

WWW i HTTP

Cilj prethodne lekcije bio je da postavi teoretske osnove za razumevanje okruženja u kome postoje web sajtovi i aplikacije. Definisani su pojmovi kompjutera i mreža, bez kojih svakako ne bi bilo ni frontend programiranja. Opisujući principe na kojima se temelji komunikacija kompjutera preko interneta, zaustavili smo se kod pojmova HTTP i WWW. Reč je o dva pojma koja imaju presudnu važnost za oblast frontend programiranja.

Šta je World Wide Web?

Većina svakodnevnih, običnih korisnika internet poistovećuje sa pregledom web sadržaja. Ipak, kada otvorite neki od programa za pregled web sadržaja (Chrome, Firefox, Safari...), vi ste zapravo pristupili samo jednom od servisa koji postoji na internetu. Reč je o servisu koji se naziva World Wide Web (skraćeno **WWW** ili jednostavno **web**).

Kao što ste u prethodnoj lekciji imali prilike da pročitate, internet je mnogo više od pregleda sadržaja različitih sajtova, omiljenih društvenih mreža ili čitanja elektronskih vesti. Ipak World Wide Web servis je uspeo da stekne status najpoznatijeg i najkorišćenijeg servisa, čije postojanje je direktno uslovljeno postojanjem svetske globalne mreže – interneta.

World Wide Web definiše se kao skup dokumenata (resursa), pri čemu je svaki takav dokument identifikovan korišćenjem specijalne adrese. Svi takvi dokumenti su međusobno povezani vezama koje se nazivaju hiperlinkovi i dostupni su preko interneta. Za pregled dokumenata WWW servisa koriste se specijalni programi, koji se nazivaju web pregledači (*engl. web browser*).



Slika 3.1. World Wide Web

Tvorac weba je Tim Berners-Li (Tim Berners-Lee), koji je 1989. godine, radeći u organizaciji CERN, kreirao prvi dokument kojim je predstavio ideju o novom internet servisu. Tim Berners-Li je 1990. godine napisao i prvi web browser i tako postavio temelje daljeg razvoja WWW servisa.

Svoje funkcionisanje WWW servis direktno zasniva na HTTP, odnosno HTTPS protokolima. Zbog toga je za adekvatno razumevanje WWW-a neophodno razumeti osnovna pravila HTTP protokola.

Šta je HTTP?

WWW i HTTP protokol su nerazdvojni pojmovi. Jednostavno, HTTP protokol razvijen je specijalno za potrebe WWW servisa, što znači da web ne bi postojao bez ovog protokola. Dalje, WWW i HTTP imaju identičnog tvorca. Radeći na novom internet servisu (WWW), Tim Berners-Li i njegov tim kreirali su i osnovne nacрте HTTP protokola.

U prethodnoj lekciji je već rečeno da je HTTP jedan od protokola aplikativnog sloja internet modela. To praktično znači da je reč o protokolu najvišeg nivoa, koji korisnici direktno upošljavaju. Da je to tako, verovatno ste i sami već primetili korišćenjem nekog web browsera. Jednostavno, prilikom posete nekog web sajta, verovatno ste primetili da adresa sajta započinje sa http ili https:

<https://www.google.com/>

Upravo je prikazana adresa jednog od najpopularnijih sajtova današnjice. Možete da primetite da ona započinje odrednicom *https*, što govori da se za komunikaciju koristi HTTP, odnosno HTTPS protokol (u suštini je reč o istom protokolu, pri čemu HTTPS predstavlja sigurniju verziju HTTP protokola; sve što u nastavku bude rečeno odnosi se na obe varijante protokola).

HTTP je takozvani *request-response* protokol, što znači da njegov način funkcionisanja počiva na međusobnom smenjivanju zahteva i odgovora između dva kompjutera. Pri tome dva kompjutera koja komuniciraju korišćenjem HTTP protokola nisu ravnopravna, već se jedan naziva **server**, a drugi **klijent**. Klijentski kompjuter je onaj sa koga je korišćenjem nekog web browsera upućen zahtev. Server je kompjuter koji takav zahtev prihvata, obrađuje i na kraju odgovara isporukom određene poruke ili resursa.

HTTP protokol definiše komunikaciju koja počiva na međusobnom smenjivanju zahteva i odgovora između servera i klijenta.



