

Java Server Faces  
Auditory Laboratory  
Example 001

## Линкови

[index.jsp](#)  
[menu.jsp](#)  
[META-INF\MANIFEST.MF](#)  
[obaveze.xhtml](#)  
[pregledObaveza.xhtml](#)  
[unosObaveza.xhtml](#)  
[WEB-INF\components\body\\_obaveze.xhtml](#)  
[WEB-INF\components\body\\_pregledObaveza.xhtml](#)  
[WEB-INF\components\body\\_unosObaveza.xhtml](#)  
[WEB-INF\components\header.xhtml](#)  
[WEB-INF\faces-config.xml](#)  
[WEB-INF\lib\javax.faces-2.1.24.jar](#)  
[WEB-INF\lib\mysql-connector-java-8.0.13.jar](#)  
[WEB-INF\planer\\_obaveza.sql](#)  
[WEB-INF\templates\template.xhtml](#)  
[WEB-INF\web.xml](#)

Пребројано је 16 ставки.

## BOOKS APP MENU

[Obaveze](#)  
[Pregled obaveza](#)  
[Unos obaveza](#)

[Menu](#)  
[Index](#)

Trenutni datum: 10.08.2021. Adresa sa koje korisnik pristupa: http://localhost:8080/049\_IP\_Vjezbe\_Materijali\_JSF\_01/obaveze.xhtml

[Pregled obaveza](#) || [Unos obaveza](#)

Obaveze za trazeni datum su:

ID obaveze	Opis	Kategorija	Adresa korisnika	Datum	Obrisi obavezu
------------	------	------------	------------------	-------	----------------

Trenutni datum: 10.08.2021. Adresa sa koje korisnik pristupa: http://localhost:8080/049\_IP\_Vjezbe\_Materijali\_JSF\_01/obaveze.xhtml

[Pregled obaveza](#) || [Unos obaveza](#)

Obaveze za traženi datum su:

ID obaveze	Opis	Kategorija	Adresa korisnika	Datum	Obrisi obavezu
1	opis	neka	adresa	2018-01-17	<a href="#">Obrisi - 1</a>
2	Ovo je nova obaveza	poslovna obaveza	Neka	2018-01-20	<a href="#">Obrisi - 2</a>
3	Polaganje ispita.	poslovna obaveza	Banja Luka	2021-01-01	<a href="#">Obrisi - 3</a>
4	Završavanje kalkulatora.	privatna obaveza	Modrica	2021-01-02	<a href="#">Obrisi - 4</a>
5	Turnir u Chicken Invadersu.	ostalo	New York	2021-01-29	<a href="#">Obrisi - 5</a>

Trenutni datum: 10.08.2021. Adresa sa koje korisnik pristupa: http://localhost:8080/049\_IP\_Vjezbe\_Materijali\_JSF\_01/pregledObaveza.xhtml

[Pregled obaveza](#) || [Unos obaveza](#)

Unesite datum za prikaz:

Trenutni datum: 10.08.2021. Adresa sa koje korisnik pristupa: http://localhost:8080/049\_IP\_Vjezbe\_Materijali\_JSF\_01/unosObaveza.xhtml

[Pregled obaveza](#) || [Unos obaveza](#)

Unesite podatke o željenoj obavezi

Opis:

