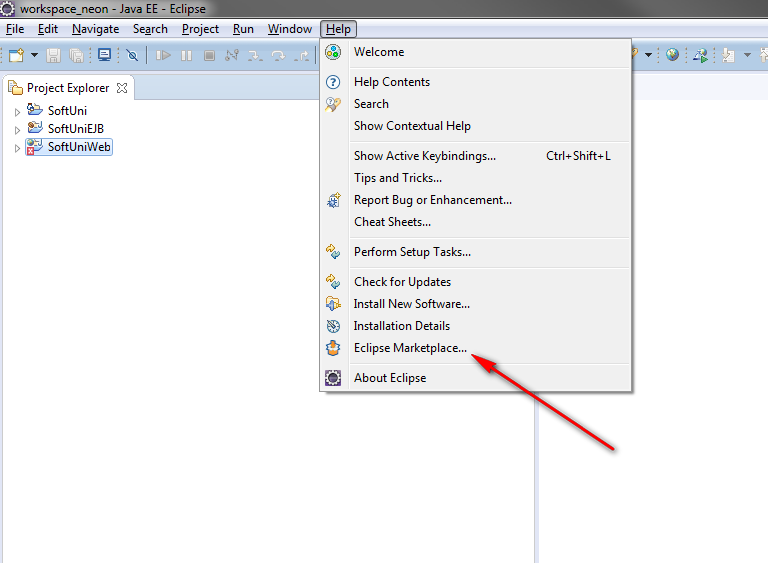
**JEE Practical Project**

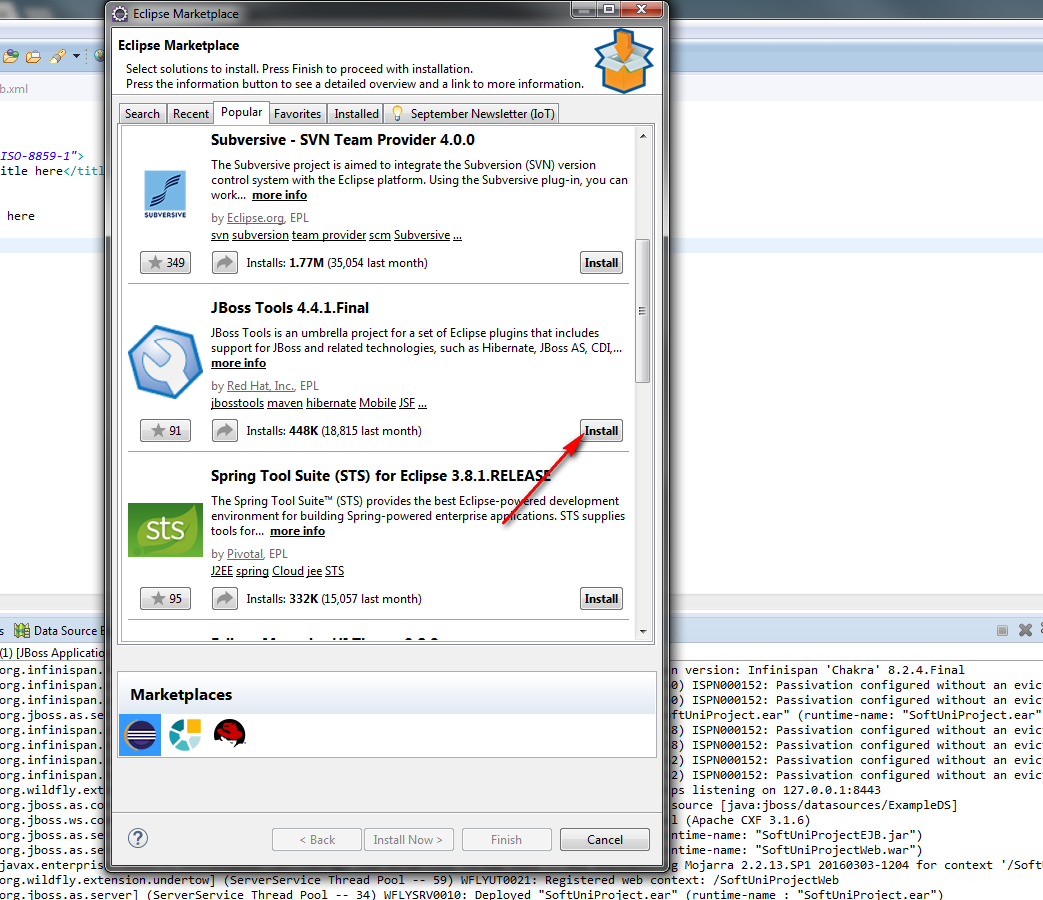
**Task**

1. **Create JavaEE application with the following functionality**
   1. Login Page - this is the initial page that the user accesses. It requires username and password, validates them and forwards the user to a success page
   2. Add Resource bundle and use it for all messages and labels in the application
   3. Add Authentication filter, that will restrict access to the secured resources
