

BAPIs and Java Methods a Powerful Combination with NWDS



Applies to:

SAP NetWeaver CE 7.1 SP5 or Higher

Summary

In several scenarios the services (BAPIs, Enterprise services) available at the moment may not fit the business requirements of a certain process. In this article we shed the light on Netweaver developer studio and how we can combine BAPI services along with Java methods to create a web service (Application Service) with new embedded Business logic bridging the gap in functionality delivered by Standard services.

Author: Ahmed Salah

Company: Certified IT Consultants

Created on: 01 January 2008

Author Bio



Ahmed Salah is on the xCCRE (Customer Centric Real Estate) product management team in CIC (Certified IT Consultants). The product is being built using CAF framework and methodologies. He has 5 Years of consulting experience in HCM, Portal Development, and SEM.

Table of Contents

| | |
|--------------------------------------|----|
| Business Scenario | 3 |
| Perquisites | 3 |
| Problem..... | 3 |
| Solution | 3 |
| Step By Step Solution..... | 4 |
| Create the Project | 4 |
| External Services | 5 |
| Modeled Services..... | 7 |
| Modeled Services Operations..... | 7 |
| Operation Parameters..... | 10 |
| Operation Data Source | 12 |
| Operation Implementation..... | 15 |
| Testing the Web Service | 19 |
| Related Content..... | 21 |
| Disclaimer and Liability Notice..... | 22 |

Business Scenario

We want to search for Business Partners by common search Criteria such as Address details, Business Partner role etc. As a result we should get a list of all those records for business partners listed by Name etc

Perquisites

1. SAP NetWeaver CE 7.1 Installed.
2. SAP Netweaver developer Studio 7.1 SP3 or Higher
3. SAP ECC 6.0.

Problem

A BAPI for searching business partners is already delivered BAPI_BUPA_SEARCH which searches partners according to an input of address search criteria; however it returns a table of business partner ids which is useless unless we remember all partner ids.

Another BAPI that takes partner ids as an input and retrieves all personnel details of that business partner is also available BAPI_BUPA_CENTRAL_GET_DETAIL. However we can't pass a list to this BAPI because it accepts only a single record and retrieves its details.

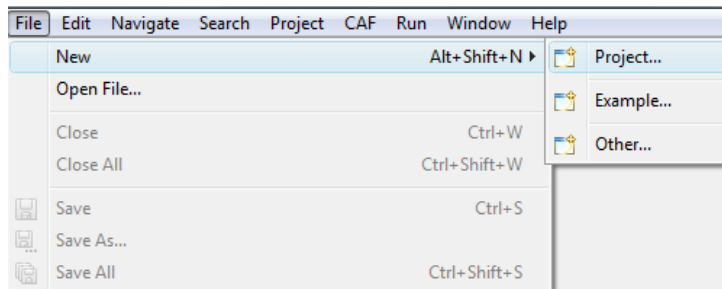
Solution

- Create a composite Service that receives our input search criteria (Address details, roles, etc...)
- Pass the input to BAPI_BUPA_SEARCH.
- Retrieve a list of BP IDs
- Pass the IDs one by one to the Next Service BAPI_BUPA_CENTRAL_GET_DETAIL
- Retrieve the results in a list and return it.

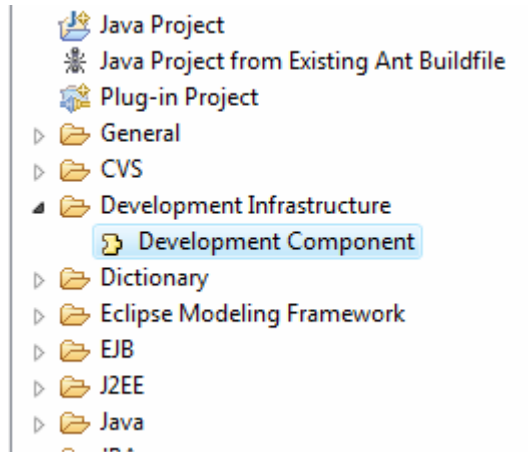
Step By Step Solution

Create the Project

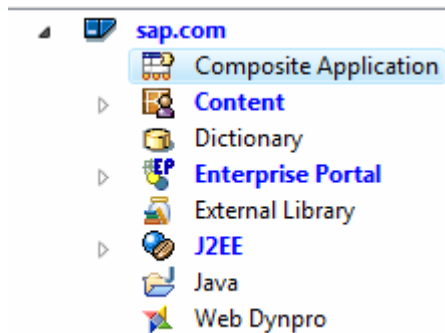
1. Launch the Netweaver Developer studio
2. Choose File → New Project



