

### 3. cas (video 2.cas)

---

1. New JPA proj
2. JPA impl. type : Disable Libr. Conf.
3. Conf. > Convert to Maven proj...
4. pom.xml after build

```
<dependencies>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.22</version>
  </dependency>
  <dependency>
    <groupId>org.hibernate.javax.persistence</groupId>
    <artifactId>hibernate-jpa-2.1-api</artifactId>
    <version>1.0.2.Final</version>
  </dependency>
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.4.22.Final</version>
  </dependency>
</dependencies>
```

5. Run as > maven install
6. ~/.m2/repo.../mysql/mysql-connector-java/.../....jar
7. Build path > Conf. Build path... > Libraries > ModulePath> JreSysLib > Edit...> Worksp. default JRE
8. Data Source Exp > new con. prof. > mysql > (+) new driver...> add jar ~/.m2/repo.../mysql/mysql-connector-java/.../....jar > rem old jar
9. fill

```
Host: nastava.is.pmf.uns.ac.rs
user: ris
password: ris2Nastava!
Schema: is2
```

10. JPA tools > Gen Entits from Tables... >....> Mapping defs. Key gener. > identity
11. model> Knjiga.java

```

...
@Entity// ← VAZNO*****
@Table(name="KNJIGA") // ← VAZNO*****
@NamedQuery(name="Knjiga.findAll", query="SELECT k FROM Knjiga k")
public class Knjiga implements Serializable {
    private static final long serialVersionUID = 1L;
....

```

12. JpaContent > persistence.xml > Connection (*bottom card switcher*)> Transaction type > Resource local

- **Populate from connection...**

13. src>**new pkg** managers> **new class** JPAUtil

```

...
public class JPAUtil {
    private static EntityManagerFactory entityManagerFactory;

    static {
        entityManagerFactory =
Persistence.createEntityManagerFactory("BibliotekaJPA");
    }

    public static EntityManager getEntityManager() {
        return entityManagerFactory.createEntityManager();
    }
}

```

14. managers>**new class** KnjigaManager

```

public class KnjigaManager {

    public Knjiga saveKnjiga(String naslov, String autor, String izdavac,
String godinaIzdanja) {
        try {
            EntityManager em = JPAUtil.getEntityManager();

            Knjiga k = new Knjiga();
            k.setAutor(autor);
            k.setGodinaIzdanja(godinaIzdanja);
            k.setIzdavac(izdavac);
            k.setNaslov(naslov);

            em.getTransaction().begin();
            em.persist(k);
            em.getTransaction().commit();
            return k;
        } catch (Exception e) {
            e.printStackTrace();

```

```

        return null;
    }
}...

```

#### 15. KnjigaManager.java

```

    public List<Knjiga> getSveKnjige(){
        EntityManager em = JPAUtil.getEntityManager();
        List<Knjiga> knjige = em.createQuery("from Knjiga",
Knjiga.class).getResultList();
        return knjige;
    }

```

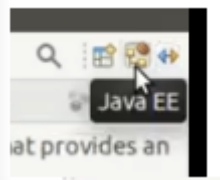
#### 16. KnjigaManager.java **Testiranje zivaca (opciono)**

```

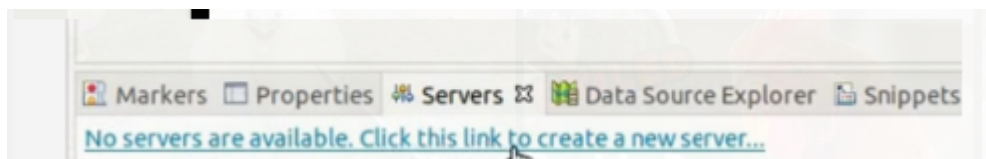
...
public static void main(String[] args) {
    KnjigaManager km = new KnjigaManager();
    // ODKOMENTARISI *****
    //      Knjiga knjiga = km.saveKnjiga("Hajdi", "Johana Spiri", "Laguna",
"1994");
    //      if(knjiga!=null)
    //          System.out.println("Knjiga je uspesno sacuvana. Id knjige je:
"+knjiga.getIdKnjige());
    // *****
    List<Knjiga> knjige = km.getSveKnjige();
    if(knjige!=null)
        for(Knjiga k:knjige)
            System.out.println(k.getNaslov()+"", "+k.getAutor());
    }
}

```

#### 17.

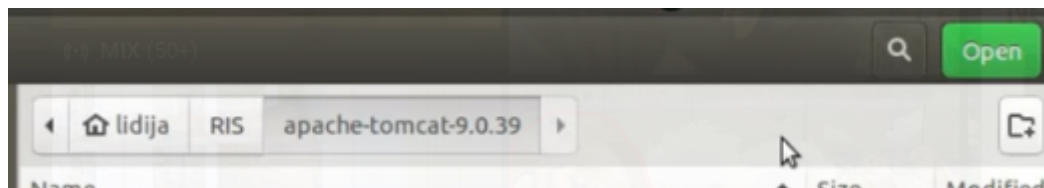


#### 18.

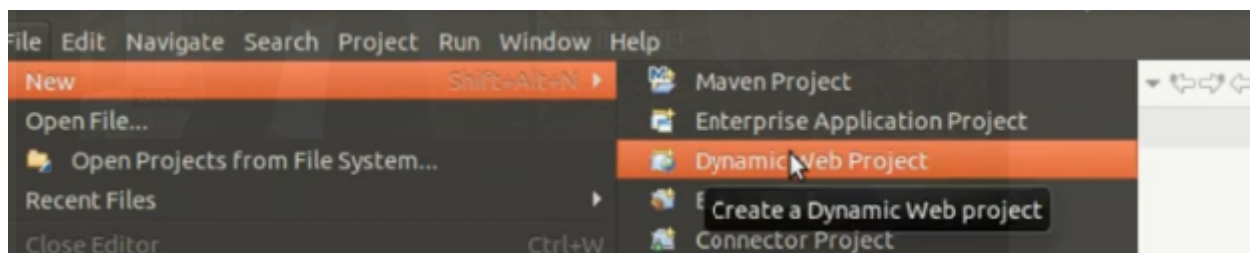




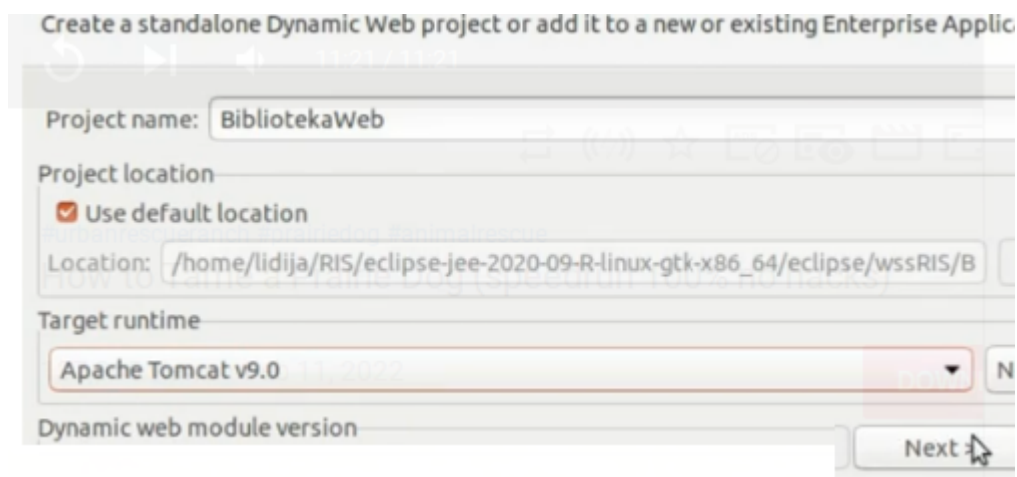
19.



20.



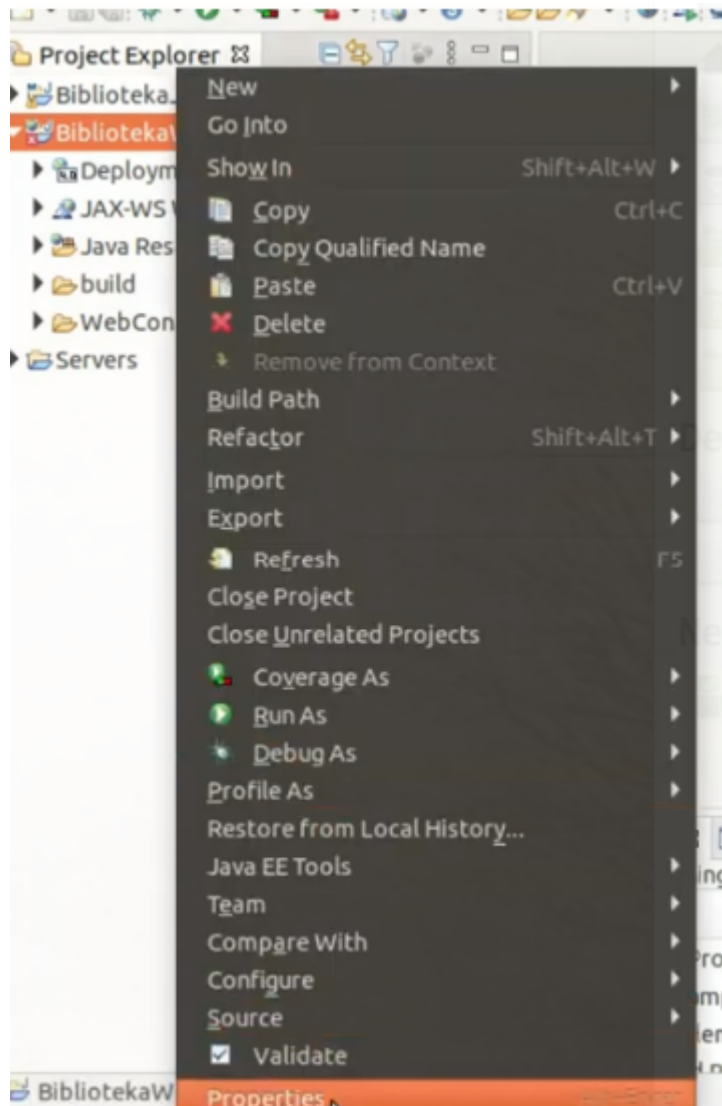
21.



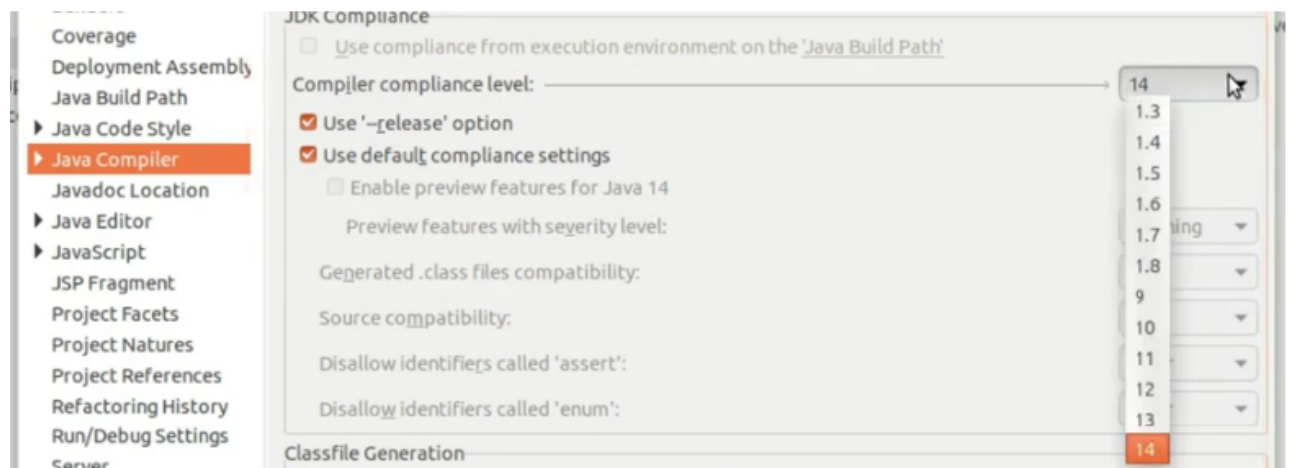
22.