   4. Add Primeface DataGrid that will display all users in the application
   5. Add functionality for creating and editing users

**Execution**

1. **Setup environment**
2. Install JDK, Eclipse and add Wildfly
3. Install Jboss tools



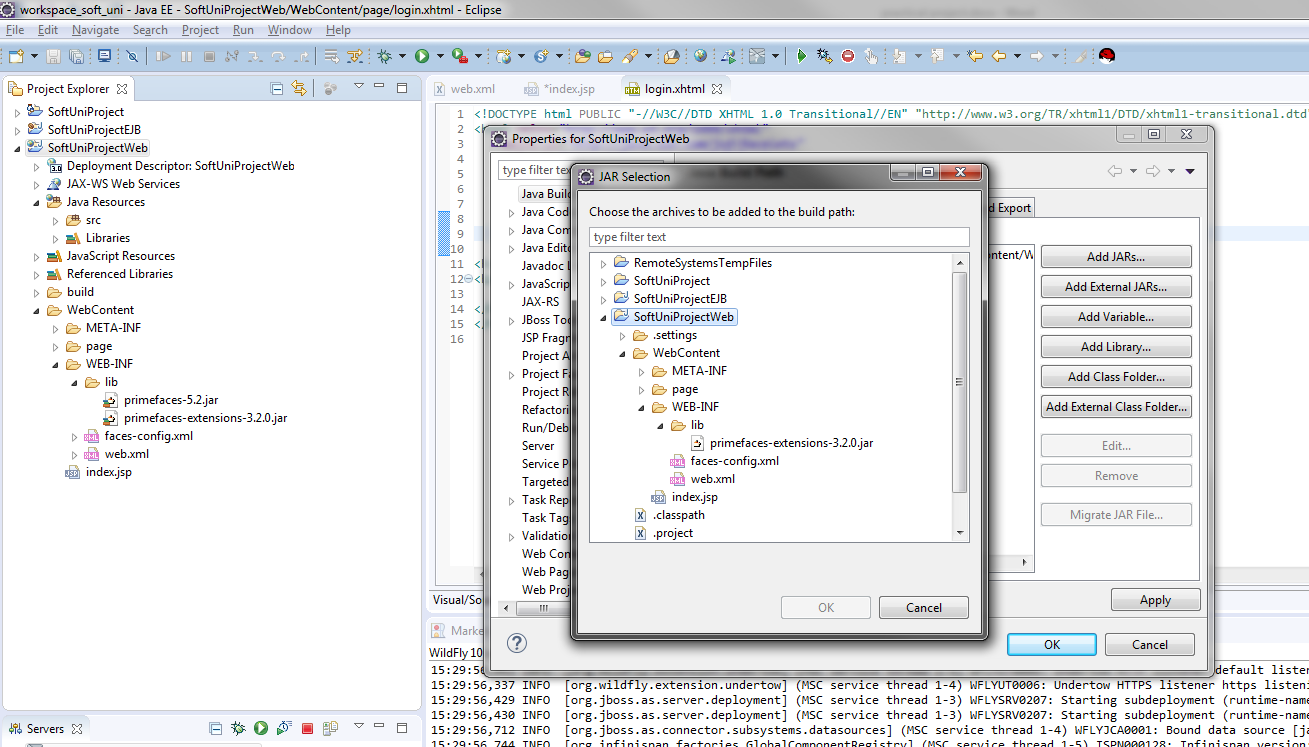


Follow the wizard and restart Eclipse

1. **Create JEE project with JSF and EJB (EJB will be used later)**
2. **Check JSF configuration in web.xml**

