**ADF\_Validation\_Framework**

**Introduction**

Oracle ADF is an extremely rich Java EE applications framework with highly rich set of UI component and business component support using drag and drop functionality. ADF provide in-build Validation and custom component to check the runtime component behavior.

**ADF Validation framework**

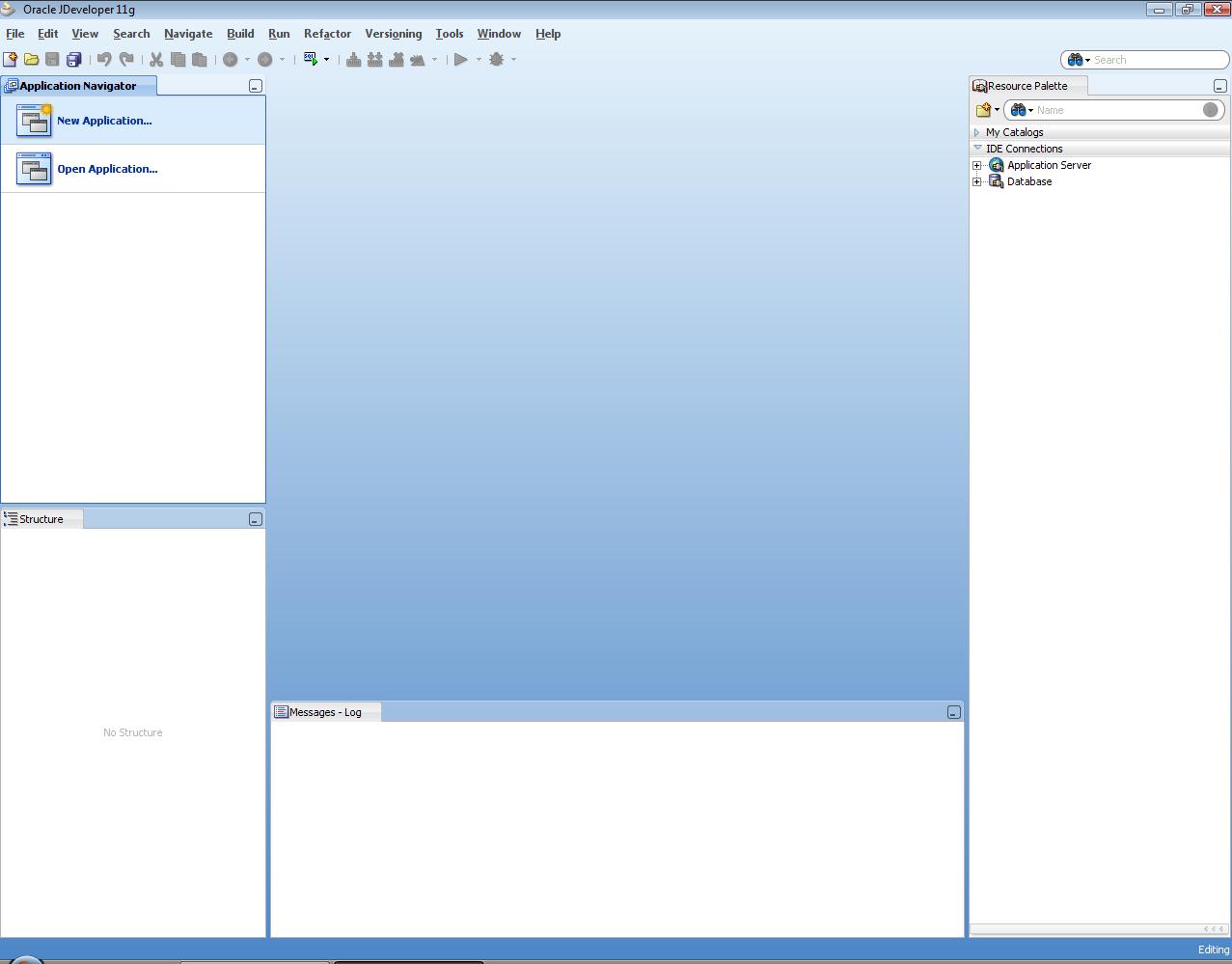
In Web application usually Data comes from the Clients, usually validated before being processed by the Server. In ADF 4 Core Tags are used for doing Validation on the ADF UI Components.