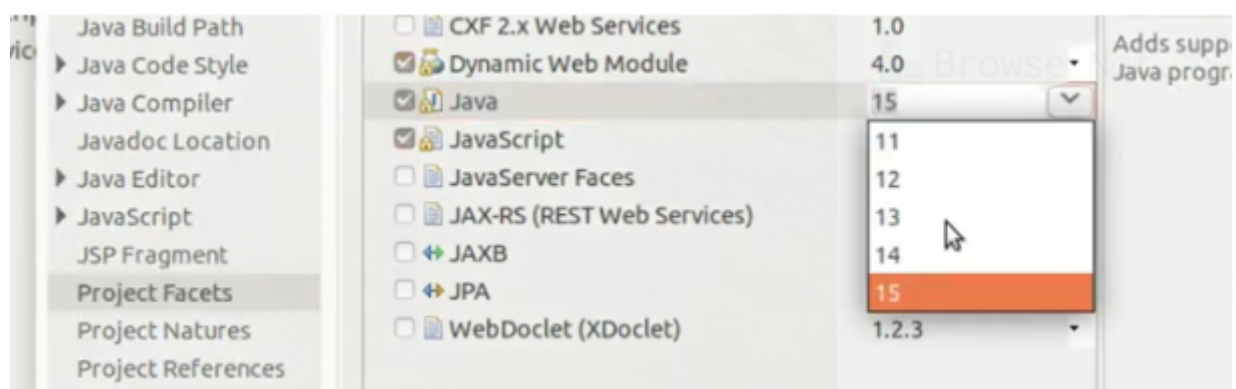
23.



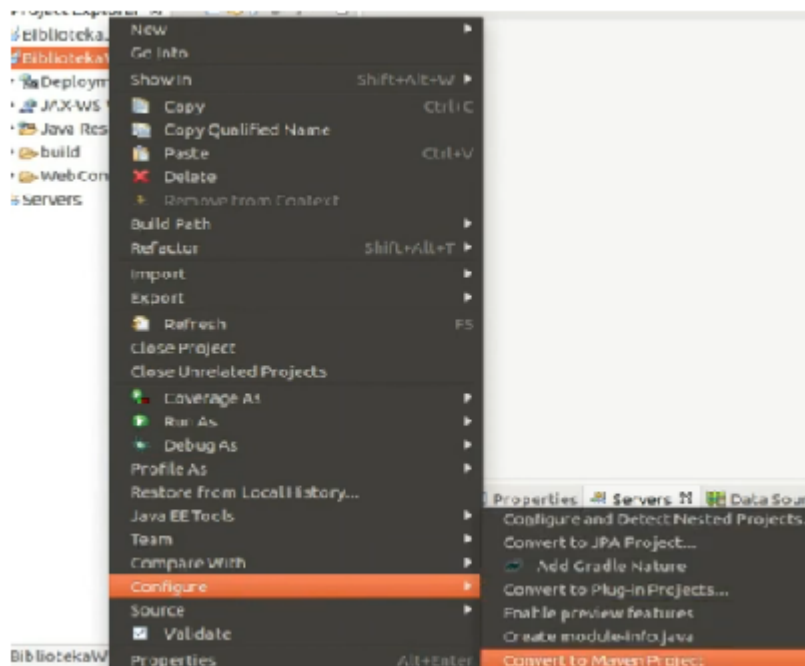
24.



25.



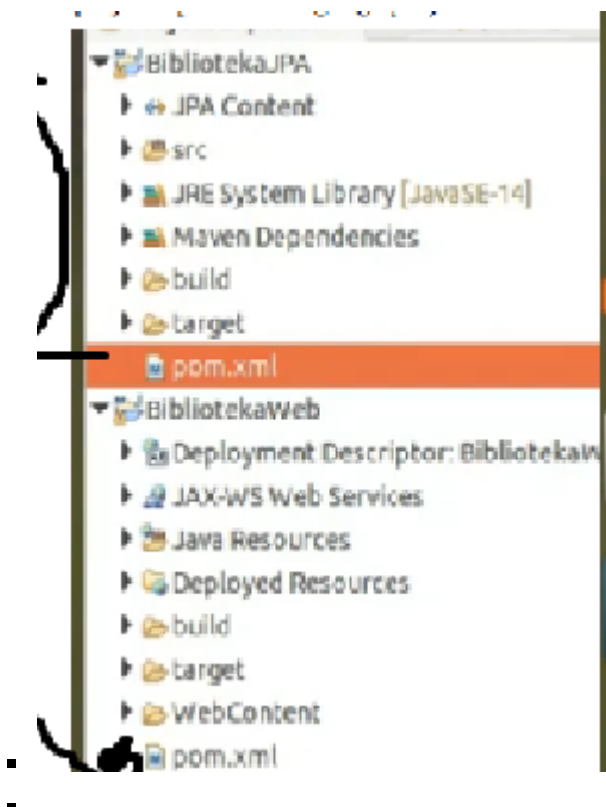
26.



27.

28. dodavanje dependencies-a u pom.xml webappa

- iz BibliotekaJPA naznaka projekta
- u BibliotekaWeb dependency

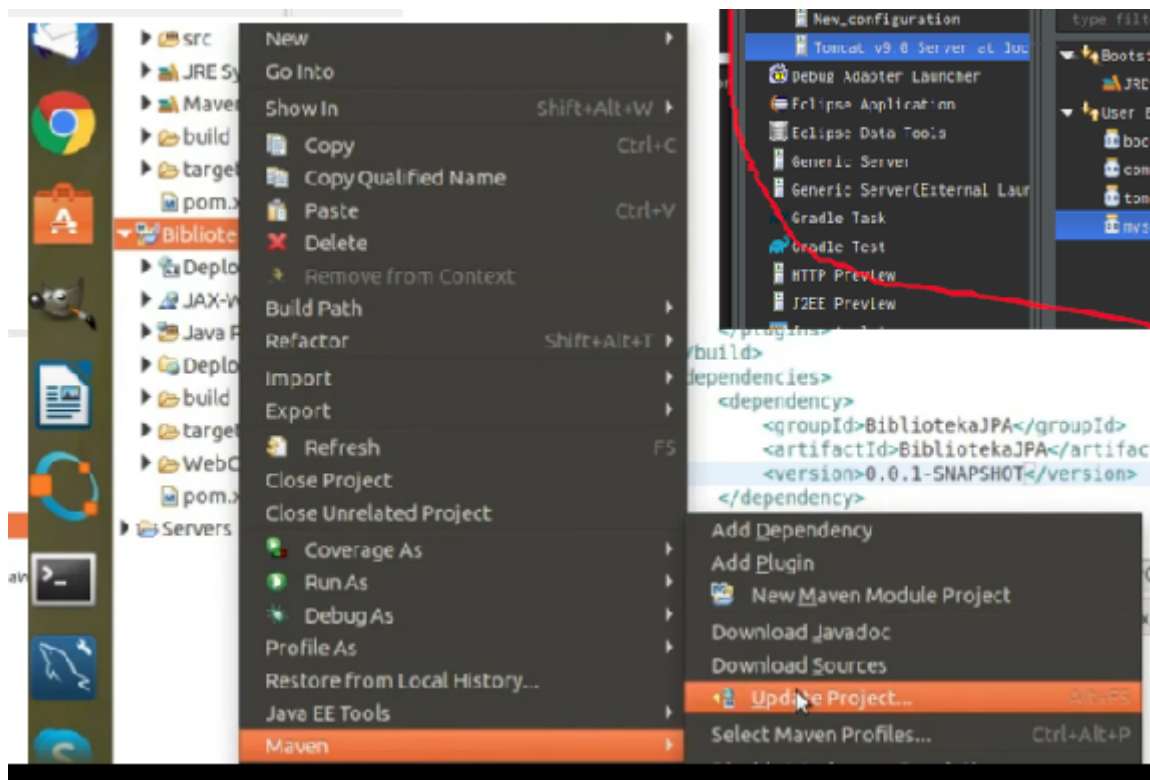


```
<dependencies>
  <dependency>
    <groupId>BibliotekaJPA</groupId>
    <artifactId>BibliotekaJPA</artifactId>
    <version>0.0.1-SNAPSHOT</version>
```

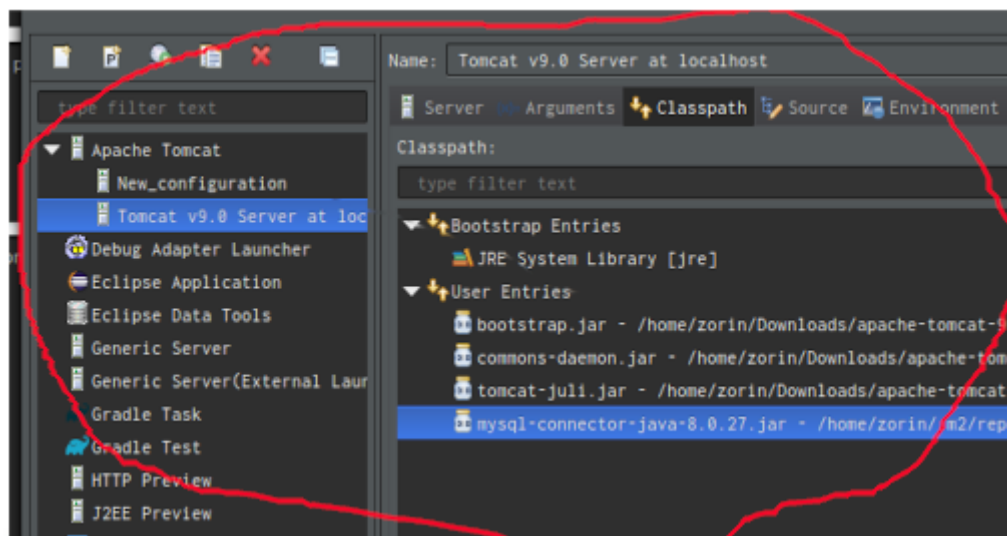
```
</dependency>
</dependencies>
```

29. (moza i ne treba) u BibliotekaWeb pom.xml

```
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>3.1.0</version>
</dependency>
```

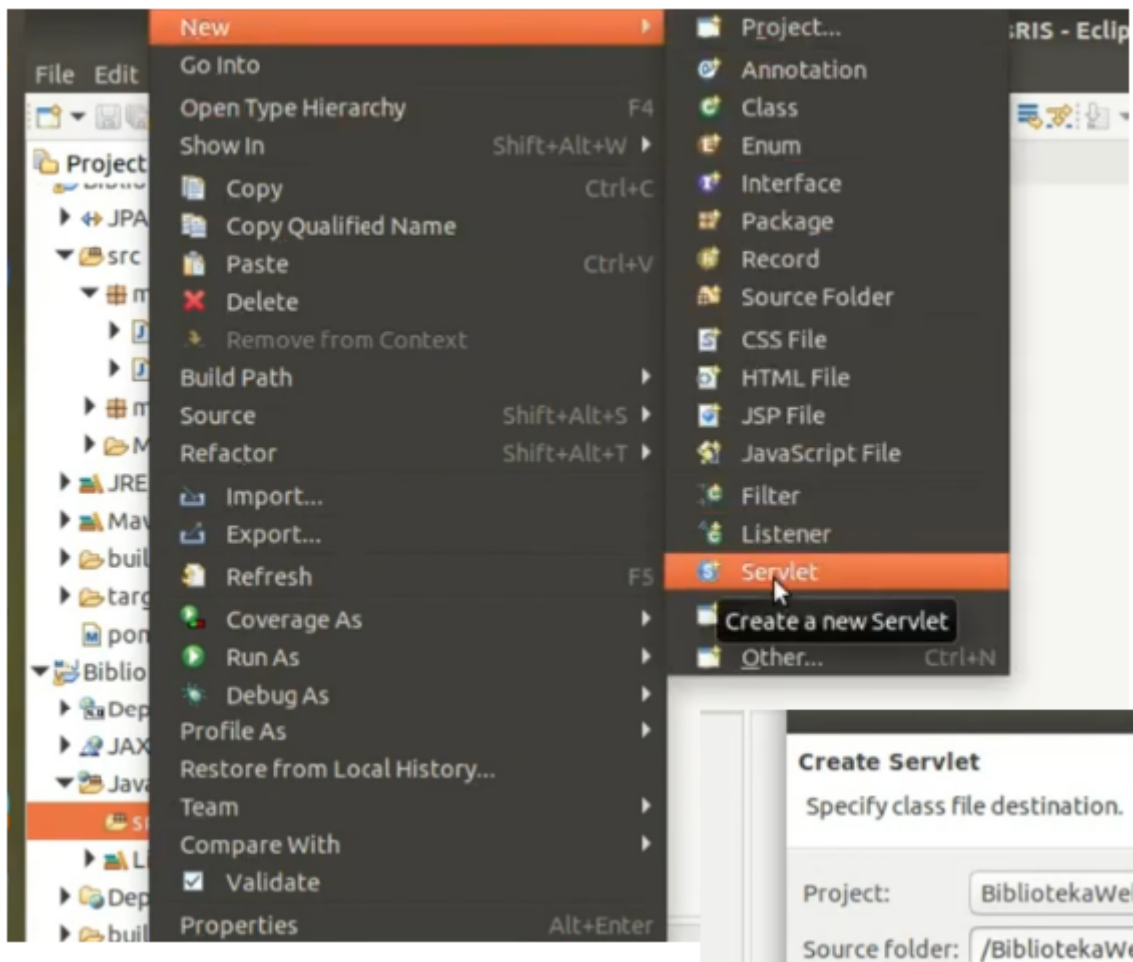


30.

