

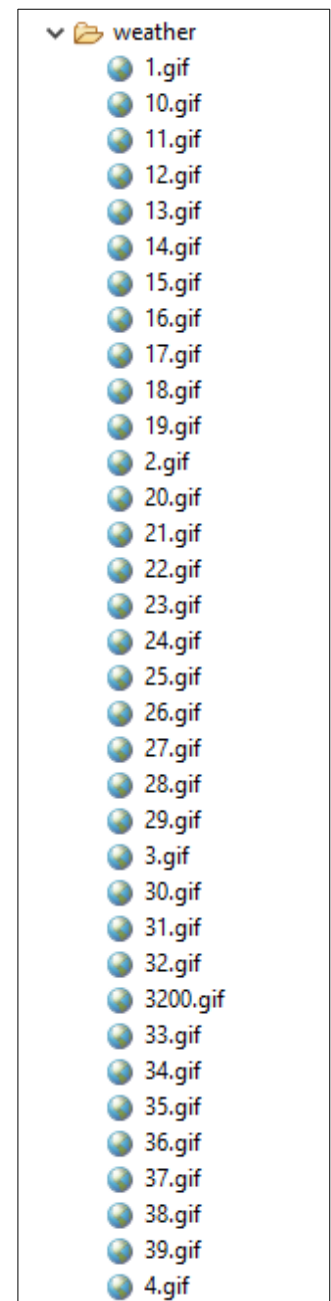
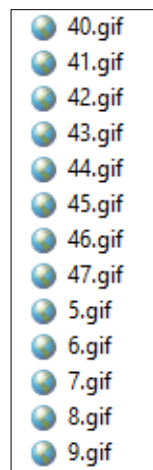
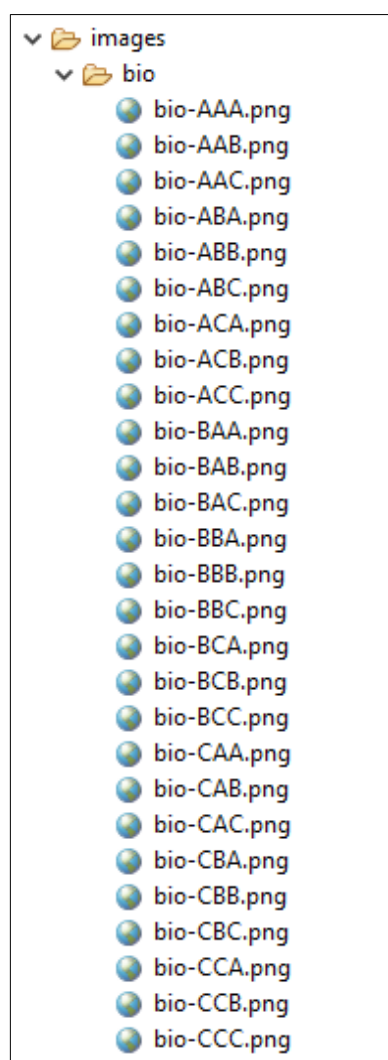
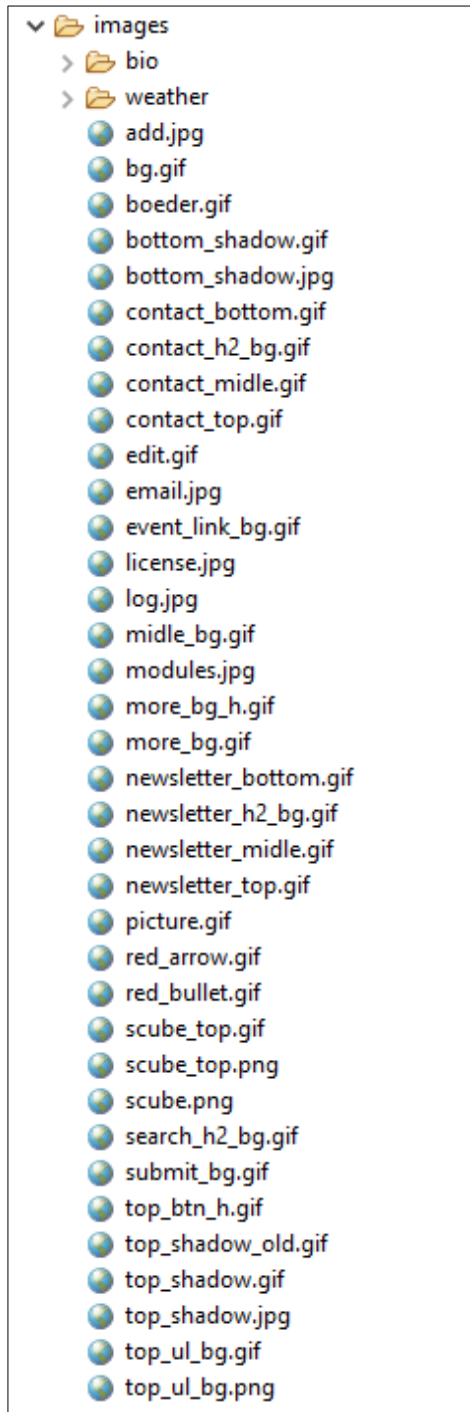
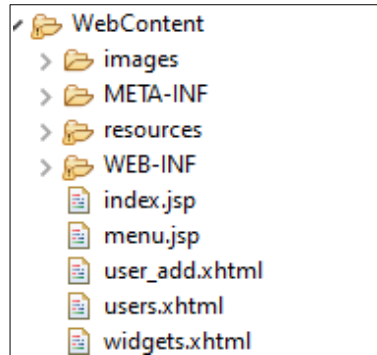
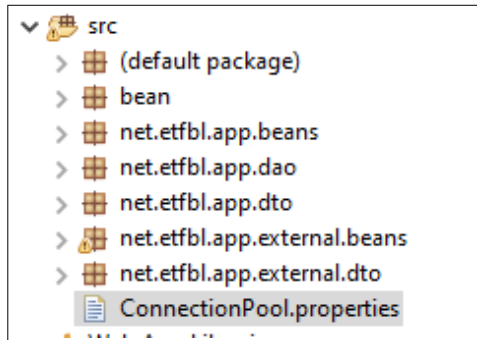
[images\bio\bio-AAA.png](#)
[images\bio\bio-AAB.png](#)
[images\bio\bio-AAC.png](#)
[images\bio\bio-ABA.png](#)
[images\bio\bio-ABB.png](#)
[images\bio\bio-ABC.png](#)
[images\bio\bio-ACA.png](#)
[images\bio\bio-ACB.png](#)
[images\bio\bio-ACC.png](#)
[images\bio\bio-BAA.png](#)
[images\bio\bio-BAB.png](#)
[images\bio\bio-BAC.png](#)
[images\bio\bio-BBA.png](#)
[images\bio\bio-BBB.png](#)
[images\bio\bio-BBC.png](#)
[images\bio\bio-BCA.png](#)
[images\bio\bio-BCB.png](#)
[images\bio\bio-BCC.png](#)
[images\bio\bio-CAA.png](#)
[images\bio\bio-CAB.png](#)
[images\bio\bio-CAC.png](#)
[images\bio\bio-CBA.png](#)
[images\bio\bio-CBB.png](#)
[images\bio\bio-CBC.png](#)
[images\bio\bio-CCA.png](#)
[images\bio\bio-CCB.png](#)
[images\bio\bio-CCC.png](#)

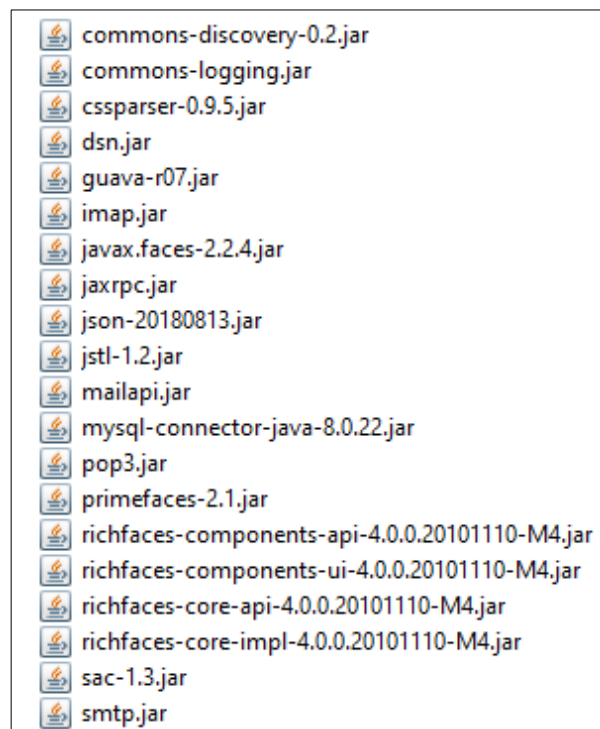
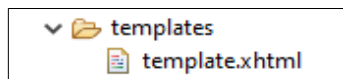
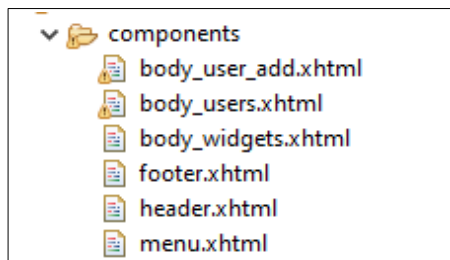
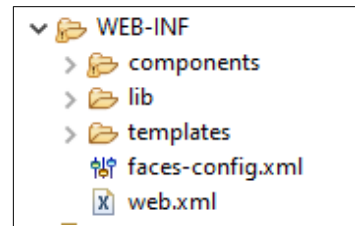
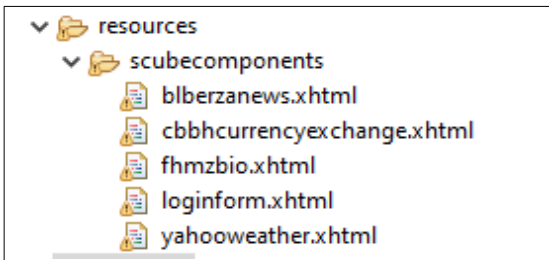
[images\boeder.gif](#)
[images\bottom shadow.gif](#)
[images\bottom shadow.jpg](#)
[images\contact bottom.gif](#)
[images\contact h2 bg.gif](#)
[images\contact midle.gif](#)
[images\contact top.gif](#)
[images\edit.gif](#)
[images\email.jpg](#)
[images\event link bg.gif](#)
[images\license.jpg](#)
[images\log.jpg](#)
[images\midle bg.gif](#)
[images\modules.jpg](#)
[images\more bg.gif](#)
[images\more bg h.gif](#)
[images\newsletter bottom.gif](#)
[images\newsletter h2 bg.gif](#)
[images\newsletter midle.gif](#)
[images\newsletter top.gif](#)
[images\picture.gif](#)
[images\red arrow.gif](#)
[images\red bullet.gif](#)
[images\scube.png](#)
[images\scube top.gif](#)
[images\scube top.png](#)
[images\search h2 bg.gif](#)
[images\submit bg.gif](#)
[images\top btn h.gif](#)
[images\top shadow.gif](#)
[images\top shadow.jpg](#)
[images\top shadow old.gif](#)
[images\top ul bg.gif](#)
[images\top ul bg.png](#)
[images\weather\svn\all-wcprops](#)
[images\weather\svn\entries](#)