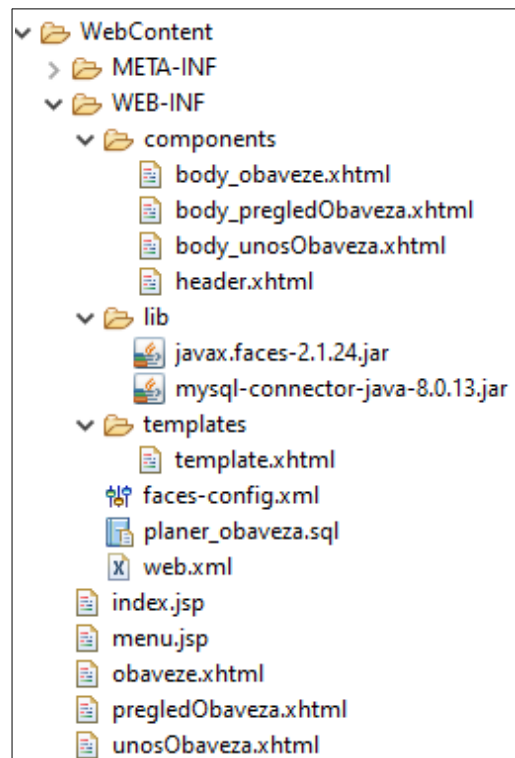
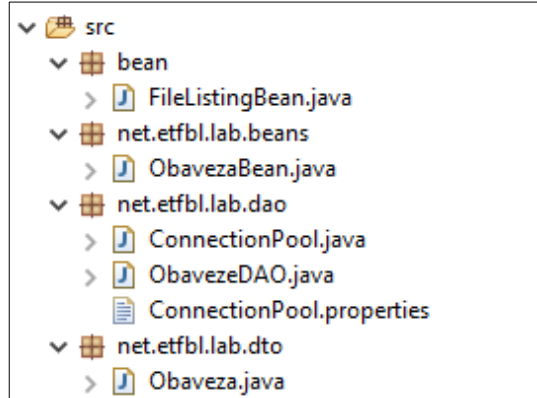
Adresa:

Datum:

Kategorija:

poslovna obaveza ▾

poslovna obaveza ▾  
poslovna obaveza  
privatna obaveza  
ostalo



## ConnectionPool.java

```
1 jdbcURL=jdbc:mysql://127.0.0.1:3306/planer_obaveza?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC
2 username=root
3 password=root
4 driver=com.mysql.cj.jdbc.Driver
5 preconnectCount=2
6 maxIdleConnections=25
7 maxConnections=25
```

## web.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3   <display-name>049_IP_Vjezbe_Materijali_JSF_01</display-name>
4   <welcome-file-list>
5     <welcome-file>index.html</welcome-file>
6     <welcome-file>index.htm</welcome-file>
7     <welcome-file>index.jsp</welcome-file>
8     <welcome-file>default.html</welcome-file>
9     <welcome-file>default.htm</welcome-file>
10    <welcome-file>default.jsp</welcome-file>
11    <welcome-file>pregledObaveza.xhtml</welcome-file>
12  </welcome-file-list>
13
14  <servlet>
15    <servlet-name>Faces Servlet</servlet-name>
16    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
17    <load-on-startup>1</load-on-startup>
18  </servlet>
19  <servlet-mapping>
20    <servlet-name>Faces Servlet</servlet-name>
21    <url-pattern>*.xhtml</url-pattern>
22  </servlet-mapping>
23  <context-param>
24    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
25    <param-value>client</param-value>
26  </context-param>
27  <context-param>
28    <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
29    <param-value>resources.application</param-value>
30  </context-param>
31
32 </web-app>
```

## faces-config.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <faces-config
4     xmlns="http://xmlns.jcp.org/xml/ns/javaee"
5     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6     xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig_2_2.xsd"
7     version="2.2">
8 </faces-config>
9
```

## planer\_obaveza.sql

```
1 CREATE DATABASE IF NOT EXISTS `planer_obaveza`;
2 USE `planer_obaveza`;
3
4 DROP TABLE IF EXISTS `obaveza`;
5
6 CREATE TABLE `obaveza` (
7   `id` int(11) NOT NULL AUTO_INCREMENT,
8   `opis` varchar(45) NOT NULL,
9   `datum` date NOT NULL,
10  `kategorija` varchar(45) NOT NULL,
11  `adresaKorisnika` varchar(45) NOT NULL,
12  PRIMARY KEY (`id`)
13 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
14
15 INSERT INTO `obaveza` VALUES (1,'opis','2018-01-17','neka','adresa'),(2,'Ovo je nova obaveza','2018-01-20','poslovna obaveza','Neka');
```

