

Filtered Search of Portal Roles, Groups and Users Using Web Dynpro for Java



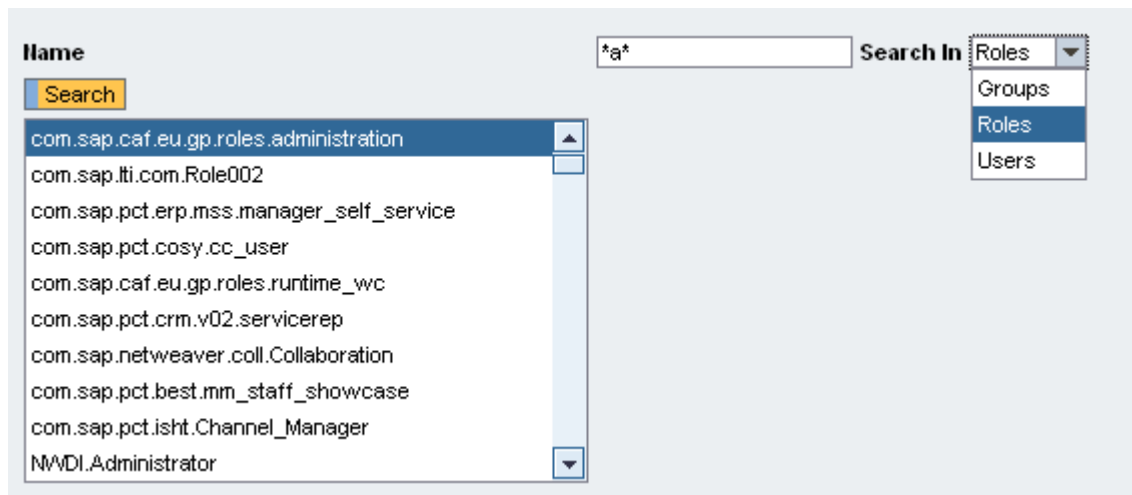
Applies to:

The code in this article is written and tested on NWDS version 7.0. For more information, visit the [User Interface Technology homepage](#). For more information, visit the [User Interface Technology homepage](#).

Summary

Sometimes it is required to fetch all the existing roles or groups or users depending upon some selection criteria. This application helps to fetch all the existing portal roles/groups/users based upon their **Unique Names** and display them in an ItemListBox UI element. The application is developed in WebDynpro for Java.

Example: You can use “a*” to select fetch all the roles that start with “a”. The final desired output is



Author: Naresh Garg

Company: L&T Infotech Pvt. Ltd, India

Created on: 6 Oct 2008

Author Bio



Naresh Garg is working as Net Weaver consultant in L&T Infotech Pvt. Ltd, India.

Table of Contents

Create Project.....	3
Create Application and Component Window and View.....	3
Add Jar File.....	4
Create Simple Type.....	4
Create Context elements.....	5
Create Layout.....	5
Implementation.....	6
Result.....	10
Related Content.....	11
Disclaimer and Liability Notice.....	12

Create Project

Create a WebDynpro project named “**SearchPortal**”

Project name:

Project Location
 Use default

Directory:

Project language: ▼

Create Application and Component Window and View

Create the application named “**SearchPortalApp**”

Name:

Package:

Authentication

Click on **next** and keep the **default values**

Component Name:

Component Package:

Source Folder:

Window Name:

Window Package:

Embed new View

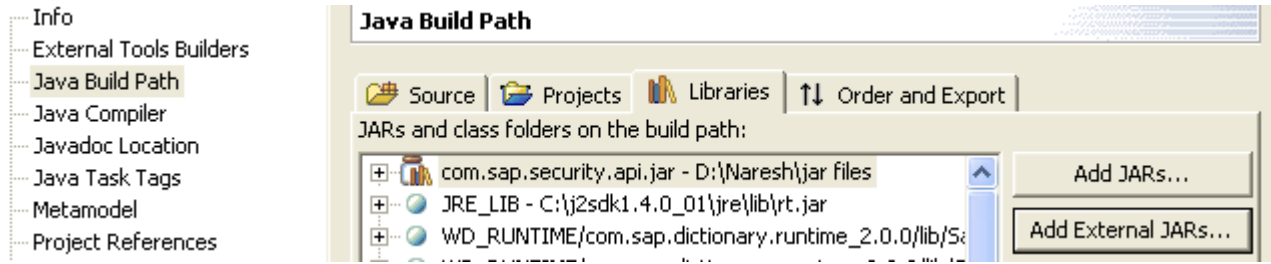
View Name:

View Package:

Add Jar File

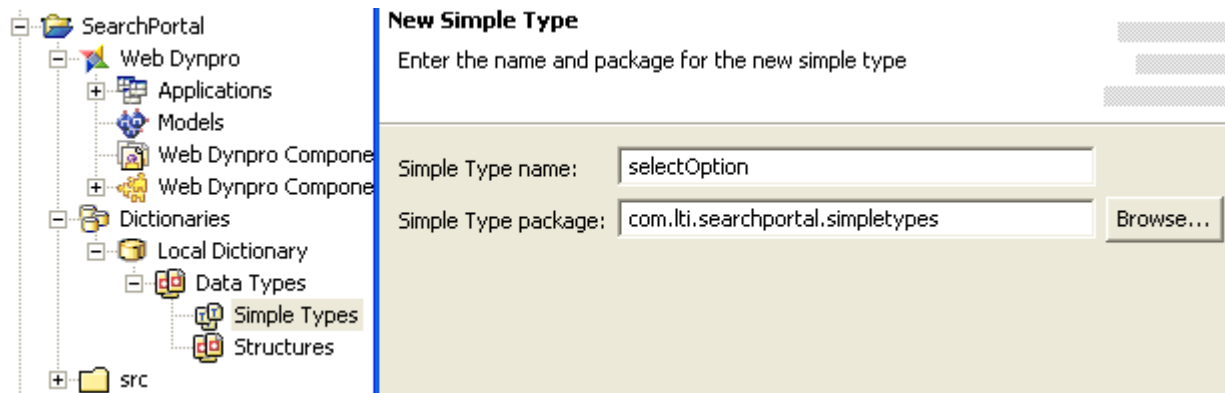
Add “com.sap.security.api” jar file to the project structure.

To do this



Create Simple Type.

Create a simple data type named “selectOption”



Add three enumerations in the “selectOptions” simple type as displayed below.

Simple Type Enumeration

Enumeration

Define an enumeration for the simple type

Generate a class representation of the enumeration

Value	Description	
Groups	Groups	New... Edit...
Roles	Roles	
Users	Users	

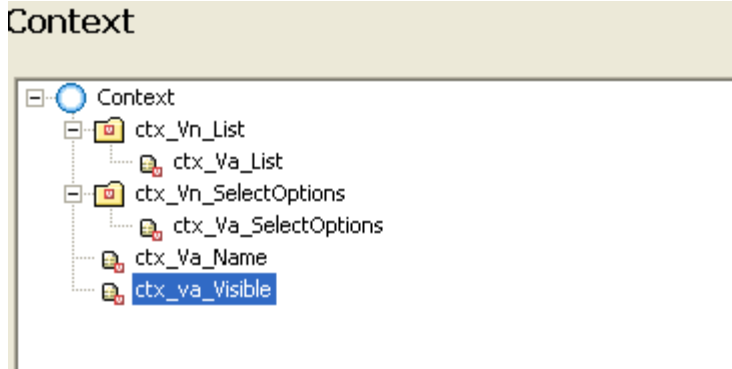
Save all metadata.

Create Context elements

Create the context elements as follow:

ctx_Va_Name, **ctx_Va_SelectOptions** and **ctx_Va_List** are of type **string**.

ctx_Va_Visible is of type “**com.sap.ide.webdynpro.uelementdefinitions.Visibility**”



Save all metadata

Create Layout

Create the layout as displayed and set various properties as follow.

InputField: dataSource = ctx_Va_Name.

DropDown: selectedKey = ctx_Vn_SelectOptions.ctx_Va_SelectOptions

ItemListBox: dataSource = ctx_Vn_List

text = ctx_Vn_List.ctx_Va_List

visible = ctx_Va_Visible

Button: create the action Search for the button, “**Search**”.