[images\weather\1.gif](#)
[images\weather\10.gif](#)
[images\weather\11.gif](#)
[images\weather\12.gif](#)
[images\weather\13.gif](#)
[images\weather\14.gif](#)
[images\weather\15.gif](#)
[images\weather\16.gif](#)
[images\weather\17.gif](#)
[images\weather\18.gif](#)
[images\weather\19.gif](#)
[images\weather\2.gif](#)
[images\weather\20.gif](#)
[images\weather\21.gif](#)
[images\weather\22.gif](#)
[images\weather\23.gif](#)
[images\weather\24.gif](#)
[images\weather\25.gif](#)
[images\weather\26.gif](#)
[images\weather\27.gif](#)
[images\weather\28.gif](#)
[images\weather\29.gif](#)
[images\weather\3.gif](#)
[images\weather\30.gif](#)
[images\weather\31.gif](#)
[images\weather\32.gif](#)
[images\weather\3200.gif](#)
[images\weather\33.gif](#)
[images\weather\34.gif](#)
[images\weather\35.gif](#)
[images\weather\36.gif](#)
[images\weather\37.gif](#)
[images\weather\38.gif](#)
[images\weather\39.gif](#)
[images\weather\4.gif](#)
[images\weather\40.gif](#)
[images\weather\41.gif](#)
[images\weather\42.gif](#)
[images\weather\43.gif](#)
[images\weather\44.gif](#)
[images\weather\45.gif](#)
[images\weather\46.gif](#)
[images\weather\47.gif](#)
[images\weather\5.gif](#)
[images\weather\6.gif](#)
[images\weather\7.gif](#)
[images\weather\8.gif](#)
[images\weather\9.gif](#)

[META-INF\MANIFEST.MF](#)
[resources\scubecomponents\blberzanews.xhtml](#)
[resources\scubecomponents\cbbhcurrenecyexchange.xhtml](#)
[resources\scubecomponents\fhmzbio.xhtml](#)
[resources\scubecomponents\loginform.xhtml](#)
[resources\scubecomponents\yahooweather.xhtml](#)
[user_add.xhtml](#)
[users.xhtml](#)
[WEB-INF\components\body_user_add.xhtml](#)
[WEB-INF\components\body_users.xhtml](#)
[WEB-INF\components\body_widgets.xhtml](#)
[WEB-INF\components\footer.xhtml](#)
[WEB-INF\components\header.xhtml](#)
[WEB-INF\components\menu.xhtml](#)
[WEB-INF\faces-config.xml](#)
[WEB-INF\lib\commons-discovery-0.2.jar](#)
[WEB-INF\lib\commons-logging.jar](#)
[WEB-INF\lib\cssparser-0.9.5.jar](#)
[WEB-INF\lib\dsn.jar](#)
[WEB-INF\lib\guava-r07.jar](#)
[WEB-INF\lib\imap.jar](#)
[WEB-INF\lib\javax.faces-2.2.4.jar](#)
[WEB-INF\lib\jaxrpc.jar](#)
[WEB-INF\lib\json-20180813.jar](#)
[WEB-INF\lib\jstl-1.2.jar](#)
[WEB-INF\lib\mailapi.jar](#)
[WEB-INF\lib\mysql-connector-java-8.0.22.jar](#)
[WEB-INF\lib\pop3.jar](#)
[WEB-INF\lib\primefaces-2.1.jar](#)
[WEB-INF\lib\richfaces-components-api-4.0.0.20101110-M4.jar](#)
[WEB-INF\lib\richfaces-components-ui-4.0.0.20101110-M4.jar](#)
[WEB-INF\lib\richfaces-core-api-4.0.0.20101110-M4.jar](#)
[WEB-INF\lib\richfaces-core-impl-4.0.0.20101110-M4.jar](#)
[WEB-INF\lib\sac-1.3.jar](#)
[WEB-INF\lib\smtp.jar](#)
[WEB-INF\templates\template.xhtml](#)
[WEB-INF\web.xml](#)
[widgets.xhtml](#)

Пребројано је 305 ставки.





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" xsi:sch
3 <display-name>047_IP_Predavanja ETF_IP_JSF2</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>faces/users.xhtml</welcome-file>
11 </welcome-file-list>
12 <servlet>
13 <servlet-name>Faces Servlet</servlet-name>
14 <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
15 <load-on-startup>1</load-on-startup>
16 </servlet>
17 <servlet-mapping>
18 <servlet-name>Faces Servlet</servlet-name>
19 <url-pattern>/faces/*</url-pattern>
20 </servlet-mapping>
21 <context-param>
22 <description>State saving method: 'client' or 'server' (=default). See JSF Specification 2.5.2</description>
23 <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
24 <param-value>client</param-value>
25 </context-param>
26 <context-param>
27 <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
28 <param-value>resources.application</param-value>
29 </context-param>
30 <listener>
31 <listener-class>com.sun.faces.config.ConfigureListener</listener-class>
32 </listener>
33 <servlet-mapping>
34 <servlet-name>Faces Servlet</servlet-name>
35 <url-pattern>*.jsf</url-pattern>
36 </servlet-mapping>
37 <servlet-mapping>
38 <servlet-name>Faces Servlet</servlet-name>
39 <url-pattern>*.faces</url-pattern>
40 </servlet-mapping>
41 </web-app>
```

ConnectionPool.properties

```
1 driver=com.mysql.jdbc.Driver
2 jdbcURL=jdbc:mysql://127.0.0.1:3306/etf?useUnicode=true&characterEncoding=UTF-8
3 username=root
4 password=root
5 preconnectCount=2
6 maxIdleConnections=25
7 maxConnections=25
```

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" version="2.2">
7   <managed-bean>
8     <managed-bean-name>userBean</managed-bean-name>
9     <managed-bean-class>net.etfb1.app.beans.UserBean</managed-bean-class>
10    <managed-bean-scope>session</managed-bean-scope>
11  </managed-bean>
12
13 </faces-config>
```

menu.jsp

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta charset="UTF-8">
6     <title>Books Application API Menu</title>
7   </head>
8   <body>
9     <h3>BOOKS APP MENU</h3>
10    <dl>
11      <dd><a href='${pageContext.request.contextPath}/faces/user_add.xhtml'>Add user</a></dd>
12      <dd><a href='${pageContext.request.contextPath}/faces/users.xhtml'>User list</a></dd>
13      <dd><a href='${pageContext.request.contextPath}/faces/widgets.xhtml'>Widget pages</a></dd>
14    </dl>
15    <br>
16    <dl>
17      <dd><a href='${pageContext.request.contextPath}/user_add.jsf'>Add user</a></dd>
18      <dd><a href='${pageContext.request.contextPath}/users.jsf'>User list</a></dd>
19      <dd><a href='${pageContext.request.contextPath}/widgets.jsf'>Widget pages</a></dd>
20    </dl>
21    <br>
22    <dl>
23      <dd><a href='${pageContext.request.contextPath}/user_add.faces'>Add user</a></dd>
24      <dd><a href='${pageContext.request.contextPath}/users.faces'>User list</a></dd>
25      <dd><a href='${pageContext.request.contextPath}/widgets.faces'>Widget pages</a></dd>
26    </dl>
27    <dl>
28      <dd><a href='${pageContext.request.contextPath}/menu.jsp'>Menu</a></dd>
29      <dd><a href='${pageContext.request.contextPath}/index.jsp'>Menu</a></dd>
30    </dl>
31  </body>
32 </html>

```

- ▼ (default package)
 - > Test.java
- ▼ bean
 - > FileListingBean.java
- ▼ net.etfbl.app.beans
 - > UserBean.java
- ▼ net.etfbl.app.dao
 - > ConnectionPool.java
 - > DAOException.java
 - > DAOUtil.java
 - > UserDao.java
- ▼ net.etfbl.app.dto
 - > User.java
- ▼ net.etfbl.app.external.beans
 - > BLBerzaBean.java
 - > CBBHBean.java
 - > FHMZBioBean.java
 - ▼ YahooWeatherBean.java
 - > YahooWeatherBean
- ▼ net.etfbl.app.external.dto
 - > Bio.java
 - > BLBerzaNews.java
 - > Currency.java
 - > WeatherDay.java
 - ConnectionPool.properties

User.java

```
1 package net.etfbl.app.dto;
2
3 public class User {
4     private int user_id;
5     private String username;
6     private String password;
7     public User(){
8
9     }
10    public User(int user_id, String username, String password) {
11        super();
12        this.user_id = user_id;
13        this.username = username;
14        this.password = password;
15    }
16    public int getUser_id() {
17        return user_id;
18    }
19    public void setUser_id(int user_id) {
20        this.user_id = user_id;
21    }
22    public String getUsername() {
23        return username;
24    }
25    public void setUsername(String username) {
26        this.username = username;
27    }
28    public String getPassword() {
29        return password;
30    }
31    public void setPassword(String password) {
32        this.password = password;
33    }
34 }
35
36
```

DAOUtil.java

```
package net.etfbl.app.dao;

import java.sql.*;

public final class DAOUtil {

    public static PreparedStatement prepareStatement(Connection connection,
        String sql, boolean returnGeneratedKeys, Object... values)
        throws SQLException {
        PreparedStatement preparedStatement = connection.prepareStatement(sql,
            returnGeneratedKeys ? Statement.RETURN_GENERATED_KEYS
            : Statement.NO_GENERATED_KEYS);
        setValues(preparedStatement, values);
        return preparedStatement;
    }

    public static void setValues(PreparedStatement preparedStatement,
        Object... values) throws SQLException {
        for (int i = 0; i < values.length; i++) {
            preparedStatement.setObject(i + 1, values[i]);
        }
    }
}
```

DAOException.java

```
package net.etfbl.app.dao;

@SuppressWarnings("serial")
public class DAOException extends Exception {

    public DAOException(String message) {
        super(message);
    }

    public DAOException(Throwable cause) {
        super(cause);
    }

    public DAOException(String message, Throwable cause) {
        super(message, cause);
    }
}
```

ConnectionPool.java

```
1 package net.etfbl.app.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("ConnectionPool");
17        String driver = bundle.getString("driver");
18        String jdbcURL = bundle.getString("jdbcURL");
19        String username = bundle.getString("username");
20        String password = bundle.getString("password");
21        int preconnectCount = 0;
22        int maxIdleConnections = 10;
23        int maxConnections = 10;
24        try {
25            preconnectCount = Integer.parseInt(
26                bundle.getString("preconnectCount"));
27            maxIdleConnections = Integer.parseInt(
28                bundle.getString("maxIdleConnections"));
29            maxConnections = Integer.parseInt(
30                bundle.getString("maxConnections"));
31        } catch (Exception ex) {
32            ex.printStackTrace();
33        }
34        try {
35            connectionPool = new ConnectionPool(driver,
36                jdbcURL, username, password,
37                preconnectCount, maxIdleConnections,
38                maxConnections);
39        } catch (Exception ex) {
40            ex.printStackTrace();
41        }
42    }
43 }
```