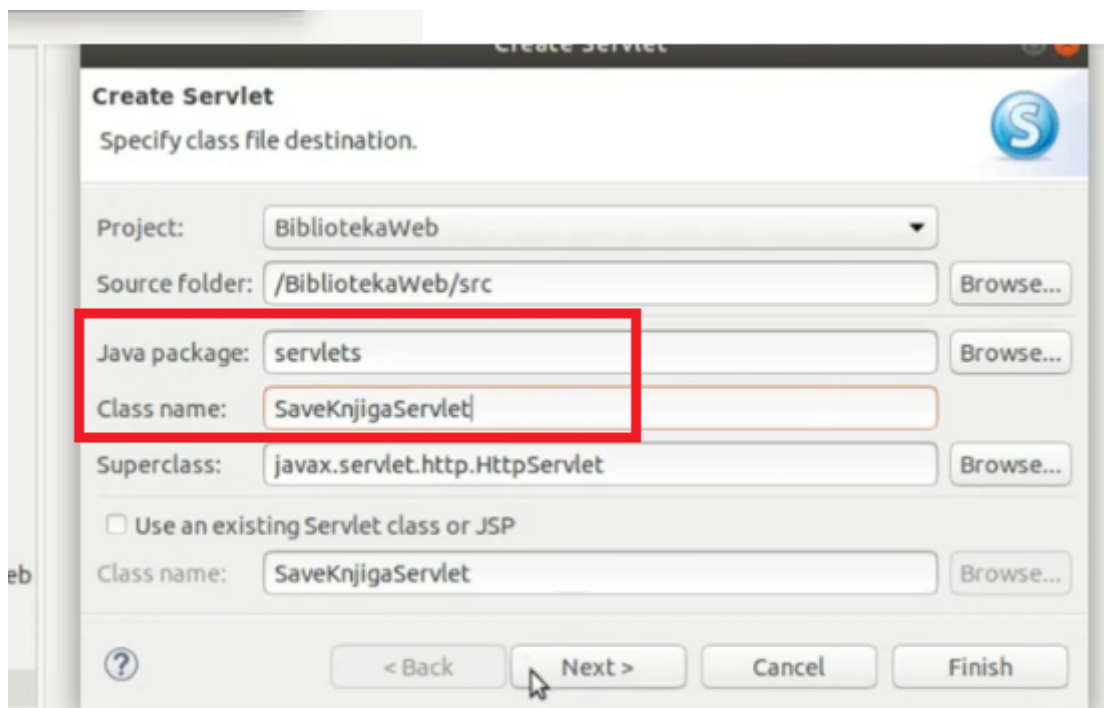


31.



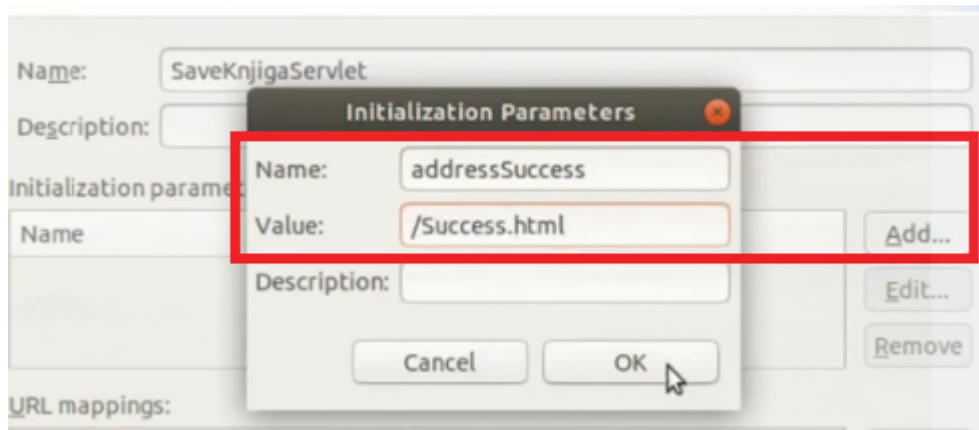


32.

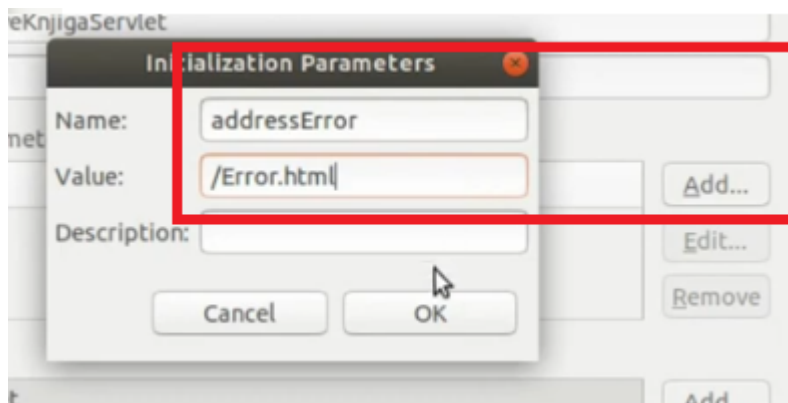


33.

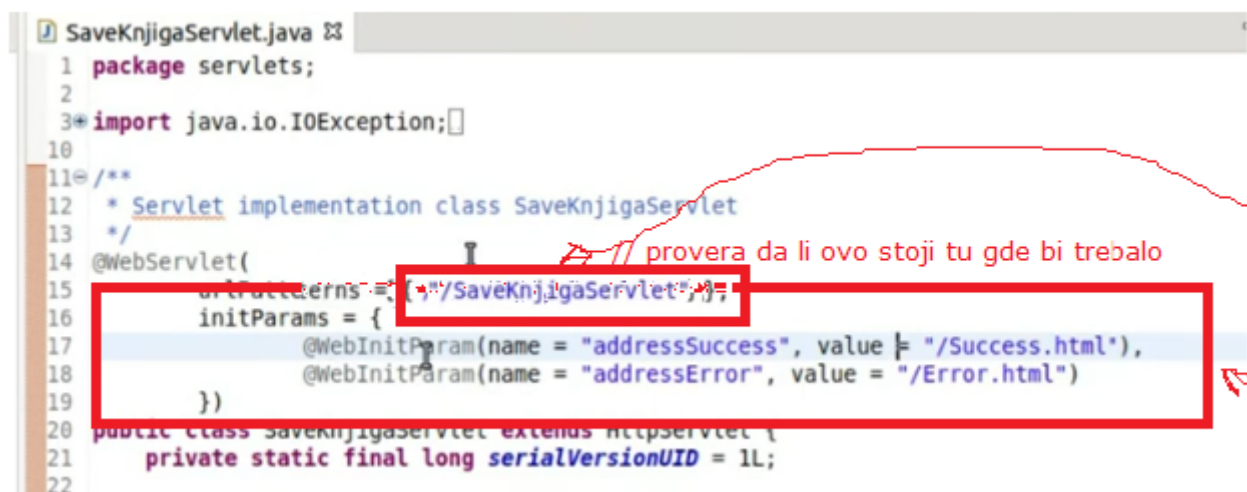




34.



35.



36.

// Ctrl + Shift + o - importovanje svega sto nije bilo importovano

37.

```

38. /**
39.  * @see HttpServlet#doPost(HttpServletRequest request,
40.  *      HttpServletResponse response)
41.  */
42. protected void doPost(HttpServletRequest request, HttpServletResponse
43. response) throws ServletException, IOException {
44.     String naslov = request.getParameter("naslovKnjige");
45.     String autor = request.getParameter("autorKnjige");
46.     String izdavac = request.getParameter("izdavacKnjige");
47.     String god = request.getParameter("godIzdKnjige");
48.
49.
50.
51.
52.
53.
54.
55.
56.
57.
58.
59.
60.
61.
62.
63.
64.
65.
66.
67.
68.
69.
70.
71.
72.
73.
74.
75.
76.
77.
78.
79.
80.
81.
82.
83.
84.
85.
86.
87.
88.
89.
90.
91.
92.
93.
94.
95.
96.
97.
98.
99.
100.

```

```

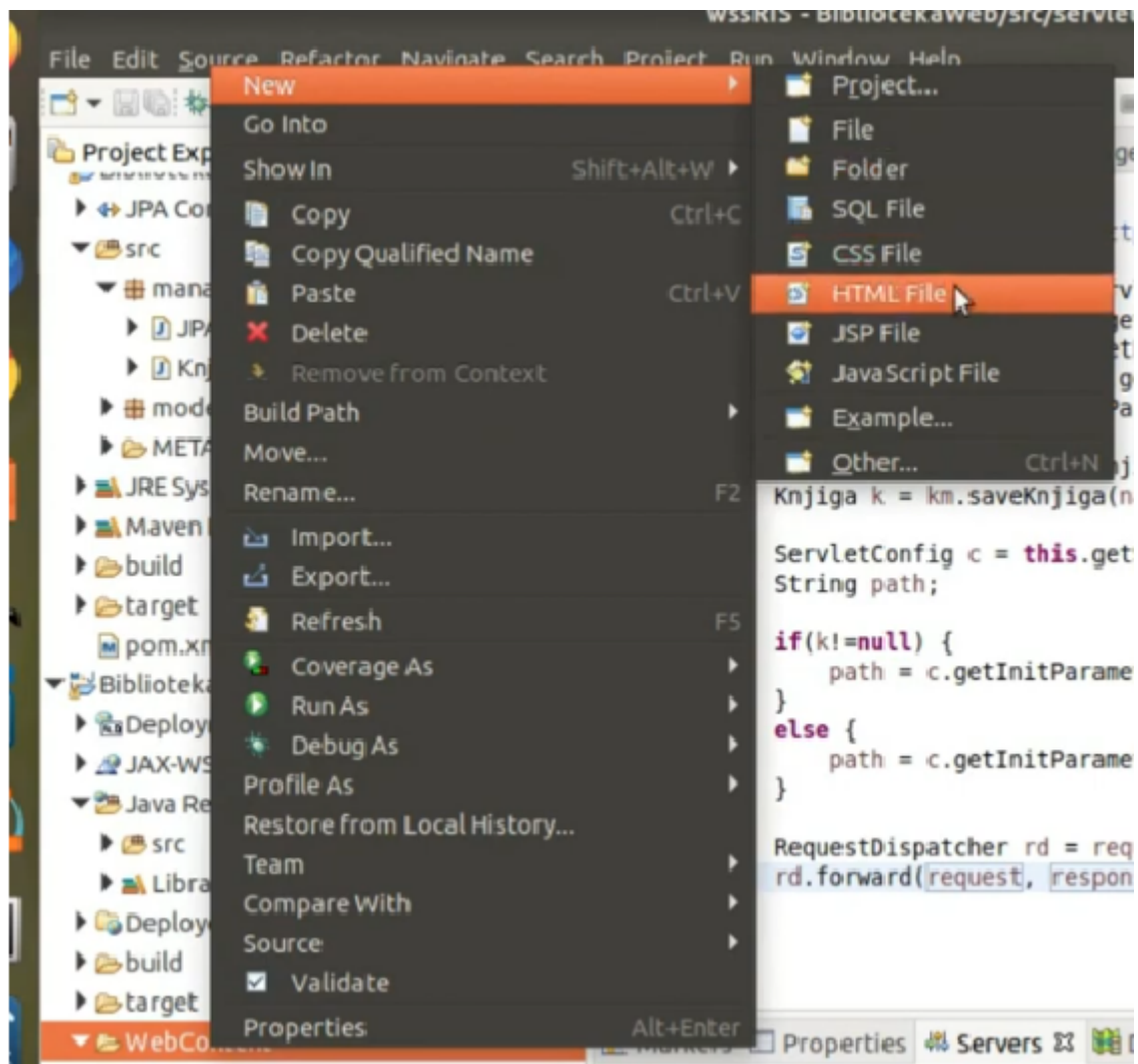
KnjigaManager km = new KnjigaManager();
Knjiga k = km.saveKnjiga(naslov, autor, izdavac, god);

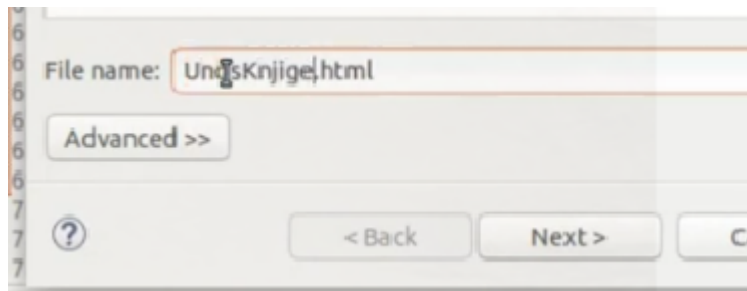
ServletConfig c = this.getServletConfig();
String path;

if(k!=null) {
    path = c.getInitParameter("addressSuccess");
}
else {
    path = c.getInitParameter("addressError");
}

RequestDispatcher rd =
request.getServletContext().getRequestDispatcher(path);
rd.forward(request, response);
}

```





40.

