



Osnove web programiranja

Prijava, odjava i keširanje

Termin 6

Sadržaj

1. Prijava sa sistema
2. Odjava sa sistema
3. Sprečavanje keširanje sadržaja na klijentu
4. Arhitektura Spring MVC

Sutup

Pokretanje primera

Importovati u Eclipse projekat u BioskopVebAplikacijaT6T7.zip

Otići u application.properties fajl i izmeniti kod tako da ukazuje na Vašu lokaciju na disku, i navedenoj putanji postaviti filmovi.txt fajl

```
#putanja do lokacije gde se nalaze filmovi txt
```

```
filmovi.pathToFile=c:\\ImePrezime\\Podaci\\filmovi.txt
```

```
filmovi.pathToFile=/imeProfila/home/ImePrezime/Podaci/filmovi.txt
```

Prijava na sistem

Rad sa autorizovanim korisnicima

- Potrebno je omogućiti pristup bioskop aplikaciji samo za autorizovane korisnike
- Ne može se isti korisnik više puta prijaviti sa različitih Klijenata
- Prilikom prijave na sistem unosi se korisničko ime i šifra
 - Prilikom prijave korisnika proverava se da li dati korisnik postoji u sistemu, da li je korisnička šifra dobro unesena, i da li je korisnik već prijavljen

Prijava na sistem

Klasa Korisnik

- Klasa Korisnik:
 - reprezentuje korisnika
 - postoje korisnici koji su administratori
 - sadrži informaciju o tome da li je korisnik prijavljen
- Pogledati klasu Korisnik

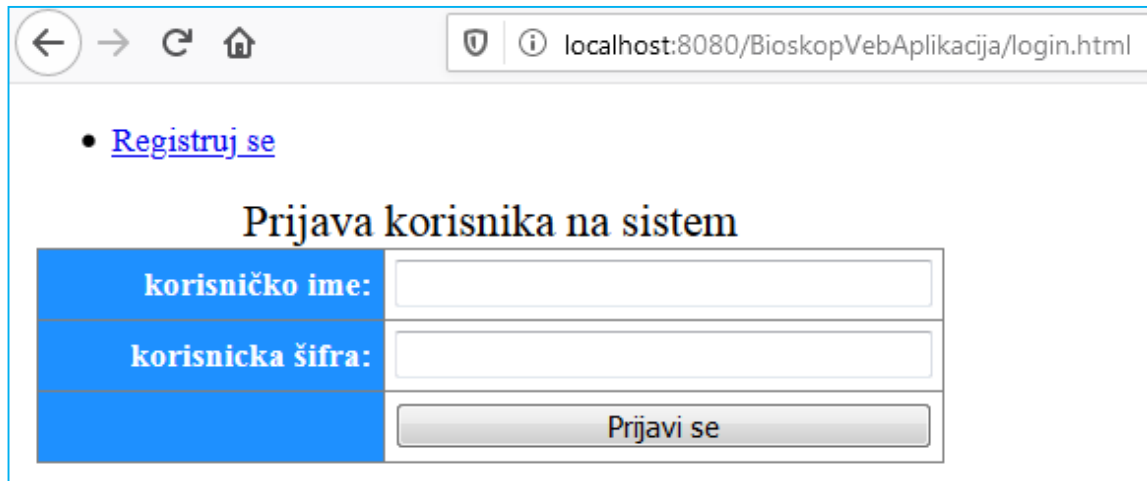
Prijava na sistem

HTML stranica za logovanje

- Kreirana je HTML stranica login.html koja omogućava unos podataka za logovanje korisnika
- login.html sadrži formu za unosa korisničkog imena i korisničke šifre, a klikom na dugme Uloguj se, odlazi se na login metodu od PrijavaOdjavaController kontrolera
- Stranica takođe sadrži link ka registraciji korisnika registracija.html (u slučaju da je prethodno neophodno da se korisnik registruje)
- HTML stranica login.html kreirana je tako se se zadrži postojeći izgled stilova aplikacije (po ugledu HTML stanicu za **registraciju korisnika**)
- Dodat je link ka login stranici u index.html