```

protected ConnectionPool(String aDriver,
    String aJdbcURL, String aUsername,
    String aPassword, int aPreconnectCount,
    int aMaxIdleConnections,
    int aMaxConnections)
    throws ClassNotFoundException, SQLException {

    freeConnections = new Vector<Connection>();
    usedConnections = new Vector<Connection>();
    driver = aDriver;
    jdbcURL = aJdbcURL;
    username = aUsername;
    password = aPassword;
    preconnectCount = aPreconnectCount;
    maxIdleConnections = aMaxIdleConnections;
    maxConnections = aMaxConnections;

    Class.forName(driver);
    for (int i = 0; i < preconnectCount; i++) {
        Connection conn = DriverManager.getConnection(
            jdbcURL, username, password);
        conn.setAutoCommit(true);
        freeConnections.addElement(conn);
    }
    connectCount = preconnectCount;
}

```

```

115     private String driver;
116
117     private String jdbcURL;
118
119     private String username;
120
121     private String password;
122
123     private int preconnectCount;
124
125     private int connectCount;
126
127     private int maxIdleConnections;
128
129     private int maxConnections;
130
131     private Vector<Connection> usedConnections;
132
133     private Vector<Connection> freeConnections;

```



```

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

```

UserDAO.java

```
1 package net.etfbl.app.dao;
2
3 import java.sql.Connection;
4
5
6
7
8
9
10
11 public class UserDAO {
12     private static ConnectionPool connectionPool = ConnectionPool.getConnectionPool();
13     private static final String SQL_INSERT = "INSERT INTO user (id, username, password) VALUES (?, ?, ?)";
14     private static final String SQL_SELECT_ALL = "SELECT * FROM user";
15     private static final String SQL_DELETE_BY_ID = "DELETE FROM user WHERE id = ?";
16
17     public static void insert(User user) throws DAOException {
18         Connection connection = null;
19         ResultSet generatedKeys = null;
20         Object values[] = { user.getUser_id(), user.getUsername(), user.getPassword() };
21         try {
22             connection = connectionPool.checkOut();
23             PreparedStatement pstmt = DAOUtil.prepareStatement(connection,
24                 SQL_INSERT, true, values);
25             int affectedRows = pstmt.executeUpdate();
26             if (affectedRows == 0)
27                 throw new DAOException(
28                     "Inserting user failed, zero rows affected.");
29             generatedKeys = pstmt.getGeneratedKeys();
30             if (generatedKeys.next())
31                 user.setUser_id(generatedKeys.getInt(1));
32             pstmt.close();
33         } catch (SQLException e) {
34             e.printStackTrace();
35         } finally {
36             connectionPool.checkIn(connection);
37         }
38     }
39
40     public static void delete(User user) throws DAOException {
41         Connection connection = null;
42         Object values[] = { user.getUser_id() };
43         try {
44             connection = connectionPool.checkOut();
45             PreparedStatement pstmt = DAOUtil.prepareStatement(connection,
46                 SQL_DELETE_BY_ID, true, values);
47             int affectedRows = pstmt.executeUpdate();
48             if (affectedRows == 0)
49                 throw new DAOException(
50                     "Deleting user failed, zero rows affected.");
51             pstmt.close();
52         } catch (SQLException e) {
53             e.printStackTrace();
54         } finally {
55             connectionPool.checkIn(connection);
56         }
57     }
58 }
```

```

40⊖ public static void delete(User user) throws DAOException {
41     Connection connection = null;
42     Object values[] = { user.getUser_id() };
43     try {
44         connection = connectionPool.checkOut();
45         PreparedStatement pstmt = DAOUtil.prepareStatement(connection,
46             SQL_DELETE_BY_ID, true, values);
47         int affectedRows = pstmt.executeUpdate();
48         if (affectedRows == 0)
49             throw new DAOException(
50                 "Deleting user failed, zero rows affected.");
51         pstmt.close();
52     } catch (SQLException e) {
53         e.printStackTrace();
54     } finally {
55         connectionPool.checkIn(connection);
56     }
57 }
58⊖ public static ArrayList<User> selectAll() {
59     ArrayList<User> users = new ArrayList<User>();
60     Connection connection = null;
61     ResultSet rs = null;
62     Object values[] = { };
63     try {
64         connection = connectionPool.checkOut();
65         PreparedStatement pstmt = DAOUtil.prepareStatement(connection,
66             SQL_SELECT_ALL, false, values);
67         rs = pstmt.executeQuery();
68         while (rs.next())
69             users.add(new User(rs.getInt("id"), rs.getString("username"), rs.getString("password")));
70         pstmt.close();
71     } catch (SQLException exp) {
72         exp.printStackTrace();
73     } finally {
74         connectionPool.checkIn(connection);
75     }
76     return users;
77 }
78 }
79

```

UserBean.java

```
1 package net.etfbl.app.beans;
2
3+ import java.util.ArrayList;
4
5
6
7
8
9 public class UserBean {
10     private User selectedUser;
11     private User newUser = new User();
12
13- public ArrayList<User> getAllUsers(){
14     return UserDAO.selectAll();
15 }
16
17- public String delete(){
18     System.out.println("selectedUser.getUser_id(): " + selectedUser.getUser_id());
19     try {
20         UserDAO.delete(selectedUser);
21     } catch (DAOException e) {
22         e.printStackTrace();
23     }
24     return null;
25 }
26
27- public String add(){
28     try {
29         UserDAO.insert(newUser);
30         newUser = new User();
31     } catch (DAOException e) {
32         e.printStackTrace();
33     }
34     return "users";
35 }
36
37 // getters and setters
38- public User getSelectedUser() {
39     return selectedUser;
40 }
41
42- public void setSelectedUser(User selectedUser) {
43     this.selectedUser = selectedUser;
44 }
45
46- public User getNewUser() {
47     return newUser;
48 }
49
50- public void setNewUser(User newUser) {
51     this.newUser = newUser;
52 }
53
54- public String login(){
55     return "true";
56 }
57 }
58
```


- ▼ net.etfbl.app.external.beans
 - > BLBerzaBean.java
 - > CBBHBean.java
 - > FHMZBioBean.java
 - > YahooWeatherBean.java
- ▼ net.etfbl.app.external.dto
 - > Bio.java
 - > BLBerzaNews.java
 - > Currency.java
 - > WeatherDay.java

Bio.java

```

1 package net.etfbl.app.external.dto;
2
3 import java.text.SimpleDateFormat;
4
5
6 public class Bio {
7     private Date date;
8     private String img;
9     private String forecast;
10    private String formattedDate;
11
12    public Date getDate() {
13        return date;
14    }
15    public void setDate(Date date) {
16        SimpleDateFormat sdf = new SimpleDateFormat("dd.MM.yyyy.");
17        formattedDate = sdf.format(date);
18        this.date = date;
19    }
20    public String getImg() {
21        return img;
22    }
23    public void setImg(String img) {
24        this.img = img;
25    }
26    public String getForecast() {
27        return forecast;
28    }
29    public void setForecast(String forecast) {
30        this.forecast = forecast;
31    }
32    public void setFormattedDate(String formattedDate) {
33        this.formattedDate = formattedDate;
34    }
35    public String getFormattedDate() {
36        return formattedDate;
37    }
38 }
39

```

BLBerzaNews.java

```
1 package net.etfbl.app.external.dto;
2
3 import java.util.Date;
4
5 public class BLBerzaNews {
6     private String title;
7     private String link;
8     private String description;
9     private Date pubDate;
10
11     public String getTitle() {
12         return title;
13     }
14     public void setTitle(String title) {
15         this.title = title;
16     }
17     public String getLink() {
18         return link;
19     }
20     public void setLink(String link) {
21         this.link = link;
22     }
23     public String getDescription() {
24         return description;
25     }
26     public void setDescription(String description) {
27         this.description = description;
28     }
29     public Date getPubDate() {
30         return pubDate;
31     }
32     public void setPubDate(Date pubDate) {
33         this.pubDate = pubDate;
34     }
35 }
36
```

WeatherDay.java

```
1 package net.etfbl.app.external.dto;
2
3 import java.util.Date;
4
5 public class WeatherDay {
6     private String code = "3200";
7     private String high;
8     private String low;
9     private String temp;
10    private Date date;
11    private String windSpeed;
12    private String windDirection;
13    private String atmosphereHumidity;
14    private String atmospherePressure;
15    private String atmosphereVisibility;
16    private String astronomySunrise;
17    private String astronomySunset;
```

```
20 public String getCode() {
21     return code;
22 }
23 public void setCode(String code) {
24     this.code = code;
25 }
26 public String getHigh() {
27     return high;
28 }
29 public void setHigh(String high) {
30     this.high = high;
31 }
32 public String getLow() {
33     return low;
34 }
35 public void setLow(String low) {
36     this.low = low;
37 }
38 public Date getDate() {
39     return date;
40 }
41 public void setDate(Date date) {
42     this.date = date;
43 }
44 public String getTemp() {
45     return temp;
46 }
47 public void setTemp(String temp) {
48     this.temp = temp;
49 }
50 public String getWindSpeed() {
51     return windSpeed;
52 }
53 public void setWindSpeed(String windSpeed) {
54     this.windSpeed = windSpeed;
55 }
56 public String getWindDirection() {
57     return windDirection;
58 }
59 public void setWindDirection(String windDirection) {
60     this.windDirection = windDirection;
61 }
62 public String getAtmosphereHumidity() {
63     return atmosphereHumidity;
64 }
```



```

41- public void setDate(Date date) {
42-     this.date = date;
43- }
44- public String getTemp() {
45-     return temp;
46- }
47- public void setTemp(String temp) {
48-     this.temp = temp;
49- }
50- public String getWindSpeed() {
51-     return windSpeed;
52- }
53- public void setWindSpeed(String windSpeed) {
54-     this.windSpeed = windSpeed;
55- }
56- public String getWindDirection() {
57-     return windDirection;
58- }
59- public void setWindDirection(String windDirection) {
60-     this.windDirection = windDirection;
61- }
62- public String getAtmosphereHumidity() {
63-     return atmosphereHumidity;
64- }
65- public void setAtmosphereHumidity(String atmosphereHumidity) {
66-     this.atmosphereHumidity = atmosphereHumidity;
67- }
68- public String getAtmospherePressure() {
69-     return atmospherePressure;
70- }
71- public void setAtmospherePressure(String atmospherePressure) {
72-     this.atmospherePressure = atmospherePressure;
73- }
74- public String getAtmosphereVisibility() {
75-     return atmosphereVisibility;
76- }
77- public void setAtmosphereVisibility(String atmosphereVisibility) {
78-     this.atmosphereVisibility = atmosphereVisibility;
79- }
80- public String getAstronomySunrise() {
81-     return astronomySunrise;
82- }
83- public void setAstronomySunrise(String astronomySunrise) {
84-     this.astronomySunrise = astronomySunrise;
85- }
86- public String getAstronomySunset() {
87-     return astronomySunset;
88- }
89- public void setAstronomySunset(String astronomySunset) {
90-     this.astronomySunset = astronomySunset;
91- }
92- }
93-

```

Currency.java