Slika 3.2. Klijent-server komunikacija

Slika 3.2. ilustruje uprošćenu komunikaciju između klijenta i servera posredstvom HTTP protokola. Sliku možete poistovetiti sa svakodnevnim korišćenjem weba. Na primer, prilikom otvaranja nekog web sajta korišćenjem web browsera (npr. Chromea) vaš kompjuter se ponaša kao HTTP klijent. Kompjuter na kome se nalazi web sajt kome pokušavate da pristupite naziva se HTTP server.

Resursi na webu

Kao što i sami znate, web je u pravom smislu te reči preplavljen sadržajem različitog tipa. Tekst, slike, audio i video zapisi, dokumenti različitog tipa samo su neki od sadržaja koji se nalaze na webu. Svi oni se objedinjeno nazivaju resursima. To praktično znači da kada na primer pokušavate da pogledate neki snimak sa YouTubea, vi zapravo upućujete zahtev za pristup određenom web resursu. U opisanoj situaciji, takav resurs je video-zapis.

Svi resursi na webu su jednoznačno određeni specijalnim adresama. Takve adrese omogućavaju razlikovanje resursa, a korisnicima obezbeđuju sistem za pristup. Prema tome <https://www.google.com> jeste primer jedne adrese web resursa.

Uniform Resource Locator (URL)

Adresa jednog web resursa se drugačija naziva *Uniform Resource Locator* ili skraćeno **URL**. Prosto rečeno, URL je ono što kucate u web browseru kada pokušavate da pristupite nekom sajtu. Po svojoj strukturi, URL je hijerarhijska sekvenca komponenti (slika 3.3).



Slika 3.3. Struktura URL adrese

Slika 3.3. ilustruje različite delove od kojih može biti sačinjen jedan URL:

- **scheme** – naziv protokola koji se koristi za komunikaciju;
- **fully qualified domain name (FQDN)** – definiše preciznu lokaciju kompjutera (hosta) unutar mreže;
- **port number** – oznaka koja definiše određeni proces ili servis koji se izvršava na kompjuteru povezanom na mrežu;
- **path** – putanja do određenog resursa na serverskom računaru;
- **resource** – naziv resursa koji se od servera zahteva;
- **query string** – opciona parametri koji se u formi parova ključeva i vrednosti mogu proslediti serveru.

Bitno je razumeti da korišćenje svih prikazanih delova URL-a nije obavezno. Drugim rečima, određeni delovi URL-a su opciona. Tako je potpuno legitimno napisati samo:

`http://www.mysite.com`

ili

<https://www.google.com>

Obe prikazane URL adrese potpuno su legitimne, dok se preostali delovi URL-a koji su prikazani slikom 3.3. mogu koristiti kada se za tako nešto javi potreba, kako bi se pristupilo specifičnom resursu unutar nekog hosta.

HTTP zahtev

Da bi se određeni zahtev uputio serveru poštovanjem pravila HTTP protokola, definisanje URL-a na kome se nalazi traženi resurs nije dovoljno. Pored URL-a, neophodno je definisati još neke informacije, upakovane u odgovarajuću formu. Tako najjednostavniji primer HTTP zahteva za jednom web stranicom može da izgleda ovako:

GET /index.html HTTP/1.1

host: www.mysite.com

Sada se sa pravom možete upitati: *Šta je ovo što je prikazano? Prilikom pristupa nekom sajtu ja nikada nisam bio u obavezi da kucam nešto ovako!*

To je zapravo potpuno tačno, ali samo zato što je formiranje prikazanog zahteva za vas u pozadini obavljao web browser (Firefox, Chrome, Safari...), koji je korišćen za upućivanje zahteva. Drugim rečima, prikazani zahtev jeste ono što browser upućuje nekom serveru kada vi u polju adrese sajta upišete: www.mysite.com.

Prikazani HTTP zahtev sadrži sve neophodne karakteristike koje su serveru potrebne kako bi mogao da formira odgovor. Ukoliko se prikazani zahtev razloži na delove, može se reći da je:

- **GET** – naziv HTTP metode;
- **/index.html** – resurs koji se zahteva;
- **HTTP/1.1** – verzija protokola;
- **host: www.mysite.com** – naziv hosta.

