

Inženjerstvo serverskog sloja – Vežbe 5

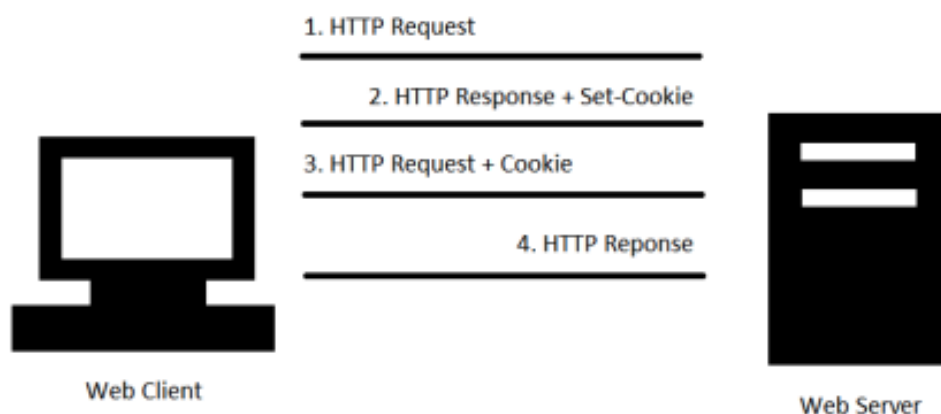
Sesija i File upload

Praćenje sesije

HTTP protokol je stateless protokol. Server ne čuva status klijenta ili korisničke sesije, zahtevi koji se šalju od strane klijenta (klijenata) su nezavisni. Kako bismo omogućili čuvanje podataka o ulogovanom korisniku postoje metode za održavanje i upravljanje sesijom.

1. 1. Praćenje sesije preko HttpSession objekta

- Za zadatak koji će biti rađen na času (tematika identična kao na prethodnim vežbama), neophodno je kreirati POJO klasu koja će nam predstavljati korisnika u našem sistemu
- prilikom uspešne prijave (login) na sistem u HttpSession objekta dodajemo korisnika **session.setAttribute("user", user)**.
- Za dobavljanje **HttpSession** objekta koristi se metoda **getSession()** iz request objekta u doGet/dopost metodi ko je tipa **HttpServletRequest**.
- Za odjavljivanje korisnika koristiti metodu **invalidate()** iz **HttpSession** objekta.



slika 1 - praćenje sesije preko cookie mehanizma

File upload

Prilikom implementacije file upload neophodno je proslediti post zahtev sa atributom **enctype: multipart/form-data format**.

Forma treba da ima attribute **method="post"** i **enctype="multipart/form-data"**. Unutar forme treba da imate input sa atribut **type="file"** i input **type="submit"**. Bitan je i atribut name npr. **name="file"**.

Na serverskoj strani, u servletima, upload je moguće realizovati na više načina, pri čemu je jedan oblik sledeći:

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    DiskFileItemFactory factory = new DiskFileItemFactory();  
    factory.setSizeThreshold(2000000);  
    ServletFileUpload upload = new ServletFileUpload(factory);  
    upload.setSizeMax(3000000);  
    try {  
        List<FileItem> items = upload.parseRequest(request);  
        for (FileItem item : items) {  
            String fieldName = item.getFieldName();  
            String fileName = item.getName();  
            File uploadedFile = new File(  
                getServletContext().getRealPath("/") + fileName);  
            System.out.println(getServletContext().getRealPath("/") + fileName);  
            item.write(uploadedFile);  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    doGet(request, response);  
}
```

Da bi naveden kod bio funkcionalan, moramo da uvedemo **Apache Commons** biblioteku, pri čemu je neophodno da dodamo **commons-io-1.1.jar** i **commons-fileupload-1.1.jar** u build path u Eclipse-u i takođe da se jar-ovi nađu u **/WEB-INF/lib** direktorijumu kako bi aplikacija radila i na serveru.

Zadaci

1. Omogućiti Logovanje na sistem preko stranice za prijavu gde će korisnik uneti svoje kredencijale (korisničko ime i lozinku). Za potrebe prijave definisati par korisnika unapred.
2. Svaki korisnik može da registruje druge korisnike, pri čemu se trenutno ulogovanom korisniku prikazuje lista korisnika koje je on registrovao.
3. Kreirati formu za upload fajla koji će sačuvati isti u samom projektu i omogućiti registraciju korisnika time što će fajl sadržati linije u formatu **korisnik:lozinka**. fajl neophodno isparsirati i kreirati korisnike u sistemu.