**\*\*LOVSwitcher Method in EmployeesRowImpl:**

/\*\*

\* Gets the attribute value for the calculated attribute DepartmentLOVSwitcher.

\* @return the DepartmentLOVSwitcher

\*/

public String getDepartmentLOVSwitcher() {

if(getSalary() != null && getSalary().doubleValue() <= 5000){

return"LOV\_DepartmentIdLessThan50";

}else{

return"LOV\_DepartmentIdGreterThan50";

}

// return (String) getAttributeInternal(DEPARTMENTLOVSWITCHER);

}

**\*\*Method Validations:**

public class EmployeesImpl extends EntityImpl {

/\*\*

\* Validation method for PhoneNumber.

\*/

**public boolean validatePhoneNumber(String phonenumber) {**

if(phonenumber != null && (phonenumber.length() < 9 || phonenumber.length() > 12)){

return false;

}

return true;

}

/\*\*

\* Validation method for Employees.

\*/

**public boolean validateEmployeesSalary() {**

BigDecimal salary = getSalary();

if(salary != null && getDepartmentId() != null && salary.doubleValue() > 50000 && getDepartmentId() <= 50){

return false;

}

return true;

}

/\*\*

Validation method for Employees.

\*/

**public boolean validateEmployees() {**

if(getFirstName() != null && getLastName() != null && getFirstName().equals(getLastName())){

return false;

}

return true;

}

**\*\*Autosuggested code for (Inputtext):**

**public static Connection getConnection() {**

try { InitialContext initialContext = new InitialContext();

DataSource ds = (DataSource) initialContext.lookup("hrDS");

Connection conn = ds.getConnection();

return conn;

} catch (Exception e) {

e.printStackTrace(); }

return null;

}

**public List firstNameItems(FacesContext facesContext, AutoSuggestUIHints autoSuggestUIHints) {**

// Add event code here...

List<SelectItem> items = new ArrayList<SelectItem>();

String sql =

"select first\_name from employees where first\_name like '" + autoSuggestUIHints.getSubmittedValue() + "%'";

PreparedStatement stat = null;

ResultSet rs = null;

Connection conn = null;

try { conn = this.getConnection();

stat = conn.prepareStatement(sql);

rs = stat.executeQuery();

while (rs.next()) {

items.add(new SelectItem(rs.getString(1), rs.getString(1))); }

} catch (Exception e) {

e.printStackTrace();

} finally {

try { rs.close();

stat.close();

conn.close();

} catch (SQLException sqle) {

sqle.printStackTrace(); } }

return null;

}

**\*\*Rest Service Code**: https://fake-json-api.mock.beeceptor.com/users

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

public class ReqresApiClient {

private static final String API\_URL = "https://fake-json-api.mock.beeceptor.com/users";

public static void main(String[] args) {

try { URL url = new URL(API\_URL);

HttpURLConnection connection = (HttpURLConnection) url.openConnection();

connection.setRequestMethod("GET");

int responseCode = connection.getResponseCode();

System.out.println("ResponseCode:"+responseCode);

if (responseCode == HttpURLConnection.HTTP\_OK) {

BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));

String inputLine;

StringBuilder response = new StringBuilder();

while ((inputLine = in.readLine()) != null) {

System.out.println(inputLine);

response.append(inputLine); }

System.out.println("---------------------------------------");

in.close();

// Print the response

System.out.println("Response: " + response.toString());

} else {

System.out.println("GET request failed. Response Code: " + responseCode); }

} catch (Exception e) {

e.printStackTrace();

} }}

**1. //this code is changing the panelbox or inputtext**

**imports:-**

import javax.faces.application.FacesMessage;

import javax.faces.context.FacesContext;

import javax.faces.event.ActionEvent;

public void clicK(ActionEvent actionEvent) {

String empID = this.getEmployeeIdBinding().getValue().toString();

if (empID.equalsIgnoreCase("1")) {

this.getOnePaneBinding().setVisible(true);

this.getSecondPanelBinding().setVisible(false);

} else if (empID.equalsIgnoreCase("2")) {

this.getOnePaneBinding().setVisible(false);

this.getSecondPanelBinding().setVisible(true);

} else {

FacesMessage Message = new FacesMessage("Enter 1 or 2"); // error massege code

Message.setSeverity(FacesMessage.SEVERITY\_INFO);

FacesContext fc = FacesContext.getCurrentInstance();

fc.addMessage(null, Message);

