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 > persistance.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());
    }
}
```

> 3. cas : find, JSP, SERVLET basic lectures skipped

> **NIJE POTREBAN ZA SPRING** :: Servers *(bottom area tab switcher)* > link... > Tomcat v9 > ...

```
Potrebno je dodati sledeci dependency u pom.xml:
<dependency>
    <groupId>jstl</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>
```

> U jsp stranicu, dodati (po potrebi):

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

- ClanManager.java > saveClan()

```
...
em.getTransaction().begin();
Kategorija k = em.find(Kategorija.class, idKategorije);
Clan c = new Clan();
c.setIme(ime);
...
c.setKategorija(k);
em.persist(c);

em.getTransaction().commit();
...
```

> 4. cas : WebContent, Date, bean select, servlet

- SaveClanServlet.java > doPost(...)

```
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"));
...
```

- UnosenjePrimeraka.jsp

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

- KategorijeBean.java

```
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;
    }
}
```

> 5. cas : ***sesije***, fora sa slanjem get zahteva uz parametar querystring url

- RazduziServlet.java > doGet(...)

```
Integer clBr = (Integer)request.getSession().getAttribute("clanskiBroj");
```

- VratiZaduzenja.java> doGet(...)

```
request.getSession().setAttribute("clanskiBroj", clBr);
```

> 6. cas : brisanje uz hibernate `em.remove(nekogModelaInstanca)`, file upload

- FileUpload.jsp

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