## unosObaveza.html

```
1 <html xmlns="http://www.w3.org/1999/xhtml"
2     xmlns:ui="http://java.sun.com/jsf/facelets"
3     xmlns:h="http://java.sun.com/jsf/html">
4   <h:head/>
5   <h:body>
6     <ui:composition template="/WEB-INF/templates/template.xhtml">
7       <ui:define name="header">
8         <ui:include src="/WEB-INF/components/header.xhtml"></ui:include>
9       </ui:define>
10      <ui:define name="body">
11        <ui:include src="/WEB-INF/components/body_unosObaveza.xhtml"></ui:include>
12      </ui:define>
13    </ui:composition>
14  </h:body>
15 </html>
```

## pregledObaveze.xhtml

```
1 <html xmlns="http://www.w3.org/1999/xhtml"
2     xmlns:ui="http://java.sun.com/jsf/facelets"
3     xmlns:h="http://java.sun.com/jsf/html">
4   <h:head/>
5   <h:body>
6     <ui:composition template="/WEB-INF/templates/template.xhtml">
7       <ui:define name="header">
8         <ui:include src="/WEB-INF/components/header.xhtml"></ui:include>
9       </ui:define>
10      <ui:define name="body">
11        <ui:include src="/WEB-INF/components/body_pregledObaveza.xhtml"></ui:include>
12      </ui:define>
13    </ui:composition>
14  </h:body>
15 </html>
```

## obaveze.xhtml

```
1 <html xmlns="http://www.w3.org/1999/xhtml"
2     xmlns:ui="http://java.sun.com/jsf/facelets"
3     xmlns:h="http://java.sun.com/jsf/html">
4     <h:head/>
5     <h:body>
6         <ui:composition template="/WEB-INF/templates/template.xhtml">
7             <ui:define name="header">
8                 <ui:include src="/WEB-INF/components/header.xhtml"></ui:include>
9             </ui:define>
10            <ui:define name="body">
11                <ui:include src="/WEB-INF/components/body_obaveze.xhtml"></ui:include>
12            </ui:define>
13        </ui:composition>
14    </h:body>
15 </html>
```

## template.xhtml

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml"
3     xmlns:ui="http://java.sun.com/jsf/facelets"
4     xmlns:h="http://java.sun.com/jsf/html"
5     xmlns:f="http://java.sun.com/jsf/core">
6     <h:head>
7         <title>Aplikacija za planiranje obaveza</title>
8     </h:head>
9     <h:body>
10        <ui:insert name="header" />
11        <ui:insert name="body" />
12    </h:body>
13 </html>
```

## header.xhtml

```
1 <html xmlns="http://www.w3.org/1999/xhtml"
2     xmlns:ui="http://java.sun.com/jsf/facelets"
3     xmlns:h="http://java.sun.com/jsf/html"
4     xmlns:f="http://java.sun.com/jsf/core">
5     <h:head />
6     <ui:composition>
7         <h:outputText value="Trenutni datum: " style="font-weight: bold;"></h:outputText>
8         <h:outputText value="${session.lastAccessedTime}">
9             <f:convertDateTime pattern="dd.MM.yyyy." type="date" />
10        </h:outputText>
11        <h:outputText value=" Adresa sa koje korisnik pristupa: " style="font-weight: bold;"></h:outputText>
12        <h:outputText value="${request.getRequestURL()}"></h:outputText>
13        <hr />
14        <h:outputLink value="pregledObaveza.xhtml">Pregled obaveza</h:outputLink> || <h:outputLink value="unosObaveza.xhtml">Unos obaveza</h:outputLink>
15    </ui:composition>
16 </html>
```

## body\_unosObaveza.xhtml

