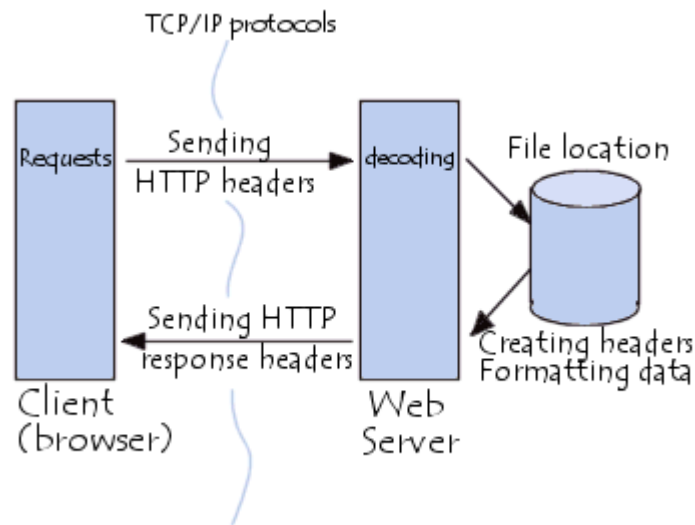


HTTP Protokol

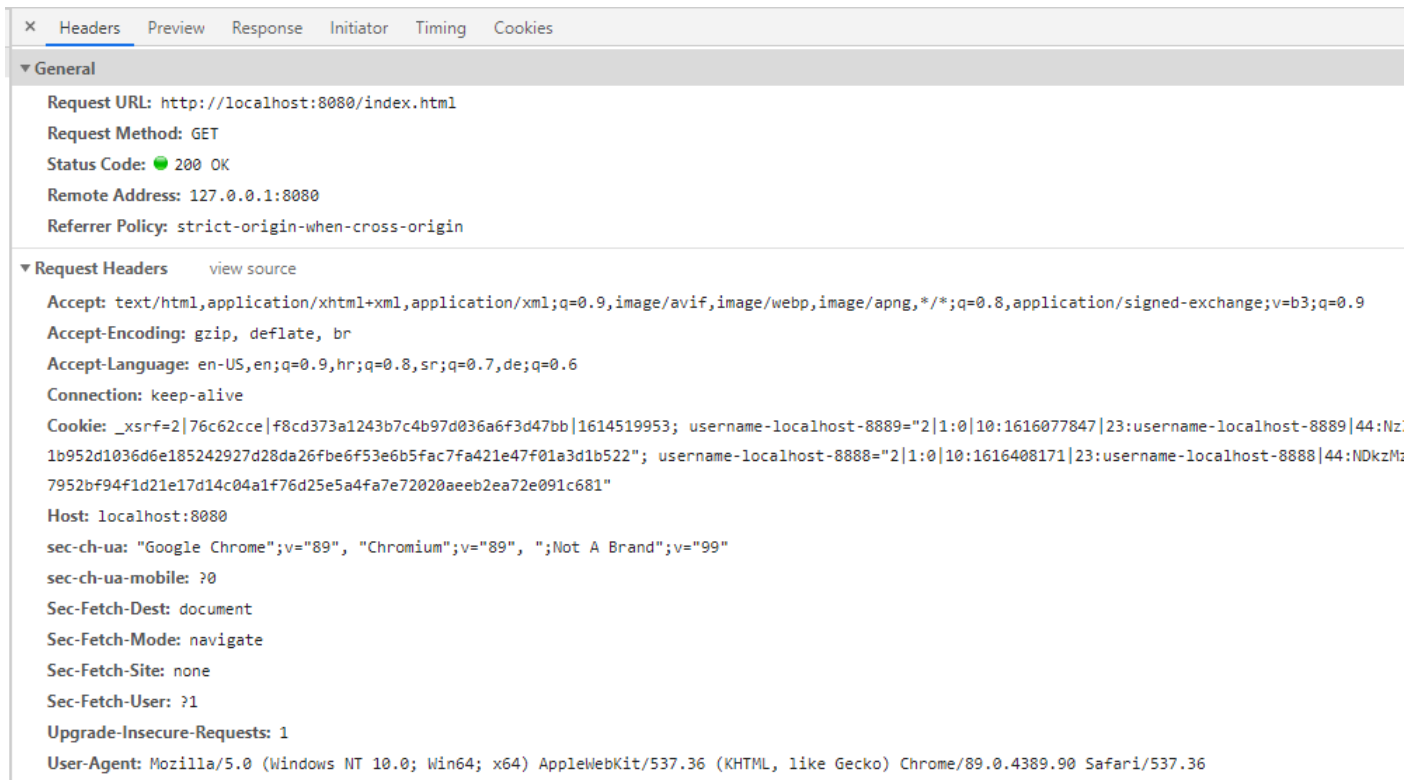
HTTP (eng. Hypertext Transfer Protocol) Komunikacija između klijenta (web browser-a) i servera Slika 1.