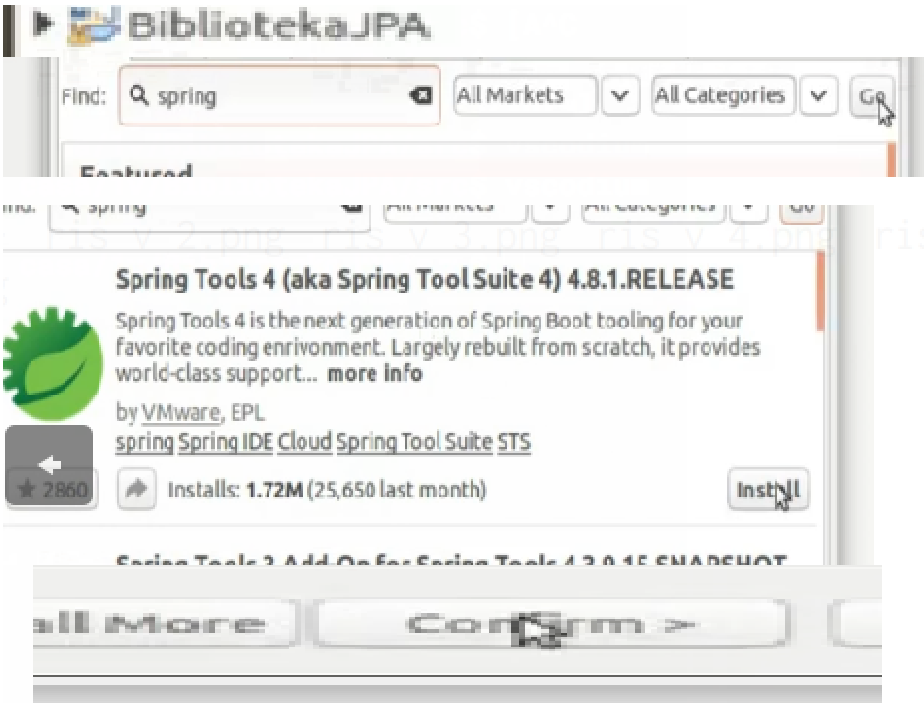
FileUploadServlet

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

8. cas

17. Help > Eclipse Marketplace...

18.



Find: spring

Spring Tools 4 (aka Spring Tool Suite 4) 4.8.1.RELEASE

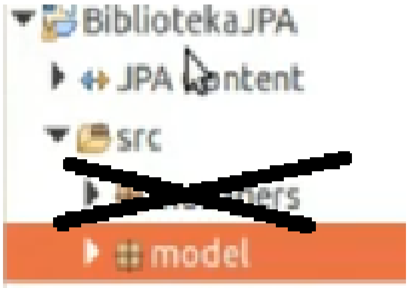
Spring Tools 4 is the next generation of Spring Boot tooling for your favorite coding enrivonment. Largely rebuilt from scratch, it provides world-class support... **more info**

by VMware, EPL

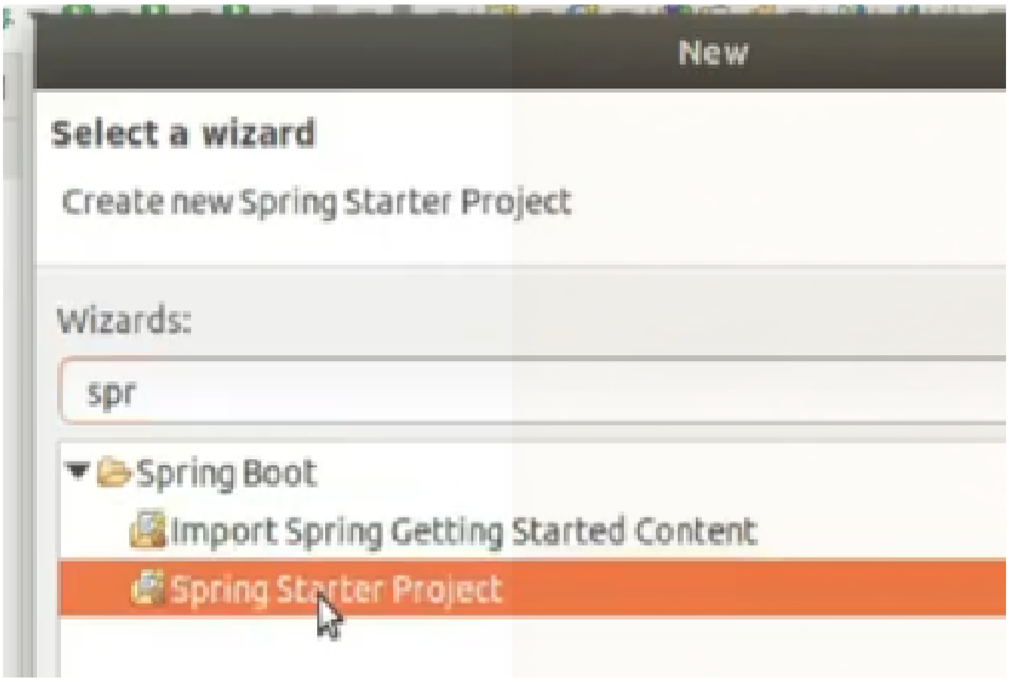
Installs: **1.72M** (25,650 last month)

BibliotekaJPA
All Markets
All Categories
Go
Featured
spring Spring IDE Cloud Spring Tool Suite STS
2860
Install
Spring Tools 3 Add-On for Spring Tools 4 3.9.15.SNAPSHOT
Install More
Confirm >

19.



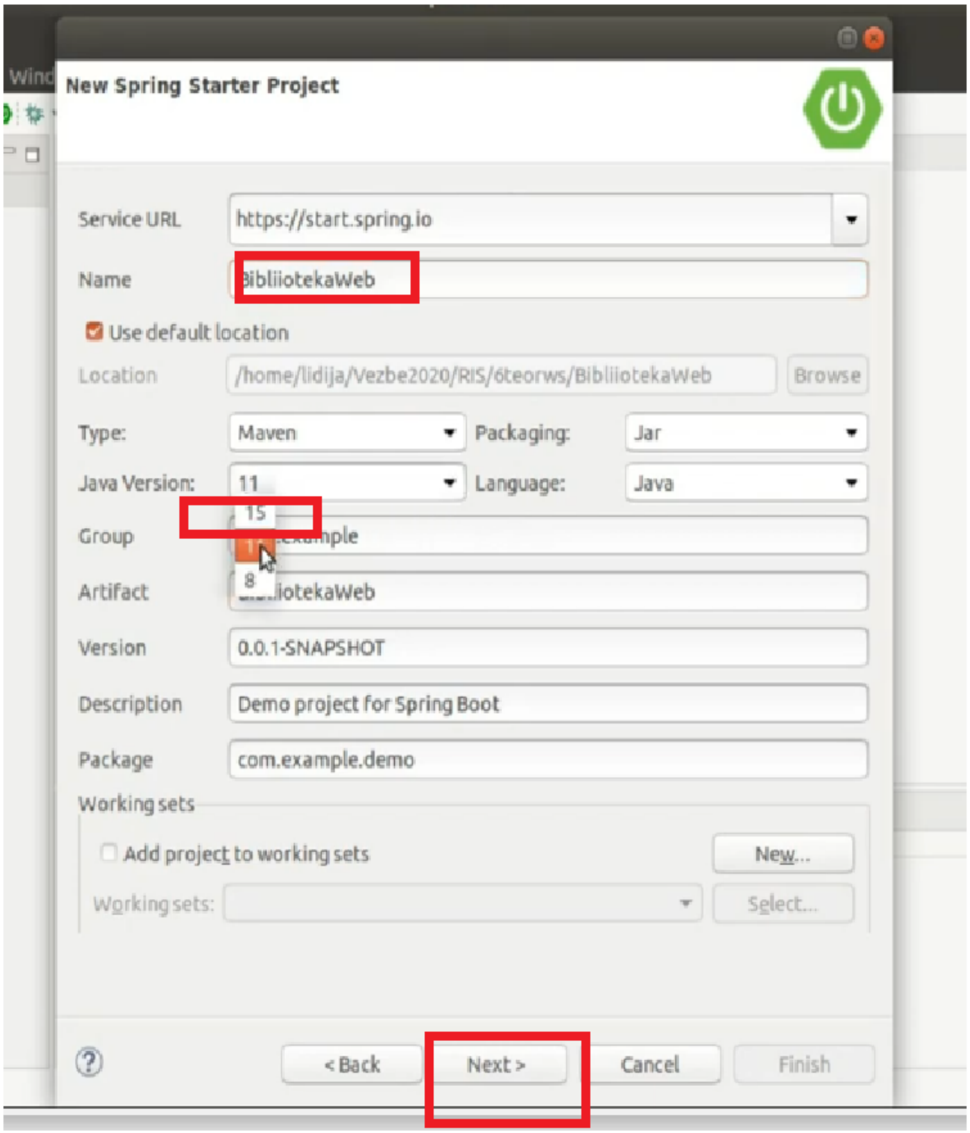
20.



Wizards: spr

New
Select a wizard
Create new Spring Starter Project
Spring Boot
Import Spring Getting Started Content
Spring Starter Project

21.



New Spring Starter Project

Service URL: https://start.spring.io

Name: BibliotekaWeb

☑ Use default location

Location: /home/lidija/Vezbe2020/RIS/6teorws/BibliiotekaWeb

Type: Maven

Packaging: Jar

Java Version: 11

Language: Java

Group: com.example

Artifact: BibliotekaWeb

Version: 0.0.1-SNAPSHOT

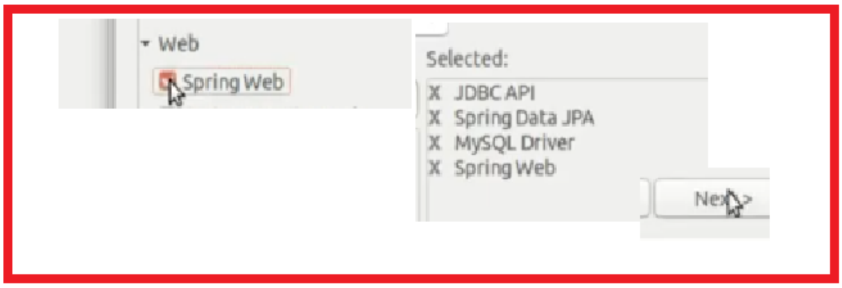
Description: Demo project for Spring Boot

Package: com.example.demo

Working sets

□ Add project to working sets

Working sets:



Web

☒ Spring Web

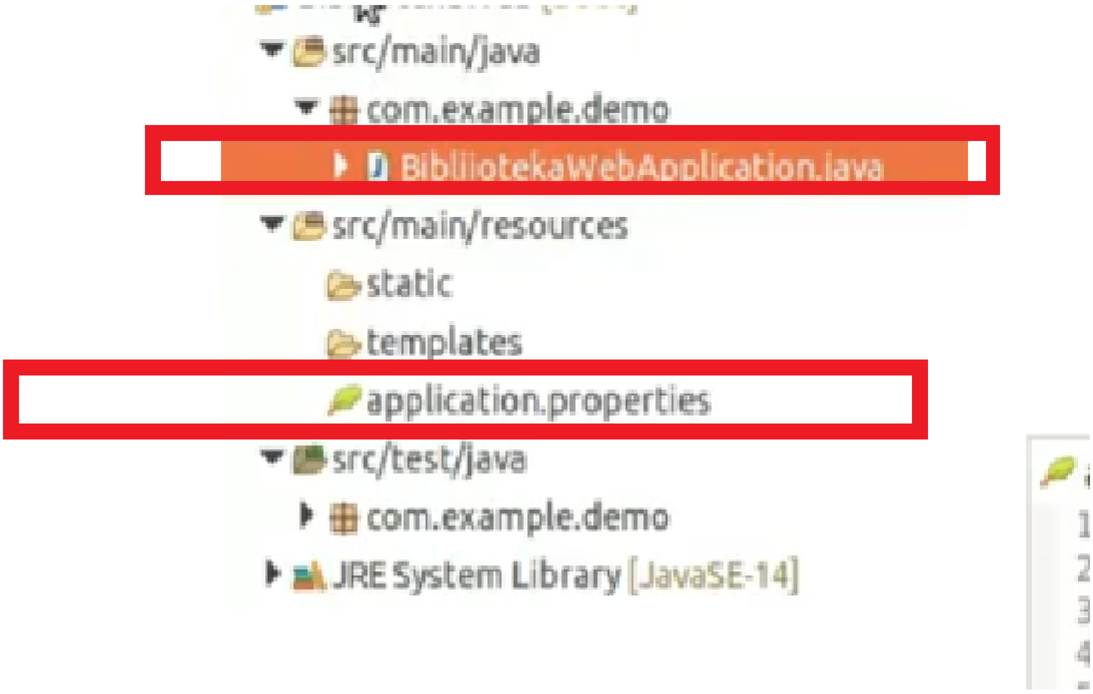
Selected:

X JDBC API
X Spring Data JPA
X MySQL Driver
X Spring Web

22.