```
1 package net.etfbl.app.external.dto;
2
3 public class Currency {
4     private int units;
5     private double middleValue;
6     private String currency;
7
8     public int getUnits() {
9         return units;
10    }
11    public void setUnits(int units) {
12        this.units = units;
13    }
14    public double getMiddleValue() {
15        return middleValue;
16    }
17    public void setMiddleValue(double middleValue) {
18        this.middleValue = middleValue;
19    }
20    public String getCurrency() {
21        return currency;
22    }
23    public void setCurrency(String currency) {
24        this.currency = currency;
25    }
26
27 }
```

BLBerzaBean.java

```
1 package net.etfbl.app.external.beans;
2
3 import java.io.IOException;
4 import java.io.InputStream;
5 import java.net.MalformedURLException;
6 import java.net.URL;
7 import java.text.ParseException;
8 import java.text.SimpleDateFormat;
9 import java.util.ArrayList;
10 import java.util.Date;
11 import java.util.List;
12 import java.util.Timer;
13 import java.util.TimerTask;
14
15 import javax.faces.bean.ApplicationScoped;
16 import javax.faces.bean.ManagedBean;
17 import javax.xml.parsers.ParserConfigurationException;
18 import javax.xml.parsers.SAXParser;
19 import javax.xml.parsers.SAXParserFactory;
20
21 import net.etfbl.app.external.dto.BLBerzaNews;
22
23 import org.xml.sax.Attributes;
24 import org.xml.sax.SAXException;
25 import org.xml.sax.helpers.DefaultHandler;
26
27 @ManagedBean(eager=true)
28 @ApplicationScoped
29 public class BLBerzaBean {
30     private ArrayList<BLBerzaNews> blBerzaNewsListSR = new ArrayList<BLBerzaNews>();
31
32     public BLBerzaBean() {
33         Timer timer = new Timer();
34         timer.scheduleAtFixedRate(new BLBerzaServiceTimerTask("http://www.blberza.com/Pages/NewsRss.aspx?Channel=berza&LangId=1", blBerzaNewsListSR),
35
36
```

```
    , blBerzaNewsListSR), new Date(), 900000);
```

```
class BLBerzaServiceTimerTask extends TimerTask {
    private String urlAddress;
    private ArrayList<BLBerzaNews> container;
    public BLBerzaServiceTimerTask(String url, ArrayList<BLBerzaNews> container) {
        this.urlAddress = url;
        this.container = container;
    }
    @Override
    public void run() {
        try {
            URL url = new URL(urlAddress);
            InputStream input = url.openStream();
            SAXParserFactory factory = SAXParserFactory.newInstance();
            factory.setNamespaceAware(false);
            SAXParser parser = factory.newSAXParser();
            parser.parse(input, new BLBerzaHandler(container));
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } catch (ParserConfigurationException e) {
            e.printStackTrace();
        } catch (SAXException e) {
            e.printStackTrace();
        }
    }
}
```

```

65 private class BLBerzaHandler extends DefaultHandler {
66     private boolean descriptionTag = false;
67     private boolean linkTag = false;
68     private boolean pubDateTag = false;
69     private boolean titleTag = false;
70     private BLBerzaNews blBerzaNews;
71     private ArrayList<BLBerzaNews> container;
72     private SimpleDateFormat sdf = new SimpleDateFormat("EEE, dd MMM yyyy HH:mm:ss +0200");
73     private SimpleDateFormat sdf2 = new SimpleDateFormat("EEE, dd MMM yyyy HH:mm:ss +0100");
74
75 public BLBerzaHandler(ArrayList<BLBerzaNews> container) {
76     this.container = container;
77     this.container.clear();
78 }
79
80 public void startElement(String uri, String localName, String qName,
81     Attributes attributes) throws SAXException {
82
83     if(qName.equals("item"))
84         blBerzaNews = new BLBerzaNews();
85     else if (qName.equals("description"))
86         descriptionTag = true;
87     else if (qName.equals("link"))
88         linkTag = true;
89     else if (qName.equals("pubDate"))
90         pubDateTag = true;
91     else if (qName.equals("title"))
92         titleTag = true;
93
94     super.startElement(uri, localName, qName, attributes);
95 }
96
97 public void endElement(String uri, String localName, String qName)
98     throws SAXException {
99
100     if(qName.equals("item"))
101         container.add(blBerzaNews);
102     else if (qName.equals("description"))
103         descriptionTag = false;
104     else if (qName.equals("link"))
105         linkTag = false;
106     else if (qName.equals("pubDate"))
107         pubDateTag = false;
108     else if (qName.equals("title"))
109         titleTag = false;
110 }
111

```



```

112 public void characters(char ch[], int start, int length)
113     throws SAXException {
114
115     if(descriptionTag){
116         String description = new String(ch, start, length);
117         blBerzaNews.setDescription(description);
118     }
119     if(linkTag){
120         String link = new String(ch, start, length);
121         blBerzaNews.setLink(link);
122     }
123     if(pubDateTag){
124         String pubDate = new String(ch, start, length);
125         Date pDate = null;
126         try {
127             pDate = sdf.parse(pubDate);
128         } catch (ParseException e) {
129             try {
130                 pDate = sdf2.parse(pubDate);
131             } catch (ParseException e1) {
132                 e1.printStackTrace();
133             }
134         }
135         blBerzaNews.setPubDate(pDate);
136     }
137     if(titleTag){
138         String title = new String(ch, start, length);
139         blBerzaNews.setTitle(title);
140     }
141 }
142
143
144 public List<BLBerzaNews> getBLBerzaNewsList() {
145     if(blBerzaNewsListSR.size()>5)
146         return blBerzaNewsListSR.subList(0, 5);
147     else
148         return blBerzaNewsListSR;
149 }
150
151
152 }
153

```

FHMZBioBean.java

```

1 package net.etfbl.app.external.beans;
2
3 import java.io.BufferedReader;
4
5
6
7
8
9
10
11
12
13
14 @ManagedBean(name="fhmzBioBean", eager=true)
15 @ApplicationScoped
16 public class FHMZBioBean {
17     private Bio day1 = new Bio();
18     private Bio day2 = new Bio();
19
20     public FHMZBioBean() {
21         Timer timer = new Timer();
22         timer.scheduleAtFixedRate(new FHMZBioServiceTimerTask(), new Date(), 900000);
23     }
24
25     public Bio getDay1() {
26         return day1;
27     }
28
29     public Bio getDay2() {
30         return day2;
31     }
32
33     private class FHMZBioServiceTimerTask extends TimerTask {
34         @Override
35         public void run() {
36             FMHZBioParser fhmzBioParser = new FMHZBioParser();
37             fhmzBioParser.parse();
38         }
39     }
40
41     private class FMHZBioParser {
42         HTMLParser kit;
43         HTMLEditorKit.Parser parser;
44         BufferedReader reader;
45
46         public FMHZBioParser(){
47             kit = new HTMLParser();
48             parser = kit.getParser();
49         }
50
51         public void parse(){
52             HTMLParserCallback callback = new HTMLParserCallback();
53             try {
54                 URL url = new URL("https://www.fhmzbih.gov.ba/latinica/BIO/bioProгноza.php");
55                 InputStream input = url.openStream();
56                 reader = new BufferedReader(new InputStreamReader(input, "UTF-8"));
57                 parser.parse(reader, callback, true);
58             } catch (Exception e) {
59                 e.printStackTrace();
60             }
61         }
62     }
63 }
64
65
66
67
68
69
70
71
72
73

```

```

private class HTMLParserCallback extends HTMLEditorKit.ParserCallback {
    private int divCount = 0;
    private int imgCount = 0;
    private SimpleDateFormat sdf = new SimpleDateFormat("d.M.yyyy");

    public HTMLParserCallback() {

    }

    public void handleStartTag(HTML.Tag tag, MutableAttributeSet attributes,
        int position) {
        if(tag == HTML.Tag.DIV){
            System.out.println("DIV");
            Enumeration<?> en = attributes.getAttributeNames();
            while (en.hasMoreElements()) {
                Object attribute = (Object) en.nextElement();
                if(attribute.toString().equals("class")){
                    System.out.println("class");
                    Object value = attributes.getAttribute(attribute);
                    System.out.println("value: " + value);
                    if(value.equals("newboxes")){
                        System.out.println("newboxes");
                        divCount++;
                    }
                }
            }
        }
    }

    public void handleEndTag(HTML.Tag tag, int position) {

    }
}

```

```

    public void handleText(char[] text, int position) {
//      String regex = "(\\d{2}.\\d{2}.\\d{4}).";
//      Matcher m = Pattern.compile(regex).matcher(new String(text));
      switch (divCount) {
        case 1:
          break;
        case 2:
          break;
        case 3:
          System.out.println("CASE 3: " + new String(text));
          day1.setForecast(new String(text));
//          if (m.find()) {
//            try {
//              day1.setDate(sdf.parse(m.group(1)));
//            } catch (ParseException e) {
//              e.printStackTrace();
//            }
//          }
          break;
        case 4:
          divCount++;
          break;
        case 5:
          day2.setForecast(new String(text));
//          if (m.find()) {
//            try {
//              day2.setDate(sdf.parse(m.group(1)));
//            } catch (ParseException e) {
//              e.printStackTrace();
//            }
//          }
          divCount++;
          break;
        default:
          break;
      }
    }
}

```

```

    public void handleSimpleTag(HTML.Tag tag, MutableAttributeSet attributes,
        int position) {
      if((tag == HTML.Tag.IMG) && (imgCount < 2)){
        imgCount++;
        Enumeration<> en = attributes.getAttributeNames();
        while (en.hasMoreElements()) {
          Object attribute = (Object) en.nextElement();
          if(attribute.toString().equals("src")){
            String value = (String) attributes.getAttribute(attribute);
            System.out.println("IMG: " + value + " imgCount: " + imgCount);
            if(imgCount==1)
              day1.setImg("https://www.fhmzbih.gov.ba/slike/DRAW/BIO/" + value.substring(value.lastIndexOf("/") + 1));
            else
              day2.setImg("https://www.fhmzbih.gov.ba/slike/DRAW/BIO/" + value.substring(value.lastIndexOf("/") + 1));
          }
        }
        System.out.println("===IMAGES===");
        System.out.println(day1.getImg());
        System.out.println(day2.getImg());
        System.out.println("===IMAGES===");
      }
    }
  }

  private class HTMLParser extends HTMLEditorKit {
    private static final long serialVersionUID = 1L;

    public HTMLEditorKit.Parser getParser() {
      return super.getParser();
    }
  }
}

```

CBBHBean.java