* **Validate Length Tag**: is used to specify the maximum and the minimum characters
* **Validate Long Range Tag**: is used to specify expected value fall between certain integer (long) ranges.
* **Validate Double Range Tag**: is used to specify excepted value fall between certain on floating Data
* **Validator Tag**: is used to specify customized validation on UI Components

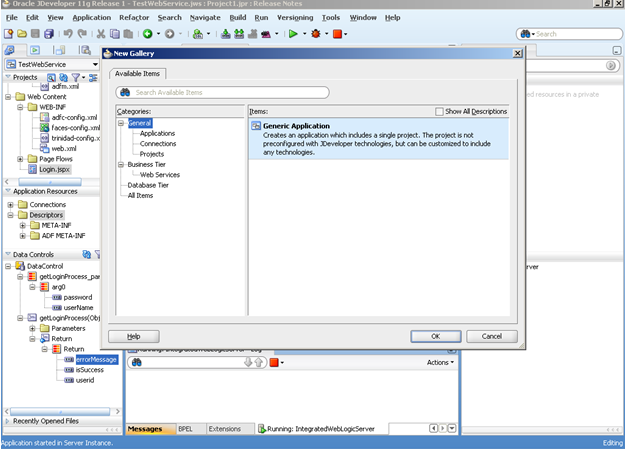
The below demo show how to use validate length tag and an email custom validation tag used to validate email feild.

**Creating new ADF application having Registration page with validations**

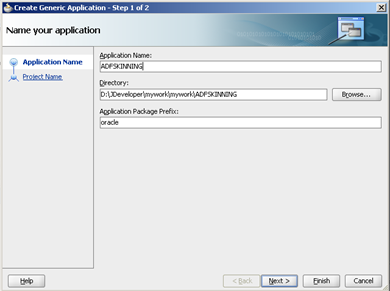
1. In order to create a new application we need to open a create application wizard. To open the wizard, in the application navigator, choose **New Application**.



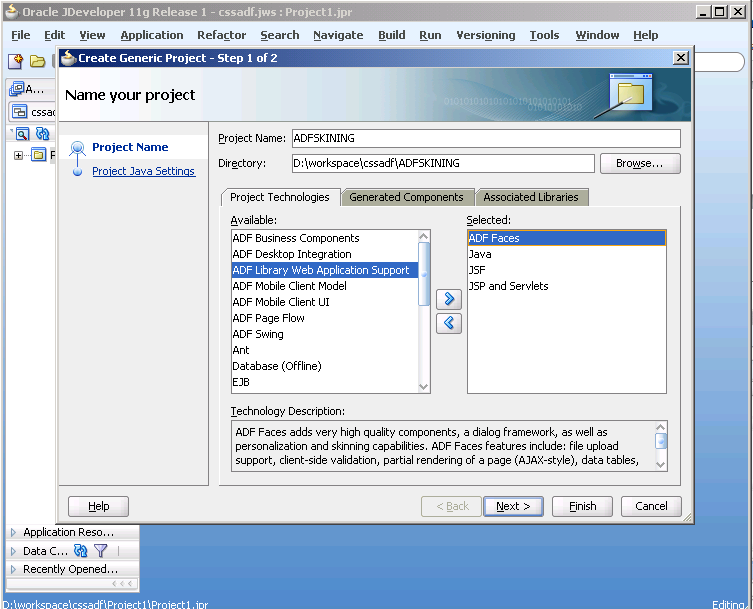
1. When you click on the **New Application** a wizard starts running and a dialog appears. Click on Generic Application and press OK button.



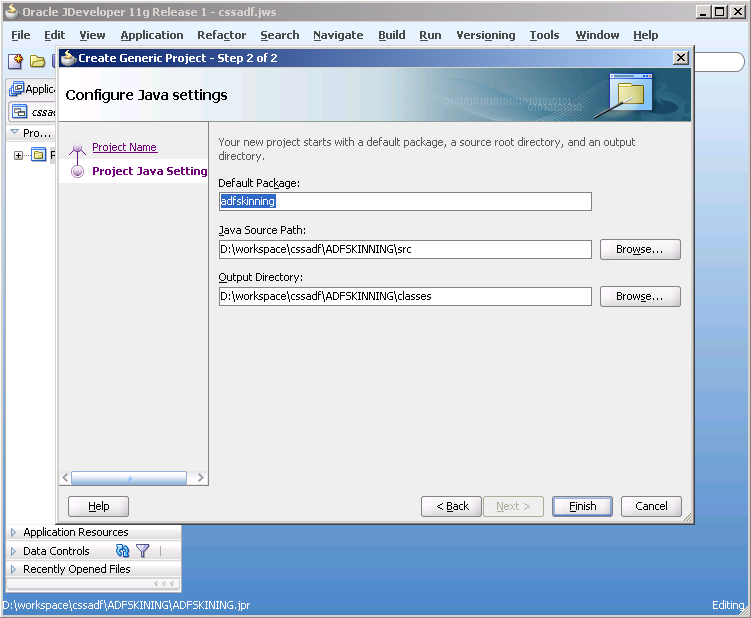
1. Enter the application name as “**ADFSKINNING*”****.* This will be the workspace and will be stored as .jws file.