Save all metadata

Implementation

Write the following code in the `wdDoInit` method:

```
IPrivateSaarchPortalAppView.ICtx_Vn_ListElement listEle =
wdContext.createCtx_Vn_ListElement();
    wdContext.nodeCtx_Vn_List().bind(listEle);

IPrivateSaarchPortalAppView.ICtx_Vn_SelectOptionsElement optionsEle =
wdContext.createCtx_Vn_SelectOptionsElement();
    wdContext.nodeCtx_Vn_SelectOptions().bind(optionsEle);
    wdContext.currentContextElement().setCtx_va_Visible(WDVisibility.NONE);
```

In the method `onActionSearch` write following code.

```
String selectedOption =
    wdContext
        .currentCtx_Vn_SelectOptionsElement()
        .getCtx_Va_SelectOptions();
wdContext.nodeCtx_Vn_List().invalidate();
wdContext.currentContextElement().setCtx_va_Visible(
    WDVisibility.VISIBLE);

//Code to search the Groups ///////////////

    if (selectedOption.equalsIgnoreCase("Groups")) {

        IGroupFactory groupFact = UFactory.getGroupFactory();

        try {
            String groupName =
                wdContext.currentContextElement().getCtx_Va_Name();
            String str = "";

            ArrayList arrList = new ArrayList();
            IPrivateSaarchPortalAppView.ICtx_Vn_ListElement listEle =
null;

            IGroupSearchFilter groupFilt =
groupFact.getGroupSearchFilter();
            groupFilt.setUniqueName(
                groupName,
                ISearchAttribute.LIKE_OPERATOR,
                false);
            ISearchResult result = groupFact.searchGroups(groupFilt);
            try {
                if (result.getState() ==
ISearchResult.SEARCH_RESULT_OK) {
                    while (result.hasNext()) {

                        String uniqId = (String) result.next();
                        IGroup thisGroup =
groupFact.getGroup(uniqId);

                        String name = thisGroup.getDisplayName();
```

```

// Code For Drop Down
listEle =
wdContext.createCtx_Vn_ListElement();
listEle.setCtx_Va_List(name);

wdContext.nodeCtx_Vn_List().addElement(listEle);
arrList.add(listEle);

}
wdContext.nodeCtx_Vn_List().bind(arrList);
} else {

wdComponentAPI.getMessageManager().reportException(
    "Group Not Found",
    false);
}
} catch (RuntimeException e) {
    wdComponentAPI.getMessageManager().reportException(
        e.getMessage(),
        false);
}

} catch (UMException e1) {

    wdComponentAPI.getMessageManager().reportException(
        e1.getMessage(),
        false);
}
}

//Code to search the roles ///////////////
if (selectedOption.equalsIgnoreCase("Roles")) {

    IRoleFactory roleFact = UMFactory.getRoleFactory();

    try {
        String roleName =
            wdContext.currentContextElement().getCtx_Va_Name();
        String str = "";

        ArrayList arrList = new ArrayList();
        IPrivateSaarchPortalAppView.ICtx_Vn_ListElement listEle =
null;

        IRoleSearchFilter roleFilt = roleFact.getRoleSearchFilter();
        roleFilt.setUniqueName(
            roleName,
            ISearchAttribute.LIKE_OPERATOR,
            false);
        ISearchResult result = roleFact.searchRoles(roleFilt);
        try {
            if (result.getState() ==
ISearchResult.SEARCH_RESULT_OK) {
                while (result.hasNext()) {

                    String uniqId = (String) result.next();

```

```

        IRole thisRole = roleFact.getRole(uniqId);
        String name = thisRole.getDisplayName();

        listEle =
wdContext.createCtx_Vn_ListElement();
        listEle.setCtx_Va_List(name);

        wdContext.nodeCtx_Vn_List().addElement(listEle);
        arrList.add(listEle);
    }
    wdContext.nodeCtx_Vn_List().bind(arrList);
} else {

wdComponentAPI.getMessageManager().reportException(
    "Role Not Found",
    false);
}
} catch (RuntimeException e) {
    wdComponentAPI.getMessageManager().reportException(
        e.getMessage(),
        false);
}
} catch (UMException e1) {

    wdComponentAPI.getMessageManager().reportException(
        e1.getMessage(),
        false);
}
}

//Code to search the Users ///////////////
if (selectedOption.equalsIgnoreCase("Users")) {

    IUserFactory userFact = UMFactory.getUserFactory();

    try {
        String userName =
            wdContext.currentContextElement().getCtx_Va_Name();
        String str = "";

        ArrayList arrList = new ArrayList();
        IPrivateSaarchPortalAppView.ICtx_Vn_ListElement listEle =
null;

        IUserSearchFilter userFilt = userFact.getUserSearchFilter();
        userFilt.setUniqueName(
            userName,
            ISearchAttribute.LIKE_OPERATOR,
            false);
        ISearchResult result = userFact.searchUsers(userFilt);
        try {
            if (result.getState() ==
ISearchResult.SEARCH_RESULT_OK) {