```

1 package net.etfb1.app.external.beans;
2
3 import java.io.BufferedReader;
4
42
43 @ManagedBean(name = "cbbhBean", eager = true)
44 @ApplicationScoped
45 public class CBBHBean {
46     private HashMap<String, Currency> currencyList = new HashMap<String, Currency>();
47     private List<SelectItem> selectItems = new ArrayList<SelectItem>();
48     private String selectedCurrency;
49     private int currencyKM;
50
51     public CBBHBean() {
52         Timer timer = new Timer();
53         timer.scheduleAtFixedRate(new CBBHCurrencyServiceTimerTask(), new Date(), 900000);
54     }
55
56     public double getConversionValue() {
57         double result = 0;
58         if (selectedCurrency != null && !selectedCurrency.equals("")) {
59             Currency selCurrency = currencyList.get(selectedCurrency);
60             result = currencyKM / selCurrency.getMiddleValue() * selCurrency.getUnits();
61         }
62         return result;
63     }
64
65     public int getCurrencyKM() {
66         return currencyKM;
67     }
68
69     public void setCurrencyKM(int currencyKM) {
70         this.currencyKM = currencyKM;
71     }
72
73     public String getSelectedCurrency() {
74         return selectedCurrency;
75     }
76
77     public void setSelectedCurrency(String selectedCurrency) {
78         this.selectedCurrency = selectedCurrency;
79     }
80
81     public List<SelectItem> getSelectItems() {
82         return selectItems;
83     };
84
85     public void setSelectItems(List<SelectItem> selectItems) {
86         this.selectItems = selectItems;
87     }
88

```

```

}
class CBBHCurrencyServiceTimerTask extends TimerTask {
    @Override
    public void run() {
        try {
            SimpleDateFormat sdf = new SimpleDateFormat("MM/dd/yyyy 00:00:00");
            String addr = "http://www.cbbh.ba/CurrencyExchange/GetXml?date="
                + URLEncoder.encode(sdf.format(new Date()), "UTF-8");
            // + URLEncoder.encode("12/27/2017 00:00:00", "UTF-8");
            String addr = "https://www.cbbh.ba/CurrencyExchange/GetJsonForPeriod?dateFrom=Tue,%2018%20Dec%202018%20
            URLConnection connection = new URL(addr).openConnection();
            connection.setRequestProperty("User-Agent",
                "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1271.95
            connection.connect();

            InputStream input = connection.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(input));
            String jsonTxt = "", line="";
            while((line = br.readLine())!=null){
                jsonTxt += line;
            }
            // SAXParserFactory factory = SAXParserFactory.newInstance();
            // factory.setNamespaceAware(false);
            // SAXParser parser = factory.newSAXParser();
            // parser.parse(input, new CBBHHandler());
            // fillSelectItems();

            JSONArray object = new JSONArray(jsonTxt);
            JSONObject wrapper = object.getJSONObject(0);
            JSONArray currencyItems = wrapper.getJSONArray("CurrencyExchangeItems");
            for (int i = 0; i < currencyItems.length(); i++) {
                JSONObject currencyItem = currencyItems.getJSONObject(i);
                String alphaCode = currencyItem.getString("AlphaCode");
                int unit = Integer.parseInt(currencyItem.getString("Units"));
                double middle = Double.parseDouble(currencyItem.getString("Middle").replace(",", "."));
                Currency temp = new Currency();
                temp.setCurrency(alphaCode);
                temp.setMiddleValue(middle);
                temp.setUnits(unit);
                currencyList.put(temp.getCurrency(), temp);
            }
            fillSelectItems();
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
}

```

```

period?dateFrom=Tue,%2018%20Dec%202018%2023:00:00%20GMT&dateTo=Tue,%2018%20Dec%202018%2023:00:00%20GMT";

(KHTML, like Gecko) Chrome/23.0.1271.95 Safari/537.11");

```


YahooWeatherBean.java

```
1 package net.etfbl.app.external.beans;
2
3 import java.io.IOException;
4
24
25 @ManagedBean(eager=false)
26 @ApplicationScoped
27 public class YahooWeatherBean {
28     private WeatherDay today;
29     private WeatherDay forecastDay1;
30     private WeatherDay forecastDay2;
31
32 public YahooWeatherBean() {
33     today = new WeatherDay();
34     forecastDay1 = new WeatherDay();
35     forecastDay2 = new WeatherDay();
36     Timer timer = new Timer();
37     timer.scheduleAtFixedRate(new YahooWeatherServiceTimerTask(), new Date(), 900000);
38 }
39
40 public WeatherDay getToday() {
41     return today;
42 }
43
44 public void setToday(WeatherDay today) {
45     this.today = today;
46 }
47
48 public WeatherDay getForecastDay1() {
49     return forecastDay1;
50 }
51
52 public void setForecastDay1(WeatherDay forecastDay1) {
53     this.forecastDay1 = forecastDay1;
54 }
55
56 public WeatherDay getForecastDay2() {
57     return forecastDay2;
58 }
59
60 public void setForecastDay2(WeatherDay forecastDay2) {
61     this.forecastDay2 = forecastDay2;
62 }
63 }
```

```

64 private class YahooWeatherServiceTimerTask extends TimerTask {
65     YahooHandler yahooHandler = new YahooHandler();
66
67     public YahooWeatherServiceTimerTask() {
68     }
69
70     @Override
71     public void run() {
72         try {
73             URL url = new URL(
74                 "http://weather.yahooapis.com/forecastrss?p=BKXX0001&u=c");
75             InputStream input = url.openStream();
76             SAXParserFactory factory = SAXParserFactory.newInstance();
77             factory.setNamespaceAware(false);
78             SAXParser parser = factory.newSAXParser();
79             parser.parse(input, yahooHandler);
80         } catch (MalformedURLException e) {
81             e.printStackTrace();
82         } catch (IOException e) {
83             e.printStackTrace();
84         } catch (ParserConfigurationException e) {
85             e.printStackTrace();
86         } catch (SAXException e) {
87             e.printStackTrace();
88         }
89     }
90 }
91

```

```

92 private class YahooHandler extends DefaultHandler {
93
94     int forecast = 0;
95     SimpleDateFormat sdf = new SimpleDateFormat("EEE, dd MMM yyyy hh:mm a z");
96     SimpleDateFormat sdf2 = new SimpleDateFormat("d MMM yyyy");
97
98     public YahooHandler() {
99
100     }
101     public void startElement(String uri, String localName, String qName,
102         Attributes attributes) throws SAXException {
103
104         if ("yweather:wind".equals(qName)) {
105             today.setWindDirection(getWindDirection(attributes.getValue("direction")));
106             today.setWindSpeed(attributes.getValue("speed"));
107         } else if ("yweather:atmosphere".equals(qName)) {
108             today.setAtmosphereHumidity(attributes.getValue("humidity"));
109             today.setAtmosphereVisibility(attributes.getValue("visibility"));
110             today.setAtmospherePressure(attributes.getValue("pressure"));
111         } else if ("yweather:astronomy".equals(qName)) {
112             today.setAstronomySunrise(attributes.getValue("sunrise"));
113             today.setAstronomySunset(attributes.getValue("sunset"));
114         } else if ("yweather:condition".equals(qName)) {
115             today.setCode(attributes.getValue("code"));
116             today.setTemp(attributes.getValue("temp"));
117             try {
118                 today.setDate(sdf.parse(attributes.getValue("date")));
119             } catch (ParseException e) {
120                 e.printStackTrace();
121             }
122         } else if ("yweather:forecast".equals(qName)) {
123             if (forecast == 0) {
124                 forecastDay1.setCode(attributes.getValue("code"));
125                 forecastDay1.setHigh(attributes.getValue("high"));
126                 forecastDay1.setLow(attributes.getValue("low"));
127                 try {
128                     forecastDay1.setDate(sdf2.parse(attributes.getValue("date")));
129                 } catch (ParseException e) {
130                     e.printStackTrace();
131                 }
132                 forecast++;
133             } else {
134                 forecastDay2.setCode(attributes.getValue("code"));
135                 forecastDay2.setHigh(attributes.getValue("high"));
136                 forecastDay2.setLow(attributes.getValue("low"));
137                 try {
138                     forecastDay2.setDate(sdf2.parse(attributes.getValue("date")));
139                 } catch (ParseException e) {
140                     e.printStackTrace();
141                 }
142                 forecast = 0;
143             }
144         }
145         super.startElement(uri, localName, qName, attributes);
146     }

```