Slika 1. HTTP protokol

HTTP zahtev

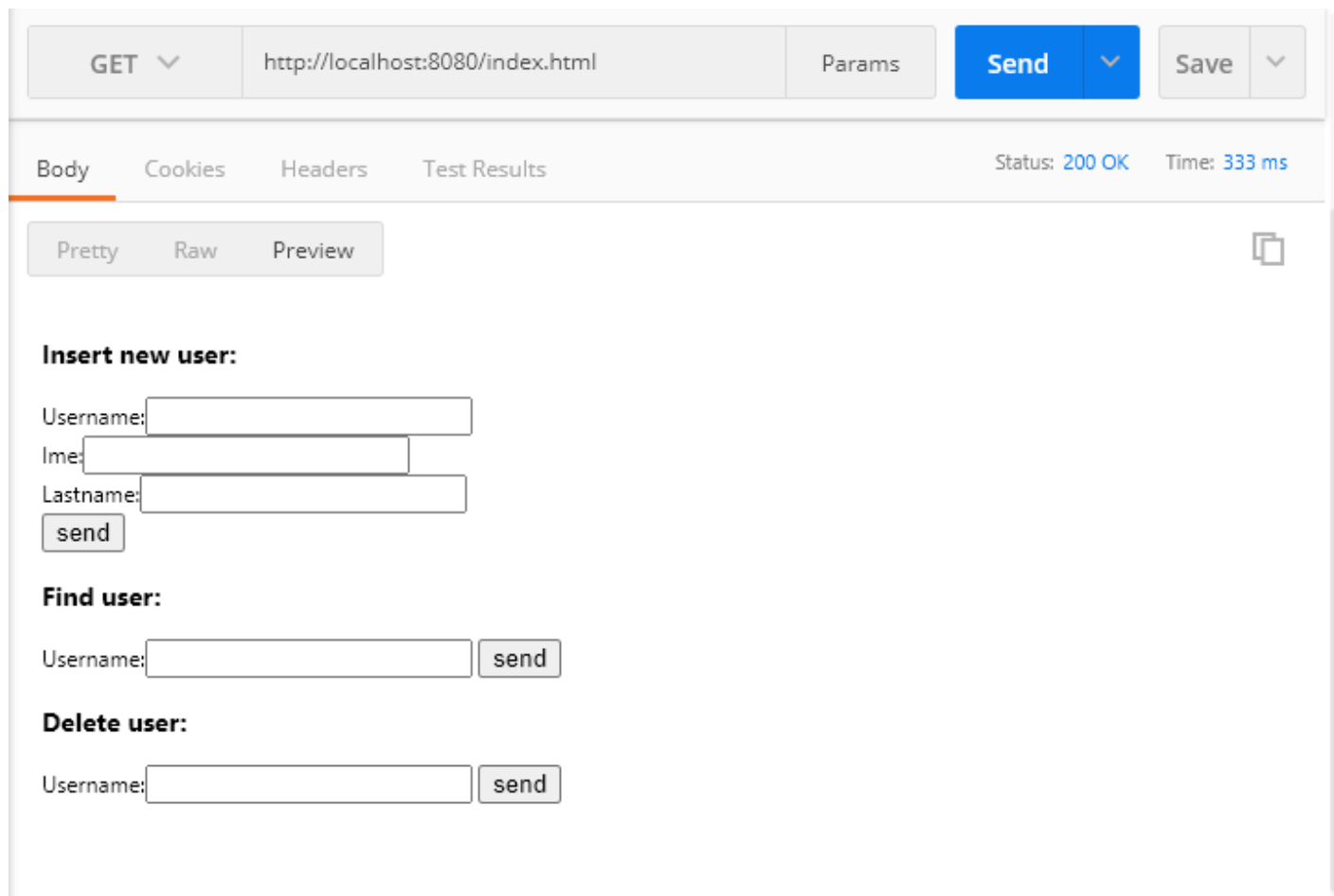
HTTP zahtev sastoji se iz linija koje šalje web čitač serveru. Slika 1.