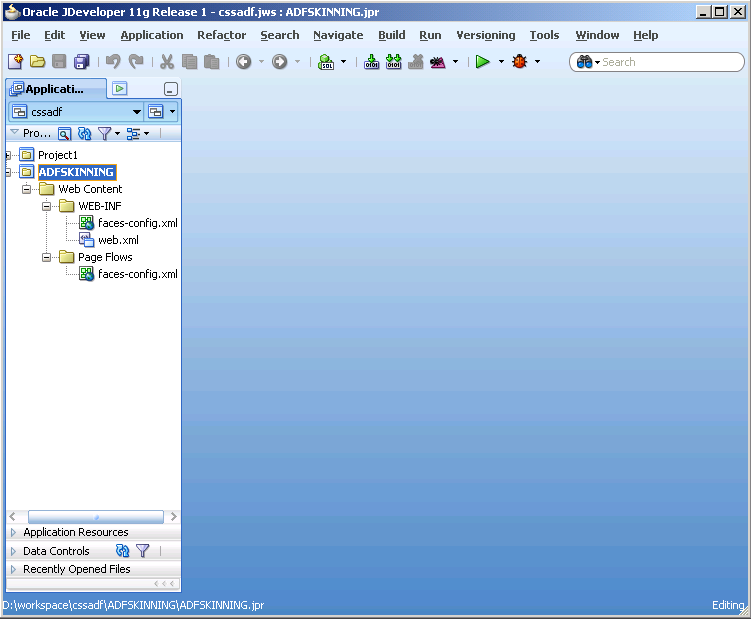
1. Enter the project name or leave it as default and then select the technology ADF Faces. Click Next to continue.



1. Enter the package name or leave it as default.

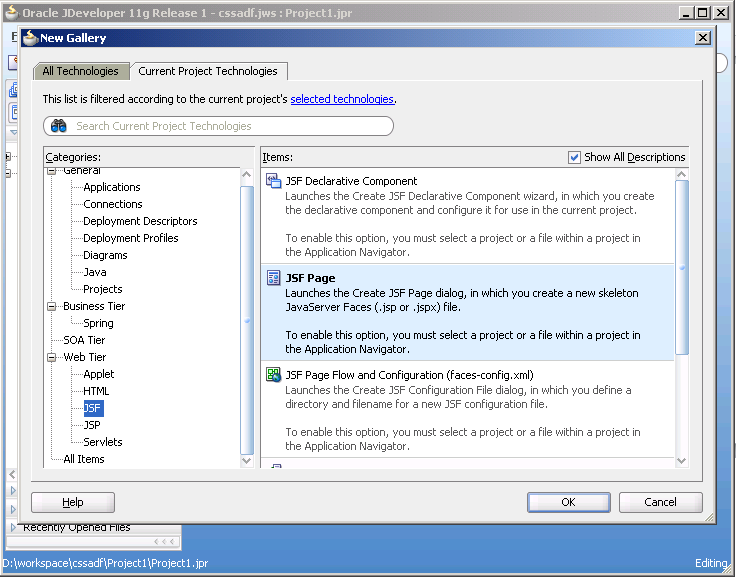


1. Click finish to create the project. The application is created and the following screen appears having one folder “**ADFSKINNING*”***.