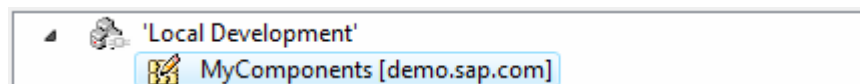
3. Choose Development Infrastructure → Development Component.



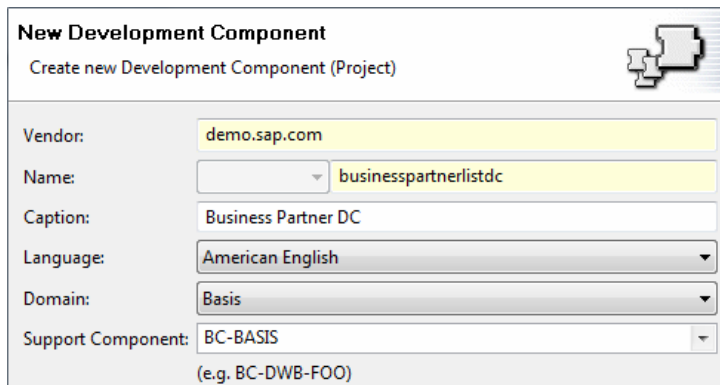
4. Choose sap.com → Composite Application



5. Choose Local development



6. Give Your DC a Name, Caption and a Support Component.



New Development Component
Create new Development Component (Project)

Vendor: demo.sap.com

Name: businesspartnerlistdc

Caption: Business Partner DC

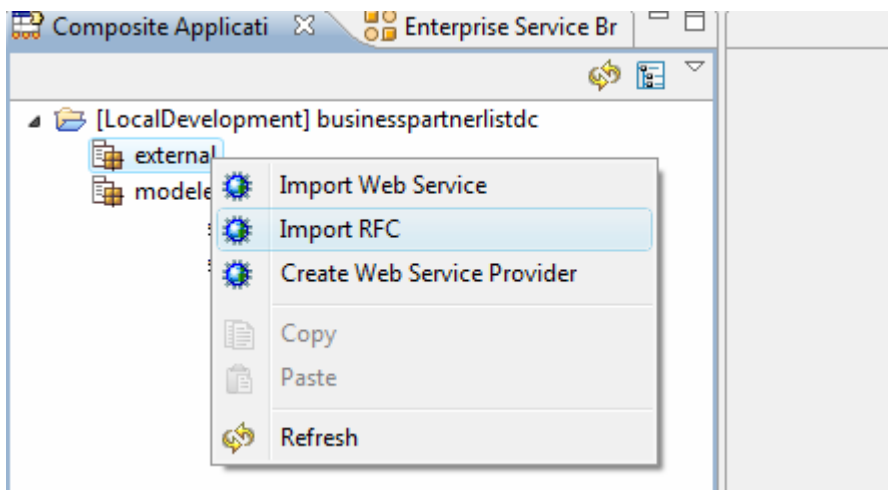
Language: American English

Domain: Basis

Support Component: BC-BASIS
(e.g. BC-DWB-FOO)

External Services

7. Go to your project in the Composite Application Perspective and right click on it.
8. Drill down to External Services.
9. Click on Import RFC.



10. Enter you SAP ABAP System Parameters

SAP Logon Information
Specify the SAP Server you want to connect to and enter the logon information

Single Server **Load Balancing**

Host Name: 192.168.1.150
System Number: 21
SAP Router:

User Account

Client: 800
Logon Name: idadmin
Password: ****
Language: EN

Note: if you don't know you sap system logon information you can get it from SAP GUI. Click on your System and then choose Change item.

