

HTTP

Maja Štula
ak. god. 2008/2009



HTTP



- **HyperText Transfer Protocol** je protokol aplikacijske razine koji upravlja komunikacijom između Web (HTTP) servera i Web (HTTP) klijenta.
- Verzije HTTP protokola:
 - HTTP/1.0: RFC 1945 (1996)
 - HTTP/1.1: RFC 2068 (1997)

Aplikacijska razina (application layer)
Prezentacijska razina (presentation layer)
Dijaloška razina (session layer)
Transportna razina (transport layer)
Mrežna razina (network layer)
Podatkovna razina (data link layer)
Fizička razina (physical layer)

Proces/Aplikacija HTTP, FTP, SMTP,
Komunikacija između dva računala TCP
Mrežna razina IP, ARP
Mrežni pristup Ethernet, PPP, Slip

2

HTTP



- HTTP protokol (server/klijent tipa) se obično prenosi preko TCP protokola
- U samom protokolu nije specificirano da se treba koristiti TCP, ali je specificirano da protokoli niže razine trebaju HTTP protokolu osigurati siguran i pouzdan prijenos između HTTP servera i klijenta
- HTTP server po *defaultu* sluša na portu 80, ali bilo koji drugi port se također može koristiti
- Često korištene aplikacije [HTTP servera](#) su Apache, IIS (*Internet Information Server*),

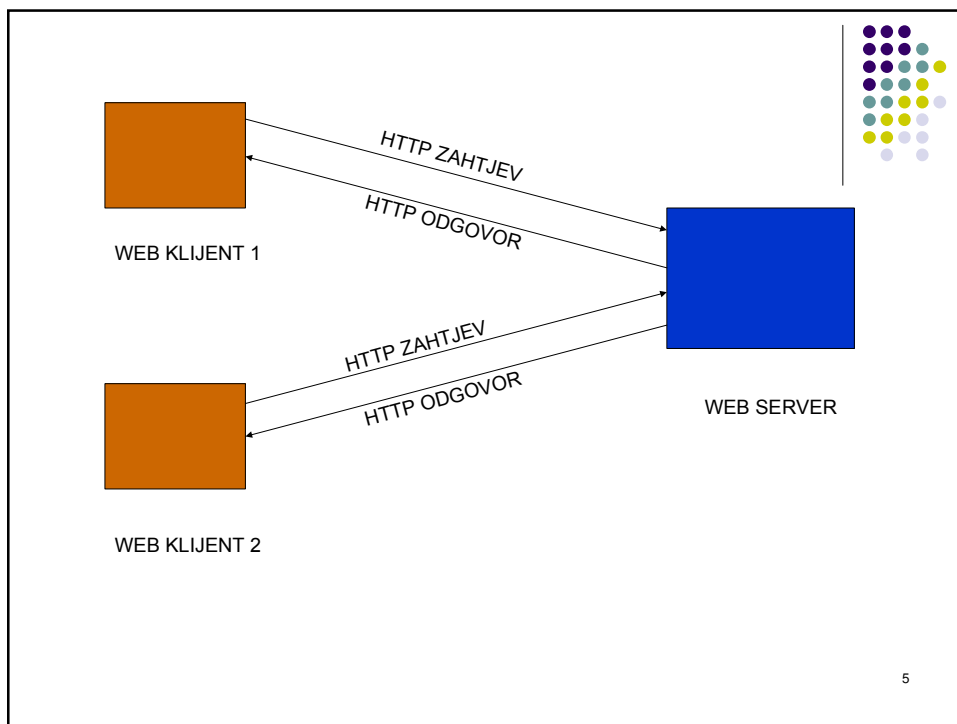
3

HTTP



- HTTP server (ili Web server ili servis) je aplikacija servera koja radi prema HTTP standardu
- HTTP klijent i HTTP server razmjenjuju HTTP **poruke** preko TCP konekcije
- Format HTTP poruka je ASCII niz karaktera odnosno tekst
- Svaka linija HTTP poruke mora završavati sa dva ASCII karaktera - CRLF “\r\n” (*line feed* (LF, \n, 10, HEX 0A) i *carriage return* (CR, \r, 13, HEX 0D)) – tim se karakterima formatira i kraj linije tekstualne datoteke na Windows OS-ovima

4



HTTP

- HTTP protokol definira:
 - Format poruke zahtjeva (*request message*)
 - Format poruke odgovora (*response message*)
 - Format zaglavlja (*header*)
 - Format statusnih kodova (*status codes*)
 - Format *cookie*-ja

6

HTTP



- HTTP protokol je *stateless* i *connectionless* protokol.
- Jedan od nedostataka HTTP protokola je to što je HTTP protokol *stateless* protokol.
- Svaki HTTP zahtjev za HTTP server je posebna naredba.

7

HTTP *cookie*



- Jedna od metoda praćenje pojedine sesije (komunikacije između klijenta i servera) je *cookie*.
- HTTP *cookie* ili Web *cookie* je skup informacija u tekstualnom obliku koje server pošalje klijentu (IE, Firefox,...) i koje klijent onda šalje nazad tom istom serveru u sklopu svakog budućeg zahtjeva.

8

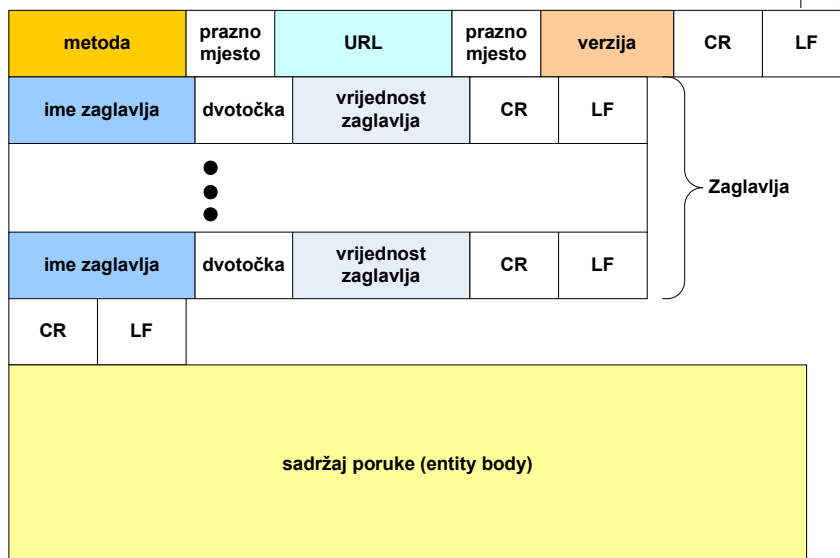
HTTP zahtjev

- HTTP zahtjev šalje klijent serveru.
- HTTP protokol točno definira kakav format treba imati ispravni HTTP zahtjev.
- Format HTTP zahtjeva je:
 - Linija zahtjeva
 - Zaglavlja (nijedno, jedno, više)
 - Prazna linija
 - Sadržaj poruke



9

HTTP zahtjev



10

HTTP zahtjev



```
POST /cgi-bin/post-query HTTP/1.0
Accept: text/html
Accept: video/mpeg
Accept: image/gif
Accept: application/postscript
User-Agent: Lynx/2.2 libwww/2.14
From:
Content-type: application/x-www-form-urlencoded
Content-length: 150
* PRAZNA LINIJA *
org=CyberWeb%20SoftWare
&users=10000
&browsers=lynx
```

```
LINIJA ZAHITJEVA
ZAGLAVLJE ACCEPT
ZAGLAVLJE ACCEPT
ZAGLAVLJE ACCEPT
ZAGLAVLJE ACCEPT
ZAGLAVLJE USER-AGENT
ZAGLAVLJE FROM
ZAGLAVLJE CONTENT-TYPE
ZAGLAVLJE CONTENT-LENGTH
PRAZNA LINIJA (\r\n)
SADRŽAJ PORUKE .....
.....
```

Linija zahtjeva sadrži HTTP metodu, prazno mjesto (*space*), URL koji se dohvaća, prazno mjesto i verziju HTTP protokola po kojoj je zahtjev složen.

Kako je HTTP protokol ASCII niz znakova kao delimiteri između dijelova HTTP poruke koriste se ASCII znakovi (*space*, *carriage return*, *line feed*)

HTTP 1.0 zahtjev ne treba imati niti jedno zaglavlje, dok HTTP 1.1 zahtjev treba¹¹ imati barem zaglavlje "Host".