```

1 <html xmlns="http://www.w3.org/1999/xhtml"
2   xmlns:ui="http://java.sun.com/jsf/facelets"
3   xmlns:h="http://java.sun.com/jsf/html"
4   xmlns:f="http://java.sun.com/jsf/core">
5   <h:head />
6   <ui:composition>
7     <hr />
8     <h:outputText value="Unesite podatke o zeljenoj obavezi"></h:outputText>
9     <hr />
10    <h:form>
11      <h:panelGrid columns="2">
12        <h:outputText value="Opis: "></h:outputText>
13        <h:inputTextarea value="#{obavezaBean.obaveza.opis}">
14        </h:inputTextarea>
15        <h:outputText value="Adresa: "></h:outputText>
16        <h:inputText value="#{obavezaBean.obaveza.adresa}">
17        </h:inputText>
18        <h:outputText value="Datum: "></h:outputText>
19        <h:inputText converter="javax.faces.DateTime" value="#{obavezaBean.obaveza.datum}"
20          <f:convertDateTime pattern="dd.mm.yyyy"/>
21        </h:inputText>
22        <h:outputText value="Kategorija: "></h:outputText>
23        <h:selectOneMenu value="#{obavezaBean.obaveza.kategorija}">
24          <f:selectItem itemValue="poslovna obaveza" itemLabel="poslovna obaveza"></f:selectItem>
25          <f:selectItem itemValue="privatna obaveza" itemLabel="privatna obaveza"></f:selectItem>
26          <f:selectItem itemValue="ostalo" itemLabel="ostalo"></f:selectItem>
27        </h:selectOneMenu>
28        <h:outputText value=""></h:outputText>
29        <h:commandButton value="Unesi obavezu" action="#{obavezaBean.upisiObavezu}"></h:commandButton>
30      </h:panelGrid>
31    </h:form>
32    <h:outputText value="#{obavezaBean.poruka}"></h:outputText>
33  </ui:composition>
34 </html>

```

## body\_pregledObaveza.xhtml

```

1 <html xmlns="http://www.w3.org/1999/xhtml"
2   xmlns:ui="http://java.sun.com/jsf/facelets"
3   xmlns:h="http://java.sun.com/jsf/html"
4   xmlns:f="http://java.sun.com/jsf/core">
5   <h:head />
6   <ui:composition>
7     <hr />
8     <h:form>
9       <h:panelGrid columns="2">
10        <h:outputText value="Unesite datum za prikaz: "></h:outputText>
11        <h:inputText value="#{obavezaBean.datum}">
12        </h:inputText>
13        <h:outputText value=""></h:outputText>
14        <h:commandButton value="Prikaži obaveze" action="#{obavezaBean.ocitajObavezeIzBaze}"></h:commandButton>
15      </h:panelGrid>
16    </h:form>
17  </ui:composition>
18 </html>

```

## body\_obaveze.xhtml

```

1 <html xmlns="http://www.w3.org/1999/xhtml"
2     xmlns:ui="http://java.sun.com/jsf/facelets"
3     xmlns:h="http://java.sun.com/jsf/html"
4     xmlns:f="http://java.sun.com/jsf/core">
5 <h:head />
6 <ui:composition>
7     <h:form>
8         <b>Obaveze za trazení datum su:</b>
9         <h:dataTable value="#{obavezaBean.obaveze}" var="obaveza"
10             style="width: 100%; align: center">
11             <h:column>
12                 <f:facet name="header">ID obaveze</f:facet>
13                 <h:outputText value="#{obaveza.id}">
14                     </h:outputText>
15             </h:column>
16             <h:column>
17                 <f:facet name="header">Opis</f:facet>
18                 <h:outputText value="#{obaveza.opis}"></h:outputText>
19             </h:column>
20             <h:column>
21                 <f:facet name="header">Kategorija</f:facet>
22                 <h:outputText value="#{obaveza.kategorija}"></h:outputText>
23             </h:column>
24             <h:column>
25                 <f:facet name="header">Adresa korisnika</f:facet>
26                 <h:outputText value="#{obaveza.adresa}"></h:outputText>
27             </h:column>
28             <h:column>
29                 <f:facet name="header">Datum</f:facet>
30                 <h:outputText value="#{obaveza.datum}"></h:outputText>
31             </h:column>
32             <h:column>
33                 <f:facet name="header">Obrisi obavezu</f:facet>
34                 <h:commandLink value="Obrisi - #{obaveza.id}"
35                     action="#{obavezaBean.obrisiObavezu}">
36                     <f:param value="#{obaveza.id}" name="id" />
37                 </h:commandLink>
38             </h:column>
39         </h:dataTable>
40     </h:form>
41 </ui:composition>
42 </html>

```



## Obaveza.java