```

148 private String getWindDirection(String direction) {
149     String directions[] = {"N", "NNE", "NE", "ENE", "E", "ESE", "SE", "SSE", "S", "SSW", "SW", "WSW", "W", "WNW", "NW", "NNW"};
150     int degree = Integer.parseInt(direction);
151     int x = (int) Math.round(degree/22.5);
152     return directions[x];
153 }
154 }
155 }

```

BOOKS APP MENU

[Add user](#)
[User list](#)
[Widget pages](#)

[Add user](#)
[User list](#)
[Widget pages](#)

[Add user](#)
[User list](#)
[Widget pages](#)

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

- [Add User](#)
- [Widgets](#)

Username

Password

UNESI

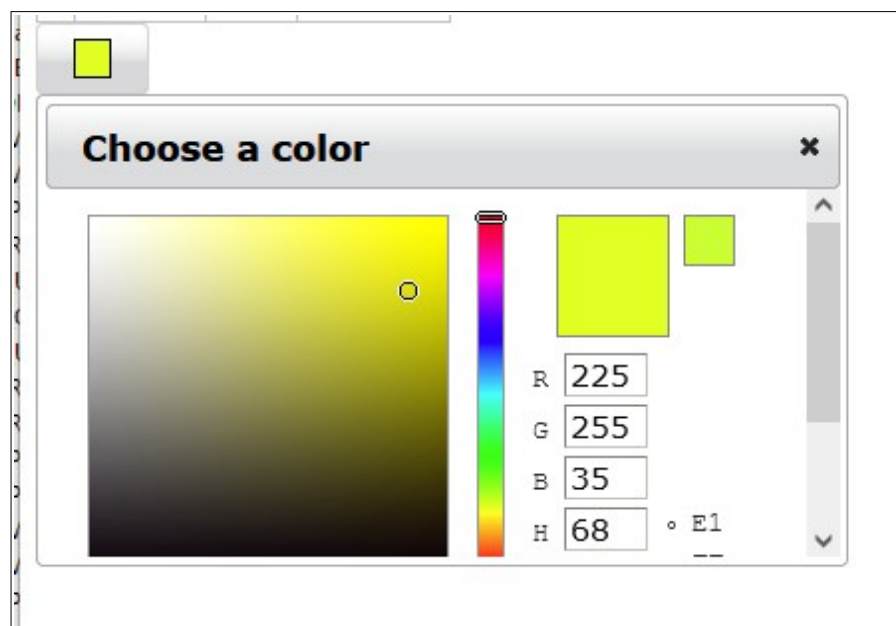
footer

- [Add User](#)
- [Widgets](#)

id	username	action	action
1	marko	DELETE	AJAX DELETE
2	janko	DELETE	AJAX DELETE



footer

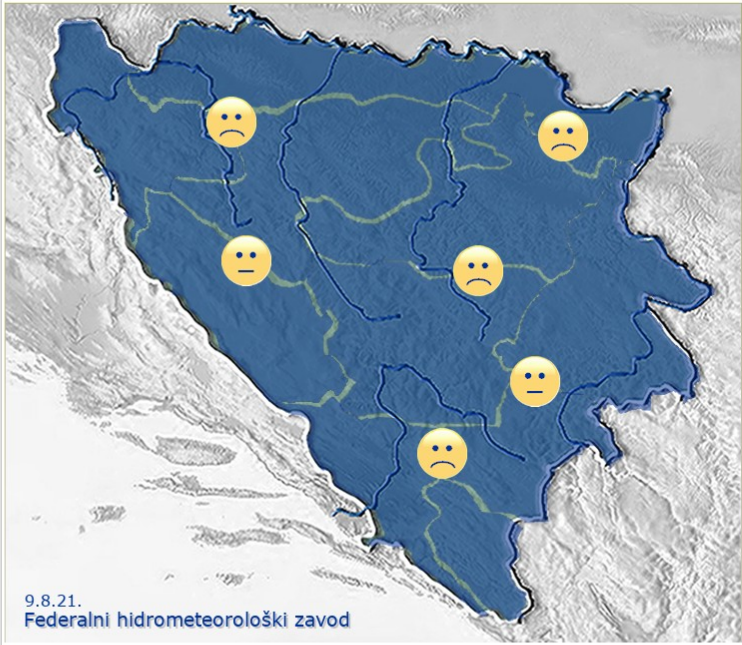


- [Add User](#)
- [Widgets](#)

BL berza - novosti
[Saopštenje za javnost Ministarstva finansija - Isplata dospjelih obaveza po osnovu emitovanih HOV](#)
[Distributivni centar a.d. Bijeljina - Privremeno zaustavljanje trgovanja](#)
[Promtes a.d. Teslić - Uvrštenje akcija PRMT-R-A na slobodno tržište - tržište akcija](#)
[Na službeno berzansko tržište uvrštena 9. emisija obveznica Republike Srpske emitovanih po osnovu stare devizne štednje \(RSDS-O-I\)](#)
[Rudnici Željezne rude Ljubia a.d. Prijedor - Uvrštenje akcija LJUB-R-A na slobodno tržište - tržište akcija](#)
 Izvor: BL berza

Iznos: KM
 Valuta:
[Izvor: CBBH](#)

Danas Sutra



9.8.21.
Federalni hidrometeorološki zavod

BIOMETEOROLOŠKA PROGNOZA ZA PONEĐELJAK 09. AUGUST 2021. GODINE. Slično kao u protekлом periodu, ugodnije tokom priepodneva. Drugi dio dana proteći će uz porast temperatura i sparno vrijeme, što će uzrokovati poteškoće hroničnim bolesnicima, stoga je poželjno umanjiti aktivnosti. Dodatni problem može nastati usljed nestabilnih vremenskih prilika u popodnevnim satima. Učesnicima u saobraćaju sugerisemo oprezniju vožnju. Negativni uticaj sparne djelimično će oslabiti sa noćnim satima, tako da će osjetljive osobe imati manje problema sa snom.

[Izvor: FHMZ](#)

Korisnicko ime:
 Password:

footer

Activate Windows
 Go to Settings to activate Windows.

BL berza - novosti
[Saopštenje za javnost Ministarstva finansija - Isplata dospjelih obaveza po osnovu emitovanih HOV](#)
[Distributivni centar a.d. Bijeljina - Privremeno zaustavljanje trgovanja](#)
[Promtes a.d. Teslić - Uvrštenje akcija PRMT-R-A na slobodno tržište - tržište akcija](#)
[Na službeno berzansko tržište uvrštena 9. emisija obveznica Republike Srpske emitovanih po osnovu stare devizne štednje \(RSDS-O-I\)](#)
[Rudnici Željezne rude Ljubija a.d. Prijedor - Uvrštenje akcija LJUB-R-A na slobodno tržište - tržište akcija](#)
 Izvor: BL berza

Iznos: KM
 Valuta:
[Izvor: CBBH](#)

BL berza - novosti

[Saopštenje za javnost Ministarstva finansija - Isplata dospjelih obaveza po osnovu emitovanih HOV](#)
[Distributivni centar a.d. Bijeljina - Privremeno zaustavljanje trgovanja](#)
[Promtes a.d. Teslić - Uvrštenje akcija PRMT-R-A na slobodno tržište - tržište akcija](#)
[Na službeno berzansko tržište uvrštena 9. emisija obveznica Republike Srpske emitovanih po osnovu stare devizne štednje \(RSDS-O-I\)](#)
[Rudnici željezne rude Ljubija a.d. Prijedor - Uvrštenje akcija LJUB-R-A na slobodno tržište - tržište akcija](#)
[Izvor: BL berza](#)

Iznos: KM

Valuta: RSD ▼

605,59
[Izvor: CBBH](#)

user_add.xhtml

```

1<html xmlns="http://www.w3.org/1999/xhtml"
2  xmlns:f="http://java.sun.com/jsf/core"
3  xmlns:ui="http://java.sun.com/jsf/facelets"
4  xmlns:h="http://java.sun.com/jsf/html"
5  xmlns:a4j="http://richfaces.org/a4j"
6  xmlns:rich="http://richfaces.org/rich">
7<h:body>
8<ui:composition template="/WEB-INF/templates/template.xhtml">
9  <f:loadBundle basename="net.etfbl.s_cube.emailparser.dictionaries.messages_#{adminBean.Lang}" var="msgs"/>
10<ui:define name="title">
11  <h:outputText value="#{msgs.app_name}" />
12</ui:define>
13<ui:define name="page_title">
14  <h:outputText value="#{msgs.page_title_logs}" />
15</ui:define>
16<ui:define name="menu">
17  <ui:include src="/WEB-INF/components/menu.xhtml"></ui:include>
18</ui:define>
19<ui:define name="body">
20  <ui:include src="/WEB-INF/components/body_user_add.xhtml"></ui:include>
21</ui:define>
22<ui:define name="footer">
23  <ui:include src="/WEB-INF/components/footer.xhtml"></ui:include>
24</ui:define>
25</ui:composition>
26
27</h:body>
28</html>

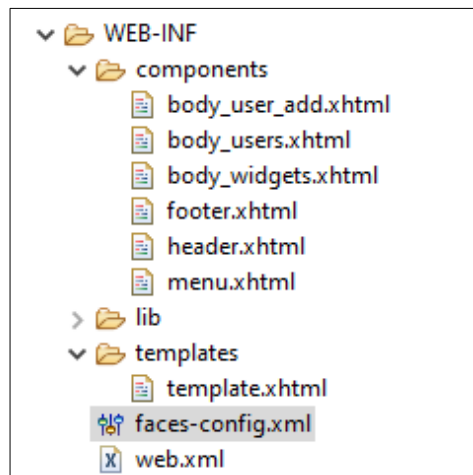
```


users.xhtml

```
1<html xmlns="http://www.w3.org/1999/xhtml"
2      xmlns:f="http://java.sun.com/jsf/core"
3      xmlns:ui="http://java.sun.com/jsf/facelets"
4      xmlns:h="http://java.sun.com/jsf/html"
5      xmlns:a4j="http://richfaces.org/a4j"
6      xmlns:rich="http://richfaces.org/rich">
7<h:body>
8<ui:composition template="/WEB-INF/templates/template.xhtml">
9<ui:define name="title">
10    <h:outputText value="#{msgs.app_name}" />
11</ui:define>
12<ui:define name="page_title">
13    <h:outputText value="#{msgs.page_title_logs}" />
14</ui:define>
15<ui:define name="menu">
16    <ui:include src="/WEB-INF/components/menu.xhtml"></ui:include>
17</ui:define>
18<ui:define name="body">
19    <ui:include src="/WEB-INF/components/body_users.xhtml"></ui:include>
20</ui:define>
21<ui:define name="footer">
22    <ui:include src="/WEB-INF/components/footer.xhtml"></ui:include>
23</ui:define>
24</ui:composition>
25
26</h:body>
27</html>
```

widgets.xhtml

```
1<html xmlns="http://www.w3.org/1999/xhtml"
2      xmlns:f="http://java.sun.com/jsf/core"
3      xmlns:ui="http://java.sun.com/jsf/facelets"
4      xmlns:h="http://java.sun.com/jsf/html"
5      xmlns:a4j="http://richfaces.org/a4j"
6      xmlns:rich="http://richfaces.org/rich">
7<h:head/>
8<h:body>
9<ui:composition template="/WEB-INF/templates/template.xhtml">
10<ui:define name="title">
11    Widgets
12</ui:define>
13<ui:define name="page_title">
14    <h:outputText value="#{msgs.page_title_logs}" />
15</ui:define>
16<ui:define name="menu">
17    <ui:include src="/WEB-INF/components/menu.xhtml"></ui:include>
18</ui:define>
19<ui:define name="body">
20    <ui:include src="/WEB-INF/components/body_widgets.xhtml"></ui:include>
21</ui:define>
22<ui:define name="footer">
23    <ui:include src="/WEB-INF/components/footer.xhtml"></ui:include>
24</ui:define>
25</ui:composition>
26
27</h:body>
28</html>
```



templates.xhtml

```

1 <html xmlns="http://www.w3.org/1999/xhtml"
2     xmlns:ui="http://java.sun.com/jsf/facelets"
3     xmlns:f="http://java.sun.com/jsf/core"
4     xmlns:h="http://java.sun.com/jsf/html"
5     xmlns:a4j="http://richfaces.org/a4j"
6     xmlns:rich="http://richfaces.org/rich" >
7 <h:head>
8 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
9 <title><ui:insert name="title" /></title>
10 </h:head>
11
12 <body>
13 <div id="topNav">
14 <ui:insert name="menu" />
15 </div>
16 <div id="body">
17 <ui:insert name="body" />
18 </div>
19 <div id="footer">
20 <ui:insert name="footer" />
21 </div>
22 </body>
23 </html>

```

menu.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 <ui:composition>
6 <h:form>
7 <ul>
8 <li><h:outputLink value="user_add.xhtml" title="Add User"><h:outputText value="Add User"/></h:outputLink></li>
9 <li><h:outputLink value="widgets.xhtml" title="Widgets"><h:outputText value="Widgets"/></h:outputLink></li>
10 </ul>
11 </h:form>
12 </ui:composition>
13 </html>
14