Prijava na sistem

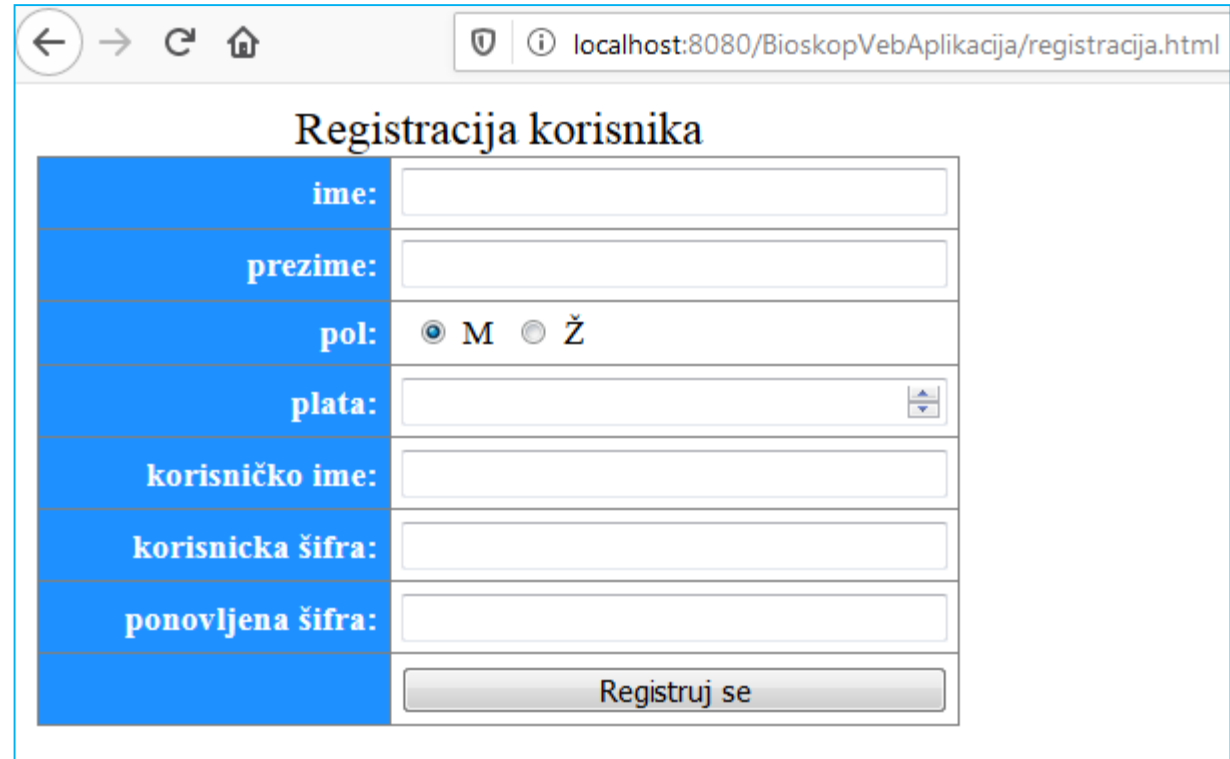
HTML stranica za logovanje



• [Registruj se](#)

Prijava korisnika na sistem

korisničko ime:	<input type="text"/>
korisnicka šifra:	<input type="password"/>
<input type="button" value="Prijavi se"/>	



Registracija korisnika

ime:	<input type="text"/>
prezime:	<input type="text"/>
pol:	<input checked="" type="radio"/> M <input type="radio"/> Ž
plata:	<input type="text"/>
korisničko ime:	<input type="text"/>
korisnicka šifra:	<input type="password"/>
ponovljena šifra:	<input type="password"/>
<input type="button" value="Registruj se"/>	