```
<dependencies>
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>jstl</artifactId>
    </dependency>
    <dependency>
        <groupId>org.apache.tomcat.embed</groupId>
        <artifactId>tomcat-embed-jasper</artifactId>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>BibliotekaJPA</groupId>
        <artifactId>BibliotekaJPA</artifactId>
        <version>0.0.1-SNAPSHOT</version>
    </dependency>
</dependencies>
```

23. Unos tomcat-a je sada besmislen



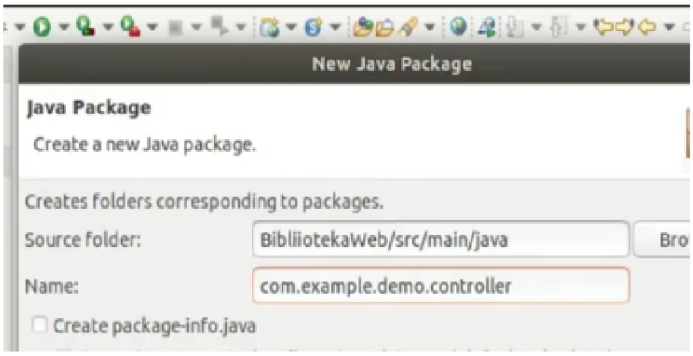
24.

```
spring.mvc.view.prefix=/
spring.mvc.view.suffix=.jsp
spring.datasource.url=jdbc:mysql://nastava.is.pmf.uns.ac.rs:3306/is2
spring.datasource.username=ris
spring.datasource.password=ris2Nastava!
server.servlet.context-path=/Library
server.port=8080
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
```

25. BibliotekaWebApplication.java

```
...
@SpringBootApplication
///////////////////
@EntityScan("model") // ←←← DODAJEMO SAMI
/////////
public class BibliiotekaWebApplication {