```

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     xmlns:rich="http://richfaces.org/rich"
6     xmlns:a4j="http://richfaces.org/a4j">
7 <ui:composition>
8     <h:form>
9 <h:panelGrid border="0" columns="2">
10 <h:panelGroup>
11 <h:outputText value="licensing server"></h:outputText>
12 </h:panelGroup>
13 <h:panelGroup>
14 <h:outputLink value="mailto:s-cube@etfbl.net">
15     <h:outputText value="Kontakt"></h:outputText>
16 </h:outputLink>
17 </h:panelGroup>
18 </h:panelGrid>
19 </h:form>
20 </ui:composition>
21 </html>
22
```

footer.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 <ui:composition>
6     footer
7 </ui:composition>
8 </html>
9
```

body_widgets.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     xmlns:rich="http://richfaces.org/rich"
6     xmlns:a4j="http://richfaces.org/a4j"
7     xmlns:scube="http://java.sun.com/jsf/composite/scubecomponents">
8 <h:head>
9 <body>
10 <h:panelGrid columns="6" border="0" width="100%" columnClasses="vertical-top,vertical-top,vertical-top,vertical-top,vertical-top,vertical-top">
11     <scube:blberzanews panelStyleClass="abc" />
12     <scube:cbhhcurrencyexchange panelStyleClass="abc" />
13     <scube:fhmzbio panelStyleClass="abc" panelWidth="450"></scube:fhmzbio>
14 </h:panelGrid>
15 <h:form>
16 <scube:loginform gridColumnClasses="" gridStyleClass="" usernameLabel="Korisnicko ime:" passwordLabel="Password:"
17     usernameValue="#{userBean.newUser.username}" passwordValue="#{userBean.newUser.password}"
18     usernameID="user" passwordID="pass" usernameRequiredMessage="Username required"
19     passwordRequiredMessage="Password required" usernameStyleClass="" passwordStyleClass=""
20     submitButtonAction="#{userBean.Login}" submitButtonText="Submit" submitButtonStyleClass="" messageesStyleClass="">
21 </h:form>
22
23 </body>
24 </html>
```

body_users.xhtml

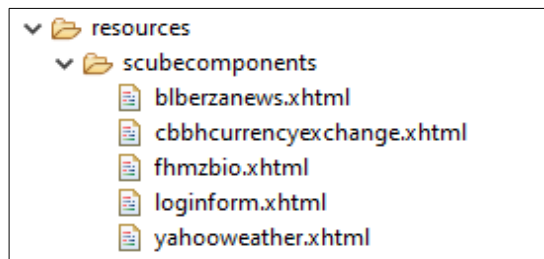
```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4     xmlns:f="http://java.sun.com/jsf/core"
5     xmlns:h="http://java.sun.com/jsf/html"
6     xmlns:a4j="http://richfaces.org/a4j"
7     xmlns:rich="http://richfaces.org/rich"
8     xmlns:ui="http://java.sun.com/jsf/facelets"
9     xmlns:prime="http://primefaces.prime.com.tr/ui">
10 <ui:composition>
11     <h:form prependId="true">
12         <rich:dataTable id="userTable" value="#{userBean.getAllUsers()}" var="user">
13             <rich:column>
14                 <f:facet name="header">
15                     <h:outputText value="id" />
16                 </f:facet>
17                 <h:outputText value="#{user.user_id}" />
18             </rich:column>
19             <rich:column>
20                 <f:facet name="header">
21                     <h:outputText value="username" />
22                 </f:facet>
23                 <h:outputText value="#{user.username}" />
24             </rich:column>
25             <rich:column>
26                 <f:facet name="header">
27                     <h:outputText value="action" />
28                 </f:facet>
29                 <h:commandLink action="#{userBean.delete}">
30                     <f:setPropertyActionListener target="#{userBean.selectedUser}" value="#{user}" />
31                     <h:outputText title="DELETE" value="DELETE" />
32                 </h:commandLink>
33             </rich:column>
34             <rich:column>
35                 <f:facet name="header">
36                     <h:outputText value="action" />
37                 </f:facet>
38                 <a4j:commandLink action="#{userBean.delete}" render="userTable" execute="@this">
39                     <f:setPropertyActionListener target="#{userBean.selectedUser}" value="#{user}" />
40                     <h:outputText title="AJAX DELETE" value="AJAX DELETE" />
41                 </a4j:commandLink>
42             </rich:column>
43         </rich:dataTable>
44     </h:form>
45     <prime:colorPicker/></prime:colorPicker>
46 </ui:composition>
47
48 </html>

```


body_user_add.xhtml

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org
3 <html xmlns="http://www.w3.org/1999/xhtml"
4     xmlns:f="http://java.sun.com/jsf/core"
5     xmlns:h="http://java.sun.com/jsf/html"
6     xmlns:a4j="http://richfaces.org/a4j"
7     xmlns:rich="http://richfaces.org/rich"
8     xmlns:ui="http://java.sun.com/jsf/facelets"
9     xmlns:prime="http://primefaces.prime.com.tr/ui">
10 <ui:composition>
11     <h:form prependId="true">
12         <h:panelGrid columns="2">
13             <h:outputText value="Username"/>
14             <h:inputText value="#{userBean.newUser.username}" />
15             <h:outputText value="Password"/>
16             <h:inputSecret value="#{userBean.newUser.password}" />
17             <h:outputText></h:outputText>
18             <h:commandButton value="UNESI" action="#{userBean.add}" />
19         </h:panelGrid>
20     </h:form>
21
22 </ui:composition>
23
24 </html>
```



blberzanews.xhtml

```
1<html xmlns="http://www.w3.org/1999/xhtml"
2  xmlns:h="http://java.sun.com/jsf/html"
3  xmlns:f="http://java.sun.com/jsf/core"
4  xmlns:ui="http://java.sun.com/jsf/facelets"
5  xmlns:rich="http://richfaces.org/rich"
6  xmlns:a4j="http://richfaces.org/a4j"
7  xmlns:composite="http://java.sun.com/jsf/composite">
8<h:head>
9  <title>This will not be present in rendered output</title>
10</h:head>
11
12<h:body>
13  <composite:interface>
14    <composite:attribute name="panelStyleClass"/>
15  </composite:interface>
16
17  <composite:implementation>
18    <h:panelGrid styleClass="{cc.attrs.panelStyleClass}">
19      <rich:panel header="{msgs.blberza_news}">
20        <h:dataTable headerClass="left-column" frame="border" columnClasses="right-column" rowClasses="odd-row,even-row" styleClass="{cc.attrs.panelStyleClass}">
21          <h:column>
22            <f:facet name="header">
23              <h:outputText value="BL berza - novosti" />
24            </f:facet>
25            <h:outputLink value="{blBerzaNews.link}" target="_blank" title="{blBerzaNews.title}"><h:outputText value="{blBerzaNews.title}"/></h:outputLink>
26            <f:verbatim><br/></f:verbatim>
27            <h:outputText value="{blBerzaNews.pubDate}">
28              <f:convertDateTime pattern="dd.MM.yyyy. hh:mm:ss" timeZone="GMT+1"/>
29            </h:outputText>
30          </h:column>
31          <f:facet name="footer">
32            <h:outputLink value="http://www.blberza.com" target="_blank" title="{msgs.blberza}"><h:outputText value="Izvor: BL berza"/></h:outputLink>
33          </f:facet>
34        </h:dataTable>
35      </rich:panel>
36    </h:panelGrid>
37  </composite:implementation>
38</h:body>
39</html>
```

```
      <h:column>
        <h:outputText value="{blBerzaNews.title}" />
      </h:column>
    </h:dataTable>
  </rich:panel>
</h:panelGrid>
</composite:implementation>
</h:body>
</html>
```


cbbhcurrenxyexchange.xhtml

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4       xmlns:h="http://java.sun.com/jsf/html"
5       xmlns:f="http://java.sun.com/jsf/core"
6       xmlns:ui="http://java.sun.com/jsf/facelets"
7       xmlns:rich="http://richfaces.org/rich"
8       xmlns:a4j="http://richfaces.org/a4j"
9       xmlns:composite="http://java.sun.com/jsf/composite">
10 <h:head>
11   <title>This will not be present in rendered output</title>
12
13 </h:head>
14
15 <h:body>
16   <composite:interface>
17     <composite:attribute name="panelStyleClass"/>
18   </composite:interface>
19
20   <composite:implementation>
21     <h:panelGrid styleClass="#{cc.attrs.panelStyleClass}"
22       <rich:panel header="#{msgs.cbbh_currency_exchange}"
23         <h:form>
24           <h:panelGrid columns="3" styleClass="#{cc.attrs.panelStyleClass}" frame="border">
25             <h:outputText value="Iznos:"/>
26             <h:inputText id="currencyKM" size="10" value="#{cbbhBean.currencyKM}"
27               <f:validateLongRange minimum="0" maximum="100000"/>
28               <f:ajax execute="@this" render="conversionValue" event="blur"/>
29             </h:inputText>
30             <h:outputText value="KM"/>
31             <h:outputText value="Valuta:"/>
32             <h:selectOneMenu value="#{cbbhBean.selectedCurrency}"
33               <f:selectItem itemValue="" itemLabel="#{msgs.cbbh_choose}" />
34               <f:selectItems value="#{cbbhBean.selectItems}" />
35               <f:ajax execute="@this" render="conversionValue" event="valueChange"/>
36             </h:selectOneMenu>
37             <h:outputText/>
38             <h:outputText value="#{msgs.cbbh_conversion}" />
39             <h:outputText id="conversionValue" value="#{cbbhBean.conversionValue}"
40               <f:convertNumber maxFractionDigits="2"/>
41             </h:outputText>
42             <h:outputText/>
43             <h:outputText/>
44             <h:outputLink value="http://www.cbbh.ba" target="_blank" title="CBBH"><h:outputText value="Izvor: CBBH"/></h:outputLink>
45           </h:panelGrid>
46         </h:form>
47       </rich:panel>
48     </h:panelGrid>
49   </composite:implementation>
50
51 </h:body>
52
53 </html>
```

fhmzbio.xhtml

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4       xmlns:h="http://java.sun.com/jsf/html"
5       xmlns:f="http://java.sun.com/jsf/core"
6       xmlns:ui="http://java.sun.com/jsf/facelets"
7       xmlns:rich="http://richfaces.org/rich"
8       xmlns:a4j="http://richfaces.org/a4j"
9       xmlns:composite="http://java.sun.com/jsf/composite">
10 <h:head>
11   <title>This will not be present in rendered output</title>
12
13 </h:head>
14
15 <h:body>
16 <f:loadBundle var="msgs" basename="net.etfbl.s_cube.dictionaries.{regUsrKorisnikBean.selectedLanguage}"></f:loadBundle>
17
18 <composite:interface>
19   <composite:attribute name="panelStyleClass"/>
20   <composite:attribute name="panelWidth"/>
21 </composite:interface>
22
23 <composite:implementation>
24   <h:panelGrid styleClass="{cc.attrs.panelStyleClass}" width="{cc.attrs.panelWidth}"
25     <rich:tabPanel switchType="client">
26       <rich:tab header="Danas">
27         <h:panelGrid columns="1" styleClass="{cc.attrs.panelStyleClass}" frame="border">
28           <h:graphicImage alt="{fhmzBioBean.day1.formattedDate}" title="{fhmzBioBean.day1.formattedDate}" value="{fhmzBioBean.day1.img}"/>
29           <h:outputText value="{fhmzBioBean.day1.formattedDate}"/>
30           <h:outputText value="{fhmzBioBean.day1.forecast}"/>
31           <hr/>
32           <h:outputLink value="http://www.fhmzbih.gov.ba" target="_blank"><h:outputText value="Izvor: FH#IZ"/></h:outputLink>
33         </h:panelGrid>
34       </rich:tab>
35       <rich:tab header="Sutra">
36         <h:panelGrid columns="1" styleClass="{cc.attrs.panelStyleClass}" frame="border">
37           <h:graphicImage alt="{fhmzBioBean.day2.formattedDate}" title="{fhmzBioBean.day2.formattedDate}" value="{fhmzBioBean.day2.img}"/>
38           <h:outputText value="{fhmzBioBean.day2.formattedDate}"/>
39           <h:outputText value="{fhmzBioBean.day2.forecast}"/>
40           <hr/>
41           <h:outputLink value="http://www.fhmzbih.gov.ba" target="_blank"><h:outputText value="Izvor: FH#IZ"/></h:outputLink>
42         </h:panelGrid>
43       </rich:tab>
44     </rich:tabPanel>
45   </h:panelGrid>
46 </composite:implementation>
47
48 </h:body>
49
50 </html>
51
```