```



```

        while (result.hasNext()) {

            String uniqId = (String) result.next();
            IUser thisUser = userFact.getUser(uniqId);
            String name = thisUser.getDisplayName();

            // Code For Drop Down
            listEle =

wdContext.createCtx_Vn_ListElement();

            listEle.setCtx_Va_List(name);

            wdContext.nodeCtx_Vn_List().addElement(listEle);
            arrList.add(listEle);

        }
        wdContext.nodeCtx_Vn_List().bind(arrList);
    } else {

        wdComponentAPI.getMessageManager().reportException(
            "User Not Found",
            false);
    }
} catch (RuntimeException e) {
    wdComponentAPI.getMessageManager().reportException(
        e.getMessage(),
        false);
}

} catch (UMException e1) {

    wdComponentAPI.getMessageManager().reportException(
        e1.getMessage(),
        false);
}

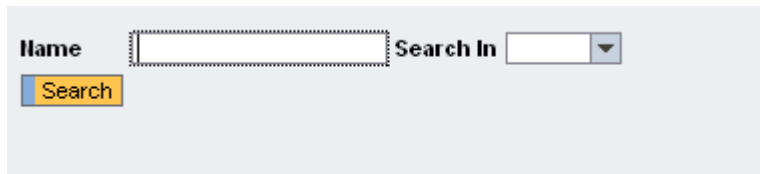
}
}

```

Save all metadata.

Result

On deploying and running the application following screen appears:



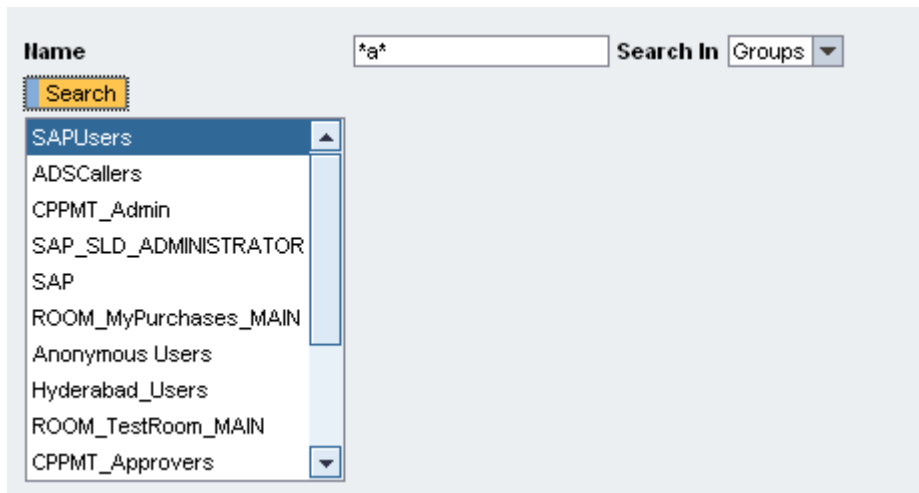
The initial search interface shows a text input field for the name, a dropdown menu for the search scope, and a search button. The name field is empty, and the search scope is set to a default value.

Enter the desired selection criteria for the name.



The search criteria are entered: the name field contains the wildcard pattern '*a*' and the search scope dropdown is open, showing options for 'Groups', 'Roles', and 'Users'.

The output is displayed



The search results are displayed in a list box. The search criteria remain the same: name '*a*' and search scope 'Groups'. The list of results includes:

- SAPUsers
- ADSCallers
- CPPMT_Admin
- SAP_SLD_ADMINISTRATOR
- SAP
- ROOM_MyPurchases_MAIN
- Anonymous Users
- Hyderabad_Users
- ROOM_TestRoom_MAIN
- CPPMT_Approvers

Related Content

<https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/00c80a68-ac51-2b10-efa6-c3ea8f984a6a>

<https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/d0eb5a9d-4256-2b10-79b9-bb0b797ccf73>

<https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/c2fad86f-0401-0010-91ac-bdb38595a735>

For more information, visit the [User Interface Technology homepage](#).

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.