**How to display empty results on a page using**

It might be very common requirement, not to display all the results first time when a page loaded,  
  
 Write a simple method as shown below in AM Impl class, drag this on to the task flow and mark it as default activity. executeEmptyRowset() will apply false criteria and bring no results.  
  
    public void initializeEmpSearchResults(){  
          // Get VO Instance and invoke executeEptyRowSet()  
          getEmpVO1().executeEmptyRowSet();  
    }

public void assetReturnInitialization(){

getAstAssetReturnsVO().clearCache();

getAstAssetReturnsVO().executeEmptyRowSet();

Row row = getAstAssetReturnsVO().createRow();

row.setAttribute("Stat", 99);

row.setAttribute("AsrtDate", "1396/12/31");

getAstAssetsVO().executeQuery();

Row assetRow = getAstAssetsVO().first();

if (assetRow != null)

{

//your logic

}

row.setAttribute("AsetId", assetRow.getAttribute("Id"));

getAstAssetsVO().getCurrentRow().setAttribute("Status", "Returned".toUpperCase());

getAstAssetReturnsVO().insertRow(row);

getAstAssetReturnsVO().setCurrentRow(row);

return ;

}

ViewObjectImpl vo = this.getSomeViewObject();

vo.executeEmptyRowSet();

# [How to press the first item of a list/table in a page on first page load?](https://stackoverflow.com/questions/41506308/how-to-press-the-first-item-of-a-list-table-in-a-page-on-first-page-load)

I am pretty new to ADF and still acquiring hands-on experience.

I'd like to call a managed bean method only once when the user initially opens a JSFF page. The method is bound to SelectionListener of a table on the page. I need the iterators to be available because the handler does some CRUD operation (below is the code for method "insertNewReturnReason").

I have read this [link](https://blogs.oracle.com/jdevotnharvest/entry/jdeveloper_11g_r2_and_12c) from Shay Shmeltzer and I curious how I can do the same programmatically.

Please let me know.

Thank you and Your help is much appreciate it.

Regards

package com.asrandisheh.mis.asset.viewcontroller;

import javax.faces.event.ActionEvent;

import oracle.adf.model.BindingContext;

import oracle.adf.model.binding.DCIteratorBinding;

import oracle.jbo.Row;

import com.asrandisheh.mis.asset.model.viewobject.AstAssetReturnsVORowImpl;

import com.asrandisheh.mis.asset.viewcontroller.JSFUtils;

import javax.faces.context.FacesContext;

import oracle.adf.model.binding.DCBindingContainer;

import oracle.jbo.server.Entity;

import org.apache.myfaces.trinidad.event.SelectionEvent;

import oracle.binding.BindingContainer;

import oracle.jbo.ViewObject;

import oracle.binding.OperationBinding;

import oracle.jbo.Key;

public class AssetReturn {

public AssetReturn() {

super();

}

public void insertNewReturnReason(SelectionEvent selectionEvent) {

// Maintain the makecurrent behavior

JSFUtils.resolveMethodExpression("#{bindings.AstAssetsVO.collectionModel.makeCurrent}", null,

new Class[]{SelectionEvent.class}, new Object[]{selectionEvent});

// Get the binding context

BindingContext bctx = BindingContext.getCurrent();

BindingContainer bindings = bctx.getCurrentBindingsEntry();

// get the current row for AST\_ASSET table

DCIteratorBinding astAssetsVOIterator = (DCIteratorBinding) bindings.get("astAssetsVOIterator");

Key parentKey = astAssetsVOIterator.getCurrentRow().getKey();

//Rollback any previously created row

OperationBinding ob= bindings.getOperationBinding("Rollback");

ob.execute();

//Set again row key as current row

astAssetsVOIterator.setCurrentRowWithKey(parentKey.toStringFormat(true));

// get the AstAssetReturnsVOIterator and create a new row with default values

DCIteratorBinding astAssetReturnsVOIterator = (DCIteratorBinding) bindings.get("AstAssetReturnsVOIterator");

ViewObject assetReturnVO = astAssetReturnsVOIterator.getViewObject();

assetReturnVO.executeEmptyRowSet();

// pre-set values for ast\_asset\_return

Row assetReturnRow = assetReturnVO.createRow();

Row assetRow = astAssetsVOIterator.getCurrentRow();

assetReturnRow.setAttribute("AsetId", assetRow.getAttribute("Id"));

assetReturnRow.setAttribute("AsrtDate", "1395/12/31");

assetReturnRow.setAttribute("Stat", 0);

// insert the new row

assetReturnVO.insertRow( assetReturnRow );

}

public void saveAssetReturn(ActionEvent actionEvent) {

// Get the bindings

BindingContext bctx = BindingContext.getCurrent();

BindingContainer bindings = bctx.getCurrentBindingsEntry();

// get the current row for AST\_ASSET table

DCIteratorBinding astAssetsVOIterator = (DCIteratorBinding) bindings.get("AstAssetsVOIterator");

astAssetsVOIterator.getCurrentRow().setAttribute("Status", "Returned");