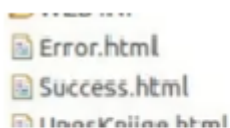
41.

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Unos nove knjige</title>
</head>
<body>
  <form action="/BibliotekaWeb/SaveKnjigaServlet" method="post">
    <table>
      <tr>
        <td>Naslov:</td>
        <td><input type="text" name="naslovKnjige"/></td>
      </tr>
      <tr>
        <td>Autor:</td>
        <td><input type="text" name="autorKnjige"/></td>
      </tr>
      <tr>
        <td>Izdavac:</td>
        <td><input type="text" name="izdovacKnjige"/></td>
      </tr>
      <tr>
        <td>Godina izdanja:</td>
        <td><input type="text" name="godIzdKnjige"/></td>
      </tr>
      <tr>
        <td><input type="submit" value="Sacuvaj"/></td>
      </tr>
    </table>
  </form>

</body>
</html>

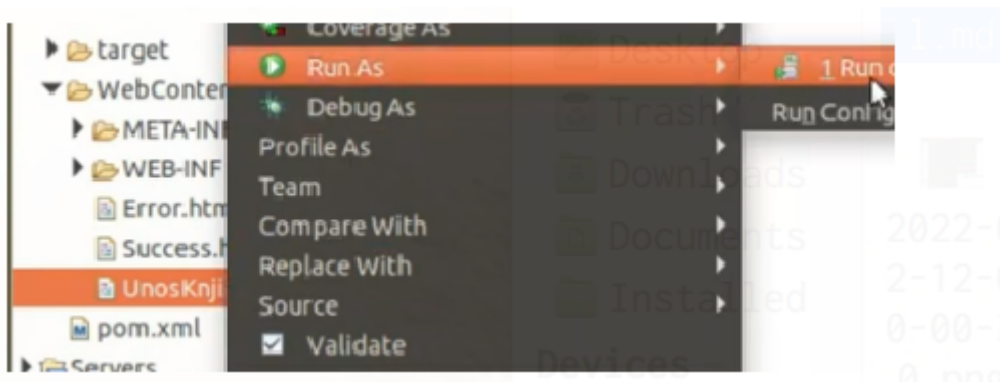
```



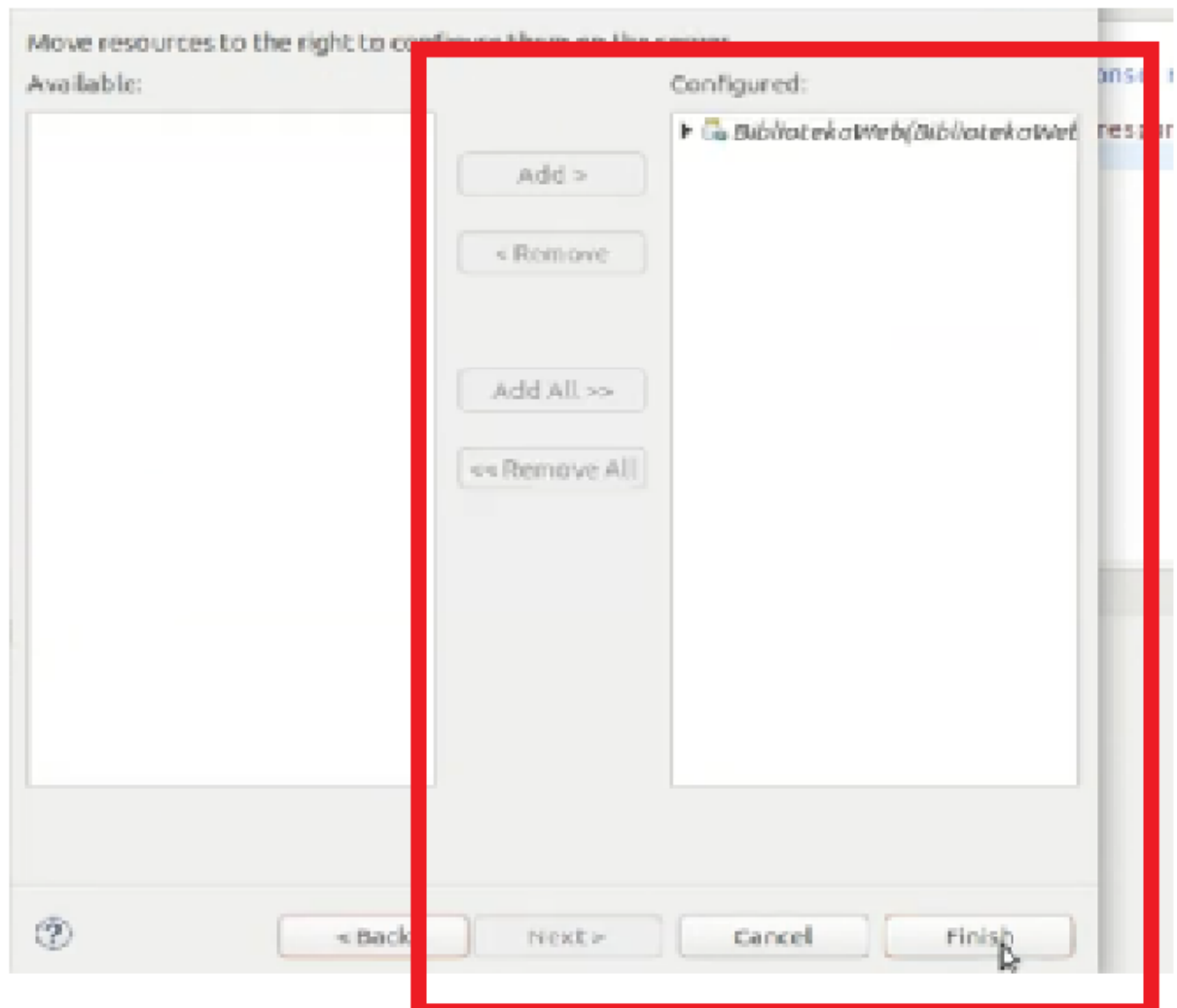
// pravimo proste html stranice za ispisivanje obavestenja

42.

43.



44.



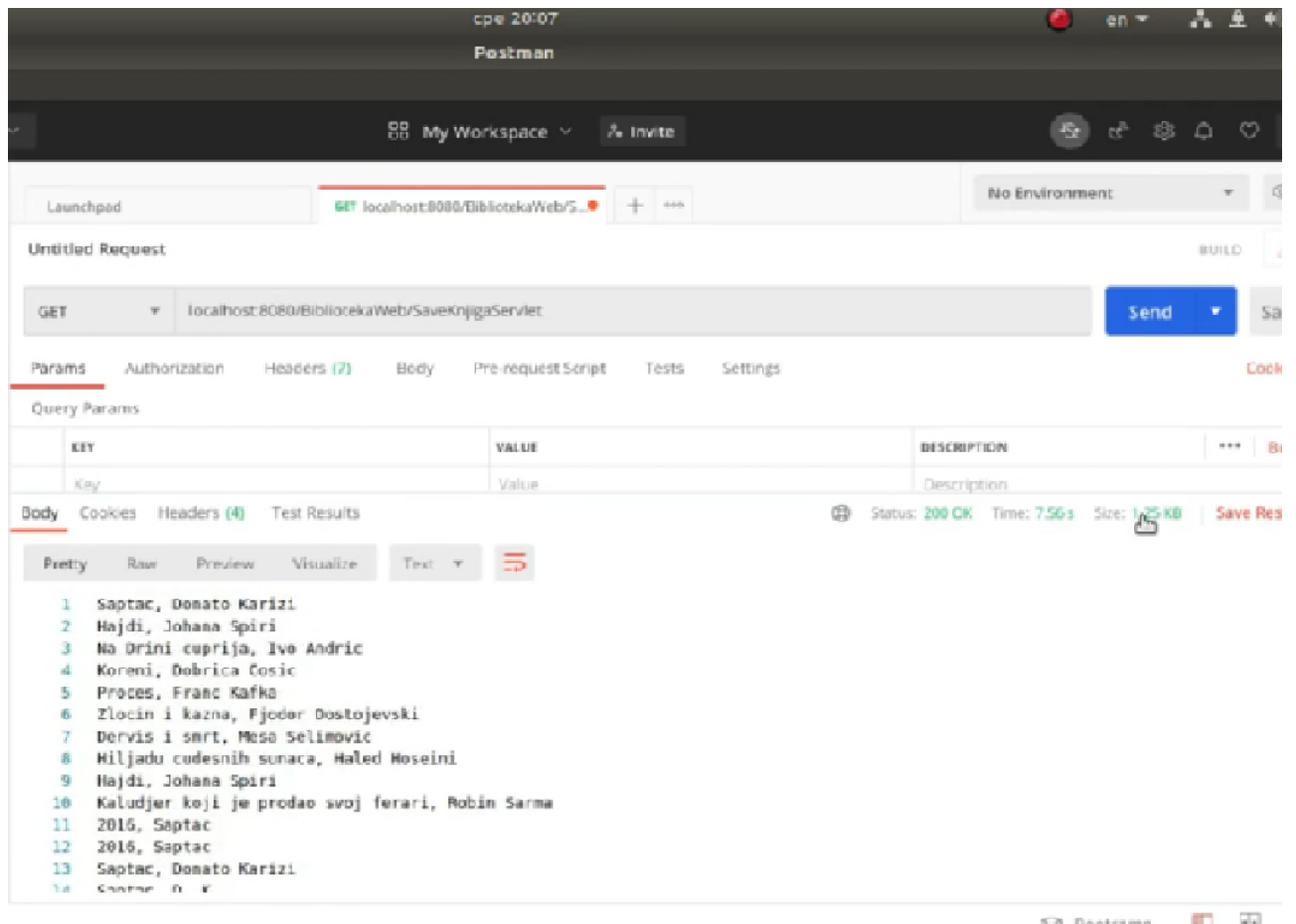
45.

```
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    KnjigaManager km = new KnjigaManager();
    List<Knjiga> knjige = km.getSveKnjige();

    PrintWriter pw = response.getWriter();
    for(Knjiga k:knjige)
        pw.write(k.getNaslov()+", "+k.getAutor()+"\n");
    pw.flush();
    pw.close();
}
```

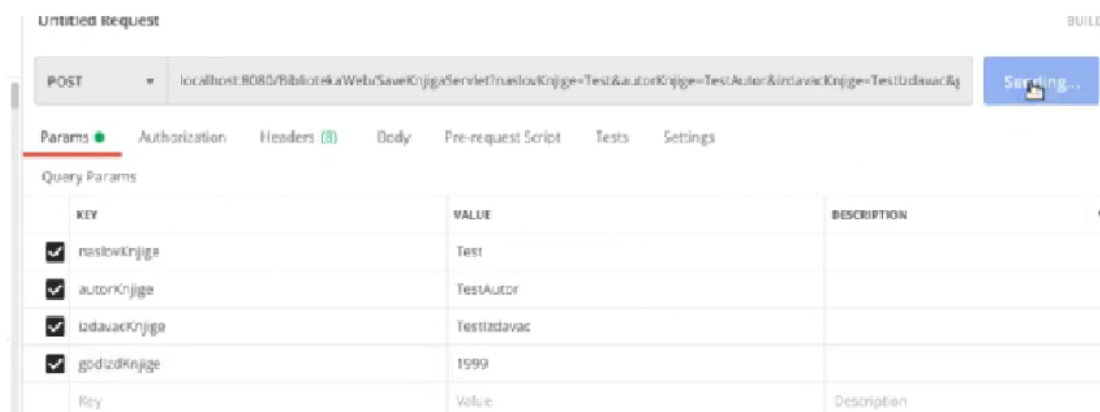
## 46. BibliotekaJPA > managers > KnjigaManager.java