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"* id=*"WebApp\_ID"* version=*"3.0"*>  <display-name>SoftUniProjectWeb</display-name>  <welcome-file-list>  <welcome-file>index.jsp</welcome-file>  </welcome-file-list>    <servlet>  <servlet-name>Faces Servlet</servlet-name>  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>  <load-on-startup>1</load-on-startup>  </servlet>  <servlet-mapping>  <servlet-name>Faces Servlet</servlet-name>  <url-pattern>\*.html</url-pattern>  </servlet-mapping>  </web-app> |

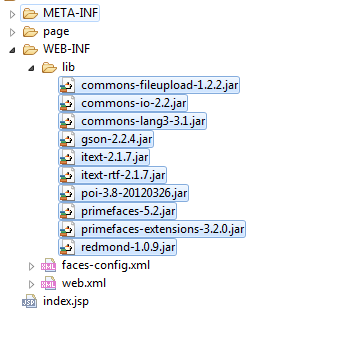
1. **Download PrimeFaces and import it into the project**

[**http://www.primefaces.org/downloads**](http://www.primefaces.org/downloads)

1. **Create login.jsp file in WebContent with the following content:**

|  |
| --- |
| <%@page contentType=*"text/html"* pageEncoding=*"UTF-8"* %>  <% response.sendRedirect(request.getContextPath() + "/page/login.html"); %> |

1. **Create folder, named page in WebContent**
2. **Create empty login.xhtml file in pages folder**
3. **Add extra libraries to the web project**



1. **Create java package and Login bean**

|  |
| --- |
| **package** bg.softuni.web;  **import** java.io.Serializable;  **import** javax.annotation.PostConstruct;  **import** javax.faces.application.FacesMessage;  **import** javax.faces.bean.ManagedBean;  **import** javax.faces.bean.RequestScoped;  **import** javax.faces.context.FacesContext;  **import** javax.inject.Inject;  **import** javax.servlet.http.HttpServletRequest;  **import** org.apache.commons.lang3.StringUtils;  /\*\*  \* The purpose of this class is to provide backend functionality of the login  \*/  @ManagedBean(name = "loginBean")  @RequestScoped  **public** **class** LoginBean **implements** Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  //beans.xml need to be added in WEB-INF in order CDI to work  @Inject  **private** HttpServletRequest request;    **private** String username;  **private** String password;    **private** **static** **final** String ***SUCCESS\_LOGIN\_REDIRECT*** = "/page/success?faces-redirect=true";  **private** **static** **final** String ***LOGIN\_PAGE\_REDIRECT*** = "/page/login?faces-redirect=true";    @PostConstruct  **public** **void** init() {  //**TODO**  }    /\*\*  \* Implement application login logic  \*  \* **@return**  \*/  **public** String login() {  **if** (StringUtils.*isBlank*(username) || StringUtils.*isBlank*(password)) {  FacesContext.*getCurrentInstance*().addMessage(**null**, **new** FacesMessage("Invalid credentials"));  **return** "";  } **else** **if**("admin".equals(username) && "123".equals(password)){  // If we want to avoid using CDI for accessing the request we can take it from FacesCondex by using:  // HttpServletRequest request = (HttpServletRequest)FacesContext.getCurrentInstance().getExternalContext().getRequest();  request.getSession().setAttribute("LOGGED\_USER", username);  **return** ***SUCCESS\_LOGIN\_REDIRECT***;  }    FacesContext.*getCurrentInstance*().addMessage(**null**, **new** FacesMessage("Invalid credentials"));  **return** "";  }  /\*\*  \* Implement application logout logic  \*  \* **@return**  \*/  **public** String logout() {  request.getSession().invalidate();  **return** ***LOGIN\_PAGE\_REDIRECT***;  }    **public** String getUsername() {  **return** username;  }  **public** **void** setUsername(String username) {  **this**.username = username;  }  **public** String getPassword() {  **return** password;  }  **public** **void** setPassword(String password) {  **this**.password = password;  }  } |