URI (URL)



- URL (*Uniform Resource Locator*) je pojam koji se vrlo često koristi umjesto pojma URI (*Uniform (Universal) Resource Identifier*) mada to nisu istoznačni pojmovi.
- URL je ustvari samo dio URI-ja (*Uniform Resource Identifier*).
- URI je standard koji definira jednoznačnu sintaksu za definiranje globalnih identifikatora dokumenata (ne samo HTML dokumenata) koji se nalaze na mreži kako bi im se preko tog jedinstvenog identifikatora moglo pristupiti.
- Globalni identifikator je string koji specificira način pristupa nekom resursu na mreži odnosno jednoznačno identifikira resurs na mreži kao što je na primjer HTML dokument.
- Korisnik u pretraživač unosi URL da bi dohvatio željenu stranicu, sliku, dokument, ... (<http://www.fesb.hr/index.html>).

URI (URL)



- Premda u početku Weba nema formalne specifikacije termin *Uniform Resource Locator* označava adresu dokumenta dok se termin URN (*Uniform Resource Name*) koristi za ime resursa. (<http://www.fesb.hr/index.html> ↔ index.html)
- Naime pokazala se potreba da se razlikuje string koji je samo ime resursa i string koji je ustvari adresa resursa (jedinствена).
- RFC 1630 iz 1994. (Tim Berners-Lee) uvodi termin URI koji najbolje definira svojstvo jedinstvenog identifikatora, ali je termin URL i dalje uglavnom u upotrebi.
- U toj prvoj specifikaciji definira se i URL i URN te URI kao formalna sintaksa za definiranje *URL-like* stringova čija je sintaksa i semantika ovisila o korištenoj shemi (protokolu).
- Također se navode sintakse URL shema (protokola) koje su tada u upotrebi bez formalne specifikacije.

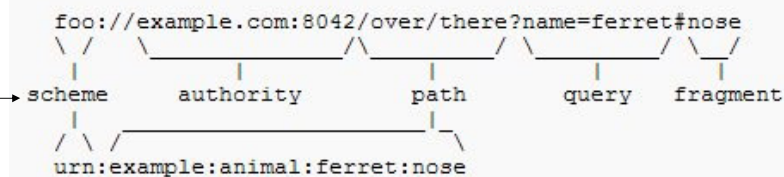
13

URI (URL)



- Naime URL omogućava pristup resursima sa različitim načinima imenovanja (*naming schemes*) tj. načinima pristupa (HTTP, FTP, e-mail) na jednostavan način. Shema je navedena prije znaka :
- Format URL-a <protokol>:<adresa>
mailto:maja.stula@fesb.hr
Za FTP, HTTP, HTTPS
<protokol>://<server>[:<port>]/<put do dokumenta>
<http://laris.fesb.hr/index.html>
<http://laris.fesb.hr:80/index.html>

Pet
komponenti
URI-ja



Trenutno registrirane URI sheme http://en.wikipedia.org/wiki/URI_scheme
ili na <http://www.iana.org/assignments/uri-schemes.html>

14

URI (URL)



- RFC 2396 iz 1998. odvaja specifikaciju URI sintakse od URL-a i time RFC-ovi 1630 i 1738 postaju zastarjeli.
- URL standard je dokumentiran u RFC 1738.
- Dijelovi RFC 1738 koji su predstavljali trenutno postojeće URL sheme nisu definirani kao zastarjeli novim RFC 2396.

15

URI (URL)



- 1999. RFC 2732 je neznatno promijenio RFC 2396 proširujući URI standard za IPV6 adrese.
- U siječnju 2005 izdan je RFC 3986 koji je trenutna verzija standarda URI-ja tako da je i RFC 2396 zastario mada sheme koje su definirane u starijim standardima (1738 i drugi) i dalje vrijede.
- U kolovozu 2002 RFC 3305 je termin URL-a proglasio zastarjelim.
- Koristi se samo da bi se naglasilo da je URI adresa koja u sebi sadrži shemu koja definira mogućnost nekakvog mrežnog pristupa dokumentu bez obzira koristi li se URI baš u svrhu adresiranja dokumenta.

16

URI referenca



- URI referenca je ili apsolutni URI ili relativna referenca dokumenta.
 - Kod HTML dokumenta vrijednost *src* atributa HTML *img* elementa je URI referenca.
 - Kod XML dokumenta nakon ključne riječi SYSTEM u DTD je URI referenca.
 - Kod XSLT vrijednost *href* atributa *xsl:import* elementa je URI referenca.

17

URI referenca



- URI referenca može imati oblik čitavog URI, ili dijela specifikacije sheme ili čak praznog stringa.
 - `http://example/resource.txt#frag01`
 - `http://somehost/absolute/URI/with/absolute/path/to/resource.txt`
 - `/relative/URI/with/absolute/path/to/resource.txt`
 - `relative/path/to/resource.txt`
 - `../../resource.txt`
 - `resource.txt`
 - `/resource.txt#frag01`
 - `#frag01`
 - *(empty string)*

18

Apsolutni ↔ Relativni URI



- Relativni URI specificira samo dio URI-ja, a ostatak URI-ja se nasljeđuje od ostatka dokumenta. Apsolutni URI sadrži potpunu adresu dokumenta.
- **Apsolutni URI**
 - `http://somehost/absolute/URI/with/absolute/path/to/resource.txt`
 - `ftp://somehost/resource.txt`
 - `urn:issn:1535-`
- **Relativni URI**
 - `/relative/URI/with/absolute/path/to/resource.txt`
 - `relative/path/to/resource.txt`
 - `../../resource.txt`
 - `resource.txt`

19

URL kodiranje



- URL standard (RFC 1738) omogućava korištenje samo dijela US-ASCII karakter skupa (brojeve, slova i neke specijalne karaktere).
- HTML standard dopušta korištenje svih karaktera ISO-8859-1 karakter skupa dok HTML verzija 4 proširuje skup dopuštenih karaktera na Unicode karakter skup.
- Karakter koji se koriste u definiranju URL-a u HTML kodu (npr. elementi `<form action="neki url">`, ``...), a koji nisu u skupu dozvoljenih karaktera za URL trebaju se kodirati. Također se trebaju kodirati i karakteri koji su dozvoljeni u URL, ali su to specijalni, rezervirani karakteri. Npr. karakter `/` se smije koristiti isključivo za definiranje puta do dokumenta (`/direktorij/poddirektorij/index.html`).
- URL kodiranje ili kodiranje sa postotcima (*percent encoding*) kodira karakter na način da ga zapiše sa njegovom ASCII vrijednošću zapisanom sa dvije heksadecimalne znamenke ispred kojih se stavlja karakter `%`.
- Npr. specijalni karakter space (razmak) se kodira sa `%20` (ili sa znakom `+`). Primjer: `http://www.fesb.hr/~kiki/proba2.htm`

20

HTTP metode



- Polje metode u HTTP liniji zahtjeva označava koja će se metoda primijeniti na objektu identificiranom sa URL-om.
- GET metoda je uvijek podržana na svim resursima identificiranim navedenim URI-jem (za sve objekte).
- Popis ostalih metoda koje su prihvatljive za neki objekt se može dobiti u HTTP odgovoru.
- Lista trenutno postojećih metoda se može proširivati (HTTP protokol je proširiv (*extensibility*)).
- Imena metoda (GET, POST, ...) su *case sensitive* – ne može se pisati Get i sl.

21

HTTP metode



- HTTP standardne metode trenutno definirane su:
 - OPTIONS
 - GET
 - HEAD
 - POST
 - PUT
 - DELETE
 - TRACE

22

OPTIONS



- OPTIONS metoda je zahtjev za informacijama o HTTP metodama koje su na raspolaganju za određeni URI.
- Ova metoda omogućava klijentu da odredi opcije i zahtjeve vezane uz resurs ili mogućnosti servera da dohvati neki resurs.
- Ukoliko odgovor servera nije greška, odgovor NE SMIJE uključivati nikakvu informaciju o objektu osim informacije koja se odnosi na komunikaciju.
- Odgovor na zahtjev s ovom metodom se ne kešira (*cache*).
- Ako je URI asterisk ("*"), OPTIONS se odnosi na cijeli server.

23

OPTIONS



- Ako URI nije asterisk, OPTIONS se odnosi na opcije koje su na raspolaganju prilikom komunikacije sa specifičnim resursom određenim sa URI-jem navedenim u poslanom zahtjevu,
- Ako HTTP zahtjev sa OPTIONS prođe kroz HTTP *proxy*, HTTP *proxy* MORA izmijeniti odgovor od servera da isključi opcije koje nisu dozvoljene preko *proxy*-ja.
- telnet klijent je generički TCP klijent preko kojeg se možete spajati i na druge servise poput HTTP.