```
public List<Knjiga> getSveKnjige(){
    EntityManager em = JPAUtil.getEntityManager();
    List<Knjiga> knjige = em.createQuery("from Knjiga",
    Knjiga.class).getResultList();
    return knjige;
}
```

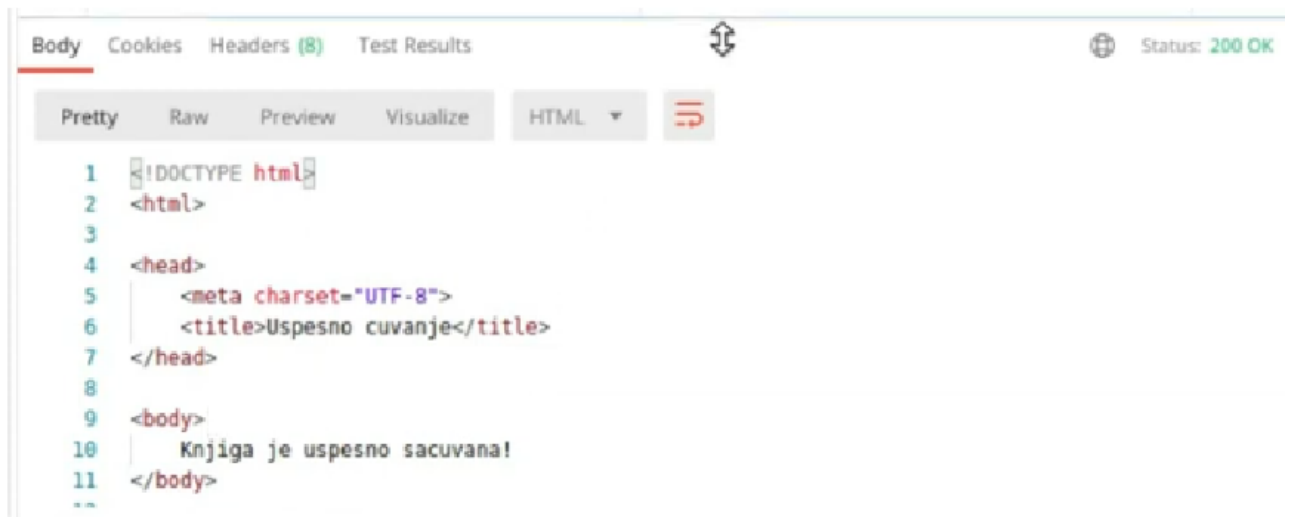


47.

// iz postman-a je moguće slati i post metodu!



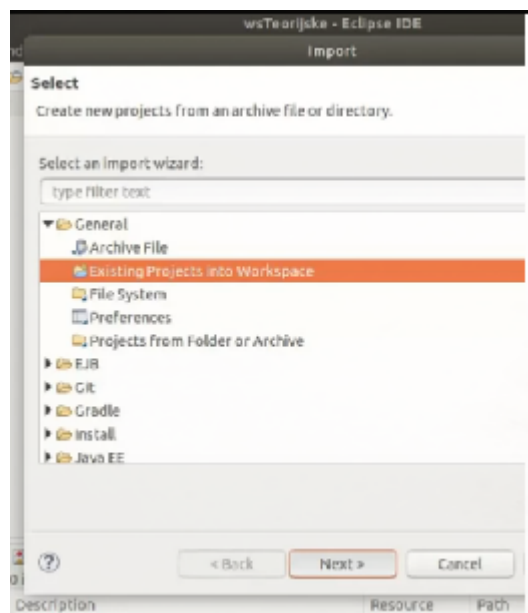
48.



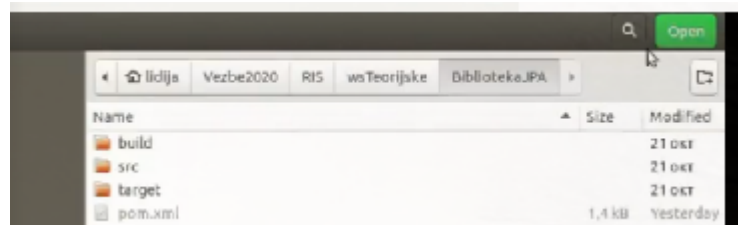
49.

v4 <- v3 <- v2

importovanje projekta

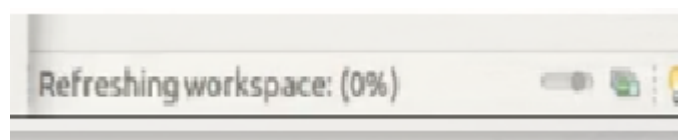


50.



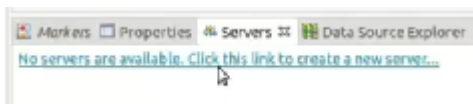
51.

52. ^ // ovo mora I za JPA I za Web

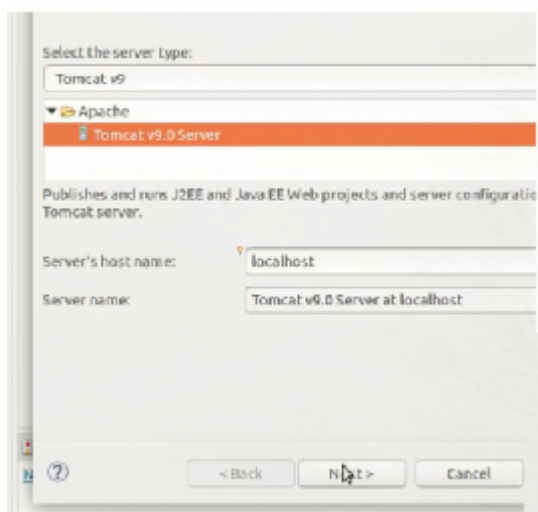


53.

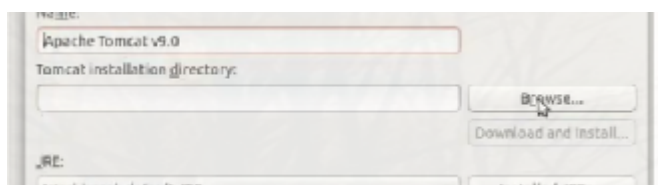
54.



55.



56.



57.



```
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    String naslov = request.getParameter("naslovKnjige");
    String autor = request.getParameter("autorKnjige");
    String izdovac = request.getParameter("izdovacKnjige");
    String god = request.getParameter("godIzdKnjige");

    KnjigaManager km = new KnjigaManager();
    Knjiga k = km.saveKnjiga(naslov, autor, izdovac, god);

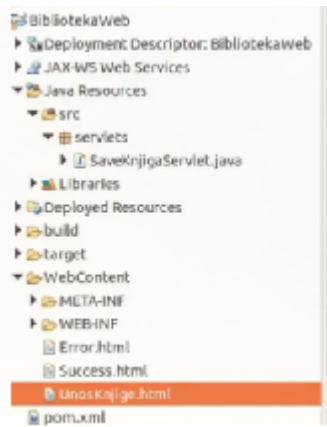
    ServletConfig c = this.getServletConfig();
    String path;

    if(k!=null) {
        path = c.getInitParameter("addressSuccess");
    }
    else {
        path = c.getInitParameter("addressError");
    }

    RequestDispatcher rd =
request.getServletContext().getRequestDispatcher(path);
```

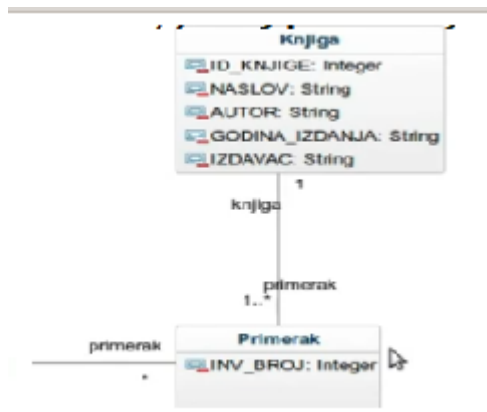


```
rd.forward(request, response);  
}
```



58.

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>Unos nove knjige</title>  
</head>  
<body>  
  <form action="/BibliotekaWeb/SaveKnjigaServlet" method="post">  
    <table>  
      <tr>  
        <td>Naslov:</td>  
        <td><input type="text" name="naslovKnjige"/></td>  
      </tr>  
      <tr>  
        <td>Autor:</td>  
        <td><input type="text" name="autorKnjige"/></td>  
      </tr>  
      <tr>  
        <td>Izdavac:</td>  
        <td><input type="text" name="izdovacKnjige"/></td>  
      </tr>  
      <tr>  
        <td>Godina izdanja:</td>  
        <td><input type="text" name="godIzdKnjige"/></td>  
      </tr>  
      <tr>  
        <td><input type="submit" value="Sacuvaj"/></td>  
      </tr>  
    </table>  
  </form>  
  
</body>  
</html>
```



59.

## 60. SaveKnjigaServlet.java

```

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    String naslov = request.getParameter("naslovKnjiga");
    String autor = request.getParameter("autorKnjiga");
    String izdavac = request.getParameter("izdavacKnjiga");
    String god = request.getParameter("godIzdKnjiga");

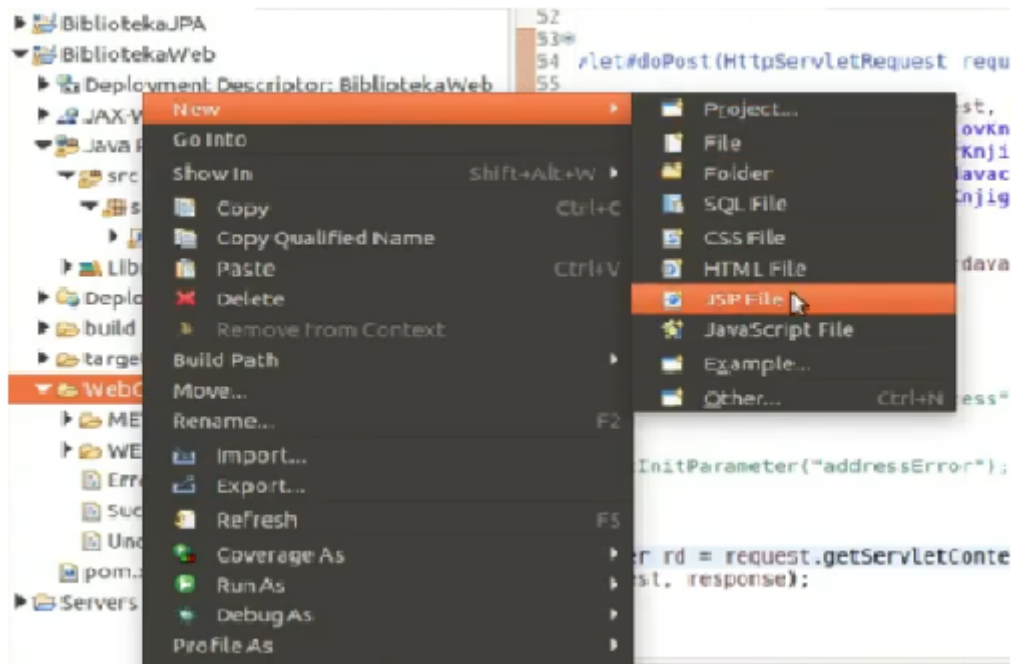
    KnjigaManager km = new KnjigaManager();
    Knjiga k = km.saveKnjiga(naslov, autor, izdavac, god);

    // ServletConfig c = this.getServletConfig();
    // String path;
    //
    // if(k!=null) {
    //     path = c.getInitParameter("addressSuccess");
    // }
    // else {
    //     path = c.getInitParameter("addressError");
    // }

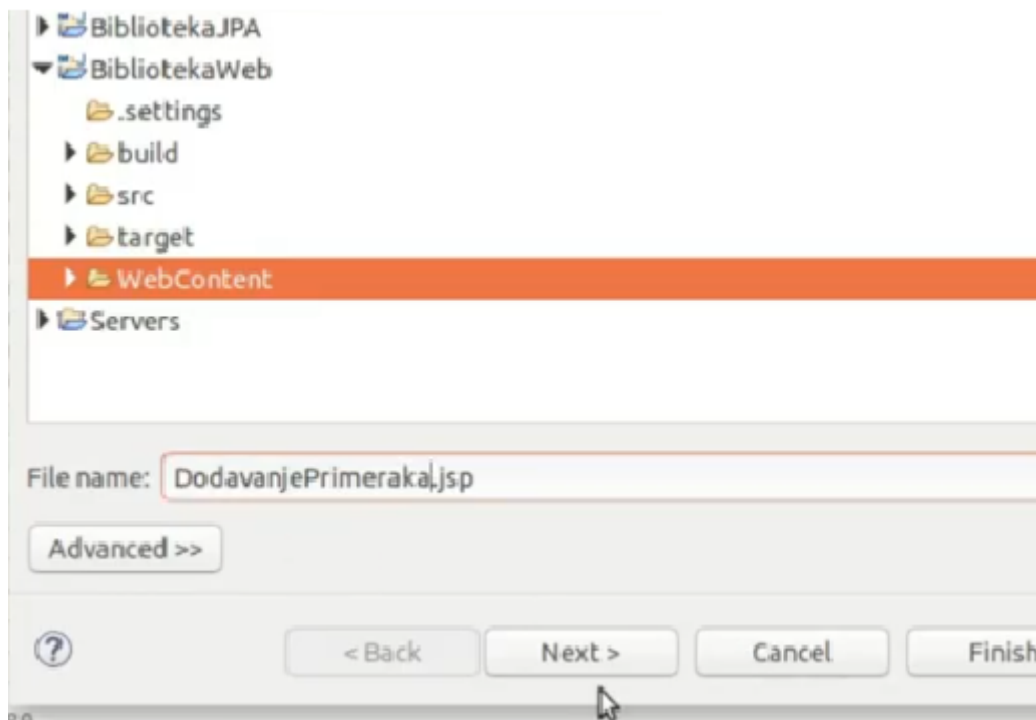
    String poruka;
    if(k!=null)
        poruka = "Knjiga je uspesno sacuvana. Id knjige je
"+k.getIdKnjige();
    else
        poruka = "Doslo je do greske. Knjiga nije sacuvana.";

    request.setAttribute("porukaKnjiga", poruka);
    request.getSession().setAttribute("knjiga", k); // <---- ovde je
namesten session

    RequestDispatcher rd =
request.getServletContext().getRequestDispatcher("/DodavanjePrimeraka.jsp"
);
    rd.forward(request, response);
}
  
```



61.