11. Click on Next

12. Enter BAPI_BUPA_SEARCH in the Function Name and then select its checkbox.

Function Name Function Group

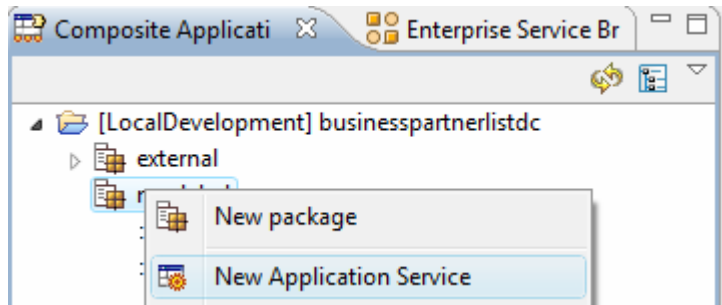
| | Function Name | Description | Function Group |
|-------------------------------------|------------------|---|----------------|
| <input checked="" type="checkbox"/> | BAPI_BUPA_SEARCH | SAP BP, BAPI: Search Business Partner fo... | BUBA_3 |

13. Click on Finish

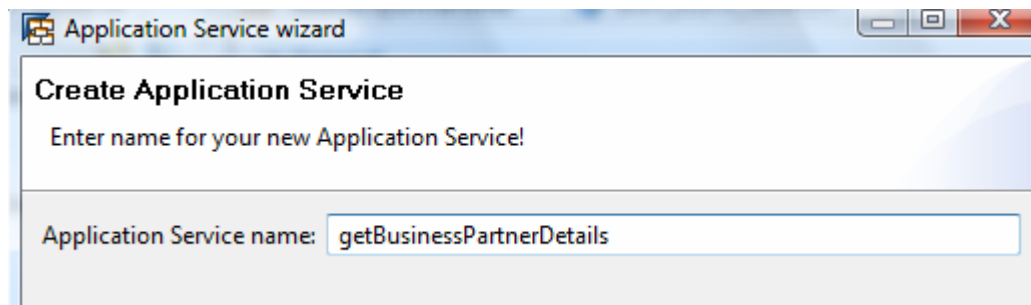
14. Repeat Steps 8-13 for Function BAPI_BUPA_CENTRAL_GETDETAIL.

Modeled Services

15. Right Click on modeled and choose New Application service.



16. Give your Service a Name getBusinessPartnerDetails.



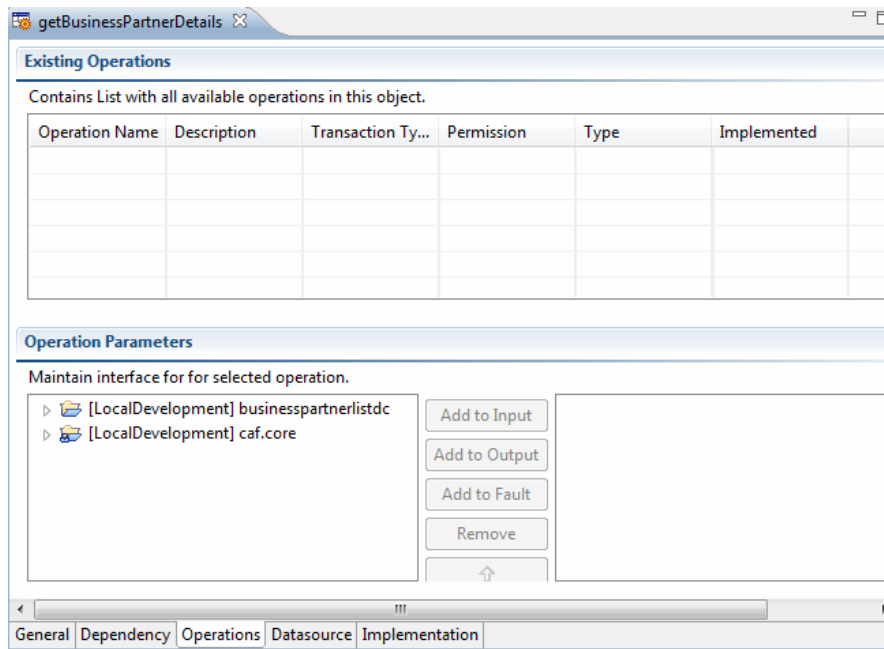
Modeled Services Operations

Now we need to create the operations that will be embedded in our application service and map them to our BAPIs. Remember 2 of our operations are already implemented (No Need to write Code), which are the BAPIs. We need to implement the third operation which will utilize the 2 BAPIs and return a list of Business Partners with their details