24

- Primjer:

"Telnetirate" se na web server kojem želite poslati zahtjev:

```
telnet marjan.fesb.hr 80
```

```
Trying 161.53.166.3...
Connected to marjan.fesb.hr.
Escape character is '^]'.
```

Nakon spajanja unesete zahtjev kao niz ASCII karaktera:

```
OPTIONS http://www.fesb.hr/~kiki/index.htm HTTP/1.0
```

Na kraju zahtjeva ide jedna prazna linija da server zna da je to kraj zahtjeva

Odgovor servera je:

```
HTTP/1.1 200 OK
Date: Mon, 12 Mar 2007 08:23:06 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-19 mod_ssl/2.8.22 OpenSSL/0.9.7e
mod_perl/1.29
Content-Length: 0
Allow: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, PATCH, PROPFIND,
PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK, TRACE
Connection: close
```

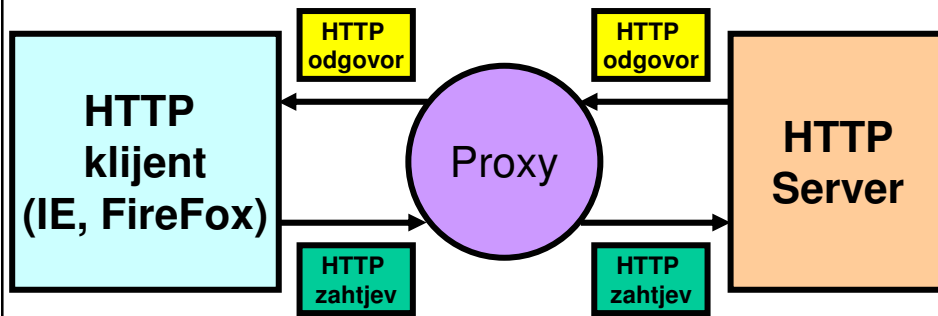
25

HTTP *proxy* (posrednik)

- *Proxy* je posrednički program koji istovremeno djeluje i kao HTTP server i kao HTTP klijent u svrhu postavljanja zahtjeva u ime drugih klijenta.
- Zahtjevi se obrađuju ili unutar *proxy*-ja ili ih se prosljeđuje drugim serverima uz moguće izmjene zahtjeva.
- (RFC 2068 HTTP/1.1) An intermediary program which acts as both a server and a client for the purpose of making requests on behalf of other clients. Requests are serviced internally or by passing them on, with possible translation, to other servers.
- *Proxy* služi za prosljeđivanje zahtjeva koje dobije na način da *proxy* pročita zahtjev (zahtijevani URI) ako želi izmjeniti dio dobivenog zahtjeva te ga proslijedi serveru kojem je zahtjev i namijenjen (piše u URI-ju).

26

HTTP proxy



27

HTTP proxy



- Implementacija HTTP proxy-ja može biti na više načina:
 - Presretački proxy (*intercepting proxy*, *transparent proxy*) je kombinacija HTTP proxy-ja i NAT (*Network Address Translation*). Konekcije klijenta se preko NAT prosleđuju proxy-ju bez znanja klijenta. Često ga koriste ISP (*Internet Service Provider*) kako bi smanjili promet koristeći zajednički keš.
 - Otvoreni proxy (*open proxy*) je proxy koji prihvata konekcije klijenata sa bilo koje IP adrese prema bilo kojem resursu (URI-ju). Zbog nedovoljne zaštite (nema ograničenja tko postavlja zahtjev i prema kome) često se zloupotrebljavaju.
 - Obrnuti proxy (*reverse proxy*) je proxy server instaliran uz HTTP server. Sav promet prema HTTP serveru ide preko proxy-ja. Omogućava dodatnu zaštitu web servera, keširanje sadržaja sa web servera na proxy-ju čime se smanjuje opterećenje web servera, može komprimirati sadržaj koji se prenosi u HTTP porukama ...
 - Razdijeljeni proxy (*split proxy*) je par proxy-ja instaliranih na dva računala koji mogu međusobno komunicirati brže nego sa web serverom i klijentom. Obično se koriste za kompresiju/dekompresiju podataka preko sporih veza poput bežične ili mobilne konekcije.

28

GET



- GET metoda dohvaća objekt identificiran sa URI-jem u HTTP zahtjevu.
- Objekt identificiran URI-jem ne treba biti isključivo samo statička HTML stranica. Ako se zatraženi URI odnosni na neki proces koji generira podatke, web server zna na koji način pozvati proces, proslijediti mu podatke klijentovog zahtjeva, te vratiti nazad klijentu podatke koje je izgenerirao proces, a ne tekst koji je zapisan u file-u kojim se pokreće proces.
- Web server mora razlikovati statički dokument od procesa. Kako to radi nije definirano u HTTP protokolu.
- Npr. neke implementacije web servera (Apache) definiraju posebni direktorij u koji se smještaju CGI (*Common Gateway Interface*) programi.
- PRIMJER:
statička stranica <http://www.fesb.hr/~kiki/proba.htm>
proces <http://www.fesb.hr/kiki-cgi/proba>

29

UVJETNI GET



- Semantika GET metode se mijenja kod uvjetnog GET-a (*conditional GET*) kada HTTP zahtjev sadrži If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match, ili If-Range polje zaglavlja.
- Uvjetni GET zahtjev omogućava prijenos objekta definiranog URI-jem samo ako su zadovoljeni uvjeti definirani u zaglavlju.
- Njime se postiže smanjivanje prometa na mreži.
- Ukoliko se koristi If-Modified-Since zaglavlje server će vratiti traženi objekt samo ukoliko se objekt promijenio nakon vremena poslanog u zahtjevu. Ukoliko se objekt nije promijenio server će vratiti odgovor sa statusnim kodom 304 (*Not Modified*) i bez sadržaja poruke (*entity body*).
- Format zaglavlja If-Modified-Since:
If-Modified-Since = "If-Modified-Since" ":" HTTP datum (*HTTP-date*)

30

HTTP formati datuma i vremena



- HTTP aplikacije podržavaju tri različita formata zapisa datuma/vremena:
 - Sun, 06 Nov 1994 08:49:37 GMT ; RFC 822 (1982), updated by RFC 1123 (1989)
 - Sunday, 06-Nov-94 08:49:37 GMT ; RFC 850, obsoleted by RFC 1036 (Standard for Interchange of USENET Messages, 1987)
 - Sun Nov 6 08:49:37 1994 ; ANSI C's asctime() format
- Prvi format je standard zapisa na Internetu. Drugi format je u upotrebi premda se bazira na zastarjelom (*obsoleted*) standardu za zapis datuma u kojem je godina predstavljena sa dvije znamenke (problem 2000). Prema HTTP/1.1 standardu i klijent i server moraju znati pročitati sva tri formata zapisa datuma zbog kompatibilnosti sa HTTP/1.0. Ali datum koji oni generiraju mora biti zapisan isključivo u prvom formatu (RFC 1123).
- HTTP/1.1 standard preporuča podršku i za druge formate zapisa jer se ponekad komunikacija ostvaruje i prema aplikacijama koje nisu HTTP aplikacije pa koriste i druge formate datuma (recimo SMTP ili NNTP).

31

HTTP formati datuma i vremena



- rfc1123-date = wkday "," SP date1 SP time SP "GMT"
- rfc850-date = weekday "," SP date2 SP time SP "GMT"
- asctime-date = wkday SP date3 SP time SP 4DIGIT
- date1 = 2DIGIT SP month SP 4DIGIT ; day month year (e.g., 02 Jun 1982)
- date2 = 2DIGIT "-" month "-" 2DIGIT ; day-month-year (e.g., 02-Jun-82)
- date3 = month SP (2DIGIT | (SP 1DIGIT)) ; month day (e.g., Jun 2)
- time = 2DIGIT ":" 2DIGIT ":" 2DIGIT ; 00:00:00 - 23:59:59
- wkday = "Mon" | "Tue" | "Wed" | "Thu" | "Fri" | "Sat" | "Sun"
- weekday = "Monday" | "Tuesday" | "Wednesday" | "Thursday" | "Friday" | "Saturday" | "Sunday"
- month = "Jan" | "Feb" | "Mar" | "Apr" | "May" | "Jun" | "Jul" | "Aug" | "Sep" | "Oct" | "Nov" | "Dec"
- "GMT" (*Greenwich Mean Time*) danas se zove i "UT" (*Universal Time*), a aplikacije to onda mogu pretvoriti u lokalno vrijeme.