1. **Add Username and Password text boxes in login.xhtml and bind them to LoginBean.java**

|  |
| --- |
| <!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:ui=*"http://java.sun.com/jsf/facelets"*  xmlns:h=*"http://java.sun.com/jsf/html"*  xmlns:f=*"http://java.sun.com/jsf/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:pe=*"http://primefaces.org/ui/extensions"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*>  <h:head>  <link href=*"#{application.contextPath}/css/common.css"* rel=*"stylesheet"* type=*"text/css"* />  <link href=*"#{application.contextPath}/css/bootstrap.css"* rel=*"stylesheet"* type=*"text/css"* />  </h:head>  <title>Soft Uni JEE</title>  <body id=*"loginBody"*>  <h:panelGroup id=*"mainPageContent"* layout=*"block"*>  <h:form styleClass=*"loginForm"*>  <h:panelGroup>  <h3 class=*"loginTitle"*>Login Page</h3>  </h:panelGroup>  <p:messages autoUpdate=*"false"* id=*"loginDialogMessages"*  globalOnly=*"false"* styleClass=*"loginMessage"* />  <h:panelGroup layout=*"block"*>  <p:inputText value=*"#{loginBean.username}"* id=*"username"*  styleClass=*"loginInputField"* required=*"true"*  requiredMessage=*"Username is required field"*>  <p:watermark for=*"username"* value=*"Username"* />  </p:inputText>  </h:panelGroup>  <br clear=*"all"* />  <br />  <h:panelGroup layout=*"block"*>  <p:password value=*"#{loginBean.password}"* id=*"password"*  styleClass=*"loginInputField"* required=*"true"*  requiredMessage=*"Password is required field"*>  <p:watermark for=*"password"* value=*"Password"* />  </p:password>  </h:panelGroup>  <br clear=*"all"* />  <br />  <h:panelGroup layout=*"block"*>  <p:commandButton ajax=*"false"* action=*"#{loginBean.login}"*  styleClass=*"btn btn-info"* value=*"Login"* />  </h:panelGroup>  </h:form>  </h:panelGroup>  </body>  </html> |

1. **Add Success Page in pages folder**

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:ui=*"http://java.sun.com/jsf/facelets"*  xmlns:h=*"http://java.sun.com/jsf/html"*  xmlns:f=*"http://java.sun.com/jsf/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:pe=*"http://primefaces.org/ui/extensions"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*>  <h:head>  <link href=*"#{application.contextPath}/css/common.css"* rel=*"stylesheet"*  type=*"text/css"* />  <link href=*"#{application.contextPath}/css/bootstrap.css"*  rel=*"stylesheet"* type=*"text/css"* />  </h:head>  <title>Soft Uni JEE</title>  <body>  <h:form styleClass=*"welcomeForm"*>  <b>Welcome #{LOGGED\_USER}</b>    <br clear=*"all"* />  <br />  <h:panelGroup layout=*"block"*>  <p:commandButton ajax=*"false"* action=*"#{loginBean.logout}"*  styleClass=*"btn btn-info"* value=*"Logout"* style="margin: *5% 0 5% 0*;" />  </h:panelGroup>  </h:form>  </body>  </html> |

1. **Add error handling**

Error handling was already added in the above examples

* 1. **In login.xhtml the following line displays any messages on the page:**

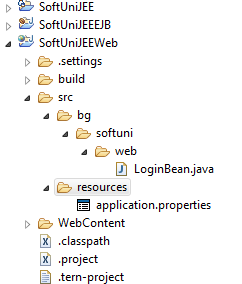
<p:messages autoUpdate=*"false"* id=*"loginDialogMessages"*globalOnly=*"false"* styleClass=*"loginMessage"* />

* 1. **In LoginBean.java, the following line adds an error message in case of invalid credentials**

FacesContext.*getCurrentInstance*().addMessage(**null**, **new** FacesMessage("Invalid credentials"));

1. **Add and Configure Resource Bundle**
   1. **Create a properties file, contains message for the page and put it into the project’s resource folder**