}

}

**2. // this code is GetSalary and HikeSalary**

**AMImpl code:-**

public void getSalary(Integer salary){

ViewObjectImpl vo = this.getEmployeesView1();

ViewCriteria vc = vo.getViewCriteria("EmployeesSalaryViewCriteria");

vo.setNamedWhereClauseParam("salaryBind", salary);

vo.applyViewCriteria(vc);

vo.executeQuery();

}

**Bean Code:-** **import:-**

import java.util.Map;

import javax.faces.event.ActionEvent;

import oracle.adf.model.BindingContext;

import oracle.binding.BindingContainer;

import oracle.binding.OperationBinding;

public void getEmployeeButton(ActionEvent actionEvent) {

// Add event code here...

Integer sal = Integer.parseInt(this.getSalaryBind().getValue().toString());

BindingContext bctx = BindingContext.getCurrent();

BindingContainer bc = bctx.getCurrentBindingsEntry();

OperationBinding ob = bc.getOperationBinding("getSalary");

Map map = ob.getParamsMap();

map.put("salary", sal);

map.put("Per", null);

ob.execute(); }

AMImpl Code:-

public void getHikeSalary(Integer salary , Integer Per){

ViewObjectImpl vo = this.getEmployeesView1();

ViewCriteria vc = vo.getViewCriteria("EmployeesSalaryViewCriteria");

vo.setNamedWhereClauseParam("salaryBind", salary);

vo.applyViewCriteria(vc);

vo.executeQuery();

if(Per!=null){

RowSetIterator rs = vo.createRowSet(null);

while(rs.hasNext()){

Row r = rs.next();

r.getAttribute("Salary");

Integer oldsalary = Integer.parseInt(r.getAttribute("Salary").toString());

Integer newsal = oldsalary + oldsalary\*Per/100;

r.setAttribute("Salary", newsal);

} }

}

**Bean Code:- import:-**

import java.util.Map;

import javax.faces.event.ActionEvent;

import oracle.adf.model.BindingContext;

import oracle.binding.BindingContainer;

import oracle.binding.OperationBinding;

public void hikeSalaryButton(ActionEvent actionEvent) {

// Add event code here...

Integer salary = Integer.parseInt(this.getSalaryBind().getValue().toString());

Integer inc = Integer.parseInt(this.getIncreaseBinding().getValue().toString());

OperationBinding ob = BindingContext.getCurrent().getCurrentBindingsEntry().getOperationBinding("getHikeSalary");

ob.getParamsMap().put("salary", salary);

ob.getParamsMap().put("Per", inc);

ob.execute();

}

**3. // this code is hoelp to print EmployeDetails in console**

**\*\* first we need to write code in amimpl in this we are getting emplyees(any view object) view and getcurrent \*\***

public void print(){

ViewObjectImpl vo = this.getEmployeesView2();

Row r = vo.getCurrentRow();

System.out.println(r.getAttribute("EmployeeId"));

System.out.println(r.getAttribute("FirstName"));

System.out.println(r.getAttribute("LastName"));

System.out.println(r.getAttribute("Email"));

System.out.println(r.getAttribute("PhoneNumber"));

System.out.println(r.getAttribute("HireDate"));

System.out.println(r.getAttribute("JobId"));

System.out.println(r.getAttribute("Salary"));

System.out.println(r.getAttribute("CommissionPct"));

System.out.println(r.getAttribute("ManagerId"));

System.out.println(r.getAttribute("DepartmentId"));

ViewObjectImpl vo1 = this.getDepartmentsView1();

Row r1 = vo1.getCurrentRow();

System.out.println(r1.getAttribute("DepartmentId"));

System.out.println(r1.getAttribute("DepartmentName"));

System.out.println(r1.getAttribute("ManagerId"));

System.out.println(r1.getAttribute("LocationId"));

}

**4. //this code is Fromdate and Todate validations**

**AMImpl code:-**

**\*\* need to create viewcriteria in getEmployeesView1 (HierDate After and HierDate Before) with Bind variable \*\***

public void ftdates(Date fro,Date tro){

ViewObjectImpl vo = this.getEmployeesView1();

ViewCriteria vc = vo.getViewCriteria("ftdate");

vo.setNamedWhereClauseParam("froms", fro);

vo.setNamedWhereClauseParam("tos", tro);

vo.applyViewCriteria(vc);

vo.executeQuery();