```
1 package net.etfbl.lab.dto;
2
3 import java.io.Serializable;
4
5
6 public class Obaveza implements Serializable {
7
8     private static final long serialVersionUID = -5152941347101008809L;
9     private int id;
10    private String opis;
11    private Date datum;
12    private String kategorija;
13    private String adresa;
14
15    public Obaveza() {
16        // TODO Auto-generated constructor stub
17    }
18
19    public Obaveza(int id, String opis, Date datum, String kategorija, String adresa) {
20        super();
21        this.id = id;
22        this.opis = opis;
23        this.datum = datum;
24        this.kategorija = kategorija;
25        this.adresa = adresa;
26    }
}
```

```
28 public int getId() {  
29     return id;  
30 }  
31  
32 public void setId(int id) {  
33     this.id = id;  
34 }  
35  
36 public String getOpis() {  
37     return opis;  
38 }  
39  
40 public void setOpis(String opis) {  
41     this.opis = opis;  
42 }  
43  
44 public Date getDatum() {  
45     return datum;  
46 }  
47  
48 public void setDatum(Date datum) {  
49     this.datum = datum;  
50 }  
51  
52 public String getKategorija() {  
53     return kategorija;  
54 }  
55  
56 public void setKategorija(String kategorija) {  
57     this.kategorija = kategorija;  
58 }  
59  
60 public String getAdresa() {  
61     return adresa;  
62 }  
63  
64 public void setAdresa(String adresa) {  
65     this.adresa = adresa;  
66 }  
67  
68 }  
69
```

## ConnectionPool.java

```
1 package net.etfbl.lab.dao;
2
3 import java.sql.*;
4
5
6 public class ConnectionPool {
7
8     public static ConnectionPool getConnectionPool() {
9         return connectionPool;
10    }
11
12    private static ConnectionPool connectionPool;
13
14    static {
15        ResourceBundle bundle =
16            PropertyResourceBundle.getBundle("net.etfbl.lab.dao.ConnectionPool");
17        String jdbcURL = bundle.getString("jdbcURL");
18        String username = bundle.getString("username");
19        String password = bundle.getString("password");
20        String driver = bundle.getString("driver");
21        int preconnectCount = 0;
22        int maxIdleConnections = 10;
23        int maxConnections = 10;
24        try {
25            Class.forName(driver);
26            preconnectCount = Integer.parseInt(
27                bundle.getString("preconnectCount"));
28            maxIdleConnections = Integer.parseInt(
29                bundle.getString("maxIdleConnections"));
30            maxConnections = Integer.parseInt(
31                bundle.getString("maxConnections"));
32        } catch (Exception ex) {
33            ex.printStackTrace();
34        }
35        try {
36            connectionPool = new ConnectionPool(
37                jdbcURL, username, password,
38                preconnectCount, maxIdleConnections,
39                maxConnections);
40        } catch (Exception ex) {
41            ex.printStackTrace();
42        }
43    }
44 }
```

```
45  protected ConnectionPool(String aJdbcURL, String aUsername,  
46      String aPassword, int aPreconnectCount,  
47      int aMaxIdleConnections,  
48      int aMaxConnections)  
49      throws ClassNotFoundException, SQLException {  
50  
51      freeConnections = new Vector<Connection>();  
52      usedConnections = new Vector<Connection>();  
53      jdbcURL = aJdbcURL;  
54      username = aUsername;  
55      password = aPassword;  
56      preconnectCount = aPreconnectCount;  
57      maxIdleConnections = aMaxIdleConnections;  
58      maxConnections = aMaxConnections;  
59  
60      for (int i = 0; i < preconnectCount; i++) {  
61          Connection conn = DriverManager.getConnection(  
62              jdbcURL, username, password);  
63          conn.setAutoCommit(true);  
64          freeConnections.addElement(conn);  
65      }  
66      connectCount = preconnectCount;  
67  }
```