    public static void main(String[] args) {
        SpringApplication.run(BibliiotekaWebApplication.class, args);
    }
}
```

// svaki paket koji pravimo mora biti podpaket glavnog paketa



New Java Package

Java Package

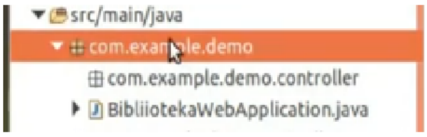
Create a new Java package.

Creates folders corresponding to packages.

Source folder: BibliiotekaWeb/src/main/java

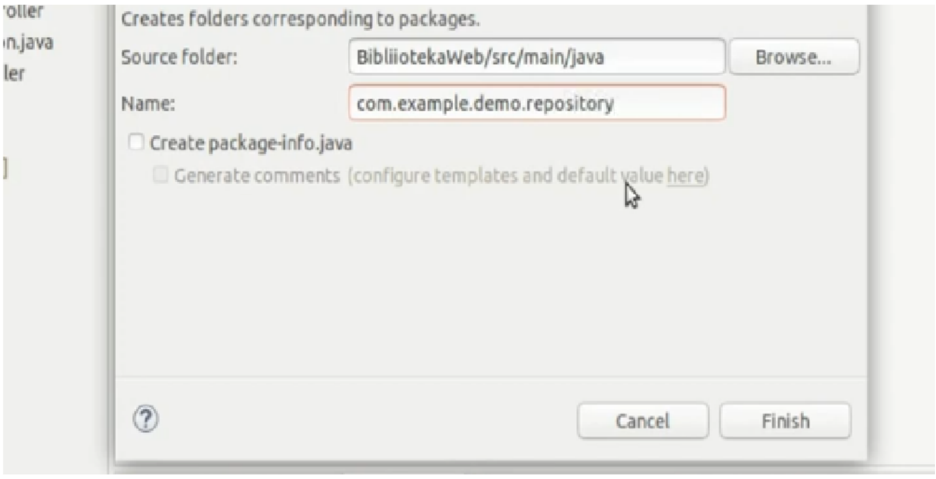
Name: com.example.demo.controller

□ Create package-info.java



src/main/java
com.example.demo
com.example.demo.controller
BibliiotekaWebApplication.java

26.



Creates folders corresponding to packages.

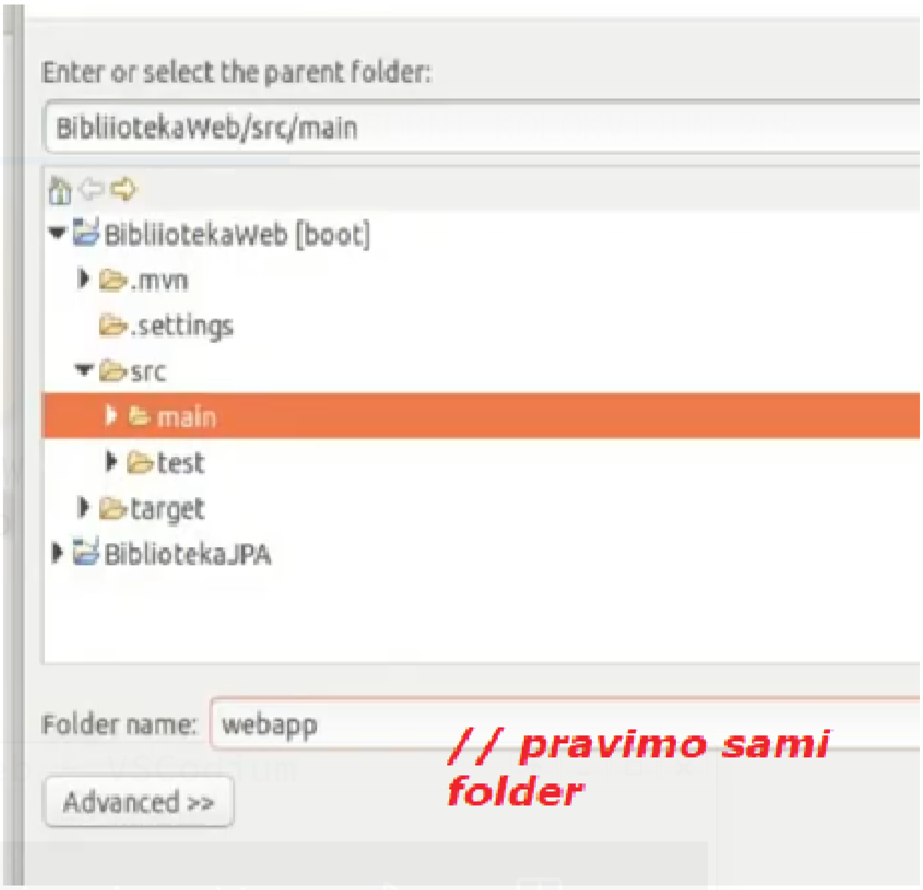
Source folder: BibliiotekaWeb/src/main/java

Name: com.example.demo.repository

□ Create package-info.java

□ Generate comments (configure templates and default value here)

27.



Enter or select the parent folder:

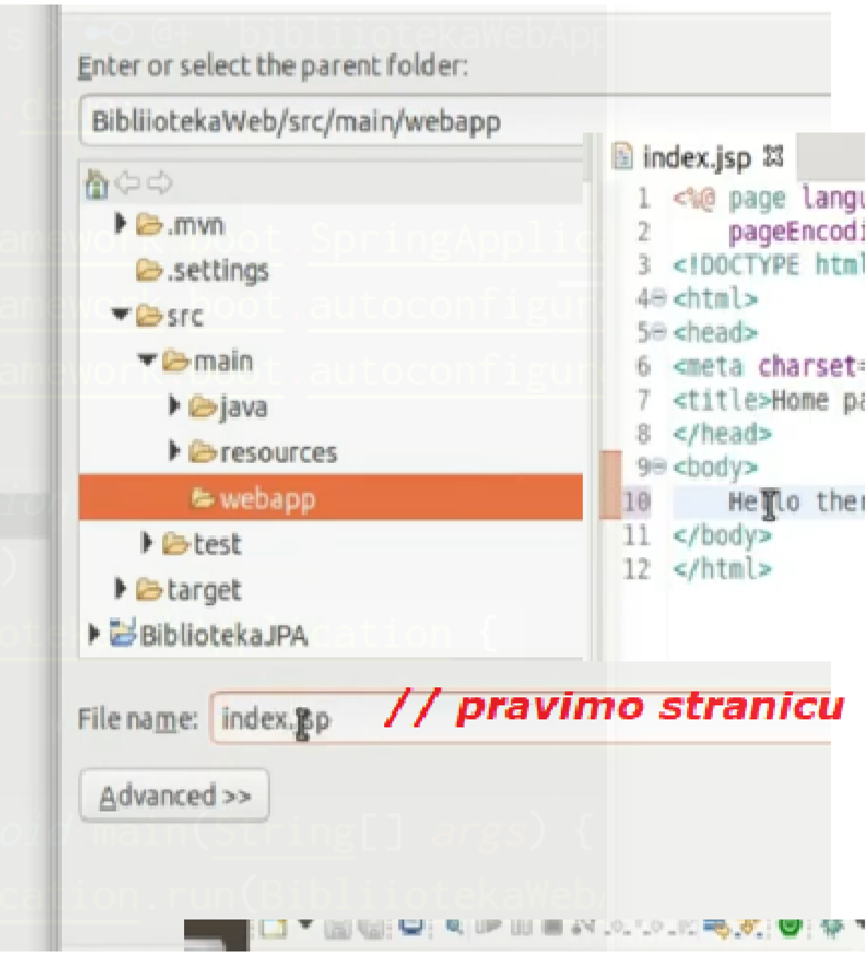
BibliiotekaWeb/src/main

BibliiotekaWeb [boot]
.mvn
.settings
src
main
test
target
BibliotekaJPA

Folder name: webapp

// pravimo sami folder

28.



Enter or select the parent folder:

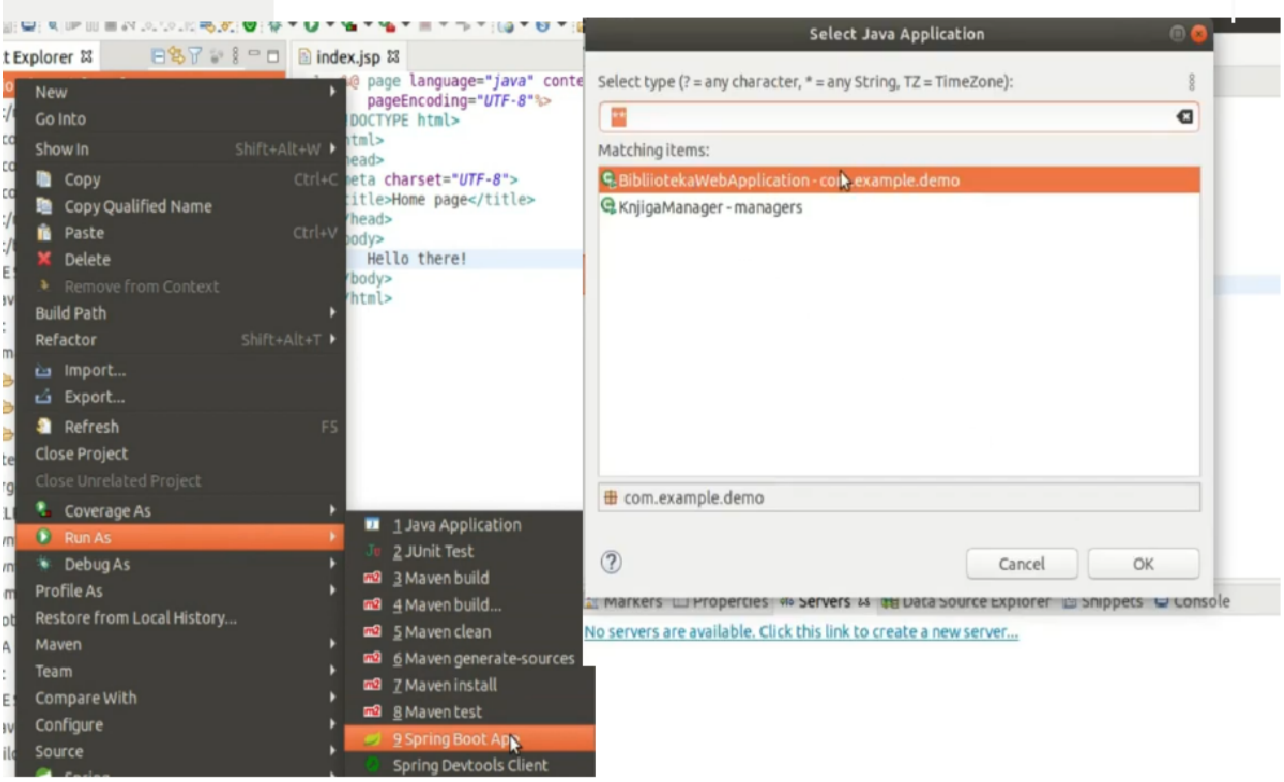
BibliiotekaWeb/src/main/webapp

File name: index.jsp // pravimo stranicu

index.jsp