OperationBinding ob = bindings.getOperationBinding("Commit");

ob.execute();

}

}

EDIT: If I use method call activity as an initial activity I get the null pointer when I try to access the ViewObject for the iterator. I have tagged the line cuzing the null pointer exception just in case.

public void assetReturnInitialization() {

// Get the binding context

BindingContext bctx = BindingContext.getCurrent();

BindingContainer bindings = bctx.getCurrentBindingsEntry();

// get the current row for AST\_ASSET table

DCIteratorBinding astAssetsVOIterator = (DCIteratorBinding) bindings.get("astAssetsVOIterator");

// get the AstAssetReturnsVOIterator and create a new row with default values

DCIteratorBinding astAssetReturnsVOIterator = (DCIteratorBinding) bindings.get("AstAssetReturnsVOIterator");

ViewObject assetReturnVO = astAssetReturnsVOIterator.getViewObject(); // NULL POINTER EXCEPTION OCCURS HERE

assetReturnVO.executeEmptyRowSet();

System.out.println("assetReturnInitialization 40");

// pre-set values for ast\_asset\_return

Row assetReturnRow = assetReturnVO.createRow();

Row assetRow = astAssetsVOIterator.getCurrentRow();

System.out.println("assetReturnInitialization 50");

assetReturnRow.setAttribute("AsetId", assetRow.getAttribute("Id"));

assetReturnRow.setAttribute("AsrtDate", "1395/12/31");

assetReturnRow.setAttribute("Stat", 0);

System.out.println("assetReturnInitialization 60");

// insert the new row

assetReturnVO.insertRow( assetReturnRow );

System.out.println("assetReturnInitialization 20");

}

[oracle-adf](https://stackoverflow.com/questions/tagged/oracle-adf)

## 3 Answers

[active](https://stackoverflow.com/questions/41506308/how-to-press-the-first-item-of-a-list-table-in-a-page-on-first-page-load?answertab=active#tab-top)[oldest](https://stackoverflow.com/questions/41506308/how-to-press-the-first-item-of-a-list-table-in-a-page-on-first-page-load?answertab=oldest#tab-top)[votes](https://stackoverflow.com/questions/41506308/how-to-press-the-first-item-of-a-list-table-in-a-page-on-first-page-load?answertab=votes#tab-top)

1

Check out af:poll component. In jsff page, do something like:

<af:poll id="p1" interval="10" pollListener="#{pageFlowScope.wci.handleFragmentLoad}">

<af:clientListener method="resetPollInterval" type="poll"/>

</af:poll>

When fragment first loads, it calls backing bean method in which you can now programmatically get a handle to table, find first row, and perform desired operations. The poll component also needs to call some javascript (clientListener) to turn off the poll once it runs on page load. This looks like:

<af:resource type="javascript">

function resetPollInterval(e) {

var src = e.getSource();

var poll = src.findComponent("p1");

if (poll != null) {

poll.setInterval( -1);