62.

63. BibliotekaWeb > pom.xml

```
<dependencies>
  <dependency>
    <groupId>BibliotekaJPA</groupId>
    <artifactId>BibliotekaJPA</artifactId>
    <version>0.0.1-SNAPSHOT</version>
  </dependency>
  <dependency>
    <groupId>jstl</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
  </dependency>
</dependencies>
```

```
</dependency>
</dependencies>
```

#### 64. BibliotekaWeb > servlets > DodavanjePrimeraka.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Unos primeraka</title>
</head>
<body>
    ${porukaKnjiga}<br/>
    <c:if test="${!empty knjiga }">
        <form action="/BibliotekaWeb/DodavanjePrimerakaServlet"
method="post">
            Broj primeraka:<input type="text" name="brPrimeraka"/><br/>
            <input type="submit" value="Sacuvaj"/>
        </form>
    </c:if>
    ${porukaPrim}
    <c:if test="${!empty invBrojevi}">
        <c:forEach items="${invBrojevi}" var="br">
            ${br},
        </c:forEach>
    </c:if>
</body>
</html>
```

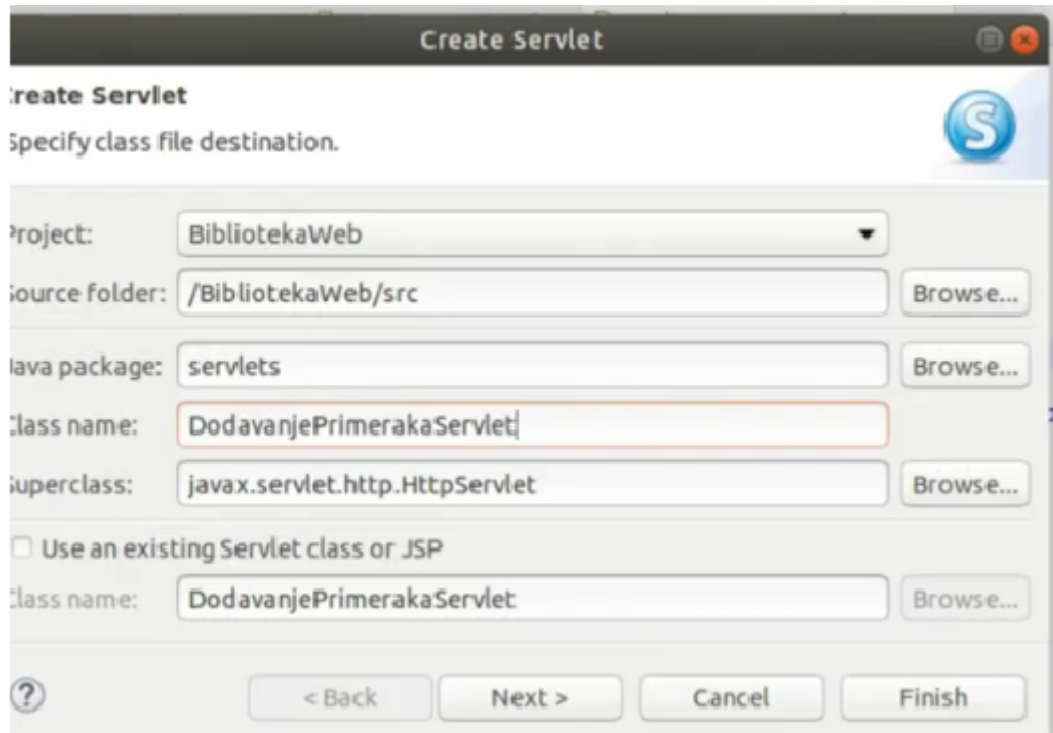
#### 65. KnjigaManager.java

```
public List<Integer> dodajPrimerke(int idKnjige, int brojPrimeraka){
    ArrayList<Integer> invBrojevi = new ArrayList<Integer>();
    try {
        EntityManager em = JPAUtil.getEntityManager();
        Knjiga k = em.find(Knjiga.class, idKnjige);
        if(k!=null) {
            em.getTransaction().begin();
            for(int i=0;i<brojPrimeraka;i++) {
                Primerak p = new Primerak();
                p.setKnjiga(k);
                em.persist(p);
                invBrojevi.add(p.getInvBroj());
            }
            em.getTransaction().commit();
        }
    }
}
```

```

    }
    return invBrojevi;
}
catch(Exception e) {
    e.printStackTrace();
    return null;
}
}

```



66.

- DodavanjePrimerakaServlet.java

```

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    KnjigaManager km = new KnjigaManager();
    int brPrim = Integer.parseInt(request.getParameter("brPrimeraka"));
    Knjiga k = (Knjiga)request.getSession().getAttribute("knjiga");

    List<Integer> invBrojevi = km.dodajPrimerke(k.getIdKnjige(), brPrim);

    String porukaPrimerci;
    if(invBrojevi!=null && invBrojevi.size()>0)
        porukaPrimerci="Primerci su uspesno dodati. Generisani su sledeci
inventarni brojevi: ";
    else
        porukaPrimerci="Doslo je do greske. Primerci nisu sacuvani.";

    request.setAttribute("porukaPrim", porukaPrimerci);
    request.setAttribute("invBrojevi", invBrojevi);

    RequestDispatcher rd =

```

```

request.getServletContext().getRequestDispatcher("/DodavanjePrimeraka.jsp"
);
    rd.forward(request, response);

}

```

## 67. DodavanjePrimeraka.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Unos primeraka</title>
</head>
<body>
    ${porukaKnjiga}<br/>
    <c:if test="${!empty knjiga }">
        <form action="/BibliotekaWeb/DodavanjePrimerakaServlet"
method="post">
            Broj primeraka:<input type="text" name="brPrimeraka"/><br/>
            <input type="submit" value="Sacuvaj"/>
        </form>
    </c:if>
    ${porukaPrim}
    <c:if test="${!empty invBrojevi}">
        <c:forEach items="${invBrojevi}" var="br">
            ${br},
        </c:forEach>
    </c:if>
</body>
</html>

```

## combox

## 68. ClanManager.java

```

public Clan saveClan(String ime, String prezime, String adresa,
    Date datumRodjenja, Date datumUpisa, int idKategorije) {
    try {
        EntityManager em = JPAUtil.getEntityManager();
        em.getTransaction().begin();

        Kategorija k = em.find(Kategorija.class, idKategorije);
        Clan c = new Clan();
        c.setIme(ime);
    }
}

```

```

        c.setPrezime(prezime);
        c.setAdresa(adresa);
        c.setDatumRodjenja(datumRodjenja);
        c.setDatumUpisa(datumUpisa);
        c.setKategorija(k);
        em.persist(c);

        em.getTransaction().commit();
        return c;
    }
    catch(Exception e) {
        e.printStackTrace();
        return null;
    }
}

```

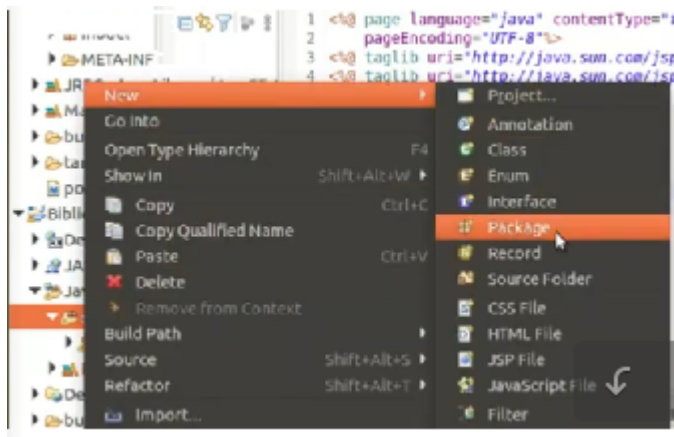
## 69. ClanManager.java

```

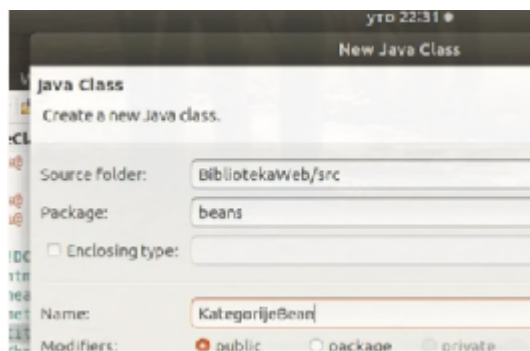
public List<Kategorija> getSveKategorije() {
    EntityManager em = JPAUtil.getEntityManager();
    List<Kategorija> kat = em.createQuery("from Kategorija k",
    Kategorija.class).getResultList();
    return kat;
}

```

70.



71.



- KategorijeBean.java



```

package beans;

import java.util.List;

import managers.ClanManager;
import model.Kategorija;

public class KategorijeBean {
    private List<Kategorija> k;

    public KategorijeBean() {
        ClanManager cm = new ClanManager();
        k = cm.getSveKategorije();
    }

    public List<Kategorija> getK() {
        return k;
    }

    public void setK(List<Kategorija> k) {
        this.k = k;
    }

}

```

## 72. SaveClan.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page import="java.util.Date"%>

...