```
<%@ page langu
      pageEncodi
<!DOCTYPE html
<html>
<head>
<meta charset=
<title>Home pa
</head>
<body>
      Hello ther
</body>
</html>
```

29.



index.jsp

```
<%@ page language="java" conte
    pageEncoding="UTF-8"%>
DOCTYPE html>
tml>
ead>
eta charset="UTF-8">
itle>Home page</title>
head>
ody>
    Hello there!
body>
html>
```

Select Java Application

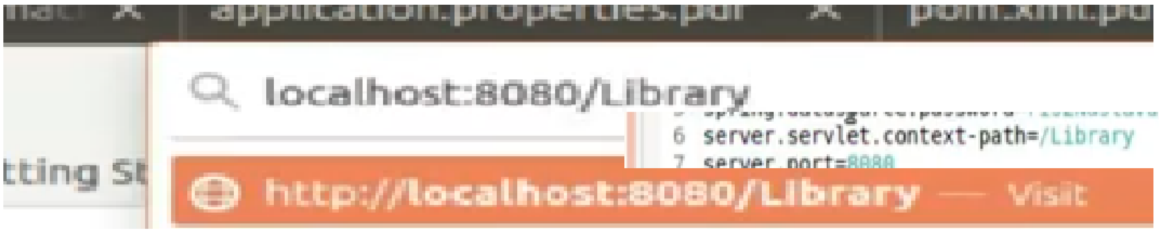
Select type (? = any character, * = any String, TZ = TimeZone):

Matching items:

BibliiotekaWebApplication - com.example.demo
KnjigaManager - managers

com.example.demo

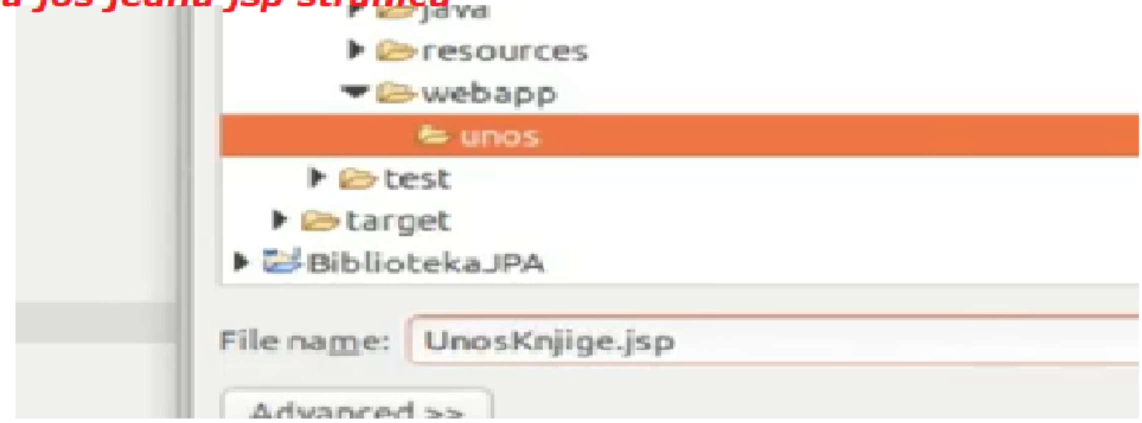
No servers are available. Click this link to create a new server...


application.properties.pdf
pom.xml.pd
tting St
localhost:8080/Library
http://localhost:8080/Library — Visit

```
6 server.servlet.context-path=/Library
7 server.port=8080
```

30. // Automatski prepoznaje postojanje index.jsp-a

// pravimo unutar stranica folder - da ne bi bilo razbacano, pa jos jednu jsp stranicu

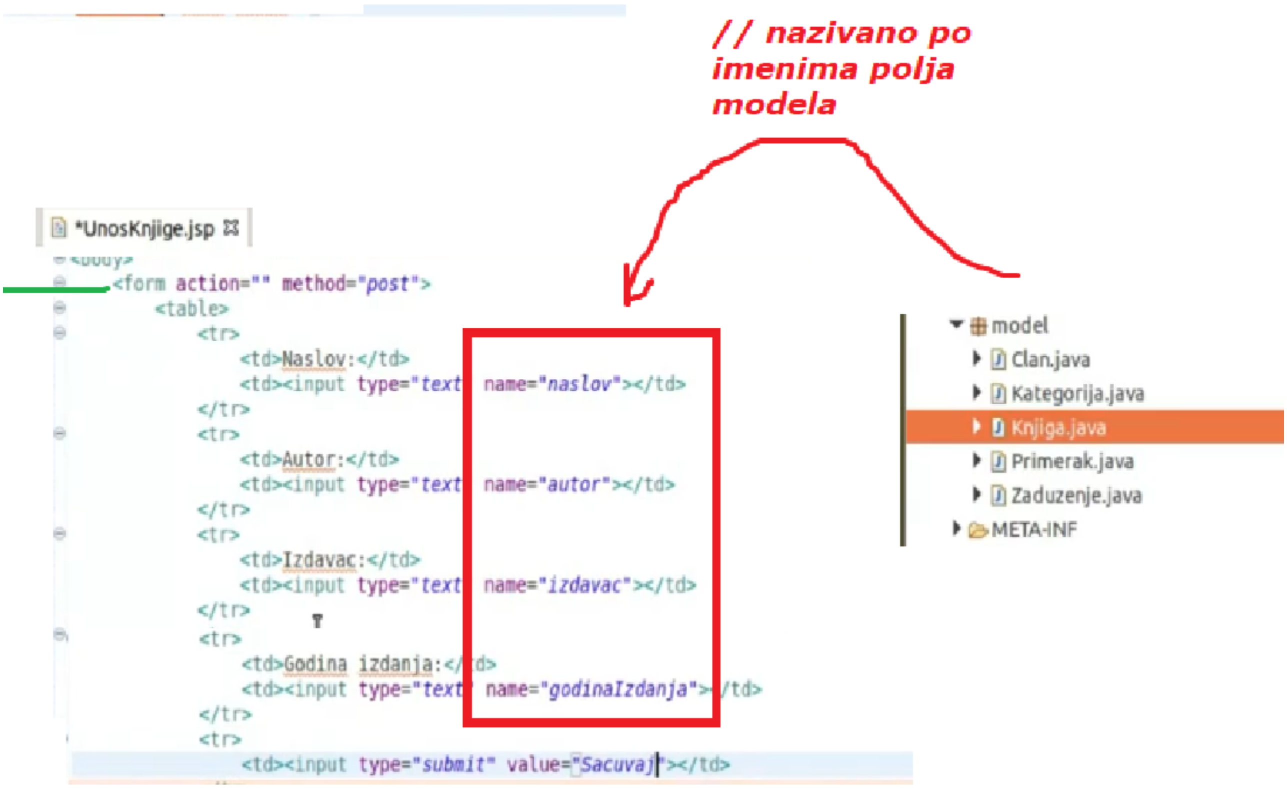


File name: UnosKnjige.jsp

31.

32.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Home page</title>
</head>
<body>
    <!-- Ovaj je link meni licno pravio problema -->
    <a href="/Library/unos/UnosKnjige.jsp">Unos knjige</a> <br>
    <!-- Ali ovaj link mi je proradio i nisam siguran u ovaj prvi -->
    <!--<a href="/Library/...nesto sto zastupa klasu servleta
controlera/...nesto sto pocinje sa get">Unos knjige</a> <br> -->
</body>

</html>
```



```
<body>
    <form action="" method="post">
        <table>
            <tr>
                <td>Naslov:</td>
                <td><input type="text" name="naslov"></td>
            </tr>
            <tr>
                <td>Autor:</td>
                <td><input type="text" name="autor"></td>
            </tr>
            <tr>
                <td>Izdavac:</td>
                <td><input type="text" name="izdavac"></td>
            </tr>
            <tr>
                <td>Godina izdanja:</td>
                <td><input type="text" name="godinaIzdanja"></td>
            </tr>
            <tr>
                <td><input type="submit" value="Sacuvaj"></td>
```