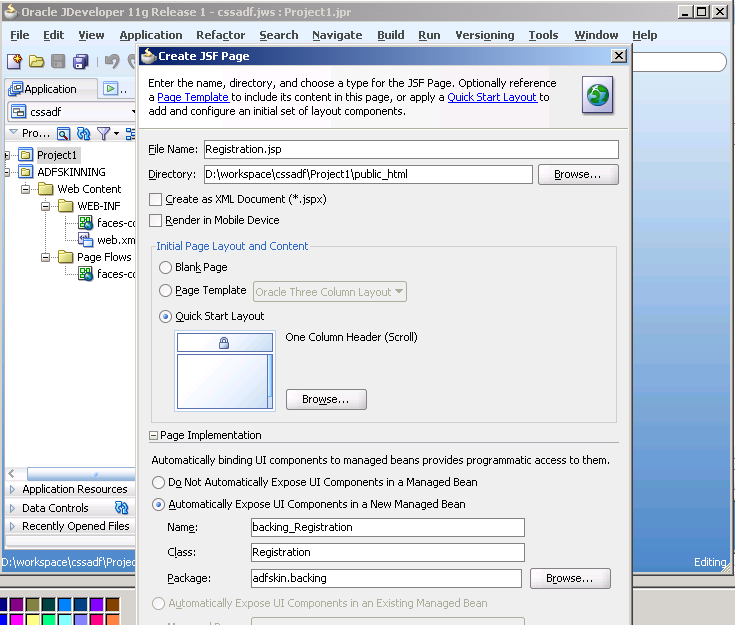


**Creating the ADF RICH User web Interface**

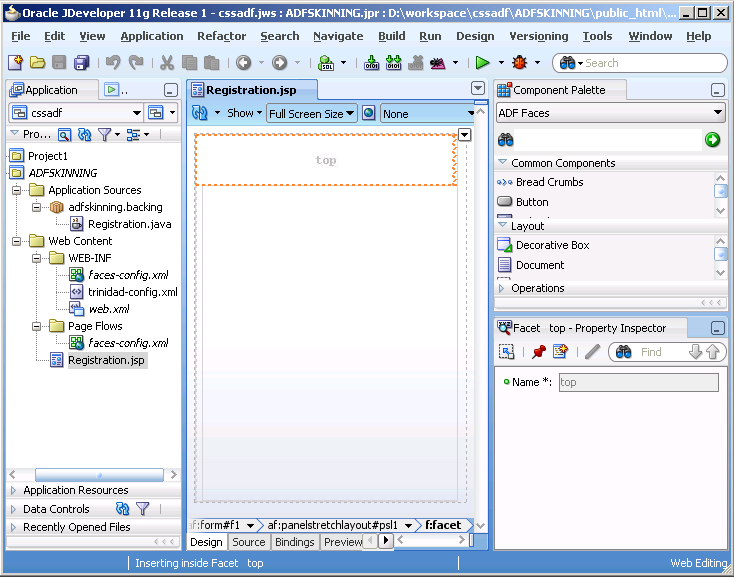
1. In Application Navigator, right click on **ADFSKINNING;** from the popup menu choose new. The New Gallery window appears and select on ADF in the left tab and click on ADF Page in the right window.



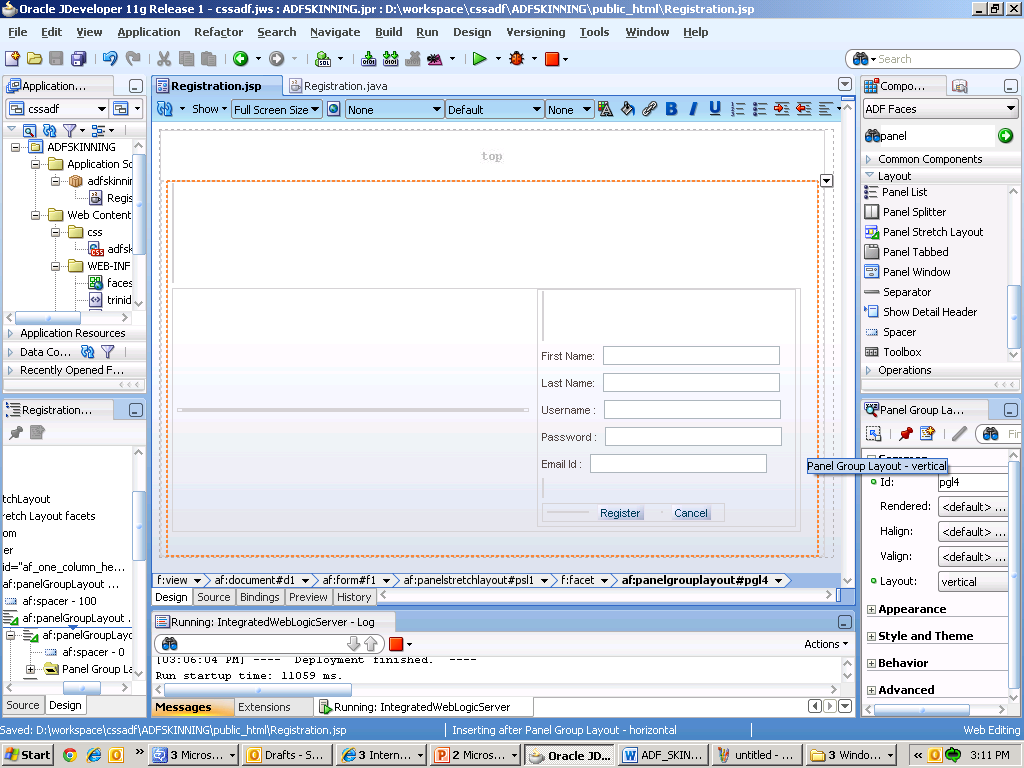
1. In this window create an ADF page Registration.jspx and choose automatically expose UI component in a new managed bean and click on Ok button.