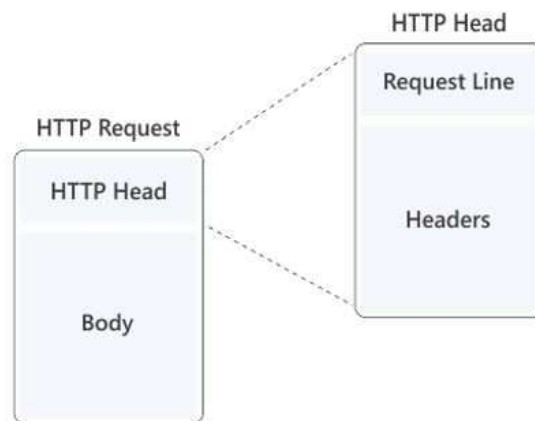
Prikazani HTTP zahtev predstavlja najosnovniji oblik jednog takvog zahteva. U stvarnosti se uglavnom koriste mnogo kompleksniji zahtevi, koji poseduju znatno više delova. Ipak, HTTP zahtev ne može imati proizvoljnu formu, već je prilikom njegovog formiranja potrebno pridržavati se nekih osnovnih pravila koja se tiču strukture HTTP zahteva.

Struktura HTTP zahteva

HTTP zahtev, koji klijent upućuje serveru, sastoji se iz dva osnovna dela i nekoliko poddelova:

- **glava** (engl. *head*);
 - početna linija ili **linija zahteva** (engl. *start line, request line*);
 - **zaglavlja** (engl. *headers*);
- **telo** (engl. *body*).,

Ukoliko bismo strukturu HTTP zahteva ilustrativno prikazali, ona bi izgledala kao na slici 3.4.



Slika 3.4. Struktura HTTP zahteva

Zahtev koji je prikazan nešto ranije bio je sačinjen iz linije zahteva i jednog zaglavlja. Pored zaglavlja koje govori kom hostu je potrebno uputiti zahtev, HTTP zahtev može sadržati i razna druga zaglavlja koja bliže određuju jedan zahtev. Tako primer jednog HTTP zahteva može da izgleda kao na slici 3.5.



Slika 3.5. Primer jednog HTTP zahteva

Slika 3.5. ilustruje primer jednog tipičnog HTTP zahteva. Zahtev započinje linijom zahteva (*engl. request line*), nakon čega je navedeno nekoliko zaglavlja koja bliže određuju sam zahtev. Telo zahteva je prazno.

Unutar tela zahteva mogu se smestiti različiti podaci, koji se zajedno sa zahtevom upućuju serveru. Iz prikazanog primera može se zaključiti da, u zavisnosti od namene zahteva, podaci unutar tela mogu, ali i ne moraju biti prisutni.

HTTP metode

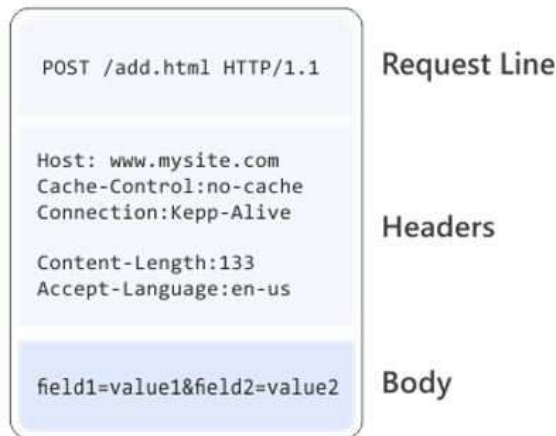
Prilikom prikaza prvog HTTP zahteva u prethodnim redovima po prvi put smo se susreli sa pojmom HTTP metoda, s obzirom da prikazani zahtev započinje navođenjem jedne takve metode (GET). Reč je o jednoj od nekoliko HTTP metoda koje se mogu koristiti prilikom formiranja HTTP zahteva. Tako su HTTP metode, pored same strukture HTTP zahteva, još jedan veoma značajan pojam, vezan za formiranje HTTP zahteva.

Za kreiranje HTTP zahteva, klijentima je na raspolaganju nekoliko različitih obrazaca komunikacije, koji se drugačije nazivaju HTTP metode. Takve metode govore serveru koja je osnovna namera zahteva.