```

69 public synchronized Connection checkOut()
70     throws SQLException {
71
72     Connection conn = null;
73     if (freeConnections.size() > 0) {
74         conn = (Connection)freeConnections.elementAt(0);
75         freeConnections.removeElementAt(0);
76         usedConnections.addElement(conn);
77     } else {
78         if (connectCount < maxConnections) {
79             conn = DriverManager.getConnection(
80                 jdbcURL, username, password);
81             usedConnections.addElement(conn);
82             connectCount++;
83         } else {
84             try {
85                 wait();
86                 conn = (Connection)freeConnections.elementAt(0);
87                 freeConnections.removeElementAt(0);
88                 usedConnections.addElement(conn);
89             } catch (InterruptedException ex) {
90                 ex.printStackTrace();
91             }
92         }
93     }
94     return conn;
95 }
96
97 public synchronized void checkIn(Connection aConn) {
98     if (aConn == null)
99         return;
100     if (usedConnections.removeElement(aConn)) {
101         freeConnections.addElement(aConn);
102         while (freeConnections.size() > maxIdleConnections) {
103             int lastOne = freeConnections.size() - 1;
104             Connection conn = (Connection)
105                 freeConnections.elementAt(lastOne);
106             try { conn.close(); } catch (SQLException ex) { }
107             freeConnections.removeElementAt(lastOne);
108         }
109         notify();
110     }
111 }
112
113 private String jdbcURL;
114 private String username;
115 private String password;
116 private int preconnectCount;
117 private int connectCount;
118 private int maxIdleConnections;
119 private int maxConnections;
120 private Vector<Connection> usedConnections;
121 private Vector<Connection> freeConnections;
122
123 }

```

## ObavezeDAO.java

```
1 package net.etfbl.lab.dao;
2
3 import java.sql.Connection;
4
5 public class ObavezeDAO {
6
7     private static final String INSERT = "INSERT INTO planer_obaveza.obaveza (opis, datum, kategorija, adresaKorisnika) VALUES (?, ?, ?, ?)";
8     private static final String SELECT = "SELECT * FROM planer_obaveza.obaveza WHERE datum=? order by datum";
9     private static final String DELETE = "DELETE FROM planer_obaveza.obaveza WHERE id=?";
10
11     public static boolean unesiObavezu(Obaveza obaveza) {
12         Connection conn = null;
13         try {
14             conn = ConnectionPool.getConnectionPool().checkout();
15             PreparedStatement stmt = conn.prepareStatement(INSERT);
16             stmt.setString(1, obaveza.getOpis());
17             stmt.setDate(2, new Date(obaveza.getDatum().getTime()));
18             stmt.setString(3, obaveza.getKategorija());
19             stmt.setString(4, obaveza.getAdresa());
20             stmt.executeUpdate();
21             stmt.close();
22             return true;
23         } catch (Exception ex) {
24             ex.printStackTrace(System.err);
25             return false;
26         } finally {
27             ConnectionPool.getConnectionPool().checkIn(conn);
28         }
29     }
30
31     public static ArrayList<Obaveza> ocitajObavezePoDatumu(String datum) {
32         ArrayList<Obaveza> obaveze = new ArrayList<>();
33         Connection conn = null;
34         try {
35             conn = ConnectionPool.getConnectionPool().checkout();
36             PreparedStatement stmt = conn.prepareStatement(SELECT);
37             stmt.setString(1, datum);
38             ResultSet rs = stmt.executeQuery();
39             while (rs.next()) {
40                 obaveze.add(
41                     new Obaveza(rs.getInt(1), rs.getString(2), rs.getDate(3), rs.getString(4), rs.getString(5)));
42             }
43             rs.close();
44             stmt.close();
45         } catch (Exception ex) {
46             ex.printStackTrace(System.err);
47         } finally {
48             ConnectionPool.getConnectionPool().checkIn(conn);
49         }
50         return obaveze;
51     }
52 }
```

