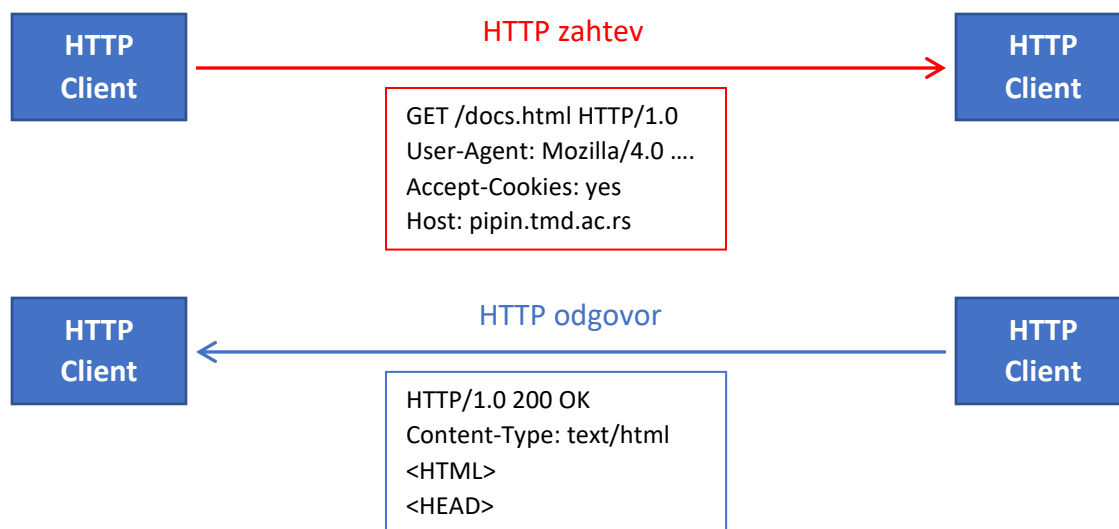


Web programiranje – Vežbe 4

HTTP

HTTP komunikacija

HTTP (Hyper Text Transfer Protocol) komunikacija je zasnovana na **zahtev/odgovor** principu, pri čemu je svaki par **zahtev/odgovor** nezavisan od ostalih, što kao posledicu ima da ne omogućava praćenje korisničke sesije.



HTTP zahtev je formatiran na sledeći način:

4. METOD /putanja HTTP/verzija
 - METOD može biti GET, POST, PUT, DELETE,...
5. Dodatni redovi
 - Atributi oblika - ime: vrednost
6. Prazan red na kraju

HTTP odgovor je formatiran na sledeći način:

1. HTTP/verzija kod tekstualni_opis
 - 200 OK, 404 Not Found, 403 Forbidden, 500 Internal Server Error,...
2. Dodatni redovi
 - Atributi oblika - ime: vrednost
3. Sadržaj datoteke

Primer – Jednostavan Web Server

Projekat **HttpWebServer** sadrži klasu **Server**. Primer ilustruje prostu implementaciju web servera koji prima **zahteve** i šalje **odgovore** korišćenjem **HTTP-a**:

- void run()
- String getResource(InputStream is)

- void sendResponse(String resource, OutputStream os)

```
// prihvataj zahteve
socket = serverSocket.accept();

// dobavi resurs zahteva
String resource = this.getResource(socket.getInputStream());

if (resource.equals(""))
    resource = "static/index.html";

// posalji odgovor
this.sendResponse(resource, socket.getOutputStream());
socket.close();
socket = null;
```

Isečak 1

U gornjem isečku koda (Isečak 1) se nalazi deo **Server.run** metode. Server konstantno prihvata **zahteve** od strane klijenta. Struktura primera je takva da se na osnovu pročitano **zahteva** uzima resurs (fajl) koji je prosleđen u url-u. Kada je **zahtev** isparsiran i resurs dobavljen (Server.getResource) salje se **odgovor** klijentu.

U donjem isečku koda (Isečak 2) se vrši parsiranje **zahteva** (iz metode **Server.getResource**) i dobavljanje traženog resursa iz url-a **zahteva**. Iz prosleđenog stream-a čitamo liniju po liniju HTTP **zahteva**. Prva linija HTTP zahteva je formatirana kao **METOD /putanja HTTP/verzija**. Iz prve linije nas interesuje koja je HTTP metoda u pitanju (u našem slučaju prolazi samo GET), i koji resurs se traži u **zahtevu**. Nakon što je resurs izvučen iz **zahteva** potrebno je da klijentu pošaljemo **odgovor**.

```
BufferedReader dis = new BufferedReader(new InputStreamReader(is));
String s = dis.readLine();
String[] tokens = s.split(" ");

// prva linija HTTP zahteva: METOD /resurs HTTP/verzija
// obradjujemo samo GET metodu
String method = tokens[0];
if (!method.equals("GET"))
    return null;

// String resursa
String resource = tokens[1];

// izbacimo znak '/' sa pocetka
resource = resource.substring(1);
```

Isečak 2

Slanje odgovora klijentu se nalazi u metodi (**Server.sendResponse**). U donjem isečku koda (Isečak 4) za prosleđeno ime resursa, prvo proveravamo da li resurs postoji. Ukoliko traženi resurs ne postoji, klijentu se vraća HTTP **odgovor** sa statusom **404 – Not Found**.

```

PrintStream ps = new PrintStream(os);

// zamenimo web separator sistemskim separatorom
resource = resource.replace('/', File.separatorChar);
File file = new File(resource);

if (!file.exists())
{
    // ako datoteka ne postoji, vratimo kod za gresku
    String errorCode = "HTTP/1.0 404 File not found\r\n"
        + "Content-type: text/html; charset=UTF-8\r\n\r\n<b>404 Not found:"
        + file.getName() + "</b>";

    ps.print(errorCode);

    // ps.flush();
    System.out.println("Could not find resource: " + file);
    return;
}

```

Isečak 3

Ukoliko resurs postoji formira se HTTP **odgovor**. Prvo se formira zaglavlje **odgovora** po formatu **HTTP/verzija kod tekstualni_opis**, a zatim se ispisuje i traženi resurs (datoteka) (Isečak 3).

```

// ispisemo zaglavlje HTTP odgovora
ps.print("HTTP/1.0 200 OK\r\n\r\n");

// a, zatim datoteku
FileInputStream fis = new FileInputStream(file);
byte[] data = new byte[8192];
int len;

while ((len = fis.read(data)) != -1)
    ps.write(data, 0, len);

```

Isečak 4

Zadaci

1. Registracija korisnika:
 - a. Napisati HTML datoteku sa formom za registraciju korisnika.
 - b. Omogućiti navigaciju do nove HTML datoteke sa početne stranice.
 - c. Omogućiti da se formira HTTP zahtev koji će registrovati korisnike pomoću forme.
 - d. Odgovor treba da sadrži HTML stranicu o uspešnoj registraciji korisnika.
2. Listing registrovanih korisnika:
 - a. Omogućiti da se formira HTTP zahtev koji će prikazati listu do sada registrovanih korisnika.
 - b. Odgovor treba da sadrži stranicu sa tabelom korisnika.