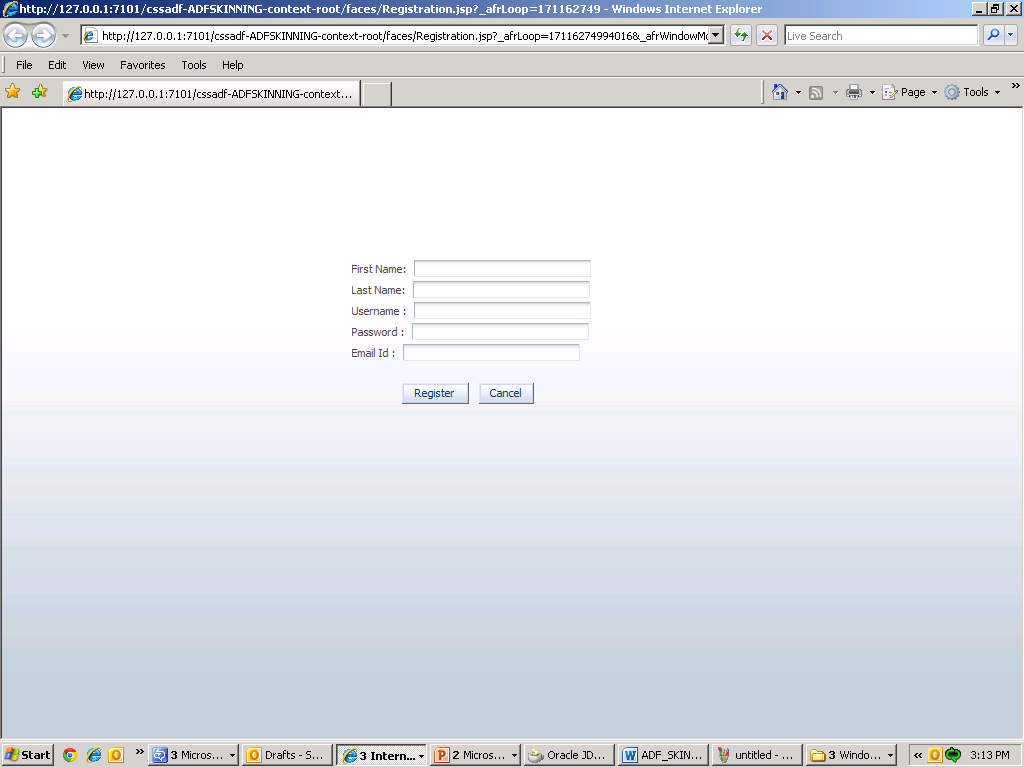
1. Now go to registration page and open it.



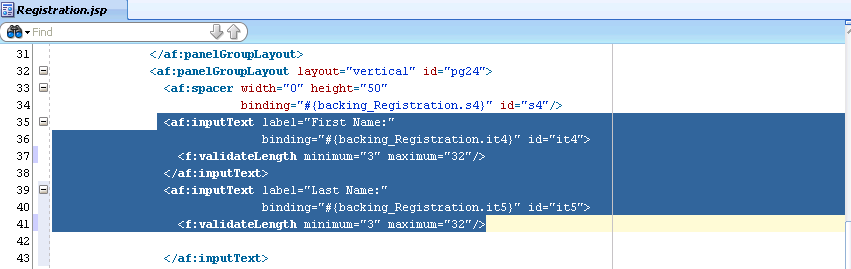
1. Design the registration page by draging the input field component for creating first name, lastname, username, email id and password fields for registration



1. Now run the Registration page. Following page is opened on running the page.



1. Now we need to add validation on the Registration page. On email field we add custom validation and on first and last name field we add minimum and maximum length validation.

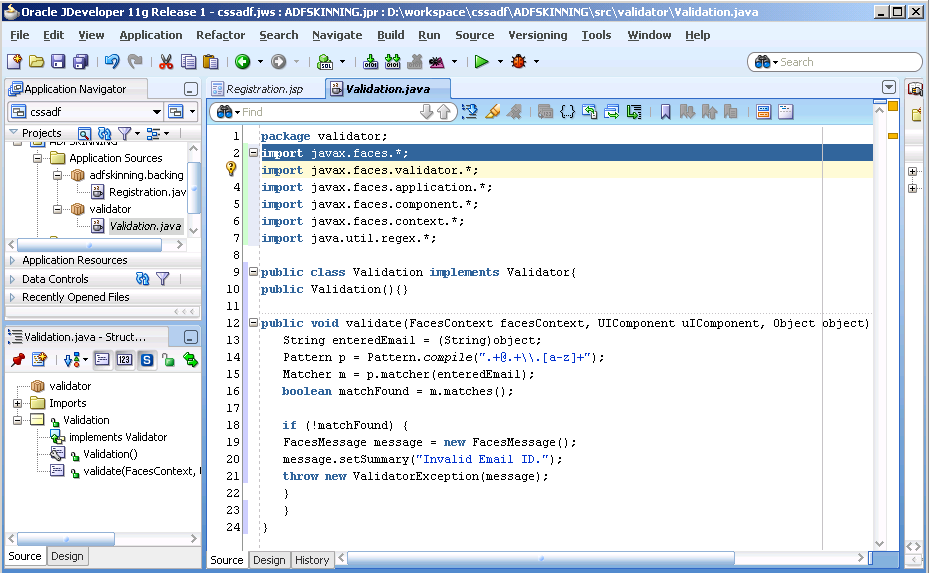


1. If validation for the component value is not supported by the standard validators then you will need to create your own validator. For example, you can create email validator to validate the email text entered by the user i.e. whether the email id is in correct format. Using custom validator you will be able to display your own validation error message.