```

37 public static ArrayList<Obaveza> ocitajObavezePoDatumu(String datum) {
38     ArrayList<Obaveza> obaveze = new ArrayList<>();
39     Connection conn = null;
40     try {
41         conn = ConnectionPool.getConnectionPool().checkout();
42         PreparedStatement stmt = conn.prepareStatement(SELECT);
43         stmt.setObject(1, datum);
44         ResultSet rs = stmt.executeQuery();
45         while (rs.next()) {
46             obaveze.add(
47                 new Obaveza(rs.getInt(1), rs.getString(2), rs.getDate(3), rs.getString(4), rs.getString(5)));
48             rs.close();
49             stmt.close();
50         } catch (Exception ex) {
51             ex.printStackTrace(System.err);
52         } finally {
53             ConnectionPool.getConnectionPool().checkIn(conn);
54         }
55     }
56     return obaveze;
57 }
58
59 public static void obrisiObavezu(int id) {
60     Connection conn = null;
61     try {
62         conn = ConnectionPool.getConnectionPool().checkout();
63         PreparedStatement stmt = conn.prepareStatement(DELETE);
64         stmt.setInt(1, id);
65         stmt.executeUpdate();
66         stmt.close();
67     } catch (Exception ex) {
68         ex.printStackTrace(System.err);
69     } finally {
70         ConnectionPool.getConnectionPool().checkIn(conn);
71     }
72 }
73 }
74

```

## ObavezeBean.java

```
1 package net.etfbl.lab.beans;
2
3 import java.io.Serializable;
4
5
6
7
8
9
10
11
12
13
14 @ManagedBean(name = "obavezaBean")
15 @SessionScoped
16 public class ObavezaBean implements Serializable {
17
18     public ObavezaBean() {
19     }
20
21     private static final long serialVersionUID = -8606687324815905107L;
22     private ArrayList<Obaveza> obaveze = new ArrayList<Obaveza>();
23     private Obaveza obaveza = new Obaveza();
24     private String datum = new String();
25     private String poruka;
26
27     public ArrayList<Obaveza> getObaveze() {
28         return obaveze;
29     }
30
31     public void setObaveze(ArrayList<Obaveza> obaveze) {
32         this.obaveze = obaveze;
33     }
34
35     public Obaveza getObaveza() {
36         return obaveza;
37     }
38
39     public void setObaveza(Obaveza obaveza) {
40         this.obaveza = obaveza;
41     }
42
43     public String getDatum() {
44         return datum;
45     }
46
47     public void setDatum(String datum) {
48         this.datum = datum;
49     }
50
51     public String getPoruka() {
52         return poruka;
53     }
54
55     public void setPoruka(String poruka) {
56         this.poruka = poruka;
57     }
58 }
```



```
59 public String ocitajObavezeIzBaze() {
60     obaveze = ObavezeDAO.ocitajObavezePoDatumu(datum);
61     return "obaveze.xhtml?faces-redirect=true";
62 }
63
64 public String upisiObavezu() {
65     boolean uspjeh = ObavezeDAO.unesiObavezu(obaveza);
66     if (uspjeh) {
67         this.obaveza = new Obaveza();
68         this.poruka = "Unos u bazu je uspjesno obavljen!";
69     }
70     return null;
71 }
72
73 public String obrisiObavezu() {
74     Map<String, String> reqMap = FacesContext.getCurrentInstance().getExternalContext().getRequestParameterMap();
75     if (reqMap.containsKey("id")) {
76         int z = Integer.parseInt(reqMap.get("id"));
77         for (Obaveza o : obaveze) {
78             if (o.getId() == z) {
79                 obaveze.remove(o);
80                 ObavezeDAO.obrisiObavezu(z);
81                 break;
82             }
83         }
84     }
85     return null;
86 }
87
88 }
```