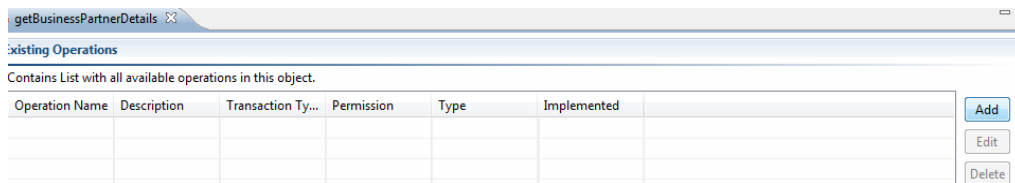
Table 1

| Service | Service Type | Corresponding Operation |
|-----------------------------|---------------------|---------------------------|
| BAPI_BUPA_SEARCH | BAPI | getBusinessPartnerIds |
| BAPI_BUPA_CENTRAL_GETDETAIL | BAPI | getBusinessPartnerDetails |
| getBusinessPartnerDetails | Application Service | getBusinessPartnerList |

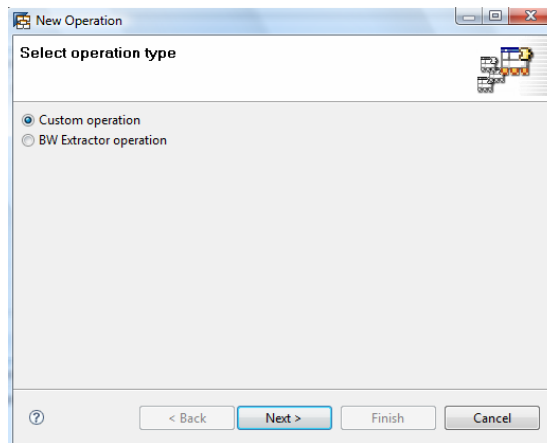
17. Go to the operations tab of your application Service



18. Click on add to create a Data Operation



19. Choose Custom Operation



20. Enter the name of the first Operation getBusinessPartnerIds

21. Enter a description for the operation

22. Choose Transaction Type Supports

23. Choose Permission Check Disable

24. Uncheck the implemented check box.

25. Refer to the following table for the Rest of Operations.

Table 2

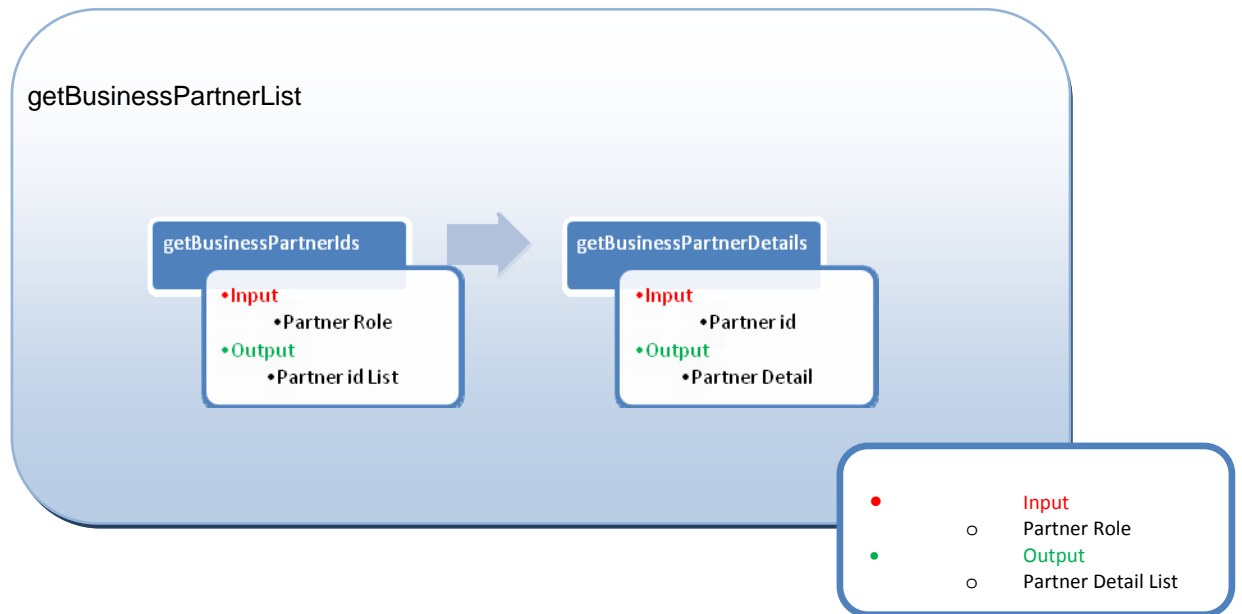
| Operation name | Transaction Type | Permission check | Implemented |
|---------------------------|------------------|------------------|-------------|
| getBusinessPartnerDetails | SUPPORTS | Disable | No |
| getBusinessPartnerList | SUPPORTS | Disable | Yes |