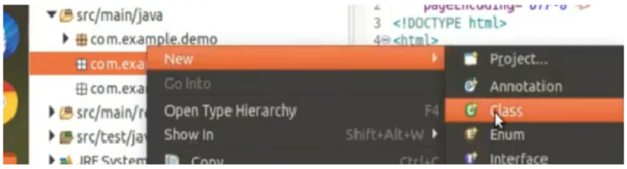
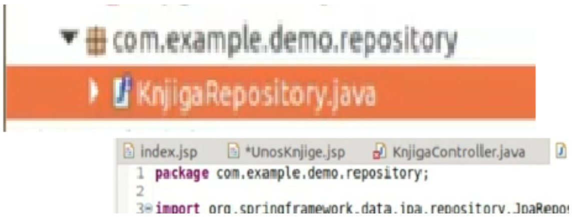
33.

34.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Unos nove knjige</title>
</head>
<body>
    <form action="/Library/knjige/saveKnjiga" method="post">
        <table>
            <tr>
                <td>Naslov:</td>
                <td><input type="text" name="naslov"></td>
            </tr>
            <tr>
                <td>Autor:</td>
                <td><input type="text" name="autor"></td>
            </tr>
            <tr>
                <td>Izdavac:</td>
                <td><input type="text" name="izdavac"></td>
            </tr>
            <tr>
                <td>Godina izdanja:</td>
                <td><input type="text" name="godinaIzdanja"></td>
            </tr>
            <tr>
                <td><input type="submit" value="Sacuvaj"></td>
            </tr>
        </table>
    </form>

</body>
</html>
```

35. index.jsp

    `<a href="/Library/clanovi/getKategorije">Unos clana</a>`

36. 

```
3 <!DOCTYPE html>
4 <html>
```

```
package com.example.demo.repository;

import org.springframework.data.jpa.repository.JpaRepo
```

    - pkgs repository i controller

```

...
public interface KnjigaRepository extends JpaRepository<Knjiga, Integer>,
KnjigaRepoSpecific {
    // prazno
}
```

37. controller > ***new class*** KnjigaController

```

import model.Knjiga;

@Controller
@RequestMapping(value="/knjige")
public class KnjigaController {

    @Autowired
    KnjigaRepository kr;

    @RequestMapping(value="/saveKnjiga", method=RequestMethod.POST)
    public String saveKnjiga(@Valid Knjiga k, Errors e, Model m,
HttpServletRequest request) {
        if(e.hasErrors())
            return "Greska";
        else
            kr.save(k);
        request.getSession().setAttribute("knjiga", k);
        m.addAttribute("poruka", "Knjiga je uspesno sacuvana.");

        return "unos/UnosPrimeraka"; // mora bez .jsp!!!
    }
```

38. **nov jsp** UnosPrimeraka.jsp

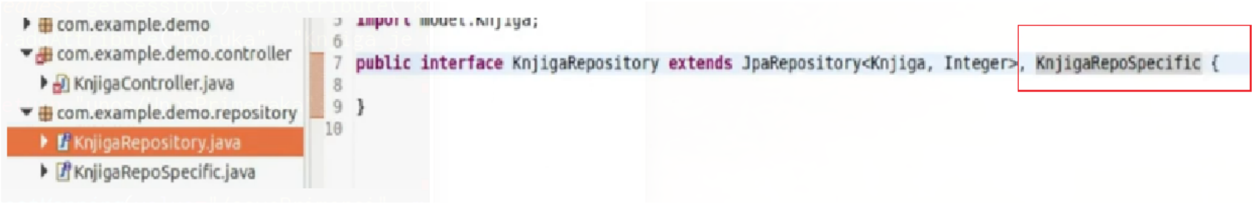
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Unos primeraka knjige</title>
</head>
<body>
${poruka}
<br>
<form action="/Library/knjige/savePrimerci" method="post">
    Broj primeraka: <input type="text" name="brPrimeraka"><br>
    <input type="submit" value="Dodaj">
</form>
<br>
<c:if test="${!empty inventarniBrojevi}">
    Primerci su uspesno sacuvani. Generisani su sledeci inventarni
brojevi:
    <c:forEach items="${inventarniBrojevi}" var="br">
        ${br}
    </c:forEach>
</c:if>
</body>
</html>
```