32

- Primjer:

"Telnetirate" se na web server kojem želite poslati zahtjev:

```
telnet marjan.fesb.hr 80
```

```
Trying 161.53.166.3...
Connected to marjan.fesb.hr.
Escape character is '^'.
```

Nakon spajanja unesete zahtjev kao niz ASCII karaktera:

```
GET http://www.fesb.hr/~kiki/index2.html HTTP/1.1
Host: www.fesb.hr
If-Modified-Since: Sun, 18 Mar 2007 00:00:00 GMT
```

Na kraju zahtjeva ide jedna prazna linija da server zna da je to kraj zahtjeva

Odgovor servera je:


```
HTTP/1.1 304 Not Modified
Date: Sun, 18 Mar 2007 07:49:24 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-19 mod_ssl/2.8.22 OpenSSL/0.9.7e
mod_perl/1.29
ETag: "16f4cd-195-3a700aec"
```

33

UVJETNI GET

- Ukoliko se koristi If-Unmodified-Since zaglavlje server će vratiti traženi objekt samo ukoliko se objekt nije promijenio nakon vremena poslanog u zahtjevu.
- Ako se objekt adresiran URI-jem promijenio nakon vremena navedenog u zaglavlju If-Unmodified-Since server ne smije napraviti traženu operaciju (ovo se zaglavlje može koristiti i sa drugim metodama, a ne samo GET metodom) i treba vratiti odgovor sa statusnim kodom 412 (*Precondition Failed*).
- Ukoliko zahtjev prema URI-ju bez zaglavlja generira odgovor različit od 2xx statusa (npr. objekt ne postoji), zaglavlje If-Unmodified-Since se ignorira.
- Ako je datum loše formatiran, zaglavlje se također ignorira.
- Format zaglavlja If-Unmodified-Since :
If-Unmodified-Since = "If-Unmodified-Since" ":" HTTP datum

34



```

"Telnetirate" se na web server kojem želite poslati zahtjev:
telnet marjan.fesb.hr 80
Trying 161.53.166.3...
Connected to marjan.fesb.hr.
Escape character is '^J'.
Nakon spajanja unesete zahtjev kao niz ASCII karaktera:
GET http://www.fesb.hr/~kiki/index2.html HTTP/1.1
Host: www.fesb.hr
If-Unmodified-Since: Mon, 15 Jan 2001 00:00:00 GMT
Na kraju zahtjeva ide jedna prazna linija da server zna da je to kraj zahtjeva
Odgovor servera je:
HTTP/1.1 412 Precondition Failed
Date: Sun, 18 Mar 2007 10:25:08 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-19 mod_ssl/2.8.22 OpenSSL/0.9.7e
mod_perl/1.29
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1

13b
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>412 Precondition Failed</TITLE>
</HEAD><BODY>
<H1>Precondition Failed</H1>
The precondition on the request for the URL /~kiki/index2.html evaluated to false.<P><HR>
<ADDRESS>Apache/1.3.33 Server at www.fesb.hr Port 80</ADDRESS>
</BODY></HTML>

```

35

UVJETNI GET



- Zaglavlje If-Match koristi se kako bi klijent mogao postaviti zahtjev za objektom uspoređujući ga sa već nekim dohvaćenim sadržajem. Na taj način npr. klijent može provjeriti da li je sadržaj njegovog *cache*-a aktualan. Ukoliko je vrijednost zaglavlja "*" to odgovara svakom sadržaju.
- Ukoliko oznaka entiteta (*entity tag*) URI-ja odgovara vrijednosti u zaglavlju If-Match ili ako je vrijednost zaglavlja "*", server vraća čitav objekt.
- Ukoliko oznaka entiteta URI-ja ne odgovara vrijednosti u zaglavlju ili ako je vrijednost zaglavlja "*", a sadržaj URI-ja prazan server ne smije napraviti traženu operaciju i treba vratiti odgovor sa statusnim kodom 412 (*Precondition Failed*).
- Ukoliko zahtjev prema URI-ju bez zaglavlja generira odgovor različit od 2xx statusa (npr. objekt ne postoji), zaglavlje If-Match se ignorira.
- Format zaglavlja If-Match:
If-Match = "If-Match" ":" ("*" | 1#entity-tag)

36

OZNAKA ENTITETA



- Oznaka entiteta (*entity tag*) koristi se za uspoređivanje dva ili više entiteta zahtijevanog resursa (URI-ja). Oznaka entiteta koristi se u zaglavljima ETag (koje server šalje klijentu sa vrijednošću oznake entiteta), If-Match, If-None-Match i If-Range. Oznaka entiteta je u biti string koji može imati dva formata:
 - entity-tag = [weak] opaque-tag
 - weak = "W/" (weakness indicator)
 - opaque-tag = quoted-string
 - Jaku oznaka entiteta (*strong entity tag*) mogu imati samo oni entiteti zahtijevanog resursa (URI-ja) koji su potpuno identični (*octet equality*). Slabu oznaka entiteta (*weak entity tag*), sa "W/" prefiksom, mogu imati samo oni entiteti zahtijevanog resursa (URI-ja) koji se mogu zamijeniti bez gubitka značenja. Oznaka entiteta treba biti jedinstvena za sve entitete određenog resursa.
 - Kod starijih verzija Apache web servera vrijednost oznake entiteta (ETag zaglavlja) se generirala na osnovu veličine datoteke, vremena zadnje promijene datoteke i sl.

37

"Telnetirate" se na web server kojem želite poslati zahtjev:
telnet marjan.fesb.hr 80

Trying 161.53.166.3...
Connected to marjan.fesb.hr.
Escape character is '^['.

Nakon spajanja unesete zahtjev kao niz ASCII karaktera:
GET <http://www.fesb.hr/~kiki/index2.html> HTTP/1.1
Host: www.fesb.hr
If-Match: "16f4cd-195-3a700aec"

Na kraju zahtjeva ide jedna prazna linija da server zna da je to kraj zahtjeva
Odgovor servera je:
HTTP/1.1 200 OK
Date: Sun, 18 Mar 2007 11:16:44 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-19 mod_ssl/2.8.22
OpenSSL/0.9.7e mod_perl/1.29
Last-Modified: Thu, 25 Jan 2001 11:15:56 GMT
ETag: "16f4cd-195-3a700aec"
Accept-Ranges: bytes
Content-Length: 405
Content-Type: text/html

<HTML>.....

38

UVJETNI GET



- Ukoliko klijent ima više entiteta dohvaćenih sa zahtijevanog resursa (URI-ja) može za svih provjeriti da li su i dalje nepromijenjeni navodeći oznake entiteta u zaglavlju If-None-Match.
- Ukoliko oznaka entiteta (*entity tag*) URI-ja odgovara barem jednoj od vrijednosti u zaglavlju If-None-Match ili ako je vrijednost zaglavlja "*", server ne smije izvršiti metodu. Ukoliko je metoda bila GET ili HEAD server mora odgovoriti sa statusnim kodom 304 (*Not Modified*), a za ostale metode treba vratiti odgovor sa statusnim kodom 412 (*Precondition Failed*).
- Ukoliko oznaka entiteta ne odgovara ni jednoj od vrijednosti u zaglavlju If-None-Match ili ako je vrijednost zaglavlja "*", a zahtijevani URI nema entiteta (sadržaja) server može obraditi zahtjeva kao da zaglavlje If-None-Match nije ni poslano.
- Format zaglavlja If-None-Match :
If-None-Match = "If-None-Match" ":" ("*" | 1#entity-tag)

39

"Telnetirate" se na web server kojem želite poslati zahtjev:

telnet marjan.fesb.hr 80

Trying 161.53.166.3...

Connected to marjan.fesb.hr.

Escape character is '^'.