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4       xmlns:h="http://java.sun.com/jsf/html"
5       xmlns:f="http://java.sun.com/jsf/core"
6       xmlns:ui="http://java.sun.com/jsf/facelets"
7       xmlns:composite="http://java.sun.com/jsf/composite">
8 <h:head>
9     <title>This will not be present in rendered output</title>
10
11 </h:head>
12
13 <h:body>
14
15     <composite:interface>
16         <composite:attribute name="gridColumnClasses"/>
17         <composite:attribute name="gridStyleClass"/>
18         <composite:attribute name="usernameLabel"/>
19         <composite:attribute name="passwordLabel"/>
20         <composite:attribute name="usernameValue"/>
21         <composite:attribute name="passwordValue"/>
22         <composite:attribute name="usernameID"/>
23         <composite:attribute name="passwordID"/>
24         <composite:attribute name="usernameRequiredMessage"/>
25         <composite:attribute name="passwordRequiredMessage"/>
26         <composite:attribute name="usernameStyleClass"/>
27         <composite:attribute name="passwordStyleClass"/>
28         <composite:attribute name="submitButtonAction"
29             method-signature="java.lang.String action()"/>
30         <composite:attribute name="submitButtonText"/>
31         <composite:attribute name="submitButtonStyleClass"/>
32         <composite:attribute name="messagesStyleClass"/>
33     </composite:interface>
```

loginform.xhtml

```

1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3  <html xmlns="http://www.w3.org/1999/xhtml"
4    xmlns:h="http://java.sun.com/jsf/html"
5    xmlns:f="http://java.sun.com/jsf/core"
6    xmlns:ui="http://java.sun.com/jsf/facets"
7    xmlns:composite="http://java.sun.com/jsf/composite">
8  <h:head>
9    <title>This will not be present in rendered output</title>
10
11 </h:head>
12
13 <h:body>
14
15   <composite:interface>
16     <composite:attribute name="gridColumnClasses"/>
17     <composite:attribute name="gridStyleClass"/>
18     <composite:attribute name="usernameLabel"/>
19     <composite:attribute name="passwordLabel"/>
20     <composite:attribute name="usernameValue"/>
21     <composite:attribute name="passwordValue"/>
22     <composite:attribute name="usernameID"/>
23     <composite:attribute name="passwordID"/>
24     <composite:attribute name="usernameRequiredMessage"/>
25     <composite:attribute name="passwordRequiredMessage"/>
26     <composite:attribute name="usernameStyleClass"/>
27     <composite:attribute name="passwordStyleClass"/>
28     <composite:attribute name="submitButtonAction"
29       method-signature="java.lang.String action()"/>
30     <composite:attribute name="submitButtonText"/>
31     <composite:attribute name="submitButtonStyleClass"/>
32     <composite:attribute name="messageesStyleClass"/>
33   </composite:interface>

```

```

<composite:implementation>
  <h:panelGrid columns="3" columnClasses="{cc.attrs.gridColumnClasses}" styleClass="{cc.attrs.gridStyleClass}">
    <h:outputText value="{cc.attrs.usernameLabel}"></h:outputText>
    <h:inputText required="true" id="{cc.attrs.usernameID}" value="{cc.attrs.usernameValue}" requiredMessage="{cc.attrs.usernameRequiredMessage}" s
    <h:message for="{cc.attrs.usernameID}" styleClass="{cc.attrs.messageesStyleClass}"></h:message>
    <h:outputText value="{cc.attrs.passwordLabel}"></h:outputText>
    <h:inputSecret required="true" id="{cc.attrs.passwordID}" value="{cc.attrs.passwordValue}" requiredMessage="{cc.attrs.passwordRequiredMessage}"
    <h:message for="{cc.attrs.passwordID}" styleClass="{cc.attrs.messageesStyleClass}"></h:message>
    <h:outputText value=""></h:outputText>
    <h:commandButton action="{cc.attrs.submitButtonAction}" value="{cc.attrs.submitButtonText}" styleClass="{cc.attrs.submitButtonStyleClass}"></h:
  </h:panelGrid>
</composite:implementation>

```

```

}" requiredMessage="{cc.attrs.usernameRequiredMessage}" styleClass="{cc.attrs.usernameStyleClass}"></h:inputText>
:message>

ue}" requiredMessage="{cc.attrs.passwordRequiredMessage}" styleClass="{cc.attrs.passwordStyleClass}"></h:inputSecret>
:message>

ext}" styleClass="{cc.attrs.submitButtonStyleClass}"></h:commandButton>

```

```
</h:body>
```

```
</html>
```

yahooweather.xhtml

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4       xmlns:h="http://java.sun.com/jsf/html"
5       xmlns:f="http://java.sun.com/jsf/core"
6       xmlns:ui="http://java.sun.com/jsf/facelets"
7       xmlns:rich="http://richfaces.org/rich"
8       xmlns:a4j="http://richfaces.org/a4j"
9       xmlns:composite="http://java.sun.com/jsf/composite">
10 <h:head>
11   <title>This will not be present in rendered output</title>
12 </h:head>
13 <h:body>
14   <f:loadBundle var="msgs" basename="net.etfbl.s_cube.dictionaries.#{regUsrKor.isntKBean.selectedLanguage}"></f:loadBundle>
15
16   <composite:interface>
17     <composite:attribute name="panelStyleClass"/>
18   </composite:interface>
19
20   <composite:implementation>
21     <h:panelGrid styleClass="#{cc.attrs.panelStyleClass}">
22       <rich:tabPanel switchType="client">
23         <rich:tab header="Today">
24           <h:panelGrid columns="1" styleClass="#{cc.attrs.panelStyleClass}" frame="border">
25             <h:outputText value="#{yahooWeatherBean.today.date}">
26               <f:convertDateTime pattern="dd.MM.yyyy." timeZone="GMT+1"/>
27             </h:outputText>
28             <h:graphicImage value="images/weather/#{yahooWeatherBean.today.code}.gif"/>
29             <h:outputText value="Temperature: #{yahooWeatherBean.today.temp} &deg;C"/>
30             <h:outputText value="Wind speed: #{yahooWeatherBean.today.windSpeed} km/h"/>
31             <h:outputText value="Wind direction: #{yahooWeatherBean.today.windDirection}"/>
32             <h:outputText value="Atmosphere humidity: #{yahooWeatherBean.today.atmosphereHumidity} %"/>
33             <h:outputText value="Atmosphere pressure: #{yahooWeatherBean.today.atmospherePressure} mb"/>
34             <h:outputText value="Atmosphere visibility: #{yahooWeatherBean.today.atmosphereVisibility}"/>
35             <h:outputText value="Astronomy sunrise: #{yahooWeatherBean.today.astronomySunrise}"/>
36             <h:outputText value="Astronomy sunset: #{yahooWeatherBean.today.astronomySunset}"/>
37             <hr/>
38             <h:outputLink value="http://weather.yahoo.com" target="_blank"><h:outputText value="Source: Yahoo! Weather"/></h:outputLink>
39           </h:panelGrid>
40         </rich:tab>
41       </rich:tabPanel>
42     </h:panelGrid>
43   </composite:implementation>
44 </h:body>
45 </html>
```

```
22 <composite:implementation>
23   <h:panelGrid styleClass="#{cc.attrs.panelStyleClass}">
24     <rich:tabPanel switchType="client">
25       <rich:tab header="Today">
26         <h:panelGrid columns="1" styleClass="#{cc.attrs.panelStyleClass}" frame="border">
27           <h:outputText value="#{yahooWeatherBean.today.date}">
28             <f:convertDateTime pattern="dd.MM.yyyy." timeZone="GMT+1"/>
29           </h:outputText>
30           <h:graphicImage value="images/weather/#{yahooWeatherBean.today.code}.gif"/>
31           <h:outputText value="Temperature: #{yahooWeatherBean.today.temp} &deg;C"/>
32           <h:outputText value="Wind speed: #{yahooWeatherBean.today.windSpeed} km/h"/>
33           <h:outputText value="Wind direction: #{yahooWeatherBean.today.windDirection}"/>
34           <h:outputText value="Atmosphere humidity: #{yahooWeatherBean.today.atmosphereHumidity} %"/>
35           <h:outputText value="Atmosphere pressure: #{yahooWeatherBean.today.atmospherePressure} mb"/>
36           <h:outputText value="Atmosphere visibility: #{yahooWeatherBean.today.atmosphereVisibility}"/>
37           <h:outputText value="Astronomy sunrise: #{yahooWeatherBean.today.astronomySunrise}"/>
38           <h:outputText value="Astronomy sunset: #{yahooWeatherBean.today.astronomySunset}"/>
39           <hr/>
40           <h:outputLink value="http://weather.yahoo.com" target="_blank"><h:outputText value="Source: Yahoo! Weather"/></h:outputLink>
41         </h:panelGrid>
42       </rich:tab>
43       <rich:tab header="Forecast">
44         <h:panelGrid columns="1" styleClass="#{cc.attrs.panelStyleClass}" frame="border">
45           <h:outputText value="#{yahooWeatherBean.forecastDay1.date}">
46             <f:convertDateTime pattern="dd.MM.yyyy." timeZone="GMT+1"/>
47           </h:outputText>
48           <h:graphicImage value="images/weather/#{yahooWeatherBean.forecastDay1.code}.gif"/>
49           <h:outputText value="Temperature: min: #{yahooWeatherBean.forecastDay1.low} &deg;C, max: #{yahooWeatherBean.forecastDay1.high} &deg;C"/>
50           <hr/>
51           <h:outputText value="#{yahooWeatherBean.forecastDay2.date}">
52             <f:convertDateTime pattern="dd.MM.yyyy." timeZone="GMT+1"/>
53           </h:outputText>
54           <h:graphicImage value="images/weather/#{yahooWeatherBean.forecastDay2.code}.gif"/>
55           <h:outputText value="Temperature: min: #{yahooWeatherBean.forecastDay2.low} &deg;C, max: #{yahooWeatherBean.forecastDay2.high} &deg;C"/>
56           <hr/>
57           <h:outputLink value="http://weather.yahoo.com" target="_blank" title="Yahoo! Weather"><h:outputText value="Source: Yahoo! Weather"/></h:outputLink>
58         </h:panelGrid>
59       </rich:tab>
60     </rich:tabPanel>
61   </h:panelGrid>
62 </composite:implementation>
63 </h:body>
64 </html>
```

Activate Windows
Go to Settings to activate Windows