39. KnjigaController

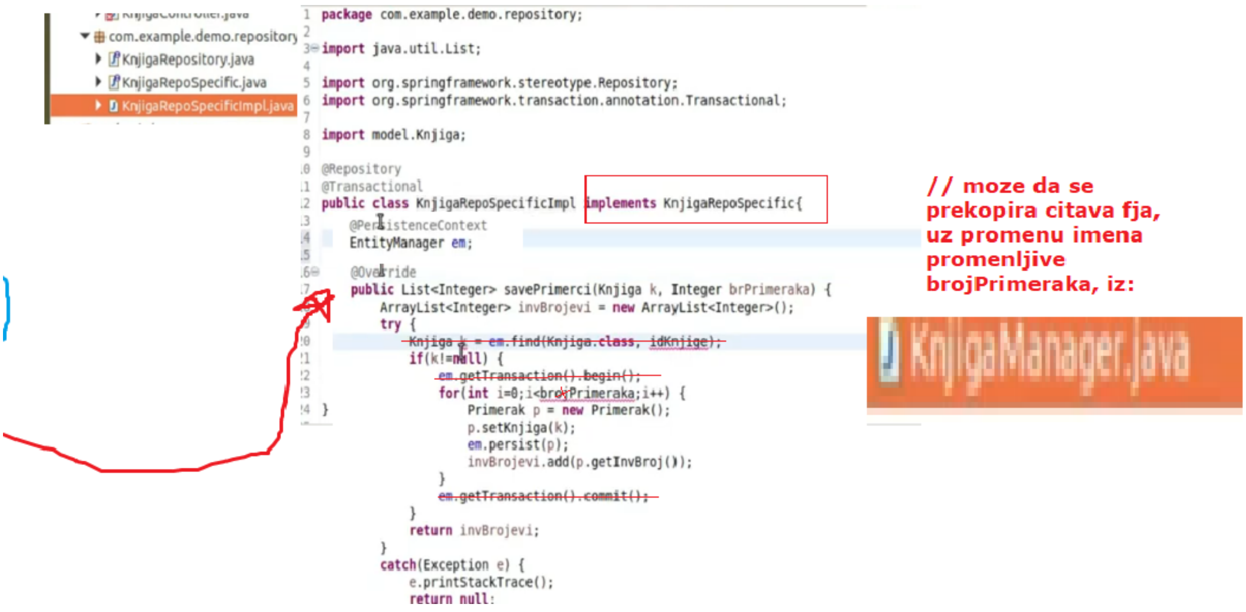
```
@RequestMapping(value="/savePrimerci", method=RequestMethod.POST)
    public String savePrimerci(Integer brPrimeraka, HttpServletRequest
request, Model m) {
        Knjiga k = (Knjiga)request.getSession().getAttribute("knjiga");
        List<Integer> invBrojevi = kr.savePrimerci(k, brPrimeraka);
        m.addAttribute("inventarniBrojevi", invBrojevi);
        return "unos/UnosPrimeraka";
    }
```

40.

```
...
public interface KnjigaRepoSpecific {

    public List<Integer> savePrimerci(Knjiga k, Integer brPrimeraka);
}
```

41. 

```
import model.Knjiga;

public interface KnjigaRepository extends JpaRepository<Knjiga, Integer>, KnjigaRepoSpecific {

}
```

42. 

```
package com.example.demo.repository;

import java.util.List;

import org.springframework.stereotype.Repository;
import org.springframework.transaction.annotation.Transactional;

import model.Knjiga;

@Repository
@Transactional
public class KnjigaRepoSpecificImpl implements KnjigaRepoSpecific{
    @PersistenceContext
    EntityManager em;

    @Override
    public List<Integer> savePrimerci(Knjiga k, Integer brPrimeraka) {
        ArrayList<Integer> invBrojevi = new ArrayList<Integer>();
        try {
            ~~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;
}
```

// moze da se prekopira citava fja, uz promenu imena promenljive brojPrimeraka, iz:

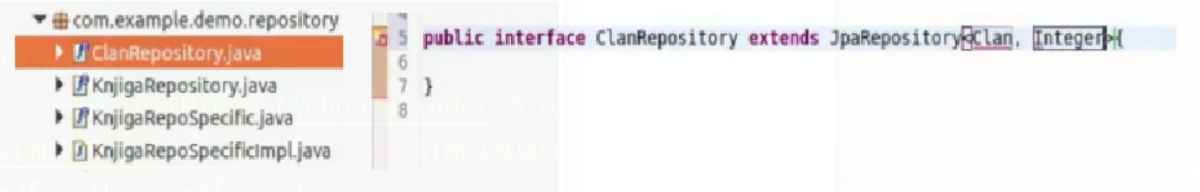
- isti kod

```
...
@Repository
@Transactional
public class KnjigaRepoSpecificImpl implements KnjigaRepoSpecific{

    @PersistenceContext
    EntityManager em;

    @Override
    public List<Integer> savePrimerci(Knjiga k, Integer brPrimeraka) {
        ArrayList<Integer> invBrojevi = new ArrayList<Integer>();
        try {
            if(k!=null) {
                for(int i=0;i<brPrimeraka;i++) {
                    Primerak p = new Primerak();
                    p.setKnjiga(k);
                    em.persist(p);
                    invBrojevi.add(p.getInvBroj());
                }
            }
            return invBrojevi;
        }
        catch(Exception e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

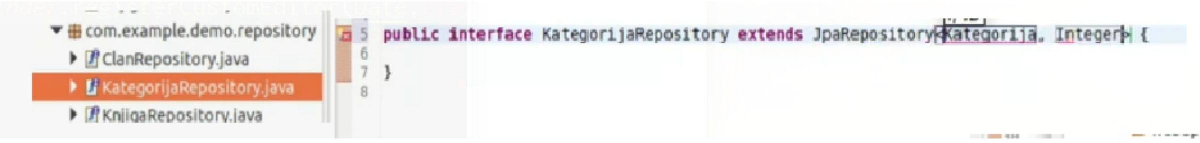
43. combo box

- **new class** ClanController

```
    @RequestMapping(value="/saveClan", method=RequestMethod.POST)
    public String saveClan(String ime, String prezime, String adresa, Date
datumRodjenja, Date datumUpisa, Integer idKat, Model m) {
        Clan c = new Clan();
        c.setAdresa(adresa);
        c.setDatumRodjenja(datumRodjenja);
        c.setDatumUpisa(datumUpisa);
        c.setIme(ime);
        c.setPrezime(prezime);

        Kategorija k = kr.findById(idKat).get();
        c.setKategorija(k);
        Clan clan = cr.save(c);

        m.addAttribute("clan", clan);
        // NIJE setAttribute!!!!!

        return "unos/UnosClana";
    }
```

- 

```
public interface ClanRepository extends JpaRepository<Clan, Integer>{

}
```

- 

```
public interface KategorijaRepository extends JpaRepository<Kategorija, Integer> {

}
```

- **Korisno za prikupljanje podataka za select**

```
@Controller
@RequestMapping(value="/clanovi")
public class ClanController {

    @Autowired
    KategorijaRepository kr;

    @Autowired
    ClanRepository cr;

    @RequestMapping(value="/getKategorije", method=RequestMethod.GET)
    public String getKategorije(HttpServletRequest request) {
        List<Kategorija> kat = kr.findAll();
        request.getSession().setAttribute("kategorije", kat);
        return "unos/UnosClana";
    }
```

- **nov jsp** UnosClana.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 clana</title>
</head>
<body>
    <form action="/Library/clanovi/saveClan" method="post">
        <table>
            <tr>
                <td>Kategorija:</td>
                <td>
                    <select name="idKat">
                        <c:forEach items="${kategorije}" var="k">
                            <option
value="${k.idkategorije}">${k.nazivkategorije}</option>
                        </c:forEach>
                    </select>
                </td>
            </tr>
            <tr>
                <td>Ime:</td>
                <td><input type="text" name="ime"></td>
            </tr>
            <tr>
                <td>Prezime:</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"></td>
            </tr>
            <tr>
                <td>Datum upisa:</td>
                <td><input type="date" name="datumUpisa"></td>
            </tr>
            <tr>
```

```
                <td><input type="submit" value="Sacuvaj"></td>
            </tr>
        </table>
    </form>
<br>
<c:if test="${!empty clan}">
Clan je uspesno sacuvan. Clanski broj je ${clan.clanskibroj}
</c:if>
</body>
</html>
```