Neke od najznačajnijih HTTP metoda su:

- **GET** – zahteva podatke određenog resursa;
- **HEAD** – zahteva samo HTTP zaglavlje, bez tela odgovora;
- **POST** – prosleđuje nov resurs serveru;
- **PUT** – prosleđuje serveru resurs kojim je potrebno zameniti već postojeći resurs;
- **DELETE** – briše resurs sa servera;
- **OPTIONS** – zahteva od servera da isporuči skup HTTP metoda koje podržava;
- **TRACE** – koristi se za testiranje i debug tako što server vraća klijentu zahtev koji je primio;
- **PATCH** – vrši parcijalnu promenu postojećeg resursa.

Pojam HTTP metoda možda nije najlakše razumeti, pogotovu ako se uzme u obzir da je reč o pojmovima koji nisu direktno vidljivi korisnicima. Zbog toga je za početnike teško da razumeju njihovu namenu. Stvarnost je takva da se jednostavnim surfovanjem webom u pozadini najviše koriste zahtevi formirani GET metodom. Na primer, kada pokušavate da pročitate članak na nekom blogu, browser će u pozadini formirati HTTP zahtev korišćenjem GET metode. U svakodnevnoj upotrebi na webu često se koristi i POST metoda HTTP protokola. Iz pregleda HTTP metoda možete da pročitate da je reč o metodi kojom se serveru prosleđuje nov resurs. Na primer, kada kreirate nalog na nekoj društvenoj mreži, browser će u pozadini iskoristiti upravo POST metodu kako bi podatke vašeg novog naloga prosledio serveru. POST je i jedna od HTTP metoda, čijim se korišćenjem upošljava i nešto ranije spomenuto telo HTTP zahteva. Naime, upotrebom POST metode podaci se serveru prosleđuju upravo korišćenjem tela HTTP zahteva (slika 3.6).



Slika 3.6. Primer jednog HTTP zahteva upućenog POST metodom

Slika 3.6. dočarava primer jednog HTTP zahteva upućenog POST metodom. Ovoga puta telo zahteva nije prazno, već se unutar njega nalaze podaci koji se prosleđuju serveru.

I preostale HTTP metode poseduju značajnu primenu na webu. Ipak, njihova upotrebna vrednost zahteva malo bolje poznavanje samog procesa razvoja web aplikacija. Stoga ćemo se u nastavku ovoga programa još nekoliko puta vratiti na osobine HTTP zahteva i primenu različitih HTTP metoda.

HTTP odgovor

U dosadašnjem toku ovoga poglavlja razmatrane su osobine zahteva koje klijenti mogu da upute serveru preko interneta, poštovanjem pravila HTTP protokola. Po prijemu zahteva, server vrši njegovu obradu i formiranje odgovora. Naravno, izgled odgovora u najvećoj meri zavisi od tipa zahteva koji je klijent uputio.

Primer jednog jednostavnog HTTP odgovora može da izgleda ovako:

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 500
```

```
<html>
<body>
```

some server response...