|  |
| --- |
| version=v0.9.37  # -- JSF validation --  # standard messages (Spec\_ 2\_5\_2\_4)  javax.faces.component.UIInput.REQUIRED=**{0}** is a required field  javax.faces.component.UIInput.REQUIRED\_detail="**{0}**" is a required field  javax.faces.convert.DateTimeConverter.CONVERSION=The date value must be in format dd.MM.yyyy  javax.faces.convert.NumberConverter.CONVERSION=The value is not a number  javax.faces.validator.DoubleRangeValidator.NOT\_IN\_RANGE=**{2}**\: the value must be between **{0}** and **{1}**  javax.faces.converter.BigIntegerConverter.BIGINTEGER=**{2}** ''**{0}**'' must be a number consisting of one or more digits.  javax.faces.converter.DoubleConverter.DOUBLE=**{2}**\ "**{0}**" must be a number.  javax.faces.converter.FloatConverter.FLOAT=**{2}**\ "**{0}**" must be a number.  javax.faces.converter.IntegerConverter.INTEGER=**{2}**\ "**{0}**" must be an integer number.  javax.faces.converter.LongConverter.LONG=**{2}**\ "**{0}**" must be an integer number  javax.faces.converter.ShortConverter.SHORT=**{2}**\ "**{0}**" must be an integer number  org.apache.myfaces.calendar.CONVERSION=The date value must be in format dd.MM.yyyy  org.apache.myfaces.calendar.CONVERSION\_detail=The date value must be in format dd.MM.yyyy. "**{1}**" is invalid.  javax.faces.validator.NOT\_IN\_RANGE=the value must be between **{0}** and **{1}**.  javax.faces.validator.DoubleRangeValidator.TYPE=**{0}** value must be a number  javax.faces.validator.LengthValidator.MAXIMUM=**{1}** Invalid field\: The length should be less than ''**{0}**'' characters  javax.faces.validator.LengthValidator.MINIMUM=**{1}** Invalid field\: The length should be greater than ''**{0}**'' characters  javax.faces.validator.LongRangeValidator.MAXIMUM=**{1}** the value must be less than **{0}**  javax.faces.validator.LongRangeValidator.MINIMUM=**{1}** the value must be greater or equal to **{0}**  javax.faces.validator.LongRangeValidator.NOT\_IN\_RANGE=**{2}** the value must be between **{0}** and **{1}**  javax.faces.validator.LongRangeValidator.TYPE=**{0}** the value must be a number  javax.faces.converter.BigDecimalConverter.DECIMAL=**{2}** ''**{0}**'' must be a decimal number.  javax.faces.converter.BooleanConverter.BOOLEAN=**{1}** ''**{0}**'' must be 'true' or 'false'.  #Error messages  login.error.invalid.credentials=Invalid Credentials  #buttons labels  button.login=Login  button.logout=Logout |



* 1. **Add application.properties in faces-config.xml**

|  |
| --- |
| <application>  <locale-config>  <default-locale>en</default-locale>  <supported-locale>bg</supported-locale>  </locale-config>  <resource-bundle>  <base-name>resources.application</base-name>  <var>msgs</var>  </resource-bundle>  <message-bundle>resources.application</message-bundle>  </application> |

* 1. **Use only labels and messages from the ResourceBundle**
     1. **In web pages:**

**#{msgs['button.login']}**

* + 1. **In Java classes:**

Add MessageUtils.java with the following contents:

|  |
| --- |
| package bg**.**softuni**.**web**.**utils**;**  **package** bg.softuni.web.utils;  **import** java.text.MessageFormat;  **import** java.util.ResourceBundle;  **import** javax.faces.application.FacesMessage;  **import** javax.faces.context.FacesContext;  /\*\*  \* Common utility methods to use message resources.  \*  \* **@author** Georgi Novakov  \*/  **public** **class** MessageUtils {  **private** **final** **static** String ***RESOURCE\_BUNDLE\_FILE*** = "resources.application";  **private** **final** **static** ResourceBundle ***RESOURCE\_BUNDLE*** = *getBundle*();  **public** **static** String getMessage(String aKey) {  **return** ***RESOURCE\_BUNDLE***.getString(aKey);  }  /\*\*  \* **@param** params positional parameters to be inserted in "{0}, {1}, etc."  \*/  **public** **static** String getMessage(String aKey, Object... params) {  **return** MessageFormat.*format*(***RESOURCE\_BUNDLE***.getString(aKey), params);  }  **public** **static** **void** addMessage(String clientId, FacesMessage msg) {  FacesContext.*getCurrentInstance*().addMessage(clientId, msg);  }  **public** **static** **void** addMessage(String aMsgId) {  *addMessage*(**null**, **new** FacesMessage(*getMessage*(aMsgId)));  }  **public** **static** **void** addMessage(String aMsgId, Object... params) {  *addMessage*(**null**, **new** FacesMessage(*getMessage*(aMsgId, params)));  }  **public** **static** **void** addErrorMessage(String aMsgId) {  *addMessage*(**null**, **new** FacesMessage(FacesMessage.***SEVERITY\_ERROR***, *getMessage*(aMsgId), **null**));  }  **public** **static** **void** addErrorMessage(String aMsgId, Object... params) {  *addMessage*(**null**, **new** FacesMessage(FacesMessage.***SEVERITY\_ERROR***, *getMessage*(aMsgId, params), **null**));  }  **public** **static** **void** addErrorMessage(String clientId, String aMsgId, Object... params) {  *addMessage*(clientId, **new** FacesMessage(FacesMessage.***SEVERITY\_ERROR***, *getMessage*(aMsgId, params), **null**));  }  **private** **static** ResourceBundle getBundle() {  **if** (FacesContext.*getCurrentInstance*() != **null**) {  **return** ResourceBundle.*getBundle*(FacesContext.*getCurrentInstance*().getApplication().getMessageBundle());  }  **return** ResourceBundle.*getBundle*(***RESOURCE\_BUNDLE\_FILE***);  }  } |