Nakon spajanja unesete zahtjev kao niz ASCII karaktera:

GET http://www.fesb.hr/~kiki/moja_stranica.htm HTTP/1.1

Host: www.fesb.hr

If-None-Match: "16f51c-2411-4386c23a"

Na kraju zahtjeva ide jedna prazna linija da server zna da je to kraj zahtjeva

Odgovor servera je:

HTTP/1.1 304 Not Modified

Date: Sun, 18 Mar 2007 18:14:25 GMT

Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-19 mod_ssl/2.8.22

OpenSSL/0.9.7e mod_perl/1.29

ETag: "16f51c-2411-4386c23a"

40





UVJETNI GET

- Zaglavlje If-Range omogućava klijentu da zahtjeva samo dijelove entiteta. Ukoliko se entitet promijenio klijent očekuje cijeli entitet, a ne samo zahtijevani dio.
- Ako klijent nema oznaku entiteta može poslati i datum (npr. dobiven u Last-Modified zaglavlju). Server zna razlikovati ETag od datuma i odgovoriti na zahtjev ispravno (tj. provjeriti ETag zahtijevanog URI-ja ili provjeriti vrijeme zadnje promjene). Uz zaglavlje If-Range obavezno klijent treba slati i Range zaglavlje.
- Ukoliko oznaka entiteta odgovara trenutnoj oznaci entiteta resursa server bi trebao vratiti dio entiteta određenog Range zaglavljem u odgovoru statusnog koda 206 (*Partial content*). Ukoliko oznaka entiteta ne odgovara trenutnoj oznaci entiteta resursa server bi trebao vratiti čitav entitet u odgovoru statusnog koda 200 (OK).
- Format zaglavlja If-Range :
If-Range = "If-Range" ":" (entity-tag | HTTP-date)

71



PARCIJALNI GET

- Semantika GET metode se mijenja i kod parcijalnog GET-a (partial GET) kada HTTP zahtjev sadrži Range polje zaglavlja.
- Parcijalni GET zahtjev dohvaća samo dio objekta, a ne čitav objekt.
- Njime se također postiže smanjivanje prometa na mreži.

42

PARCIJALNI GET



- Sadržaj HTTP poruke (entity-body) je “samo” niz bajtova pa se pojam bajta može koristiti za ograničavanje dijelova sadržaja HTTP poruke.
- Ako server podržava Range zaglavlje kod običnog GET-a (bez uvjeta) vraća dio entiteta određen u Range zaglavlju uz statusni kod 206 (*Partial Content*) umjesto 200 (*OK*). Ako je zahtjev uvjetni GET odgovor ovisi i o uvjetima.
- Format zaglavlja Range :
Range = "Range" ":" ranges-specifier
- Format specifikatora raspona (*ranges-specifier*) (vrijednosti mogu ići do 0 do 9999):
 - bytes=0-499 (vrati prvih 500 bajtova)
 - bytes=500-999 (vrati drugih 500 bajtova)
 - bytes=-500 (vrati zadnjih 500 bajtova)
 - bytes=0-0,-1 (vrati samo prvi i zadnji bajt)

43

```
telnet marjan.fesb.hr 80
```

```
Trying 161.53.166.3...
Connected to marjan.fesb.hr.
Escape character is '^]'.
```

```
GET http://www.fesb.hr/~kiki/moja_stranica.htm HTTP/1.1
Host: www.fesb.hr
Range:bytes=-10
```

```
Odgovor servera je:
HTTP/1.1 206 Partial Content
Date: Sun, 18 Mar 2007 19:08:50 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-19 mod_ssl/2.8.22
      OpenSSL/0.9.7e mod_perl/1.29
Last-Modified: Fri, 25 Nov 2005 07:50:18 GMT
ETag: "16f51c-2411-4386c23a"
Accept-Ranges: bytes
Content-Length: 10
Content-Range: bytes 9223-9232/9233
Content-Type: text/html
```

```
></HTML>
```

44

HEAD



- HEAD metoda je po svemu identična GET metodi osim što server u odgovoru NE SMIJE vratiti tijelo poruke (to je sve ono što ide iza zaglavlja).
- Meta podaci (*metainformation*) iz HTTP zaglavlja odgovora trebaju biti isti kod HEAD i kod GET zahtjeva.
- Ova se metoda koristi kada se žele dohvatiti meta podaci o objektu bez prenošenja cijelog objekta. Često se koristi za testiranje linkova - da li im se može pristupiti, da li postoje, da li su nedavno promijenjeni.

45

POST



- POST metoda se koristi kada želimo da server prihvati objekt (sadržaj poruke) HTTP zahtjeva kao dio koji je podređen zahtijevanom resursu identificiranom preko URI-ja. POST je napravljen kao jedinstvena uniformna metoda koja se koristi u sljedećim slučajevima:
 - Kada želimo postojećem resursu dodati neku oznaku;
 - Kada se šalje poruka na neku mailing listu, ili newsgupu i sl.;
 - Za isporuku bloka podataka isporučenih sa forme nekom procesu za obradu tih podataka;
 - Dodavanju novih slogova u bazu.

46

POST



- Koja od prethodno navedenih funkcija će se i izvršiti nakon što server dobije HTTP zahtjev s POST metodom određeno je samim serverom i obično ovisi o URI-ju koji se zahtjeva.
- Akcija izvršena na osnovu POST HTTP zahtjeva će možda rezultirati resursom koji se ne može identificirati sa URI-jem. U tom slučaju HTTP odgovor treba imati kod ili 200 (*OK*) ili 204 (*No Content*) ovisno o tome uključuje li odgovor podatke koji su rezultat akcije ili ne.
- Ako je resurs stvoren na serveru kao posljedica POST metode tada odgovor servera treba imati statusni kod 201 (*Created*), treba sadržavati podatke o statusu zahtjeva i *Location* zaglavlje koje upućuje na stvoreni resurs.

47

Location zaglavlje



- Zaglavlje Location koristi se za redirekciju klijenta na lokaciju različitu od one navedene u URI-ju zahtjeva.
- Za odgovore sa statusnim kodom 201 Location je lokacija resursa kreiranog kao odgovor na klijentov zahtjev.
- Za odgovore sa statusnim kodovima 3xx lokacija bi trebala biti URI na koji server upućuje klijenta umjesto zahtijevanog URI-ja.
- Vrijednost zaglavlja je apsolutni URI.
- Format zaglavlja Location:
Location = "Location" ":" absoluteURI

(<meta http-equiv="Refresh" content="2; URL=http://merkur.fesb.hr/fesbX/">
redirekcija (ne servera) već klijent na osnovu HTML meta taga sam zahtijeva novi URI)

48

telnet marjan.fesb.hr 80

Trying 161.53.166.3...
Connected to marjan.fesb.hr.
Escape character is '^'.

GET <http://www.fesb.hr>
Host: www.fesb.hr

Odgovor servera je:
HTTP/1.1 200 OK
Date: Mon, 19 Mar 2007 07:16:40 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-19 mod_ssl/2.8.22 OpenSSL/0.9.7e
mod_perl/1.29
Last-Modified: Wed, 11 Oct 2006 10:34:06 GMT
ETag: "18063-6d1-452cc89e"
Accept-Ranges: bytes
Content-Length: 1745
Content-Type: text/html

```
<html>
<head>
<title>FESB</title>
<meta http-equiv="Content-Type" content="text/html;">
<meta http-equiv="Refresh" content="2; URL=http://merkur.fesb.hr/fesbX/">
```

49

Redirekcija

- Redirekcija preko HTML meta taga (`http-equiv="Refresh"`) nije implementirana u HTTP protokolu. To je dio HTML standarda i klijenti koji dohvaćaju HTML stranice znaju što napraviti kada u HTML kodu naiđu na ovaj tag.
- Redirekciju preko HTTP protokola treba implementirati na strani server. Način implementacije ovisi o web serveru, o URI-ju koji se redirektira i sl. Npr. korištenjem `.htaccess` datoteke.
- <http://www.webconfs.com/how-to-redirect-a-webpage.php>

50

telnet marjan.fesb.hr 80

Trying 161.53.166.3...
Connected to marjan.fesb.hr.
Escape character is '^'.