```
</body>
</html>
```

Prilikom formiranja HTTP odgovora, baš kao što je to bio slučaj i prilikom formiranja zahteva, postoje određena pravila, unapred definisana HTTP protokolom. Drugim rečima, može se reći da i HTTP odgovor poseduje određenu strukturu, koja je veoma slična HTTP zahtevu.

Struktura HTTP odgovora

Struktura HTTP odgovora gotovo je identična strukturi HTTP zahteva. Čine je:

- **glava** (*engl. head*);
 - početna linija ili **statusna linija** (*engl. start line, status line*);
 - **zaglavlja** (*engl. headers*);
- **telo** (*engl. body*).

Osnovna i najznačajnija razlika između strukture HTTP zahteva i odgovora tiče se početne linije, koja se kod HTTP odgovora drugačije naziva statusna linija. Ona se sastoji od:

- verzije protokola (npr. HTTP/1.1),
- statusnog koda;
- statusnog teksta.

Primer jednostavnog HTTP odgovora, prikazanog iznad, ilustrativno se može razložiti na celine kao na slici 3.7.



Slika 3.7. Primer jednog HTTP odgovora

Na slici 3.7. se može videti da serverski odgovor započinje statusnom linijom, koja sadrži informaciju o uspešnosti obrade zahteva koji je uputio klijent. Ovo je jedna od najznačajnijih informacija koju server isporučuje klijentu, naravno pored opcionih podataka unutar tela.

HTTP statusni kodovi

Za označavanje uspešnosti obrade klijentskog zahteva, HTTP protokol poznaje veliki broj statusnih kodova, podeljenih u nekoliko grupa. U prikazanom primeru server unutar statusne linije navodi kod 200, što znači da je klijentski zahtev uspešno obrađen. Statusni kod jeste osnovna informacija na osnovu koje klijent može znati kako je protekla obrada zahteva upućenog serveru. Različite grupe statusnih odgovora, koje se često nazivaju i statusne klase, jesu:

- **1xx** – informacije;
- **2xx** – uspešno izvršavanje serverske logike pri obradi zahteva;
- **3xx** – preusmeravanja (*engl. redirections*);
- **4xx** – klijentske greške;
- **5xx** – serverske greške.
- Neki od najznačajnijih konkretnih HTTP statusnih kodova su:
- **200 (OK)** – obrada zahteva je uspešna i server je isporučio odgovor;
- **201 (Created)** – obrada zahteva je uspešna i pritom je kreiran novi resurs;
- **301 (Moved Permanently)** – resurs koji se zahteva je premešten na novu adresu;
- **302 (Found)** – resurs koji se zahteva je privremeno premešten na drugu adresu;
- **304 (Not Modified)** – resurs se nije promenio od prethodnog pristupa; koristi se prilikom keširanja resursa kako bi klijent mogao da proveri da li postoji ažurnija verzija resursa;
- **400 (Bad Request)** – server nije razumeo zahtev;
- **401 (Unauthorized)** – pristup resursu zahteva autorizaciju;
- **403 (Forbidden)** – pristup resursu je zabranjen;
- **404 (Not Found)** – server ne može da pronađe traženi resurs;
- **500 (Internal Server Error)** – greška na serveru prilikom obrade zahteva.

Sa nekim od prikazanih HTTP statusnih kodova sigurno ste se već susretali. Na primer, verovatno ste nekada unutar browsera videli ovakvu ili sličnu stranicu (slika 3.8).



404. That's an error.

The requested URL /chrome/sdgsdggsgdgsdg was not found on this server. That's all we know.



Slika 3.8. Primer 404 stranice

Slika 3.8. ilustruje primer stranice koja se dobija u slučaju da server ne može da pronađe traženi resurs. Naravno, prikazana stranica je ono što vidi korisnik, dok browser u pozadini, na početku HTTP odgovora, unutar statusne linije dobija status: **404 NOT FOUND**.

Pitanje

Koji HTTP statusni kod se koristi da označi uspešno obrađen zahtev?

- 404
- 201
- **200**
- 300

Objašnjenje:

Statusni kod 200, koji HTTP server isporučuje klijentu, označava da je obrada zahteva uspešna i da je server isporučio odgovor.

Rezime

- World Wide Web se definiše kao skup dokumenata (resursa), pri čemu je svaki takav dokument identifikovan korišćenjem specijalne adrese.
- Tvorac weba je Tim Berners-Li, koji je 1989. godine, radeći u organizaciji CERN, kreirao prvi dokument kojim je predstavio ideju o novom internet servisu.
- Svoje funkcionisanje WWW servis direktno zasniva na HTTP, odnosno HTTPS protokolima.
- HTTP protokol razvijen je specijalno za potrebe WWW servisa, što znači da web ne bi postojao bez ovog protokola.
- HTTP je takozvani *request-response* protokol, što znači da njegov način funkcionisanja počiva na međusobnom smenjivanju zahteva i odgovora između dva kompjutera.
- U komunikaciji kompjutera korišćenjem HTTP protokola razlikuju se dva tipa uređaja: klijenti i serveri.
- Klijentski kompjuter je onaj sa koga je korišćenjem nekog web browsera upućen zahtev.

- Server je kompjuter koji zahtev prihvata, obrađuje i na kraju odgovara isporukom određene poruke ili resursa.
- Svi resursi na webu su jednoznačno određeni specijalnim adresama, koje se nazivaju Uniform Resource Locator ili skraćeno URL.
- HTTP komunikacija se zasniva na međusobnom smenjivanju zahteva i odgovora; pritom zahtevi i odgovori poseduju određenu strukturu, koja je utvrđena pravilima HTTP protokola.



linkgroup