Slika 1.

HTTP odgovor

HTTP odgovor se sastoji iz linija koje šalje server veb čitaču. Slika 2.



Slika 2.

Komunikaciju uvek započinje klijent.

Zadatak

Proširiti primer HTTPProtokol.zip. Funkcionalnosti koje su implementirane prošle nedelje slobodno koristiti u ovom zadatku.

Zadatak se sastoji od klijentskog dela aplikacije index.html i serverskog dela aplikacije Httpd program implementirano u C#. Trenutno zadatak sa časa radi preko dodajUsrname.html forme, sadržaj iz dodajUsrname.html prebaciti u index.html fajl

Pojašnjenje koda

Klijentska strana:

Klijentska strana aplikacije je index.html stranica. Za razliku od prošle nedelje: klijentski deo aplikacije je bila standardna konzolna aplikacija napisano isto u C#. Stranica index.html će sadržati forme za unos novog korisnika, pretragu korisnika i brisanje korisnika. Ova stranica je statička i vraća se kao resurs sa serverske strane. Ovo znači da nećete iz fajl sistem da pristupate stranice index.html tako što odradite dvoklik na fajl, ovaj način je pogrešan. Već morate da prosledite zahtev serveru (Httpd projektu) koji će onda ovu stranicu da vrati

kao resurs. Zahtev prosleđujete tako što u adresnoj liniji upisati localhost:8080/zahtev. Zahtev može da bude resurs u vidu naziva statičke stranice koje se nalazi na serveru, ili get naredba koja se prosledi preko formi. Get naredbe imaju oblik action?input_name1=input_value1&input_name2=input_value2 ... itd. Action je vrednost atributa action koji se definiše unutar form taga i ima oblik action="<http://localhost:8080/akcija>".

Na primer, svaka funkcionalnost će imati zasebnu formu:

- forma za unos novog korisnika:
 - action="<http://localhost:8080/dodaj> (<http://localhost:8080/dodaj>)"
- forma za pretragu:
 - action="<http://localhost:8080/pretraga> (<http://localhost:8080/pretraga>)"
- forma za brisanje:
 - action="<http://localhost:8080/obrisi> (<http://localhost:8080/obrisi>)"

Napomena: forma mora da ima definisan atribut method="get" (ili da se uopšte ne definiše metod vrednost, default je get) kako bi se ovaj sadržaj iščitao na serverskoj strani. Tako je implementiran zadatak (pogledati GetResource metodu na liniji 100 u Program.cs)

Serverska strana:

(Program.cs) Početak je isti kao prošle nedelje, definiše se endpoint (ovog puta se zauzima port 8080) i definisanje server socketa. Napomena: ako je slučajno zauzet port od strane druge aplikacije (procesa) slobodno promenite u sledeći slobodan 8081. Vodite računa da ovu izmenu primenite i unutar formi koje su definisane u index.html. Program je pokrenut u beskonačnoj petlji i prisluškuje zahteve.

Unutar Run metode treba da implementiramo funkcionalnosti zadatka. Preko StreamReader-a očekuju se zahtevi od strane klijenta(klijenata) i sa StreamWriter-om server odgovara klijentu tj. šalje eng. response. Metoda GetResource izdvaja resurs iz zaglavlja zahteva.

Zadatak je implementiran ako korisnik prosledi localhost:8080/ tj. prazan zahtev automatski se prosleđuje resurs, fajl index.html. Fajl index.html je home page (uglavnom) veb sajta i nalazi u istom root direktorijumu zajedno sa Httpd programom.

```
if (resource.Equals(""))  
    resource = "index.html";
```

Na liniji 70 (Program.cs) piše se dalje proverava šta je poslato od strane klijenta i u zavisnosti od toga primenjuje se odgovarajuća akcija (dodaj, pretraga, obiris).

Prilikom vraćanja odgovora server obavezno prosleđuje response u sledećem obliku:

- na vrhu se nalazi HTTP/verzija kod tekstualni_opis
- potom ide prazan red
- nakon toga ide dinamički sadržaj html stranice