}

**Bean Code:-**

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.faces.application.FacesMessage;

import javax.faces.context.FacesContext;

import oracle.adf.model.BindingContext;

import oracle.adf.view.rich.component.rich.input.RichInputDate;

import oracle.binding.BindingContainer;

import oracle.binding.OperationBinding;

public String myMethod() {

Object fro = this.getFrom().getValue();

Object tro = this.getTo().getValue();

if(fro==null&&tro!=null){

FacesMessage Message = new FacesMessage("Enter From Date");

Message.setSeverity(FacesMessage.SEVERITY\_INFO);

FacesContext fc = FacesContext.getCurrentInstance();

fc.addMessage(null, Message);

}

if(fro!=null&&tro==null){

FacesMessage Message = new FacesMessage("Enter To Date");

Message.setSeverity(FacesMessage.SEVERITY\_INFO);

FacesContext fc = FacesContext.getCurrentInstance();

fc.addMessage(null, Message);

}

if(fro==null&&tro==null){

FacesMessage Message = new FacesMessage("Enter From and TO Date");

Message.setSeverity(FacesMessage.SEVERITY\_INFO);

FacesContext fc = FacesContext.getCurrentInstance();

fc.addMessage(null, Message);

}

if(fro!=null&&tro!=null){

SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");

String from = sdf.format(fro);

String to = sdf.format(tro);

Date frod;

Date trod;

try {

frod = sdf.parse(from);

trod = sdf.parse(to);

if(frod.after(trod)){

FacesMessage Message = new FacesMessage("From date should be before To date ");

Message.setSeverity(FacesMessage.SEVERITY\_INFO);

FacesContext fc = FacesContext.getCurrentInstance();

fc.addMessage(null, Message);

}

if(frod.before(trod)){

BindingContext bctx=BindingContext.getCurrent();

BindingContainer bc = bctx.getCurrentBindingsEntry();

OperationBinding ob = bc.getOperationBinding("ftdates");

ob.getParamsMap().put("fro", fro);

ob.getParamsMap().put("tro", tro);

ob.execute();

}

} catch (ParseException e)

{

}

}

return null;

}

**5. // this code is search the FirstName Details**

\*\* to create viewcriteria for FirstName with BindVariable \*\*

AMImpl code:-

public void vc(String naam){

ViewObjectImpl vo = this.getEmployeesView1();

ViewCriteria vc = vo.getViewCriteria("name");

vo.setNamedWhereClauseParam("ln", naam);

vo.applyViewCriteria(vc);

vo.executeQuery();

}

**Bean Cone:-**

public String myMethod() {

String s = this.getLname().getValue().toString();

BindingContainer bc = BindingContext.getCurrent().getCurrentBindingsEntry();

OperationBinding ob = bc.getOperationBinding("vc");

ob.getParamsMap().put("naam", s);

ob.execute();

return null; }

**\*\*6// this code is actionEvent for button insert row:**

**Bean code:-**

public void insertRow(ActionEvent ace) {

BindingContext bc = BindingContext.getCurrent();

BindingContainer bindingContainer = bc.getCurrentBindingsEntry();

OperationBinding ob = bindingContainer.getOperationBinding("Create");

ob.execute();

RichPopup.PopupHints a = new RichPopup.PopupHints();

this.getPopupBind().show(a);

}

**\*\*this code is dailogListener for insert row:**

public void addDilogListenerMethod(DialogEvent dialogEvent) {

// Add event code here...

if (dialogEvent.getOutcome() == DialogEvent.Outcome.ok) {

//if(outcome != null && Outcome.ok){

BindingContext bc = BindingContext.getCurrent();

BindingContainer bindingContainer = bc.getCurrentBindingsEntry();

OperationBinding ob =

bindingContainer.getOperationBinding("Commit");

ob.execute();

}

AdfFacesContext.getCurrentInstance().addPartialTarget(tableRefresh);

}

**\*\* 7 // this is prog edit popup code:**

public void editEmployees(ActionEvent aec) {

RichPopup.PopupHints edit = new RichPopup.PopupHints();

this.getEditPopupBind().show(edit);

}

**\*\* this is prog edit method using dailog event**

public void editEmpDL(DialogEvent dialogEvent) {

// Add event code here...

if (dialogEvent.getOutcome() == DialogEvent.Outcome.ok) {

//if(outcome != null && Outcome.ok){

BindingContext bc = BindingContext.getCurrent();