<c:set var="today" value="<%=new Date() %>"/>
<jsp:useBean id="katBean" class="beans.KategorijeBean" scope="session">
</jsp:useBean>

    <form action="/BibliotekaWeb/SaveClanServlet" method="post">
        <table>
            <tr><td>Kategorija:</td>
                <td><select name="idKat">
                    <c:forEach items="${katBean.k}" var="kat">
                        <option
value="${kat.idkategorije}">${kat.nazivkategorije}</option>
                    </c:forEach>

```

```

        </select>
        </td>
    </tr>
    <tr>
        <td>Ime:</td>
        <td><input type="text" name="ime"/></td>
    </tr>
    <tr>
        <td>Prezimeme:</td>
        <td><input type="text" name="prezime"/></td>
    </tr>
    <tr>
        <td>Adresa:</td>
        <td><input type="text" name="adresa"/></td>
    </tr>
    <tr>
        <td>Datum rodjenja:</td>
        <td><input type="date" name="datumRodjenja"/>(yyyy-MM-dd)
    </td>
    </tr>
    <tr>
        <td>Datum upisa:</td>
        <td><input type="date" name="datumUpisa"
value='<fmt:formatDate pattern="yyyy-MM-dd"
        value="{today}"/>' />(yyyy-MM-dd)</td>
    </tr>
    <tr><td><input type="submit" value="Sacuvaj"/></td></tr>
</table>

</form>
${poruka}
<c:if test="${!empty clan }">${clan.clanskibroj }</c:if>

</body>
</html>

```

### 73. SaveClanServlet.java

```

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    String ime = request.getParameter("ime");
    String prezime = request.getParameter("prezime");
    String adresa = request.getParameter("adresa");
    Integer idK = Integer.parseInt(request.getParameter("idKat"));

    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    Date datumRodjenja=null, datumUpisa=null;

```

```

try {
    datumRodjenja=sdf.parse(request.getParameter("datumRodjenja"));
    datumUpisa=sdf.parse(request.getParameter("datumUpisa"));
} catch (ParseException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

ClanManager cm= new ClanManager();
Clan c = cm.saveClan(ime, prezime, adresa, datumRodjenja, datumUpisa,
idK);

String poruka;
if(c!=null)
    poruka="Clan je uspesno sacuvan. Id clana je ";
else
    poruka="Doslo je do greske. Clan nije sacuvan.";

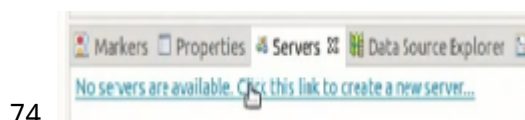
request.setAttribute("poruka", poruka);
request.setAttribute("clan", c);

RequestDispatcher rd =
request.getRequestDispatcher("/SaveClan.jsp");
rd.forward(request, response);
}

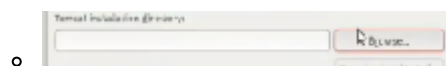
```

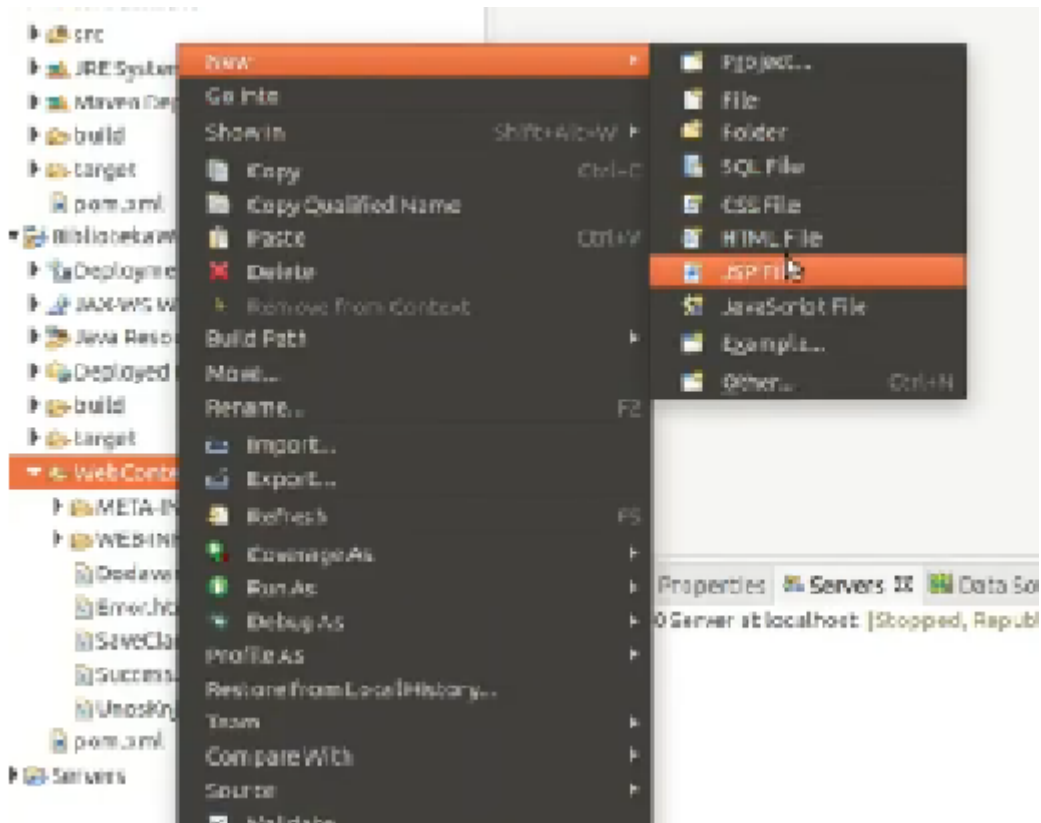
## 5. vezbe

---



76. ubaciti skinut apache tomcat 9



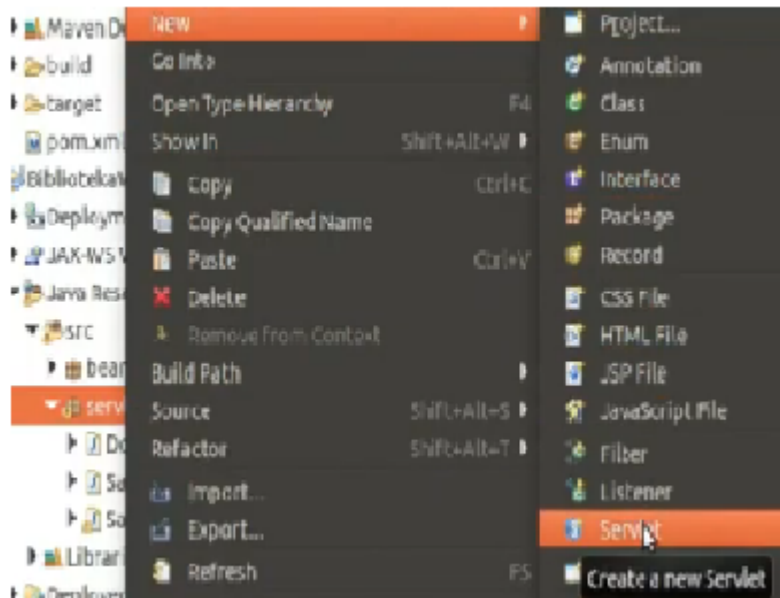


77.

78. ubacivanje u UnosZaduzenja.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Unos novog zaduzenja</title>
</head>
<body>
    <form action="/BibliotekaWeb/UnosZaduzenjaServlet" method="post">
        Clanski broj: <input type="text" name="clanskiBroj"/><br/>
        Inventarni broj: <input type="text" name="invBroj"/><br/>
        <input type="submit" value="Sacuvaj"/>
    </form>
</body>
</html>
```



79.

## 80. UnosZaduzenjaServlet.java

```
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    Integer clBr =
Integer.parseInt(request.getParameter("clanskiBroj"));
    Integer invBr = Integer.parseInt(request.getParameter("invBroj"));

    ClanManager cm = new ClanManager();
    Zaduzenje z = cm.saveZaduzenje(clBr, invBr);

    String poruka;
    if(z!=null)
        poruka="Zaduzenje je uspesno uneto.";
    else
        poruka="Doslo je do greske. Zaduzenje nije sacuvano.";

    request.setAttribute("poruka", poruka);
    request.getSession().setAttribute("zaduzenje", z);

    RequestDispatcher rd =
request.getRequestDispatcher("/PrikazZaduzenja.jsp");
    rd.forward(request, response);
}
```

- ima i varijanta gde je odmah na UnosZaduzenja.jsp ispisano,

```
RequestDispatcher rd = request.getRequestDispatcher("/UnosZaduzenja.jsp");
```

**Ispis na istoj  
stranici  
---- ili ----  
ispis na nekoj  
drugoj**

```
RequestDispatcher rd = request.getRequestDispatcher("/prikazZaduzenja.jsp");
rd.forward(request, response);
```

**u tom slučaju**

- ima i varijanta gde je prvo radila bez

```
request.setAttribute("poruka", poruka);  
request.setAttribute("zaduzenje", z);
```

## 81. BibliotekaJPA > managers > ClanManager.java

```
public Zaduzenje saveZaduzenje(int clBroj, int invBroj) {  
    try {  
        EntityManager em = JPAUtil.getEntityManager();  
        em.getTransaction().begin();  
  
        Clan c = em.find(Clan.class, clBroj);  
        Primerak p = em.find(Primerak.class, invBroj);  
  
        Zaduzenje z = new Zaduzenje();  
        z.setClan(c);  
        z.setPrimerak(p);  
        z.setDatumZaduzenja(new Date());  
  
        em.persist(z);  
  
        em.getTransaction().commit();  
        return z;  
    }  
    catch(Exception e) {  
        e.printStackTrace();  
        return null;  
    }  
}
```

## 82. PrikazZaduzenja.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"  
pageEncoding="UTF-8"%>  
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>  
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>  
  
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>Prikaz zaduzenja</title>  
</head>  
<body>  
    ${poruka}  
    <c:if test="${!empty zaduzenje}">  
        Zaduzen je clan ${zaduzenje.clan.ime } ${zaduzenje.clan.prezime },  
        ${zaduzenje.clan.clanskibroj }.<br/>  
        Zaduzen je primerak knjige ${zaduzenje.primerak.knjiga.naslov },  
        ${zaduzenje.primerak.knjiga.autor}, inventarni broj:  
        ${zaduzenje.primerak.invBroj } <br/>
```

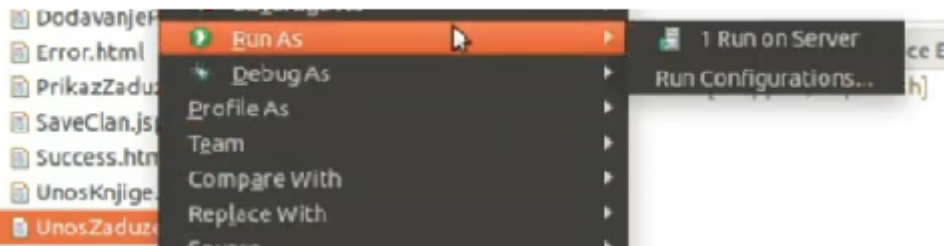
```

        <br/>
        <a href="/BibliotekaWeb/VratiZaduzenjaServlet?
clanskiBroj=${zaduzenje.clan.clanskibroj}">Prikaz svih zaduzenja</a>
    </c:if>

</body>
</html>

```

možemo  
pokrenuti



83.

84. VratiZaduzenjaServlet.java

```

protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    Integer clBr = Integer.parseInt(request.getParameter("clanskiBroj"));

    ClanManager cm = new ClanManager();
    List<Zaduzenje> z = cm.getSvaZaduzenja(clBr);

    request.setAttribute("zaduzenja", z);
    request.getSession().setAttribute("clanskiBroj", clBr);
    RequestDispatcher rd =
request.getServletContext().getRequestDispatcher("/PrikazZaduzenja.jsp");
    rd.forward(request, response);
}

```

85. BibliotekaJPA > managers > ClanManager.java

```

public List<Zaduzenje> getSvaZaduzenja(int clBr){
    EntityManager em = JPAUtil.getEntityManager();
    List<Zaduzenje> zad = em.createQuery("select z from Zaduzenje z where
z.clan.clanskibroj= :clBroj", Zaduzenje.class)
        .setParameter("clBroj", clBr).getResultList();
    return zad;
}

```

86. PrikazZaduzenja.jsp dopuna neka nakon vec implementiranog koda

```

<c:if test="${!empty zaduzenja}">
    <table border="1">

```



```

        <tr>
            <th>Naslov</th>
            <th>Autor</th>
            <th>Izdavac</th>
            <th>Godina izdanja</th>
            <th>Inventarni broj</th>
            <th>Datum zaduzenja</th>
            <th>Razduzenje</th>
        </tr>
        <c:forEach items="${zaduzenja}" var="z">
            <tr>
                <td>${z.primerak.knjiga.naslov }</td>
                <td>${z.primerak.knjiga.autor }</td>
                <td>${z.primerak.knjiga.izdavac }</td>
                <td>${z.primerak.knjiga.godinaIzdanja }</td>
                <td>${z.primerak.invBroj }</td>
                <td>${z.datumZaduzenja }</td>
                <td>
                    <c:if test="${empty z.datumVracanja }">
                        <a href="/BibliotekaWeb/RazduziServlet?
id=${z.id}">Razduzi</a>
                    </c:if>
                    <c:if test="${!empty z.datumVracanja }">
                        ${z.datumVracanja}
                    </c:if>
                </td>
            </tr>
        </c:forEach>
    </table>
</c:if>

```

## 87. RazduziServlet.java

```

protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    Integer id = Integer.parseInt(request.getParameter("id"));
    ClanManager cm = new ClanManager();

    boolean ok = cm.razduzi(id);
    if(ok) {
        Integer clBr =
(Integer)request.getSession().getAttribute("clanskiBroj");
        List<Zaduzenje> zad = cm.getSvaZaduzenja(clBr);
        request.setAttribute("zaduzenja", zad);
    }
    RequestDispatcher rd =
request.getServletContext().getRequestDispatcher("/PrikazZaduzenja.jsp");
    rd.forward(request, response);
}

```

## 88. BibliotekaJPA > ClanManager.java

```
public boolean razduzi(int id) {
    try {
        EntityManager em = JPAUtil.getEntityManager();
        em.getTransaction().begin();

        Zaduzenje z = em.find(Zaduzenje.class, id);
        z.setDatumVracanja(new Date());
        em.merge(z);

        em.getTransaction().commit();
        return true;
    }
    catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}
```

## 89. VratiZaduzenjaServlet.java