GET <http://www.fesb.hr/~kiki/redirekcija.html> HTTP/1.1
Host: www.fesb.hr

Odgovor servera je:
HTTP/1.1 301 Moved Permanently
Date: Mon, 19 Mar 2007 07:47:36 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-19 mod_ssl/2.8.22 OpenSSL/0.9.7e mod_perl/1.29
Location: <http://www.morsko-prase.hr/2006/index.php>
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1

```
139
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>301 Moved Permanently</TITLE>
</HEAD><BODY>
<H1>Moved Permanently</H1>
The document has moved <A HREF="http://www.morsko-prase.hr/2006/index.php">here</A>.<P>
<HR>
<ADDRESS>Apache/1.3.33 Server at www.fesb.hr Port 80</ADDRESS>
</BODY></HTML>
```

51

SLANJE PODATAKA SA FORME POST METODOM

HTTP
metoda

- POST metoda često se koristi za slanje sadržaja forme.
- Podaci sa forme se šalju u sadržaju poruke (*entity-body*).
- Na strani servera podaci se procesu (obično se radi o nekakvom procesu koji će podatke obraditi) šalju kao standardni ulaz procesa (stdin).
- Primjer: <http://www.fesb.hr/~kiki/proba1.htm>

```
<form METHOD="POST" ACTION="http://www.fesb.hr/kiki-cgi/proba1">
  <INPUT TEXT NAME="polje"><br>
  <INPUT TEXT NAME="polje2"><br>
  <INPUT TEXT NAME="polje3"><br>
  <input type="submit" value="POKRENI PROGRAM" size="1" >
</form>
```

52

telnet marjan.fesb.hr 80

Trying 161.53.166.3...
Connected to marjan.fesb.hr.
Escape character is '^'.

POST <http://www.fesb.hr/kiki-cgi/proba1> HTTP/1.0
Content-length: 26

&polje=1&polje2&polje3=3

Odgovor servera je:

HTTP/1.1 200 OK

Date: Mon, 19 Mar 2007 08:18:24 GMT

Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-19 mod_ssl/2.8.22 OpenSSL/0.9.7e mod_perl/1.29

Connection: close

Content-Type: text/html

<H1>Rezultati</H1>Poslali ste:<p>

 <code> = </code>

 <code>polje = 1</code>

 <code>polje2 = </code>

 <code>polje3 = 3

</code>

Connection closed by foreign host.



53

Content-length zaglavlje

- Zaglavlje Content-Length sadrži veličinu sadržaja poruke izraženu u obliku decimalnog broja okteta (bajtova) sadržaja poruke, koja se šalje primatelju poruke ili u slučaju HEAD metode veličinu sadržaja koji bi se poslao da se radi o GET metodi.
- HTTP aplikacije bi trebale koristiti ovo polje da odrede veličinu sadržaja neovisno o tipu sadržaja (tekst, slika,). Primatelj poruke mora biti u stanju odrediti kraj poruke kod HTTP/1.1 protokola.
- Format zaglavlja Content-Length :
Content-Length = "Content-Length" ":" 1*DIGIT



54

Duljina poruke



- Kada poruka ima sadržaj duljina sadržaja se određuje na jedan od sljedećih načina:
 1. Svaki odgovor koji ne smije imati sadržaj (statusni kodovi 1xx, 204 i 304, te odgovor na HEAD zahtjev) završava sa prvom praznom linijom nakon zaglavlja.
 2. Ukoliko zaglavlje Transfer-Encoding definira da je poruka razbijena na dijelove (Transfer-Encoding: chunked) duljina se dobiva iz oznake dijelova poruke.
 3. Ako je navedeno zaglavlje Content-Length tada je duljina sadržaja poruke vrijednost Content-Length zaglavlja.
 4. Ukoliko je tip sadržaja "multipart/byteranges" (Content-type: multipart/byteranges; boundary=THIS_STRING_SEPARATES) duljina se određuje delimiterima dijelova poruke.
 5. Kada server zatvori konekciju. (Time se može definirati samo duljina odgovora, a ne zahtjeva jer server ne smije zatvoriti vezu dok ne pošalje odgovor.)

55

SLANJE PODATAKA SA FORME GET METODOM

HTTP
metoda



- Sadržaj forme se može slati i korištenjem GET metode.
- Tada podaci idu kao dio URI u formi
`?podatak1=1&podatak2=2&podatak3=3`
- Dio URI-ja iza imena resursa koji se dohvaća počinje od znaka upitnika i ima definirani format.
- Taj dio se na strani server sprema u okolinsku varijablu (*environment variable*) koju procesi koji se izvode mogu dohvatiti i pročitati njen sadržaj, ali to nije dio HTTP protokola. URI-ji koji sadrže dio ?... se nazivaju upitni URI-ji (*query URI*) i server je zadužen za to da ih ispravno procesira.
- Primjer: <http://www.fesb.hr/~kiki/proba2.htm>

```
<form METHOD="GET" ACTION="http://www.fesb.hr/kiki/cgi/proba2">  
<INPUT TEXT NAME="polje"><br>  
<INPUT TEXT NAME="polje2"><br>  
<INPUT TEXT NAME="polje3"><br>  
<input type="submit" value="POKRENI PROGRAM" size="1" >  
</form>
```

56

telnet marjan.fesb.hr 80

Trying 161.53.166.3...
Connected to marjan.fesb.hr.
Escape character is '^['.

GET <http://www.fesb.hr/kiki-cgi/proba2?polje=1&polje2=2&polje3=3> HTTP/1.0

Odgovor servera je:

HTTP/1.1 200 OK
Date: Mon, 19 Mar 2007 08:59:24 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-19 mod_ssl/2.8.22 OpenSSL/0.9.7e
mod_perl/1.29
Connection: close
Content-Type: text/html

```
<H1>Rezultati forme</H1>Poslali ste:<p>  
<li> <code>polje = 1</code>  
<li> <code>polje2 = 2</code>  
<li> <code>polje3 = 3</code>  
</ul>  
Connection closed by foreign host.
```

57

PUT

- PUT metoda traži od servera da priložene podatke pohrani pod zahtijevanim URI-jem. Ako zahtijevani URI već postoji tada se smatra da je to zahtjev za *update*-om postojećeg URI-ja.
- Ako je novi resurs kreiran server odgovara sa kodom 201 (*Created*).
- Ako je resurs *update*-iran server odgovara s 200 (*OK*) ili s 204 (*No Content*).
- Ukoliko resurs nije mogao biti kreiran server treba odgovoriti sa određenom greškom.
- HTTP protokol ne definira kako PUT zahtjev utiče na stanje servera.

58

DELETE



- DELETE metoda se koristi da bi se serveru uputio zahtjev za brisanjem resursa identificiranog URI-jem.
- Klijent ne može garantirati da je akcija izvršena čak i ako je server vratio odgovor u kojem indicira da je akcija izvršena.
- Odgovor servera treba biti 200 (*OK*) ako odgovor uključuje podatke koji opisuju status servera, 202 (*Accepted*) ako je zahtjev prihvaćen, ali još nije izvršen, ili 204 (*No Content*) ako je odgovor OK, ali ne uključuje nikakve podatke koji opisuju status servera.

59

TRACE

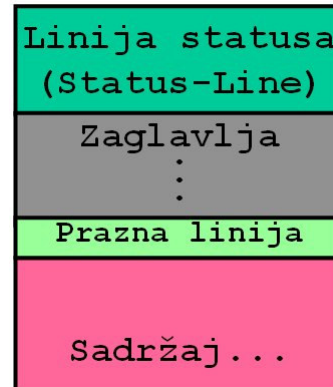


- TRACE metoda poziva povratnu petlju poruka zahtjeva koje je poslao klijent. Konačni primatelj zahtjeva treba primljenu poruku vratiti nazad klijentu kao podatkovni dio 200 (*OK*) odgovora. Konačni primatelj je ili originalni server ili prvi *proxy* ili *gateway* ispred servera. TRACE zahtjev ne smije nositi podatke.
- TRACE zahtjev omogućava klijentu da vidi što se prima od onoga što je klijent poslao na drugoj strani lanca (*request chain*) tj. što server prima. Ti se podaci koriste za testiranje veze.
- U slučaju uspjeha, odgovor treba sadržavati cijeli poruku zahtjeva i to kao podatkovni dio, a Content-Type treba biti "message/http".