Steps to create a custom validator:

1. Create a class that implements the javax.faces.validator.Validator interface.

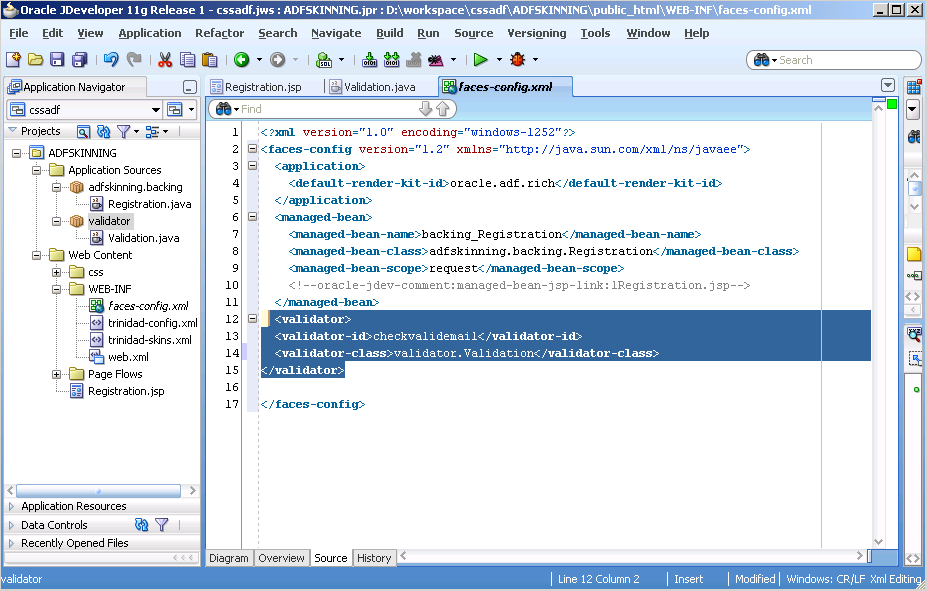
2. Implement the validate() method (defined in Validator interface).



Code for email validation

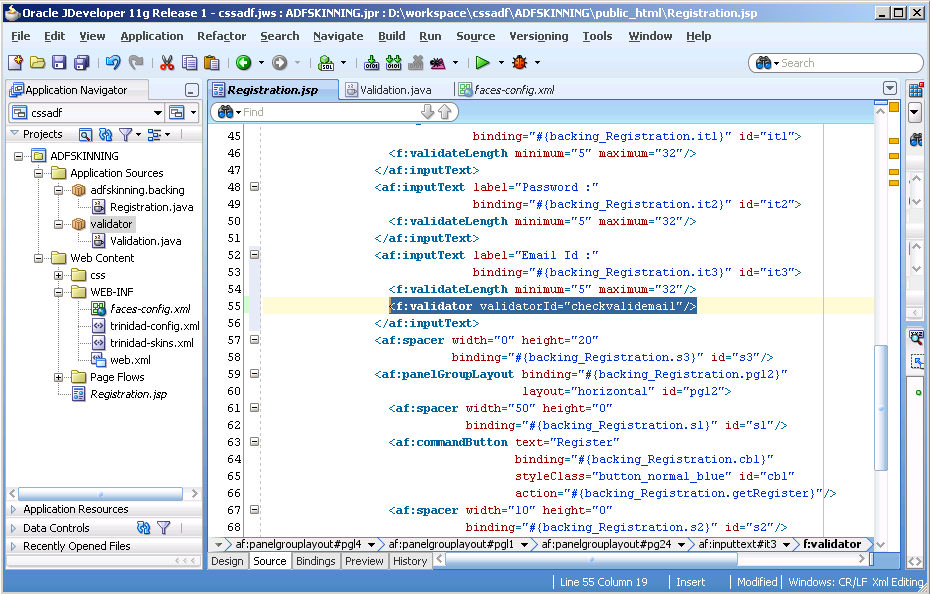
|  |
| --- |
| package validator;  import javax.faces.validator.\*;  import javax.faces.application.\*;  import javax.faces.component.\*;  import javax.faces.context.\*;  import java.util.regex.\*;  public class Validation implements Validator{  public Validation(){}  public void **validate(FacesContext facesContext, UIComponent uIComponent, Object object) throws ValidatorException{**  String enteredEmail = (String)object;  Pattern p = Pattern.compile(".+@.+\\.[a-z]+");  Matcher m = p.matcher(enteredEmail);  boolean matchFound = m.matches();  if (!matchFound) {  FacesMessage message = new FacesMessage();  message.setSummary("Invalid Email ID.");  throw new ValidatorException(message);  }  }  } |