```
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    Integer clBr = Integer.parseInt(request.getParameter("clanskiBroj"));

    ClanManager cm = new ClanManager();
    List<Zaduzenje> z = cm.getSvaZaduzenja(clBr);

    request.setAttribute("zaduzenja", z);
    request.getSession().setAttribute("clanskiBroj", clBr);
    RequestDispatcher rd =
request.getServletContext().getRequestDispatcher("/PrikazZaduzenja.jsp");
    rd.forward(request, response);
}
```

**zelim da prikaz  
zaduzenja izgleda  
ovako:**

Zaduzen je clan Pera Peric, 9.  
Zaduzen je primerak knjige Hajdi, Johana Spiri, inventarni broj: 4  
[Prikaz svih zaduzenja](#)

90.

*// ali ovaj deo nakon otvaranja zaduzenja nestane  
// razlog je taj sto se klikom na link prikaza se pravi novi  
zahtev i parametar zaduzenje se anulira*

91.

*// ovo necemo da prilagodimo, jer nam je  
njegova svrha kratkotrajna*

92.

*// expression  
language  
implicitno poziva  
parametar  
zaduzenje iz  
sesije*

o

### 93. UnosZaduzenjaServlet.java

```
request.getSession().setAttribute("zaduzenje", z);
```

### 94. PrikazZaduzenja.jsp

```
...  
<body>  
    ${poruka}  
    <c:if test="${!empty zaduzenje}">  
...  

```

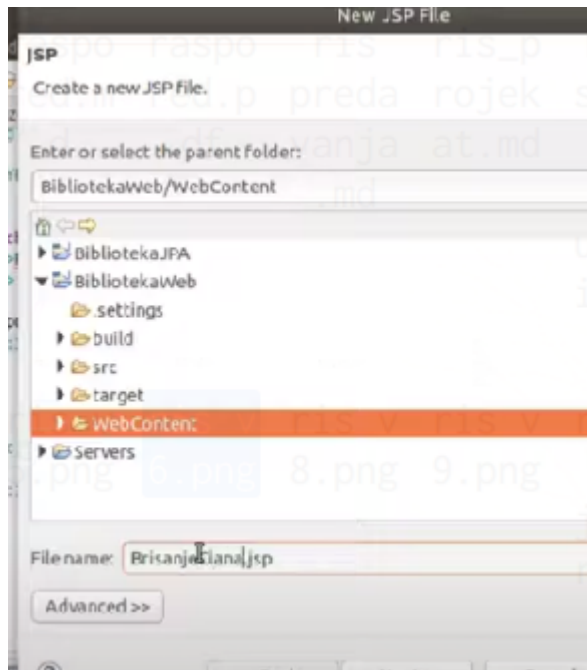
*// resenje je da  
za parametar  
zaduzenje  
angazujemo  
session*

*// expression  
language  
implicitno poziva  
parametar  
zaduzenje iz  
sesije*

95.

v6

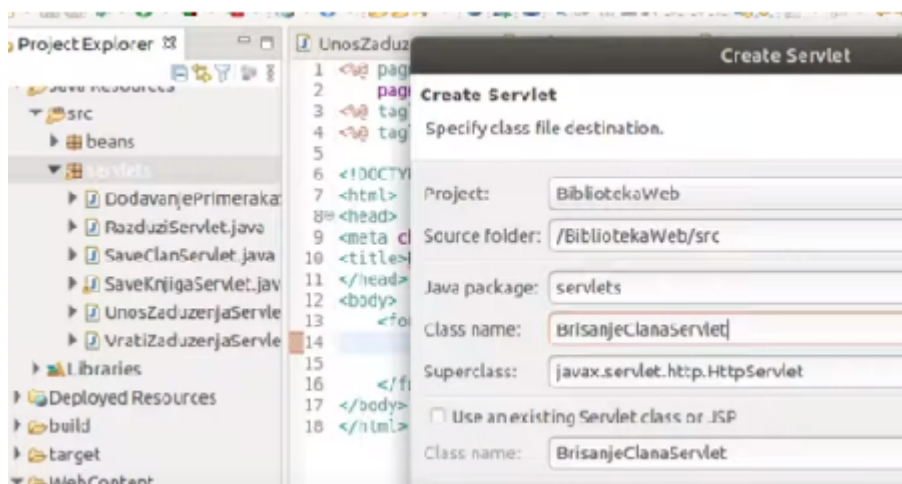
---



96.

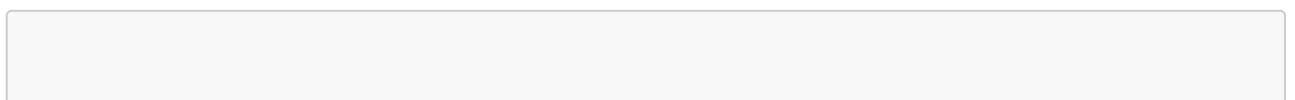
97. BrisanjeClana.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
...
<body>
    <form action="/BibliotekaWeb/BrisanjeClanaServlet" method="post">
        Clanski broj: <input type="text" name="clanskiBroj">
        <br/>
        <input type="submit" value="Obrisi">
    </form>
    ${msg}
</body>
...
```



98.

99. BrisanjeClanaServlet.java



```

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    Integer clBr = Integer.parseInt(request.getParameter("clanskiBroj"));
    ClanManager cm = new ClanManager();
    boolean ok = cm.deleteClan(clBr);

    if(ok)
        request.setAttribute("msg", "Clan i njegova zaduzenja su
obrisani.");
    else
        request.setAttribute("msg", "Doslo je do greske, brisanje nije
uspelo.");
    RequestDispatcher rd =
request.getServletContext().getRequestDispatcher("/BrisanjeClana.jsp");
    rd.forward(request, response);

}

```

#### 100. BibliotekaJPA > managers > ClanManager.java

```

public boolean deleteClan(Integer clBroj) {
    try {
        EntityManager em=JPAUtil.getEntityManager();
        em.getTransaction().begin();

        Clan c = em.find(Clan.class, clBroj);

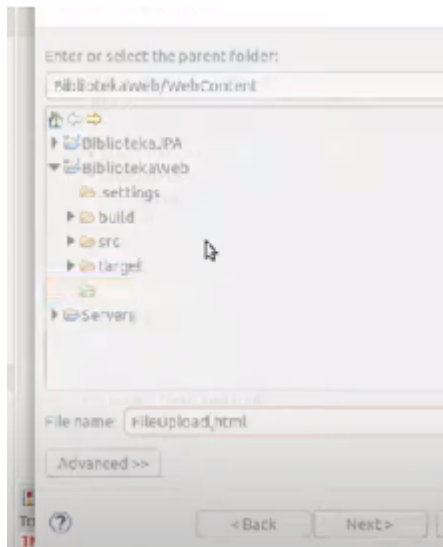
        //////////////// NEISPRAVNO
        // List<Zaduzenje> zaduzenjaClana = getSvaZaduzenja(clBroj);

        List<Zaduzenje> zaduzenjaClana = em.createQuery("
            + "select z from Zaduzenje z where "
            + "z.clan.clanskibroj=:br", Zaduzenje.class)
            .setParameter("br", clBroj)
            .getResultList();

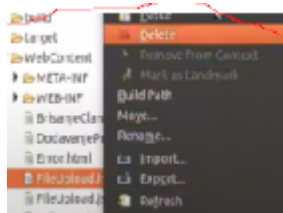
        for(Zaduzenje z:zaduzenjaClana) {
            em.remove(z);
        }
        em.remove(c);
        em.getTransaction().commit();
        return true;
    }
    catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

## # rad sa fajlovima



1.

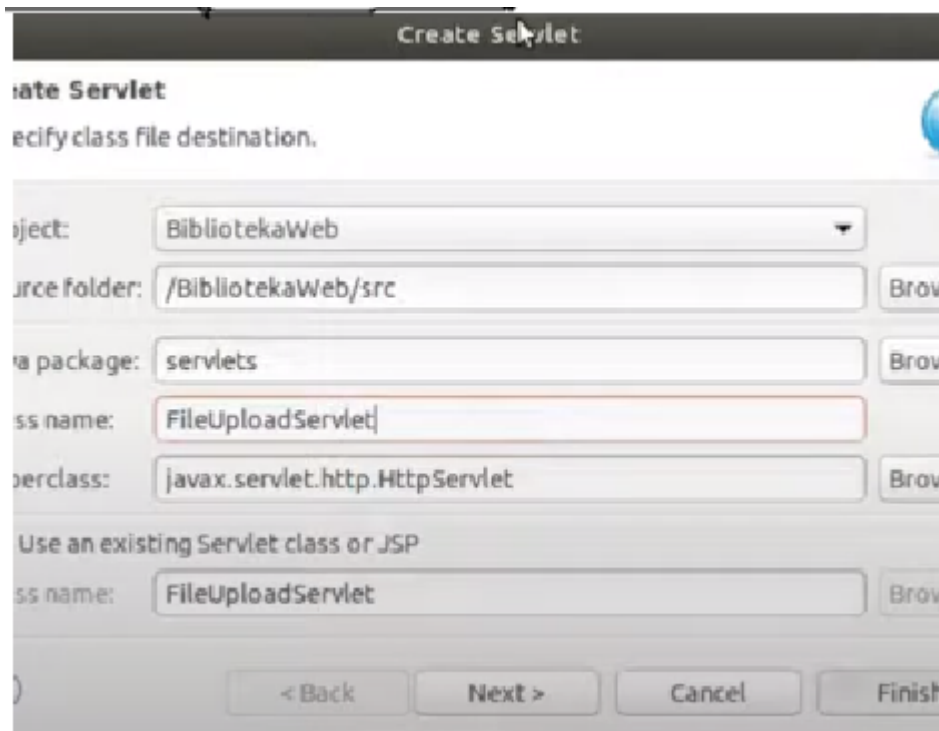


2.

3. `<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>`

4. FileUpload.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Upload file</title>
</head>
<body>
    <form action="/BibliotekaWeb/FileUploadServlet" method="post"
    enctype="multipart/form-data">
        Description: <input type="text" name="description"/><br/>
        <input type="file" name="file"/> <br/>
        <input type="submit" value="Sacuvaj"/>
    </form>
    ${poruka}
</body>
</html>
```



5.

6. FileUploadServlet.java

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@MultipartConfig(
    fileSizeThreshold=5242880, //5MB
    maxFileSize=20971520L, //20MB
    maxRequestSize=41943040L //40MB
)
@WebServlet("/FileUploadServlet")
public class FileUploadServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String poruka;
        try {
            String description = request.getParameter("description");
            Part filePart = request.getPart("file");
            String
fileName=Paths.get(filePart.getSubmittedFileName()).toString();

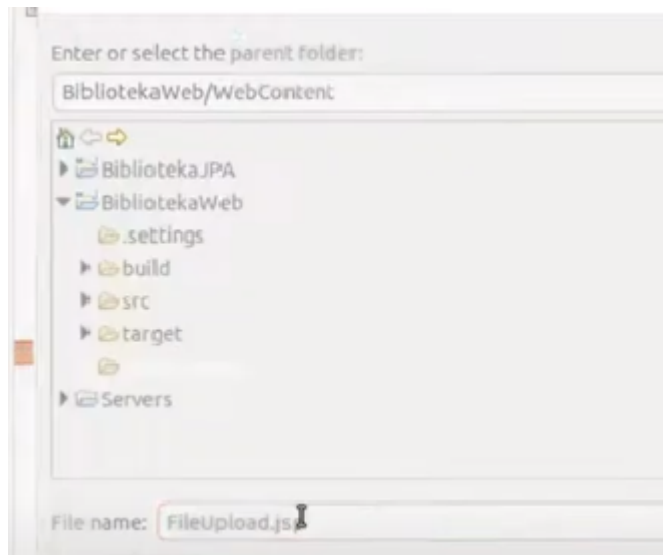
            java.io.InputStream fileContent = filePart.getInputStream();

            String uploadPath =
System.getProperty("user.home")+File.separator+fileName;
            File file = new File(uploadPath);
            OutputStream out = new FileOutputStream(file);
            file.createNewFile();
            IOUtils.copy(fileContent, out);
            fileContent.close();
            out.close();
            poruka = "Slika je uspesno sacuvana.";
        } catch (Exception e) {
```



```
        // TODO Auto-generated catch block
        e.printStackTrace();
        poruka = "Doslo je do greske. Slika nije sacuvana.";
    }

    RequestDispatcher rd =
request.getRequestDispatcher("/FileUpload.jsp");
    request.setAttribute("poruka", poruka);
    rd.forward(request, response);
}
```



1.