60

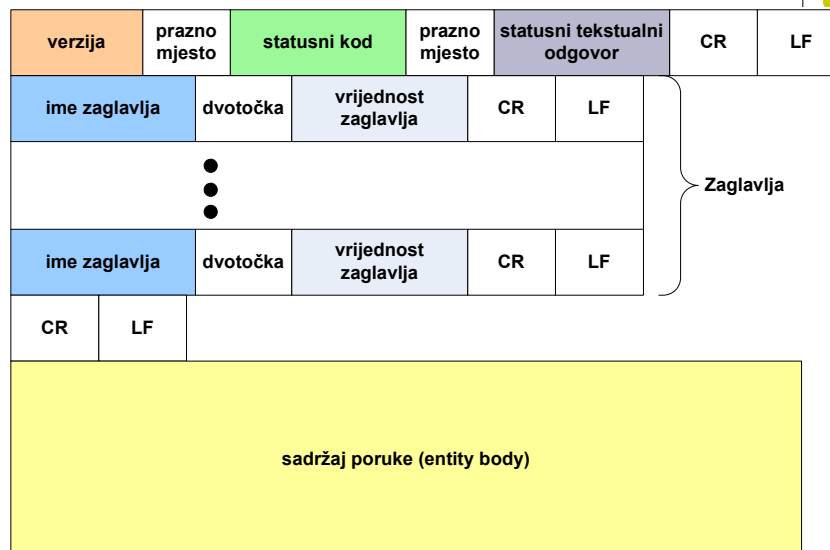
HTTP odgovor

- HTTP odgovor šalje server klijentu.
- HTTP protokol točno definira kakav format treba imati ispravni HTTP odgovor.
- Format HTTP odgovora je:
 - Linija statusa
 - Zaglavlja
 - Prazna linija
 - Sadržaj poruke



61

HTTP odgovor



62

HTTP odgovor



```
HTTP/1.0 200 OK
Date: Wednesday, 02-Feb-95 23:04:12 GMT
Server: NCSA/1.3
MIME-version: 1.0
Last-modified: Monday, 15-Nov-93 23:33:16 GMT
Content-type: text/html
Content-length: 2345
* PRAZNA LINIJA *
<HTML><HEAD><TITLE> ...
```

```
LINIJA STATUSA
ZAGLAVLJE DATE
ZAGLAVLJE SERVER
ZAGLAVLJE MIME-VERSION
ZAGLAVLJE LAST-MODIFIED
ZAGLAVLJE CONTENT-TYPE
ZAGLAVLJE CONTENT-LENGTH
PRAZNA LINIJA (\r\n)
SADRŽAJ PORUKE .....
```

63

STATUSNI KODOVI



- Linija statusa HTTP odgovora sadrži statusni kod koji opisuje odgovor servera na HTTP zahtjev.
- Format statusnog koda su tri cijela broja (npr. 200) koji identificiraju tip odgovora server.
- Statusni tekstualni odgovor (*Reason-Phrase*) je kratki tekstualni opis statusnog koda.
- Statusni kod je namijenjen računalu dok je statusni tekstualni odgovor namijenjen ljudskom korisniku stoga web klijent ne mora razumjeti statusni tekstualni odgovor.
- Prva znamenka statusnog koda određuje klasu odgovora servera. Trenutno je definirano 5 klasa odgovora servera (statusni kodovi 1xx, 2xx, 3xx, 4xx i 5xx).

64

STATUSNI KODOVI



- Kodovi koji su oblika 1xx (*Informational*) su Informacijski kodovi i označavaju odgovor koji se sastoji samo od linije statusa i zaglavlja. S obzirom da HTTP/1.0 nije definirao 1xx statusne kodove, server ne smije poslati 1xx odgovor HTTP/1.0 klijentu.
- Trenutni informacijski kodovi su 100 (*Continue*) i 101 (*Switching Protocols*).
- Informacijski kod 100 znači da klijent može nastaviti sa svojim zahtjevom. Ovim odgovorom server informira klijenta da ja primio dio zahtjeva i da ga još nije obradio. Klijent bi trebao nastaviti slati ostatak zahtjeva ili ukoliko je već poslao čitav zahtjev ignorirati ovaj odgovor servera. Server mora na kraju poslati konačni odgovor nakon što je primio čitav zahtjev.

65

STATUSNI KODOVI



- Kodovi koji su oblika 2xx (*Successful*) su Uspješni kodovi koji kažu da je zahtjev klijenta uspješno primljen i prihvaćen (200 (*OK*), 201 (*Created*), 202 (*Accepted*), 203 (*Non-Authoritative Information*), 204 (*No Content*), 205 (*Reset Content*), 206 (*Partial Content*)).
- Kodovi koji su oblika 3xx (*Redirection*) su Redirekcijski kodovi koji označavaju da klijent odnosno njegov korisnički agent (*user agent*) trebaju poduzeti još neke akcije da bi dobili zahtijevani URI. Te akcije se mogu izvršiti bez interakcije sa korisnikom samo ako je metoda zahtjeva GET ili HEAD. Korisnički agent ne smije vršiti redirekciju zahtjeva automatski više od 5 puta (300 (*Multiple Choices*), 301 (*Moved Permanently*), 302 (*Moved Temporarily*), 303 (*See Other*), 304 (*Not Modified*), 305 (*Use Proxy*)).

66

STATUSNI KODOVI



- Kodovi oblika 4xx (*Client Error*) označavaju grešku na strani klijenta. Kada server smatra da se na strani klijenta desila greška u svom odgovoru treba uključiti opis greške, da li je trenutna ili trajna i sl. Ako klijent šalje podatke serveru preko TCP-a (uglavnom je ta situacija) server treba čekati prijem potvrde (TCP ACK) od klijenta da je primio odgovor prije no što server zatvori konekciju jer ako klijent nastavi slati podatke, a veza je zatvorena server će mu poslati RTS (*reset*) paket koji može izbrisati klijentov ulazni bafer prije no što je aplikacija (IE, FireFox) pročitala i interpretirala podatke.
- 400 (*Bad Request*), 401 (*Unauthorized*), 402 (*Payment Required*), 403 (*Forbidden*), 404 (*Not Found*), 405 (*Method Not Allowed*), 406 (*Not Acceptable*), 407 (*Proxy Authentication Required*), 408 (*Request Timeout*), 409 (*Conflict*), 410 (*Gone*), 411 (*Length Required*), 412 (*Precondition Failed*), 413 (*Request Entity Too Large*), 414 (*Request-URI Too Long*), 415 (*Unsupported Media Type*)

67

STATUSNI KODOVI



- Kodovi oblika 5xx (*Server Error*) označavaju grešku na strani servera. Kada šalje odgovor server treba uključiti u podatkovni dio objašnjenje opisa greške, da li je trenutna ili trajna i sl. Klijent treba prikazati primljenu grešku (500 (*Internal Server Error*), 501 (*Not Implemented*), 502 (*Bad Gateway*), 503 (*Service Unavailable*), 504 (*Gateway Timeout*), 505 (*HTTP Version Not Supported*)).

68

STATUSNI KODOVI



- 200 OK – zahtjev je uspio. Informacija (podatkovni dio HTTP odgovora) koja se vraća ovisi o metodi zahtjeva.
 - za GET to je zahtijevani URL.
 - za HEAD HTTP sadrži samo neka zaglavlja, nema podatkovnog dijela.
 - za POST su to podaci dobiveni kao rezultat izvršavanja neke akcije
 - za TRACE je to zahtjev koji je klijent poslao.
- 400 Bad Request – server nije razumio klijentov zahtjev. Klijent ne smije ponovo serveru poslati zahtjev u istom obliku.

69

```
telnet marjan.fesb.hr 80
```

```
Trying 161.53.166.3...
Connected to marjan.fesb.hr.
Escape character is '^'.
```

```
GET http://www.fesb.hr/~kiki/nepostojecastranica.html HTTP/1.1
Host: www.fesb.hr
```

Odgovor servera je:

```
HTTP/1.1 404 Not Found
Date: Sun, 25 Mar 2007 15:10:37 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-19 mod_ssl/2.8.22
      OpenSSL/0.9.7e mod_perl/1.2.9
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1
```

```
124
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>404 Not Found</TITLE></HEAD><BODY>
<H1>Not Found</H1><P>The requested URL /~kiki/nepostojecastranica.html was not found
on this server.<P><HR><ADDRESS>Apache/1.3.33 Server at www.fesb.hr Port
80</ADDRESS>
</BODY></HTML>
```

70



telnet marjan.fesb.hr 80

Trying 161.53.166.3...
Connected to marjan.fesb.hr.
Escape character is '^'.

GET <http://www.fesb.hr/kiki-cgi/sgreskom> HTTP/1.1
Host: www.fesb.hr

Odgovor servera je:
HTTP/1.1 500 Internal Server Error
Date: Sun, 25 Mar 2007 15:23:11 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-19 mod_ssl/2.8.22 OpenSSL/0.9.7e
mod_perl/1.29
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1
24b
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD><TITLE>500 Internal Server Error</TITLE>
</HEAD><BODY><H1>Internal Server Error</H1>The server encountered an internal error or
misconfiguration and was unable to complete your request.<P>Please contact the server
administrator, www@fesb.hr and inform them of the time the error occurred, and anything you
might have done that may have caused the error.<P>More information about this error may be
available in the server error log.<P><HR><ADDRESS>Apache/1.3.33 Server at www.fesb.hr
Port 80</ADDRESS></BODY></HTML>

71

HTTP zaglavlja

- HTTP zaglavlja se dijele na:
 - općenita zaglavlja (*general-header*),
 - zaglavlja zahtjeva (*request-header*),
 - zaglavlja odgovora (*response-header*),
 - zaglavlja entiteta tj. podatkovnog dijela HTTP poruke (*entity-header*)
- Sva zaglavlja imaju isti format. Svako zaglavlje se sastoji od imena nakon kojeg ide dvotočka (":") i vrijednost zaglavlja. Imena zaglavlja su case-insensitive.