**Replace**

FacesContext.getCurrentInstance().addMessage(null, new FacesMessage("Invalid credentials"));

**With**

MessageUtils.*addErrorMessage*("login.error.invalid.credentials");

In LoginBean.java and add login.error.invalid.credentials to application.properties

1. **Add and configure Authentication filter**
   1. **Create AuthenticationFilter.java with the following content**

|  |
| --- |
| **package** bg.softuni.web.filter;  **import** java.io.IOException;  **import** java.io.Serializable;  **import** java.util.List;  **import** java.util.Properties;  **import** javax.inject.Inject;  **import** javax.servlet.Filter;  **import** javax.servlet.FilterChain;  **import** javax.servlet.FilterConfig;  **import** javax.servlet.RequestDispatcher;  **import** javax.servlet.ServletException;  **import** javax.servlet.ServletRequest;  **import** javax.servlet.ServletResponse;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpSession;  /\*\*  \* **@author** gnovakov  \*/  **public** **class** AuthenticationFilter **implements** Filter, Serializable {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;    **public** **static** **final** String ***PATH\_INDEX*** = "/index.jsp";  **public** **static** **final** String ***PATH\_LOGIN*** = "/page/login.html";  **public** **static** **final** String ***PATH\_LOGOUT*** = "/page/logout.jsp";    **public** **void** init(FilterConfig filterConfig) **throws** ServletException {  }  **public** **void** destroy() {  }  /\*\*  \* **@param** request  \* **@param** response  \* **@param** chain  \* **@throws** IOException  \* **@throws** ServletException  \*/  **public** **void** doFilter(ServletRequest request, ServletResponse response, FilterChain chain)  **throws** IOException, ServletException {  HttpServletRequest httpRequest = (HttpServletRequest) request;  String requestedPath = httpRequest.getRequestURI().substring(httpRequest.getContextPath().length());  /\*  \* Skip the action with path "index.jsp" - there is no logged user, but  \* the user is redirected to login screen  \*/  **if** (***PATH\_INDEX***.equals(requestedPath)) {  chain.doFilter(request, response);  **return**;  }  /\*  \* Skip the action with path "/login.login.html" - there is no logged  \* user, but the user is trying to log in  \*/  **if** (***PATH\_LOGIN***.equals(requestedPath)) {  chain.doFilter(request, response);  **return**;  }    /\*  \* Get logged user from the HttpSession  \*/  HttpSession session = httpRequest.getSession();  String loggedUser = (String) session.getAttribute("LOGGED\_USER");  /\*  \* Redirect to login page if there is no logged user and trying to access protected resource  \*/  **if** (loggedUser == **null**) {  RequestDispatcher requestDispatcher = request.getRequestDispatcher(***PATH\_LOGIN***);  requestDispatcher.forward(request, response);  **return**;  } **else** {  chain.doFilter(request, response);  **return**;  }  }  } |

* 1. **Configure AuthenticationFilter.java in web.xml**

|  |
| --- |
| <filter>  <filter-name>AuthenticationFilter</filter-name>  <filter-class>bg.softuni.web.filter.AuthenticationFilter</filter-class>  </filter>  <filter-mapping>  <filter-name>AuthenticationFilter</filter-name>  <url-pattern>/page/\*</url-pattern>  </filter-mapping> |