login.html

Prijava na sistem

Kontroler za prijavu i odjavu

- Kreiran je kontroler PrijavaOdjavaController koji omogućava prijavljivanje korisnika na sistem i odjavu korisnika sa sistema
 - Kontroler je dostupa preko URL – PrijavaOdjava
 - Obezbeđen je pristup ServletContext objektu i inicijalizaciju kontrolera
 - **Da li u prijava odjava PrijavaOdjava kontroleru treba da se kreira mapa korisnika?**
 - **Kako izmeniti kod init metoda da se samo jednom definiše mapa korisnika za celu aplikciju?**
- Kreirana su Handler metode.getLogin i postLogin koja omogućavaju prijavu korisnika na sistem za URL Login
 - ako za uneseno korisničko ime ne postoji korisnik, nije odgovarajuća korisnička šifra prikazaće se stranica koja ispisuje grešku prilikom logovanja (neispravno korisničko ime ili lozinka ili korisnik je već negde prijavljen).
 - ako za korisničko ime je već prijavljen na sistem na drugom Klijentu ili je korisnik već prijavljen na sistem na istom klijentu prikazaće se stranica koja ispisuje grešku prilikom logovanja (korisnik je već prijavljen na sistem morate se prethodno odjaviti)
 - ako za uneseno korisničko ime postoji korisnik u sistemu i on nije već prijavljen, prijavice se korisnik (ubaciće se korisnik u sesiju) i prikazaće se se stranica za prikaz svih filmova

PrijavaOdjavaController i login

Prijava na sistem

Izmena za autorizovane korisnike

- Omogućen je pristup bioskop aplikaciji samo za autorizovane korisnike
 - Za kontroler FilmoviController u svakoj Handler metodi proverava se da li korisnik postoji u sesiji, ako ne postoji preusmerava se poziv na login.html stanicu
- na svakoj HTML stranici aplikacije ispisan je ulogovanon korisnik

pristup autorizovanih

Odjava sa sistema

Odjava

Svaka aplikacija ima login i logout

- Kako izgleda logout procedura?
- Korisnik inicira odjavu
- Šta bi trebalo da se dogodi kada korisnik pritisne logout dugme?
 - Obrišemo sesiju
 - Cookie više nije validan
 - Vratimo korisnika na login stranicu

Odjava sa sistema

Odjava i uništavanje sesije

Realizovana je odjava sa sistema u kontroleru za prijavu i odjavu.

- Kreirana je Handler metoda logout koja omogućava odjavu korisnika sa sistema
 - ako je korisnik nije prijavljen na sistem na tom klijentu, prikazuje se stranica koja ispisuje grešku prilikom logovanja (korisnik nije prijavljen).
 - ako je korisnik prijavljen na sistem na tom klijentu, odjavljuje se korisnik (izbacuje se korisnika iz sesije i uništava se sesija) i prikazuje se stranica za logovanje.
- Omogućena je odjava sa sistema na svakoj HTML stranici bioskop aplikacije
 - Za kontroler FilmoviController u svakoj Handler metodi koja vraća HTML sadržaj ubačen je link za odjavu pored ispisa korisnika koji je prijavljen

Odjava sa sistema

Odjava i uništavanje sesije

- `Request.getSession() == Request.getSession(true)`
- `Request.getSession(false)` – pribavlja sesiju, a ako nema sesije neće kreirati novu sesiju
- sesija se poništava – `session.invalidate()`
- Klasa `Korisnik` sadrži informaciju o prijavi/odjavi
- Šta ako se korisnik odjavi, pa pritisne Back u navigatoru?

PrijavaOdjavaController logout

- Obrisati local cache u navigatoru za BioskopVebAplikaciju

Sprečavanje keširanje sadržaja na klijentu

no-cache

- Šta ako se korisnik odjavi, pa pritisne Back u navigatoru?
- Treba postaviti dodatne parametre HTTP zaglavlja za HTML dokument koji se vraća korisniku
- podesiti Cache-Control na no-cache u zaglavlju odgovora HTTP objekta
 - `response.setHeader("Cache-Control", "private, no-cache, no-store, must-revalidate");`
`// HTTP 1.1`
 - `response.setHeader("Pragma", "no-cache");` `// HTTP 1.0.`
 - `response.setDateHeader("Expires", 0);` `// Proxies.`
- META tag sa istim sadržajem u HTML kodu
 - `<META HTTP-EQUIV="Cache-Control" CONTENT="no-cache">`
 - `<META HTTP-EQUIV="Pragma" CONTENT="no-cache">`
 - `<META HTTP-EQUIV="Expires" CONTENT="0">`

no-cache

Arhitektura Spring MVC

Arhitektura Spring MVC aplikacije sa procesom rada– bez detalja