BindingContainer bindingContainer = bc.getCurrentBindingsEntry();

OperationBinding ob =

bindingContainer.getOperationBinding("Commit");

ob.execute();

}

AdfFacesContext.getCurrentInstance().addPartialTarget(tableRefresh);

}

**\*\* 8 // delete popup actionevent**

public void deleteEmployee(ActionEvent ace) {

RichPopup.PopupHints deletePopup = new RichPopup.PopupHints();

this.deletePopupBind.show(deletePopup);

}

**\*\*delete popup dailogevent**

public void deleteEmpDL(DialogEvent dialogEvent) {

// Add event code here...

BindingContext bc = BindingContext.getCurrent();

BindingContainer bindingContainer = bc.getCurrentBindingsEntry();

if (dialogEvent.getOutcome() == DialogEvent.Outcome.ok) {

OperationBinding deleteOperation =

bindingContainer.getOperationBinding("Delete");

if (deleteOperation != null) {

deleteOperation.execute();

if (!deleteOperation.getErrors().isEmpty()) {

FacesMessage errorMessage = new FacesMessage(FacesMessage.SEVERITY\_ERROR,

"Error- ", "An error occured while deleteing the record");

FacesContext.getCurrentInstance().addMessage(null, errorMessage);

return; }

}

OperationBinding commitOperation = bindingContainer.getOperationBinding("Commit");

if (commitOperation != null) {

commitOperation.execute();

if (!commitOperation.getErrors().isEmpty()) {

FacesMessage errorMessage =new FacesMessage(FacesMessage.SEVERITY\_ERROR,

"Error- ","An error occured while commiting the transaction ");

FacesContext.getCurrentInstance().addMessage(null, errorMessage);

return; }

}

System.out.println("The record was success fully deleted");

AdfFacesContext.getCurrentInstance().addPartialTarget(tableRefresh);

} }

**\*\*9 // decrease10% salary for all employees at once**

public void decrease10PerSalary(ActionEvent ace) {

System.out.println("Entering into increaseSalaryACL-----------------");

BindingContext bc = BindingContext.getCurrent();

BindingContainer bindingContainer = bc.getCurrentBindingsEntry();

DCIteratorBinding empIter =

(DCIteratorBinding)bindingContainer.get("EmployeesViewIterator");

ViewObject vo = empIter.getViewObject();

Row[] allRowsInRange = vo.getAllRowsInRange();

for (Row row : allRowsInRange) {

Number oldSalary = (Number)row.getAttribute("Salary");

//double newSalary = oldSalary.doubleValue()\*(1-10.0/100);

row.setAttribute("Salary", oldSalary.subtract(oldSalary.multiply(0.1)));

System.out.println("Exit From decrease10PerSalary---------------------");

}

vo.executeQuery();

AdfFacesContext.getCurrentInstance().addPartialTarget(tableRefresh);

FacesMessage successMessage =

new FacesMessage(FacesMessage.SEVERITY\_INFO, "Success: ",

"Salary dicreased 10% for all Employees!");

FacesContext.getCurrentInstance().addMessage(null, successMessage);

}

\*\* 10// **increase10% salary for all employees at once**

public void increaseSalaryACL(ActionEvent actionEvent) {

// Add event code here...

System.out.println("Entering into increaseSalaryACL-----------------");

BindingContext bc = BindingContext.getCurrent();

BindingContainer bindingContainer = bc.getCurrentBindingsEntry();

DCIteratorBinding empIter =

(DCIteratorBinding)bindingContainer.get("EmployeesViewIterator");

ViewObject vo = empIter.getViewObject();

Row[] allRowsInRange = vo.getAllRowsInRange();

double hikeSalary = 10.0;

for (Row row : allRowsInRange) {

Number salary = (Number)row.getAttribute("Salary");

System.out.println("Current Sala" + salary);

// double newSala = salary.doubleValue()\*(1+hikeSalary/100);

// System.out.println("New Salary is :"+newSala);

row.setAttribute("Salary", salary.add(salary.multiply(0.1)));

}

vo.executeQuery();

AdfFacesContext.getCurrentInstance().addPartialTarget(tableRefresh);

FacesMessage successMessage =

new FacesMessage(FacesMessage.SEVERITY\_INFO, "Success: ",

"Salary increased 10% for all Employees!");

FacesContext.getCurrentInstance().addMessage(null, successMessage);

}

**\*\* 11 login code using DB table bean**

public void loginButtonACL(ActionEvent actionEvent) {

// Add event code here...