1. **List users in DataTable**
   1. **Create UserDto class with the following contents**

|  |
| --- |
| **package** bg.softuni.dto;  **import** java.io.Serializable;  **public** **class** UserDto **implements** Serializable {    **private** **static** **final** **long** ***serialVersionUID*** = 1L;    **private** String username;  **private** String password;  **private** String firstName;  **private** String lastName;  **private** String email;  **public** UserDto() {  **super**();  }  **public** UserDto(String username, String password, String firstName, String lastName, String email) {  **super**();  **this**.username = username;  **this**.password = password;  **this**.firstName = firstName;  **this**.lastName = lastName;  **this**.email = email;  }  **public** String getUsername() {  **return** username;  }  **public** **void** setUsername(String username) {  **this**.username = username;  }  **public** String getPassword() {  **return** password;  }  **public** **void** setPassword(String password) {  **this**.password = password;  }  **public** String getFirstName() {  **return** firstName;  }  **public** **void** setFirstName(String firstName) {  **this**.firstName = firstName;  }  **public** String getLastName() {  **return** lastName;  }  **public** **void** setLastName(String lastName) {  **this**.lastName = lastName;  }  **public** String getEmail() {  **return** email;  }  **public** **void** setEmail(String email) {  **this**.email = email;  }  } |

* 1. **Create session managed bean which will hold all users in an ArrayList**

|  |
| --- |
| **package** bg.softuni.web.beans;  **import** java.util.ArrayList;  **import** java.util.List;  **import** javax.annotation.PostConstruct;  **import** javax.faces.bean.ManagedBean;  **import** javax.faces.bean.SessionScoped;  **import** org.apache.commons.lang3.StringUtils;  **import** bg.softuni.dto.UserDto;  @ManagedBean(name = "loginBean")  @SessionScoped  **public** **class** UsersBean {  **private** List<UserDto> users;  @PostConstruct  **public** **void** init() {  // initialize the users list  users = **new** ArrayList<UserDto>();  // create user  UserDto initUser = **new** UserDto("admin", "123", "Georgi", "Novakov", "georgi@gmail.com");  // add user the created user to the list  users.add(initUser);  }  **public** UserDto validateUser(String username, String password) {  **if** (StringUtils.*isBlank*(username) || StringUtils.*isBlank*(password)) {  **return** **null**;  }  **for** (UserDto currentUser : users) {  **if** (username.equals(currentUser.getUsername()) && password.equals(currentUser.getPassword())) {  **return** currentUser;  }  }  **return** **null**;  }  **public** List<UserDto> getUsers() {  **return** users;  }  **public** **void** setUsers(List<UserDto> users) {  **this**.users = users;  }  } |

* 1. **Modify LoginBean to use UsersBean**

|  |
| --- |
| package bg**.**softuni**.**web**.**beans**;**  **import** java**.**io**.**Serializable**;**  **import** javax**.**annotation**.**PostConstruct**;**  **import** javax**.**faces**.**application**.**FacesMessage**;**  **import** javax**.**faces**.**bean**.**ManagedBean**;**  **import** javax**.**faces**.**bean**.**ManagedProperty**;**  **import** javax**.**faces**.**bean**.**RequestScoped**;**  **import** javax**.**faces**.**context**.**FacesContext**;**  **import** javax**.**inject**.**Inject**;**  **import** javax**.**servlet**.**http**.**HttpServletRequest**;**  **import** org**.**apache**.**commons**.**lang3**.**StringUtils**;**  **import** bg**.**softuni**.**dto**.**UserDto**;**  **import** bg**.**softuni**.**web**.**utils**.**MessageUtils**;**  /\*\*  \* The purpose of this class is to provide backend functionality of the login  \*/  @ManagedBean**(**name **=** "loginBean"**)**  @RequestScoped  public class LoginBean **implements** Serializable **{**  private static final long serialVersionUID **=** 1L**;**  // beans.xml need to be added in WEB-INF in order CDI to work  @Inject  private HttpServletRequest request**;**  private String username**;**  private String password**;**  @ManagedProperty**(**"#{usersBean}"**)**  private UsersBean usersBean**;**  private static final String SUCCESS\_LOGIN\_REDIRECT **=** "/page/success?faces-redirect=true"**;**  private static final String LOGIN\_PAGE\_REDIRECT **=** "/page/login?faces-redirect=true"**;**  @PostConstruct  public void init**()** **{**  // TODO  **}**  /\*\*  \* Implement application login logic  \*  \* **@return**  \*/  public String login**()** **{**  UserDto user **=** usersBean**.**validateUser**(**username**,** password**);**  **if** **(null** **==** user**)** **{**  MessageUtils**.**addErrorMessage**(**"login.error.invalid.credentials"**);**  **return** ""**;**  **}** **else** **{**  // If we want to avoid using CDI for accessing the request we can  // take it from FacesCondex by using:  // HttpServletRequest request =  // (HttpServletRequest)FacesContext.getCurrentInstance().getExternalContext().getRequest();  request**.**getSession**().**setAttribute**(**"LOGGED\_USER"**,** user**);**  **return** SUCCESS\_LOGIN\_REDIRECT**;**  **}**  **}**  /\*\*  \* Implement application logout logic  \*  \* **@return**  \*/  public String logout**()** **{**  request**.**getSession**().**invalidate**();**  **return** LOGIN\_PAGE\_REDIRECT**;**  **}**  public String getUsername**()** **{**  **return** username**;**  **}**  public void setUsername**(**String username**)** **{**  **this.**username **=** username**;**  **}**  public String getPassword**()** **{**  **return** password**;**  **}**  public void setPassword**(**String password**)** **{**  **this.**password **=** password**;**  **}**  public UsersBean getUsersBean**()** **{**  **return** usersBean**;**  **}**  public void setUsersBean**(**UsersBean usersBean**)** **{**  **this.**usersBean **=** usersBean**;**  **}**  **}** |