}

}

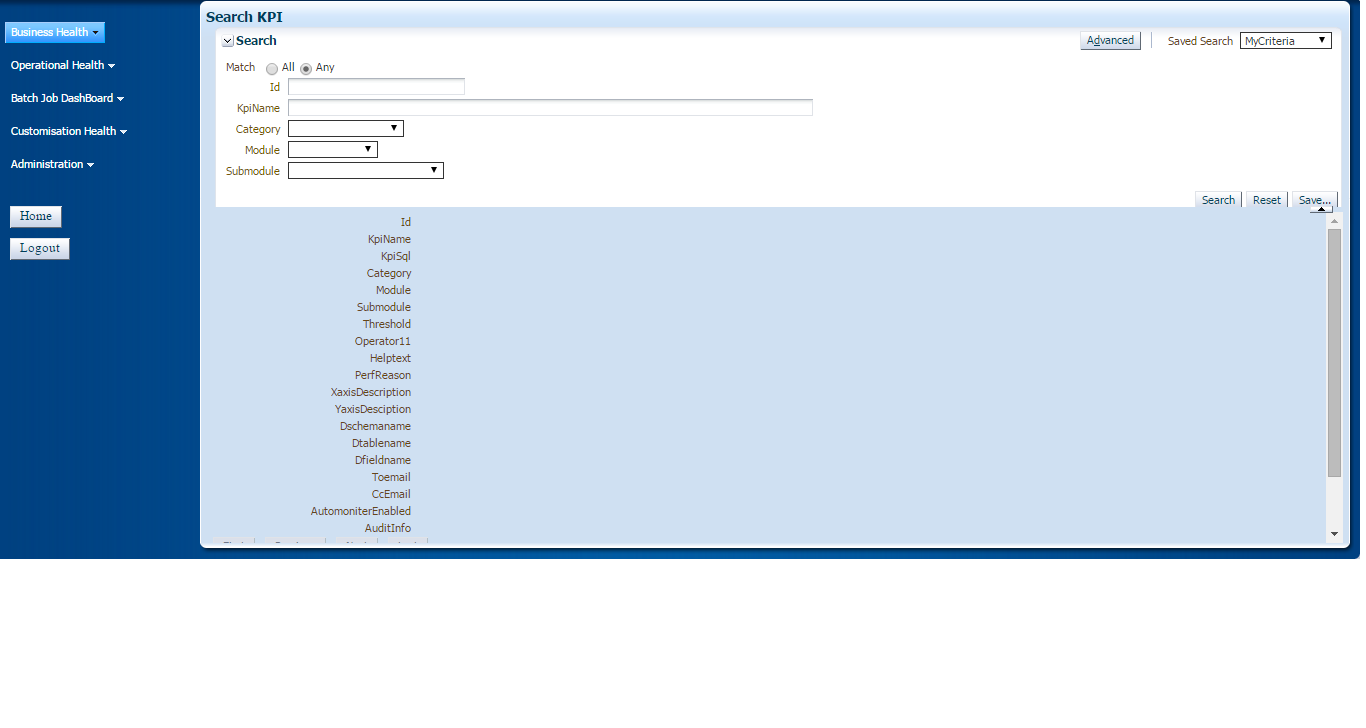
</af:resource>

This is one way to do it if you can't do it with a method call in the task flow, which is usually preferable.

Found it! I needed to add the iterators in the pageDef of the Method Call Activity. So I just had to right-click on the MCA and then went to pageDef and added iterators I needed to the executable list.

# [Oracle ADF: Refresh Form data](https://stackoverflow.com/questions/29504412/oracle-adf-refresh-form-data)

I am developing a web app using Oracle ADF. I have a bounded task flow. In that I have a search page like below.



I have created the above two forms using view object data controls.

Searching is performing well. But my problem is when I go some where else in my application using menus provided left side and come back to the search page , the page is not getting refreshed. I am getting a search page that contains old search results. At this point of time if I am trying to make any changes am getting some error called **"Another user with this id already modifed data ...."**. After this error my app is not running. Means what ever am trying to do its showing the same error.

So I need to make this: "When ever the user come to this form, He should get fresh form. It should not contain old search results.

Please help me. How do I achieve this.

Thank you.

## 2 Answers

[active](https://stackoverflow.com/questions/29504412/oracle-adf-refresh-form-data?answertab=active#tab-top)[oldest](https://stackoverflow.com/questions/29504412/oracle-adf-refresh-form-data?answertab=oldest#tab-top)[votes](https://stackoverflow.com/questions/29504412/oracle-adf-refresh-form-data?answertab=votes#tab-top)

0

There are 2 ways of doing it:

1) Set your task flow as ISOLATED, from Task Flow Overview tab -> Behaviour -> Share Data Control with calling task flow -> unchecked (or isolated, if you are using JDev 12c)

This will ensure you always start FRESH when accessing the page, but it will potentially create a performance overhead because entire View Object cache will be recreated (requeried) on page load. Nevertheless, it is the quickest solution.

2) You may create a default Method Call Activity in your task flow from where you may call a AM's custom method that resets the view criteria. The method will be placed on application module's implementation class and it may look like this:

public void initTaskFlow() {

this.getViewObject1().executeEmptyRowSet();

}

This will clean the result data. If you want to reset the querying parameters as well, you can use this example:

<http://www.jobinesh.com/2011/04/programmatically-resetting-and-search.html>

[share](https://stackoverflow.com/a/29509161)[improve this answer](https://stackoverflow.com/posts/29509161/edit)

answered Apr 8 '15 at 8:06

[[https://i.stack.imgur.com/AYdV1.jpg?s=32&g=1](https://stackoverflow.com/users/4107262/florin-marcus)](https://stackoverflow.com/users/4107262/florin-marcus)

[Florin Marcus](https://stackoverflow.com/users/4107262/florin-marcus)

**1,587**189

* Thanks for your time. I have checked your first solution. But am already having the same in my bounded taskflow. Under Transaction I have Always Begin New Transaction. And Share Data Control with calling task flow is unchecked. Let me try with second solution. – [Abdul](https://stackoverflow.com/users/2771888/abdul) [Apr 8 '15 at 8:36](https://stackoverflow.com/questions/29504412/oracle-adf-refresh-form-data#comment47175593_29509161)
* The fact that first solution is not working is abnormal, it means we are missing something... The only thing I can think of is perhaps there is a backing bean on session scope, holding bindings for af:table and af:query components... – [Florin Marcus](https://stackoverflow.com/users/4107262/florin-marcus) [Apr 8 '15 at 11:47](https://stackoverflow.com/questions/29504412/oracle-adf-refresh-form-data#comment47183044_29509161)

add a comment

0

When you made any changes to any viewObject then excute this viewObject to match entity state and viewState , i think excuting viewObject will solve your issue

Ashish