1. Register this validator in the configuration file "faces-confix.xml".

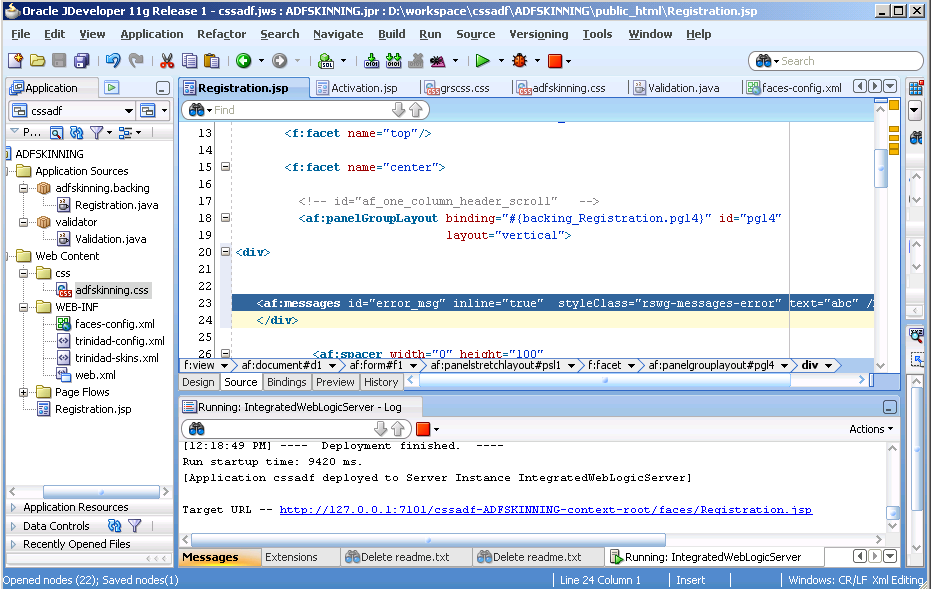


|  |
| --- |
| <?xml version="1.0" encoding="windows-1252"?>  <faces-config version="1.2" xmlns="http://java.sun.com/xml/ns/javaee">  <application>  <default-render-kit-id>oracle.adf.rich</default-render-kit-id>  </application>  <managed-bean>  <managed-bean-name>backing\_Registration</managed-bean-name>  <managed-bean-class>adfskinning.backing.Registration</managed-bean-class>  <managed-bean-scope>request</managed-bean-scope>  <!--oracle-jdev-comment:managed-bean-jsp-link:1Registration.jsp-->  </managed-bean>  **<validator>**  **<validator-id>checkvalidemail</validator-id>**  **<validator-class>validator.Validation</validator-class>**  **</validator>**  </faces-config> |

1. Use this validator by <f:validator/> tag in the JSP page.

****

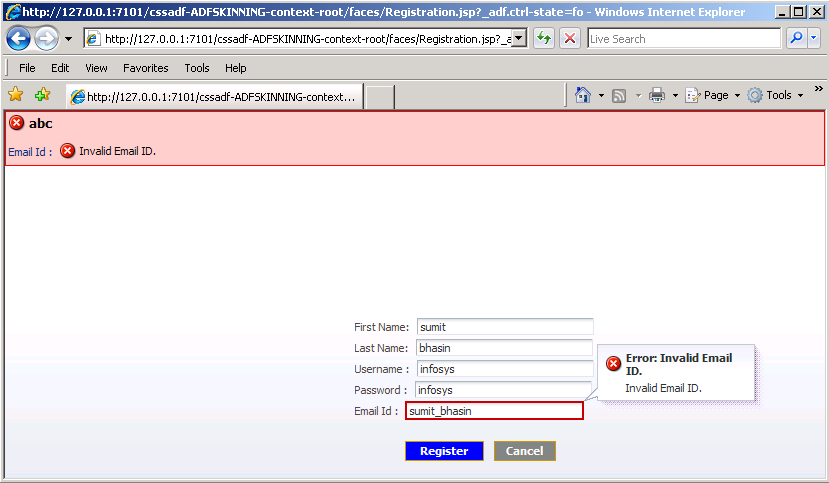
1. Use <af:messages> tag in the jsp page to display the error message.