* 1. **Create listUsers.xhtml and add DataTable to list all users**

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns=*"http://www.w3.org/1999/xhtml"*  xmlns:ui=*"http://java.sun.com/jsf/facelets"*  xmlns:h=*"http://java.sun.com/jsf/html"*  xmlns:f=*"http://java.sun.com/jsf/core"*  xmlns:p=*"http://primefaces.org/ui"*  xmlns:pe=*"http://primefaces.org/ui/extensions"*  xmlns:c=*"http://java.sun.com/jsp/jstl/core"*>  <h:head>  <link href=*"#{application.contextPath}/css/common.css"* rel=*"stylesheet"* type=*"text/css"* />  <link href=*"#{application.contextPath}/css/bootstrap.css"* rel=*"stylesheet"* type=*"text/css"* />  </h:head>  <title>Soft Uni JEE</title>  <body id=*"commonBody"*>  <h:form styleClass=*"commonForm"*>  <h:panelGroup layout=*"block"*>  <b>Welcome #{LOGGED\_USER.firstName} #{ LOGGED\_USER.lastName}</b>  <p:commandButton ajax=*"false"* action=*"#{loginBean.logout}"*  styleClass=*"btn btn-info"* value=*"#{msgs['button.logout']}"*  style="margin: *5% 5% 5% 5%*;" />  </h:panelGroup>  <h:panelGroup layout=*"block"*>  <p:dataTable  var=*"user"*  value=*"#{usersBean.users}"*  rows=*"10"*  widgetVar=*"usersTable"* paginator=*"true"*  paginatorTemplate=*"{CurrentPageReport} {FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink} {RowsPerPageDropdown}"*  paginatorPosition=*"bottom"*  rowsPerPageTemplate=*"5,10,15"*  styleClass=*"listUsersTable"*  id=*"listUsersTable"* >  <f:facet name=*"header"*>  <p:outputPanel>  <h:outputText value=*"Search all fields:"* />  <p:inputText id=*"globalFilter"* onkeyup="PF('usersTable').filter()"  style="width:*150px*" placeholder=*"Enter keyword"* />  </p:outputPanel>  </f:facet>  <p:column headerText=*"#{msgs['user.username']}"*  filterBy=*"#{user.username}"* filterMatchMode=*"startsWith"*>  <h:outputText value=*"#{user.username}"* />  </p:column>  <p:column headerText=*"#{msgs['user.firstname']}"*  filterBy=*"#{user.firstName}"*  filterMatchMode=*"startsWith"*>  <h:outputText value=*" #{user.firstName}"* />  </p:column>  <p:column headerText=*"#{msgs['user.lastname']}"*  filterBy=*"#{user.lastName}"*  filterMatchMode=*"startsWith"*>  <h:outputText value=*" #{user.lastName}"* />  </p:column>  <p:column headerText=*"#{msgs['user.email']}"*  filterBy=*"#{user.email}"*  filterMatchMode=*"startsWith"*>  <h:outputText value=*" #{user.email}"* />  </p:column>  </p:dataTable>  </h:panelGroup>  </h:form>  </body>  </html> |