72

HTTP općenita zaglavlja



- Općenita zaglavlja se odnose i na HTTP zahtjev i na HTTP odgovor, ali nisu vezana uz entitet tj. podatkovni dio HTTP poruke.
- Odnose se isključivo na HTTP poruku čiji su dio. Ovaj tip HTTP zaglavlja se može proširiti (dodati novo zaglavlje) samo promjenom HTTP protokola.
- To su zaglavlja:
 - Cache-Control, Connection, Date, Pragma, Transfer-Encoding, Upgrade, Via
- Zaglavlje koje HTTP server ili HTTP klijent nije u stanju razumjeti tretira se kao zaglavlja entiteta.

73

HTTP zaglavlja zahtjeva



- Zaglavlja zahtjeva omogućavaju klijentu da serveru u sklopu zahtjeva pošalje dodatne informacije o zahtjevu ili o samom klijentu.
- Ovaj tip HTTP zaglavlja se može proširiti (dodati novo zaglavlje) samo promjenom HTTP protokola.
 - Accept, Accept-Charset, Accept-Encoding, Accept-Language
 - Authorization, From, Host
 - If-Modified-Since, If-Match, If-None-Match, If-Range, If-Unmodified-Since
 - Max-Forwards, Proxy-Authorization, Range, Referer, User-Agent

74

HTTP zaglavlja odgovora



- Zaglavlja odgovora omogućavaju serveru da proslijedi dodatne informacije koje se ne mogu smjestiti u liniju statusa.
- Te informacije su neke informacije o serveru i o mogućnostima pristupa resursu identificiranom URI.
- Ovaj tip HTTP zaglavlja se može proširiti (dodati novo zaglavlje) samo promjenom HTTP protokola.
 - Age, Location, Proxy-Authenticate, Public, Retry-After, Server, Vary, Warning, WWW-Authenticate

75

HTTP zaglavlja entiteta



- Zaglavlja entiteta tj. podatkovnog dijela HTTP poruke definiraju opcionalne metapodatke o entitetu poruke ili ako HTTP poruka nema entiteta ti podaci su o resursu koji je identificiran URI-jem.
- Ovaj tip HTTP zaglavlja se može proširiti (dodati novo zaglavlje) i bez promjene HTTP protokola zahvaljujući *extension-header* mehanizmu. Međutim ne podrazumijeva se da su ta polja razumljiva primatelju HTTP poruke tj. da primatelj zna interpretirati sadržaj tog polja. Sva polja zaglavlja koja primatelj ne razumije treba ignorirati, a *proxy*-iji ih trebaju jednostavno prosljediti dalje.
 - Allow, Content-Base, Content-Encoding, Content-Language
 - Content-Length, Content-Location, Content-MD5, Content-Range
 - Content-Type, ETag, Expires, Last-Modified
 - extension-header (extension-header = message-header)

76

HTTP 1.0 ↔ HTTP 1.1



- Glavni nedostatak HTTP 1.0 protokola bila je nepostojana konekcija (*non-persistent connection*) između klijenta i servera.
- To je značilo da se za svaki pojedini zahtjev otvarala nova konekcija (TCP) preko koje su se slali podaci i čim bi server odgovorio na zahtjev klijent bi zatvorio vezu. Time se opterećivao server (vrijeme potrebno da uspostavi veza, potrebna memorija i sl.), a i promet na mreži je bio veći (TCP FIN i sl.).
- Kod HTTP 1.1 moguće je preko jedne konekcije razmijeniti više zahtjev-odgovor parova.

77

MIME



- MIME (*Multipurpose Internet Mail Extensions*) je standard (RFC 1521) koji se (pored ostalog) koristi da bi pretraživač znao interpretirati tj. prikazati dohvaćeni resurs.
- Standard je originalno napravljen za definiranje formata prilog (attachment) elektroničke pošte, ali se danas koristi i kod drugih Internet aplikacija.
- HTTP koristi Internet *Media Types* (noviji naziv za MIME *Content-Types*) (samo jedan mali dio MIME-a) kojim server i klijent dogovaraju tip podataka.
- Korisnik u pretraživač unosi URI da bi dohvatio željenu stranicu, sliku, dokument,
- Pretraživač zatraži od servera stranicu koju server isporuči pretraživaču koristeći HTTP protokol
- MIME povezuje niz bajtova koje klijent dobiva sa tipom kodiranja odnosno MIME “kaže” klijentu što ti bajtovi znače.

78

MIME



- Kada HTTP server šalje informacije nazad klijentu u zaglavlju uključuje i MIME tip po kojem klijent zna o kojem se tipu informacija radi.
- Zaglavlje Content-Type opisuje tip sadržaja poruke ili u slučaju HEAD metode, tip sadržaja koji bi bio poslan u GET zahtjevu.
- Format zaglavlja Content-Type :
Content-Type = "Content-Type" ":" media-type
Npr: Content-Type: text/html; charset=ISO-8859-4
- Zaglavlje Content-Encoding se koristi kao modifikator tipa sadržaja poruke. Ukoliko je zaglavlje prisutno njegova vrijednost definira dodatna kodiranja sadržaja tj. načine dekodiranja sadržaja kako bi se dobio originalni sadržaj čiji tip je naveden u polju Content-Type. Ovo se zaglavlje najčešće koristi kod komprimiranih objekata tako da ovo polje nosi podatak o načinu kompresije, a Content-Type nosi podatak o načinu kodiranja objekta nakon dekompresije.
- Format zaglavlja Content-Encoding:
Content-Encoding = "Content-Encoding" ":" 1#content-coding

79

```
telnet marjan.fesb.hr 80
```

```
Trying 161.53.166.3...
Connected to marjan.fesb.hr.
Escape character is '^['.
```

```
GET http://www.fesb.hr/~kiki/proba.gz HTTP/1.1
Host: www.fesb.hr
```

```
Odgovor servera je:
HTTP/1.1 200 OK
Date: Sun, 25 Mar 2007 16:48:45 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-19 mod_ssl/2.8.22 OpenSSL/0.9.7e
mod_perl/1.29
Last-Modified: Mon, 19 Apr 2004 07:04:23 GMT
ETag: "16f4c8-17d-408379f7"
Accept-Ranges: bytes
Content-Length: 381
Content-Type: text/html
Content-Encoding: x-gzip
```

```
f;dom.htm
KnÅ0=9LČŽKTÅŁ
  ºa8J=^ĐN  ôŁ$5ôĐXÁÉBcČK ``VpČ
id|ŸdŮUšëižuČû=3ŁĐS2UrŮ4WÉVWA{ŦmŦisYOÇ2čä,ŦŦZÉ~Ž`ŸdLm4hlm´ey
L'Ŧ°AŦšŠ~É=Z .....
```

80

MIME



- Format MIME tipova
 - text/plain, text/html, image/gif, application/pdf, ...
- Format kodiranja
 - base64, x-gzip
- Dokument se pretraživaču isporuči sa ove dvije informacije
 - ako je potrebno dokument se dekodira (base64-decode, gzip)
 - za tip imamo tri mogućnosti
 - pretraživač zna sam prikazati tip (HTML, GIF,...)
 - pretraživač zna koja aplikacija može prikazati dokument te je pozove da prikaže dokument (Adobe, Word,...)
 - pretraživač ne zna kako da prikaže određeni tip i ponudi korisniku da dokument spremi na disk

81

MIME



- Media-type vrijednosti registrira Internet Assigned Number Authority (IANA). Postupak registracije novog tipa sadržaja definirani je u RFC 2048.
- <http://www.iana.org/assignments/media-types/>
- Npr. za sliku je trenutno registrirano preko 30 različitih načina kodiranja slike (image/gif, image/tiff, image/jpeg, image/jpx, image/jpm, image/png, ...), za tekst preko 40 različitih načina (text/html, text/css, text/csv, text/plain, text/xml, text/rtf, text/xml, ...).

82