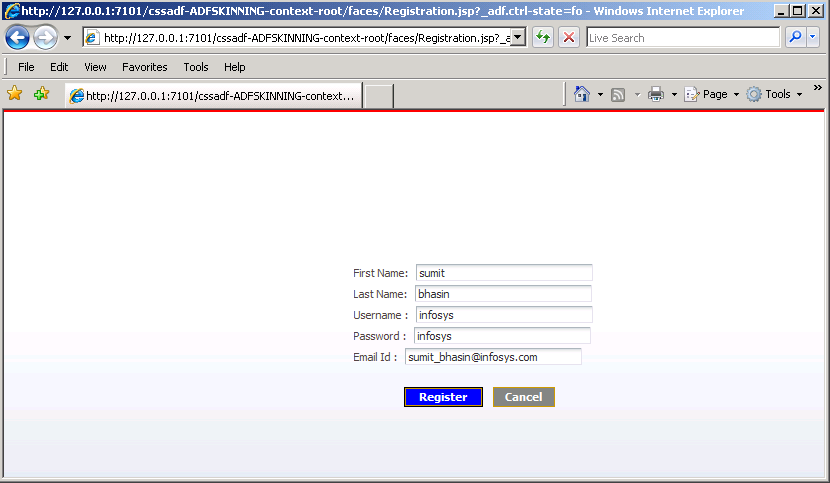


|  |
| --- |
| <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  "http://www.w3.org/TR/html4/loose.dtd">  <%@ page contentType="text/html;charset=UTF-8"%>  <%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%>  <%@ taglib uri="http://java.sun.com/jsf/html" prefix="h"%>  <%@ taglib uri="http://xmlns.oracle.com/adf/faces/rich" prefix="af"%>  <f:view>  <af:document id="d1" binding="#{backing\_Registration.d1}">  <af:form id="f1" binding="#{backing\_Registration.f1}">    <af:panelStretchLayout topHeight="50px" id="psl1"  binding="#{backing\_Registration.psl1}">  <f:facet name="top"/>  <f:facet name="center">    <!-- id="af\_one\_column\_header\_scroll" -->  <af:panelGroupLayout binding="#{backing\_Registration.pgl4}" id="pgl4"  layout="vertical">  <div>      <af:messages id="error\_msg" inline="true" styleClass="rswg-messages-error" text="abc" />  </div>    <af:spacer width="0" height="100"  binding="#{backing\_Registration.s6}" id="s6"/>  <af:panelGroupLayout layout="horizontal" id="pgl1"  binding="#{backing\_Registration.pgl1}">  <af:panelGroupLayout binding="#{backing\_Registration.pgl3}"  id="pgl3">  <af:spacer width="350" height="0"  binding="#{backing\_Registration.s5}" id="s5"/>  </af:panelGroupLayout>  <af:panelGroupLayout layout="vertical" id="pg24">  <af:spacer width="0" height="50"  binding="#{backing\_Registration.s4}" id="s4"/>  <af:inputText label="First Name:"  binding="#{backing\_Registration.it4}" id="it4">  <f:validateLength minimum="3" maximum="32"/>  </af:inputText>  <af:inputText label="Last Name:"  binding="#{backing\_Registration.it5}" id="it5">  <f:validateLength minimum="3" maximum="32"/>    </af:inputText>  <af:inputText label="Username :"  binding="#{backing\_Registration.it1}" id="it1">  **<f:validateLength minimum="5" maximum="32"/>**  </af:inputText>  <af:inputText label="Password :"  binding="#{backing\_Registration.it2}" id="it2">  <f:validateLength minimum="5" maximum="32"/>  </af:inputText>  <af:inputText label="Email Id :"  binding="#{backing\_Registration.it3}" id="email">  <f:validateLength minimum="5" maximum="32"/>  **<f:validator validatorId="checkvalidemail"/>**  </af:inputText>    <af:spacer width="0" height="20"  binding="#{backing\_Registration.s3}" id="s3"/>  <af:panelGroupLayout binding="#{backing\_Registration.pgl2}"  layout="horizontal" id="pgl2">  <af:spacer width="50" height="0"  binding="#{backing\_Registration.s1}" id="s1"/>  <af:commandButton text="Register"  binding="#{backing\_Registration.cb1}"  styleClass="button\_normal\_blue" id="cb1"  action="#{backing\_Registration.getRegister}"/>  <af:spacer width="10" height="0"  binding="#{backing\_Registration.s2}" id="s2"/>  <af:commandButton text="Cancel"  binding="#{backing\_Registration.cb2}"  styleClass="button\_normal\_grey" id="cb2"/>  </af:panelGroupLayout>  </af:panelGroupLayout>  </af:panelGroupLayout>  </af:panelGroupLayout>  </f:facet>  </af:panelStretchLayout>  </af:form>  </af:document>  </f:view> |

**Output Flow:-**

Now run the Registration.jsp page and on entering wrong email address an error message will populate on the screen on click on register button and on entering correct email address registration process will successfully executed .





**Reference(s)**

1. Oracle Jdeveloper 11g Handbook, Dr. Avrom Roy-Faderman, Peter Koletzke, Dr. Paul Dorsey

2. Jdeveloper 11g HELP menu