjemalec mora poslati telo zahtevka (navadno s POST metodo).

- **2xx Success – Uspeh**: Strežnik je uspešno prejel, razumel in sprejel zahtevek, zahtevek je pravilen.
- **200 OK:** Standardni odgovor za uspešne HTTP zahtevke.
- 202 Accepted: Zahtevek za obdelavo je bil sprejet, a obdelava se ni še zaključila.
- 204 No Content: Strežnik je zadostil zahtevku, a v odgovoru ni nikakršnega telesa.

HTTP - Vrstica Stanja 3xx

- **3xx Redirection – Preusmeritev:** Odjemalec mora izvesti dodatna dejanja za zadostitev zahtevku.

- 301 Moved Permanently: Ta in nadaljnje zahtevke bo treba preusmeriti na drug URI (ki ga določamo v glavi Location).
- **302 Found:** Zahtevani vir se začasno nahaja na različnem URI-ju. Je najpogosteje uporabljena koda.
- **303 See Other** (od HTTP/1.1 dalje): Odgovor zahtevku najdeš na drugem URI-ju z metodo GET.
- **304 Not Modified**: Če odjemalec pošlje strežniku zahtevek za določen vir in zahtevek za isto stran ponovno pošlje v zelo kratkem času, mu strežnikov odgovor s to kodo sporoči, da se od zadnjega zahtevka vir ni spremenil.
-

HTTP - Vrstica Stanja 4xx

- **4xx Client Error - Napaka odjemalca**: Zahtevek ni pravilen ali strežnik ga ne more sprejeti.

- **400 Bad Request:** Strežnik ne zadovolji zahtevka zaradi sintaktičnih nepravilnosti.
- **401 Unauthorized:** Podobno kot 403/Forbidden, namenjen za uporabo, ko je avtentikacija možna, a ni uspela.
- **403 Forbidden:** Zahteveke je legitimen, a strežnik ga noče zadovoljiti. V nasprotju s kodo 401 Unauthorized, avtentikacija nima nikakršnega učinka.
- **404 Not Found:** Zahtevani vir ni bil najden, a bo v prihodnje mogoče na voljo.
- 405 Method Not Allowed: Zahtevek vsebuje z nedovoljeno metodo. Npr. to se dogaja, kadar odjemalec napačno uporablja metodo GET za posredovanje podatki, za katere je dovoljena le metoda POST.
- 408 Request Timeout: Rok za pošiljanje zahtevka se je iztekel in strežnik je zaključil povezavo.
- 410 Gone: Označi, da vir ni več na voljo in ne bo na voljo niti v prihodnosti.

HTTP - Vrstica Stanja 5xx

- **5xx Server Error - Napaka strežnika**: Strežniku ni uspelo zadovoljiti zahtevku, čeprav je ta pravilen.

- **500 Internal Server Error:** Splošne napake, brez podrobnosti.
- 501 Not Implemented: Strežnik ne more zadovoljiti metodi zahtevka.
- **503 Service Unavailable**: Nedosegljiva storitev; strežnik začasano ne more nuditi storitve. Gre le za začasno stanje.
- 505 HTTP Version Not Supported: Strežnik ne podpira HTTP različice zahtevka.

HTTP - Sporočilo odgovora – Glava (I)

- Podobno kot pri vrstici zahtevka, imamo v sporočilu odgovora dele, ki jim pravimo glave (*headers*): to so vrstice, ki vsebujejo dodatne informacije o konfiguracji strežnika in o lastnostih vira. Vsaka informacija je podana v svoji vrstici, in sicer v obliki:
- ime glave:vrednost
- Npr. strežnik lahko pošlje lastnosti datoteke:

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.27 (Unix)  (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: »3f80f-1b6-3e1cb03b«
Accept-Ranges: bytes
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8
```

Trajnost HTTP povezave

- V HTTP/0.9 in HTTP/1.0, odjemalec pošlje zahtevek strežniku in strežnik nanj odgovori, zatem se povezava zapre. HTTP/1.1 pa je podprta tudi *trajna povezava*. To omogoča odjemalcu, da pošlje dodatne zahtevke, ne da bi bilo treba ponovno vspostavljati povezavo. To tudi omogoča pošiljanje več zahtevkom pred sprejemom odgovora.
- Tovrstno metodo včasih imenujemo »pipelining« ali »keep alive«. Nekatere odjemalci in nekateri strežniki, ki trdijo da uporabljajo HTTP/1.0 lahko podpirajo tudi trajnost povezave in strežniki, ki podpirajo HTTP/1.1, imajo lahko izklopljeno podporo za le-to. Če strežnik v svojem odgovoru poda glavo »Connection: close«, to pomeni, da bo povezavo zaprl takoj po odgovoru na prvi zahtevek.
- Tako HTTP strežniki kot odjemalci lahko zaprejo povezavo kadarkoli iz kakršnega koli razloga. To naredi HTTP protokol idealen za spletno okolje, kjer se spletne strani enega strežnika pogosto povezujejo na spletne strani z drugih strežnikov.
- Zapiranje HTTP/1.1 povezave je lahko precej daljše opravilo (od 200 milisekund do nekaj sekund) kot zapiranje HTTP/1.0 povezave, ker prva navadno rabi trajajoče zapiranje (da se primerno obdelajo še tavajoči paketki še ne odgovorjenih zahtevkov) medtem, ko druga se lahko zapre takoj, ko je poln odgovor poslan.