System.out.println("Entering into loginButtonACL...............................");

String us = this.getUs();

String ps = this.getPs();

BindingContainer bindingContainer = BindingContext.getCurrent().getCurrentBindingsEntry();

DCIteratorBinding loginIter = (DCIteratorBinding)bindingContainer.get("LoginView1Iterator");

ViewObject loginVO = loginIter.getViewObject();

ViewCriteriaManager criteriaManager = loginVO.getViewCriteriaManager();

ViewCriteria criteria = criteriaManager.getViewCriteria("LoginViewCriteria");

loginVO.setNamedWhereClauseParam("usernameBind", us);

loginVO.setNamedWhereClauseParam("psBind", ps);

loginVO.applyViewCriteria(criteria);

loginVO.executeQuery();

if(loginVO.hasNext()){

Row next = loginVO.next();

String name = (String)next.getAttribute("Username");

ADFContext.getCurrent().getSessionScope().put("usernameBind", name);

System.out.println("RoeCount"+loginVO.getEstimatedRowCount());

}

System.out.println("Exit from loginButtonACL...............................");

}

**\*\* 12 // Decimal validation for salary bean code**

import javax.faces.application.FacesMessage;

import javax.faces.context.FacesContext;

import javax.faces.event.ValueChangeEvent;

import javax.faces.validator.ValidatorException;

public void decimalValue(ValueChangeEvent valueChangeEvent) {

// Add event code here...

Object newValue = valueChangeEvent.getNewValue();

if(newValue != null){

String input = newValue.toString();

if (input.matches("\\d+\\.\\d{3}")) {

System.out.println("Valid input : "+input);

// Input matches the required format

} else {

//throw new ValidatorException(new FacesMessage("Invalid format. The number must have three decimal places."));

String message = "Invalid format. The number must have three decimal places.";

FacesMessage errorMsg = new FacesMessage(FacesMessage.SEVERITY\_ERROR, "", message);

FacesContext.getCurrentInstance().addMessage("", errorMsg);

}

}else{

throw new ValidatorException(new FacesMessage("Input cannot be null."));

} }

**email reguler expression**

[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}

**\*\* 13 // Custom LOV Bean**

import java.util.ArrayList;

import java.util.List;

import javax.faces.event.ActionEvent;

import javax.faces.model.SelectItem;

public class CustomBean {

private String selectedValue;

private List<SelectItem> staicValue;

private String outBind;

private String Message;

public CustomBean() {

super();

staicValue = new ArrayList<SelectItem>();

staicValue.add(new SelectItem("IN", "India"));

staicValue.add(new SelectItem("US", "United States"));

staicValue.add(new SelectItem("P", "PUNE"));

}

// when we select any value in select one choice than

// that key and value should be changed dynamically

public void selectValueACL(ActionEvent actionEvent) {

// Add event code here...

System.out.println("Enter...............selectValueACL");

String selectedValue = this.getSelectedValue();

if (selectedValue != null) {

for (SelectItem item : staicValue) {

if (item.getValue().equals(selectedValue)) {

this.setMessage(selectedValue);

setOutBind(item.getLabel());

System.out.println("Selected Value......." + selectedValue);

System.out.println("Selected Item=======" + item.getLabel());

return;

} } }

System.out.println("Exit\*\*\*\*\*\*\*\*\*selectValueACL");

} }

**\*\* execute emptytable by loading the page AMIMPL code:**

public void emptydepartmentForm(){

ViewObjectImpl departmentView = this.getDepartmentView();

departmentView.executeEmptyRowSet();

}

**\*\* 14 // search departments AMIPL code**

public void getDepartments(Number iD){

System.out.println("Entering into getDepartments");

ViewObjectImpl departmentView = this.getDepartmentView();

ViewCriteria criteria = departmentView.getViewCriteria("DepartmentViewCriteria");

departmentView.setNamedWhereClauseParam("deptIdBind", iD);

departmentView.appendViewCriteria(criteria);

departmentView.executeQuery();

System.out.println("Exit from getDepartments");

}

**\*\* bean code**

public void invokeEmployees(ActionEvent ace) {

// add here code

Number id = this.getDepId();

BindingContainer bc =

BindingContext.getCurrent().getCurrentBindingsEntry();