44. ClanController - obrada datuma iz UnosClana.jsp

```
    @InitBinder
    public void initBinder(WebDataBinder binder) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        sdf.setLenient(true); // koriscenje heuristika za parsiranje u
date iz str
        binder.registerCustomEditor(Date.class, new CustomDateEditor(sdf,
true));
    }
}
```

9. vezbe - brzi nacin za rad sa formama

45. ClanController - spremamo formi tip koji ce da se salje

```
    @ModelAttribute("clan")
    public Clan getClan() {
        return new Clan();
    }
```

46. listamo za combo box

```
    @ModelAttribute("kategorije")
    public List<Kategorija> getKategorije(){
        return kr.findAll();
    }
```

47. Redirekcije

```
    @RequestMapping(value="/unosClana", method=RequestMethod.GET)
    public String unosClana() {
        return "unos/UnosClana1";
    }
```

- za index.jsp

```
<a href="/Library/clanovi/unosClana">Unos clana / data binding</a> <br>
```

48. **nova klasa i paket** converter/KategorijaKonverter.java

   - ima 2 nacina

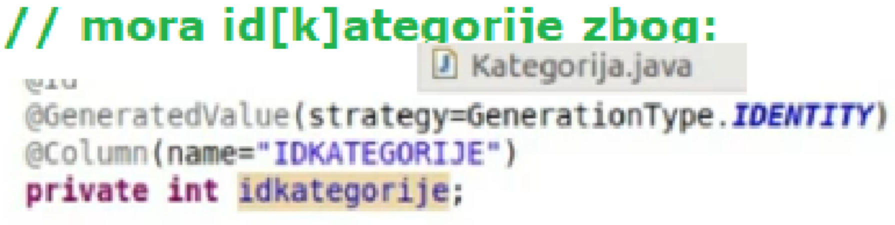
```
public class KategorijaKonverter implements Converter<String, Kategorija>
{

    KategorijaRepository kr;

    public KategorijaKonverter(KategorijaRepository kr) {
        this.kr = kr;
    }

    @Override
    public Kategorija convert(String source) {
        int katid=-1;
        try {
            katid = Integer.parseInt(source);
        }
        catch (NumberFormatException e) {
            throw new
ConversionFailedException(TypeDescriptor.valueOf(String.class),
                    TypeDescriptor.valueOf(Kategorija.class), katid,
null);
        }
        Kategorija k = kr.findById(katid).get();
        return k;
    }

}
```

49. **novi jsp** UnosClana1.jsp

```
<%@ page import="java.util.Date" %>
<%@ 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"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Unos clana</title>
</head>
<body>
    <c:set var="today" value="<%=new Date()%>"/>
    <form:form action="/Library/clanovi/saveClan1" method="post"
modelAttribute="clan">
        <table>
            <tr>
                <td>Kategorija:</td>
                <td>
<!--                    <form:select name="idKat" path="kategorija">
                        <form:options items="${kategorije}"
itemValue="idkategorije"
                                            itemLabel="nazivkategorije"/>
                    </form:select>
-->
                    <form:select path="kategorija" items="${kategorije}"
itemValue="idkategorije"
                                            itemLabel="nazivkategorije"/>
                </td>
            </tr>
            <tr>
                <td>Ime:</td>
                <td><form:input type="text" path="ime"/></td>
            </tr>
            <tr>
                <td>Prezime:</td>
                <td><form:input type="text" path="prezime"/></td>
            </tr>
            <tr>
                <td>Adresa:</td>
                <td><form:input type="text" path="adresa"/></td>
            </tr>
            <tr>
                <td>Datum rodjenja:</td>
                <td><form:input type="date" path="datumRodjenja"/></td>
            </tr>
            <tr>
                <td>Datum upisa:</td>
                <td><form:input type="date" path="datumUpisa"
value='<fmt:formatDate
                            pattern="yyyy-MM-dd" value="${today}"/>'/>
</td>
            </tr>
            <tr>
                <td><input type="submit" value="Sacuvaj"></td>
            </tr>
        </table>
    </form:form>
<br>
```

```
<c:if test="${!empty clanSaved}">
${poruka} ${clanSaved.clanskibroj}
</c:if>
</body>
</html>
```

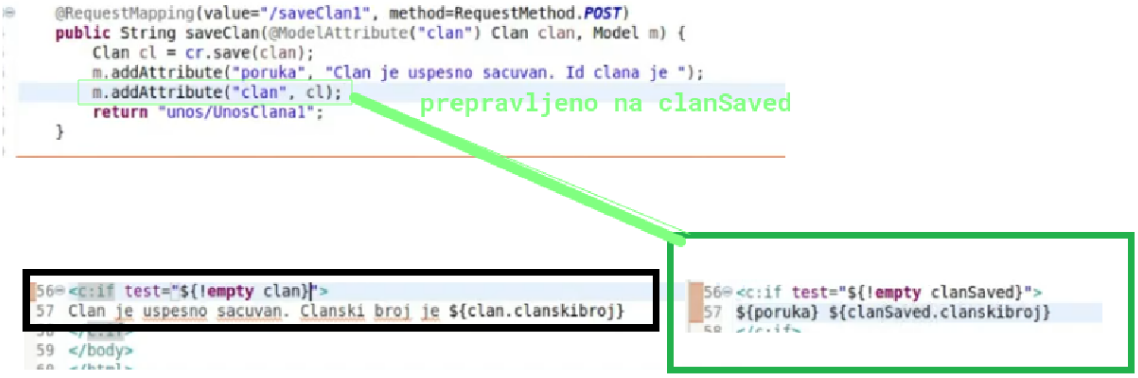


```
// mora id[k]ategorije zbog:
@GeneratedValue(strategy=GenerationType.IDENTITY)
@Column(name="IDKATEGORIJE")
private int idkategorije;
```

-
- Cuvanje

```
@RequestMapping(value="/saveClan1", method=RequestMethod.POST)
public String saveClan(@ModelAttribute("clan") Clan clan, Model m) {
    Clan cl = cr.save(clan);
    m.addAttribute("poruka", "Clan je uspesno sacuvan. Id clana je ");
    m.addAttribute("clanSaved", cl);
    return "unos/UnosClana1";
}
```

- ovo je vec implementirano



```
@RequestMapping(value="/saveClan1", method=RequestMethod.POST)
public String saveClan(@ModelAttribute("clan") Clan clan, Model m) {
    Clan cl = cr.save(clan);
    m.addAttribute("poruka", "Clan je uspesno sacuvan. Id clana je ");
    m.addAttribute("clan", cl);
    return "unos/UnosClana1";
}
```

prepravljeno na clanSaved

```
<c:if test="${!empty clan}">
Clan je uspesno sacuvan. Clanski broj je ${clan.clanskibroj}
</body>
```

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