OperationBinding ob = bc.getOperationBinding("getDepartments");

ob.getParamsMap().put("iD", id);

ob.execute(); }

**\*\*15 // navigation between jspx pages without controlflowcase bean code**

public void gotoPageB(ActionEvent ace){

System.out.println("Enter PageA");

ExternalContext context = FacesContext.getCurrentInstance().getExternalContext();

HttpServletRequest request = (HttpServletRequest)context.getRequest();

String contextPath = request.getContextPath()+"/faces/PageB.jspx";

System.out.println("Temp : "+contextPath);

try {

context.redirect(contextPath);

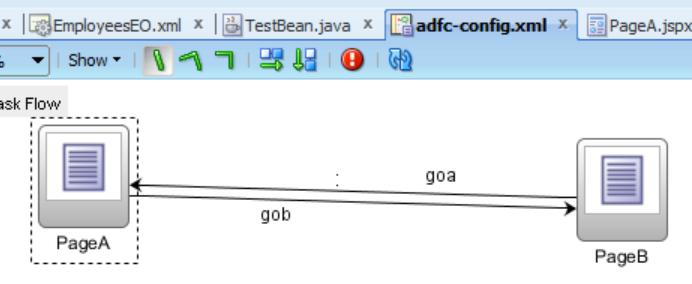
} catch (IOException e) {

System.out.println("Exception is logout()");

}

System.out.println("Exit PageA");

}



**\*\* With controflowcase**

public String goBAC(){

System.out.println("PageB");

return "gob"; }

**\*\*16 // Declarative coponents bean code:**

private Object getValueObject(String expr, Class returnType){

FacesContext fc = FacesContext.getCurrentInstance();

ELContext elctx = fc.getELContext();

ExpressionFactory elFactory = fc.getApplication().getExpressionFactory();

ValueExpression valueExpr = elFactory.createValueExpression(elctx, expr, returnType);

return valueExpr.getValue(elctx);

}

public void deleteDialogListener(DialogEvent dialogEvent) {

System.out.println("Enter deleteDialogListener");

BindingContainer bindings = BindingContext.getCurrent().getCurrentBindingsEntry();

UIXDeclarativeComponent myDeleteCmp = (UIXDeclarativeComponent)getValueObject("#{comp}", UIXDeclarativeComponent.class);

System.out.println("UIXDeclarativeComponent : "+myDeleteCmp);

String s = (String)myDeleteCmp.getAttributes().get("OperationId");

System.out.println("myDeleteCmp .. "+s);

OperationBinding ob = bindings.getOperationBinding(s);

System.out.println("Operation Biniding "+ob);

ob.execute();

ob = bindings.getOperationBinding("Commit");

ob.execute();

System.out.println("Exit deleteDialogListener");

}

\*\* 17 // max salary vlidation progrmmatically

/\*\*

\* Sets <code>value</code> as attribute value for SALARY using the alias name Salary.

\* @param value value to set the SALARY

\*/

public void setSalary(BigDecimal value) {

ViewRowImpl departmentsVO1 = this.getDepartmentsVO1();

if(departmentsVO1!=null){

Number MaxSalary = (Number)departmentsVO1.getAttribute("Maxsalary");

if(value.compareTo((BigDecimal)MaxSalary)>0){

throw new JboException ("Salary is above max salary"+"------"+MaxSalary);} }

setAttributeInternal(SALARY, value);

}

RichPopup.PopupHints errorPopup = new RichPopup.PopupHints(); errorPopup.add(RichPopup.PopupHints.HintTypes.HINT\_ALIGN, RichPopup.PopupHints.AlignTypes.ALIGN\_BEFORE\_END); this.getPopupBind().show(errorPopup);

-----------------------------------------------

private void redirectToPage(String page) {

ExternalContext context =

FacesContext.getCurrentInstance().getExternalContext();

String contextPath = context.getRequestContextPath() + page;

System.out.println("Redirecting to :" + contextPath);

try {

context.redirect(contextPath);

} catch (IOException e) {

System.err.println("Redirection error : " + e.getMessage());

}

}

------------------------------------------------------------

private void displayErrorMessage(String message) {

FacesMessage errorMsg =

new FacesMessage(FacesMessage.SEVERITY\_ERROR, "Error", message);

FacesContext.getCurrentInstance().addMessage(null, errorMsg);

System.out.println(message);

}