26. Now you should have something that looks like this

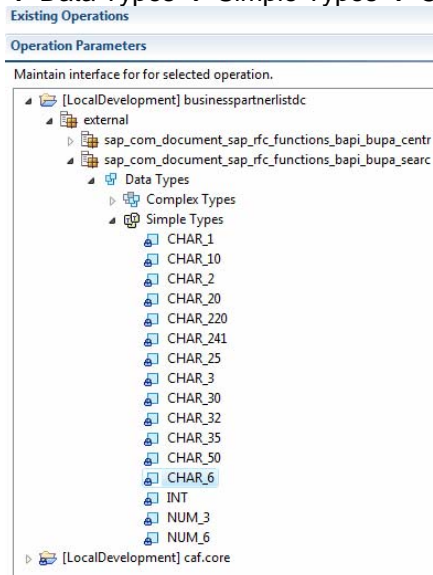
| Operation Name | Description | Transaction Ty... | Permission | Type | Implemented |
|-----------------------------|------------------------------|-------------------|-----------------------------|--------|-----------------------------|
| ● getBusinessPartnerDetails | The Operation Returns a ... | Supports | <input type="checkbox"/> No | Custom | <input type="checkbox"/> No |
| ● getBusinessPartnerIds | The operation return the ... | Supports | <input type="checkbox"/> No | Custom | <input type="checkbox"/> No |
| ● getBusinessPartnerList | | Supports | <input type="checkbox"/> No | Custom | <input type="checkbox"/> No |

Operation Parameters

Now we need to define input and output parameters for our defined operations. For the sake of simplicity we will search our business partners using roles.

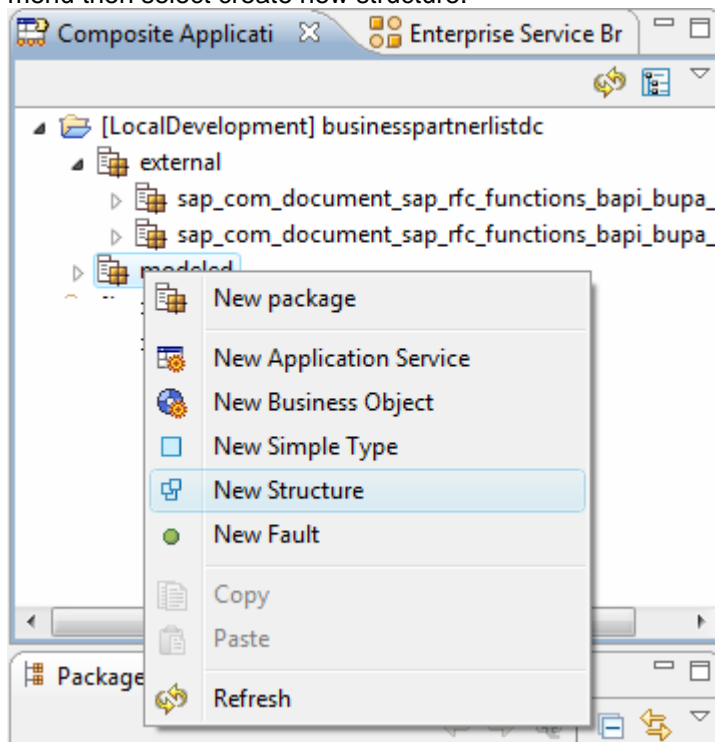


27. Click on getBusinessPartnerIds Operation under the Operation tab.
28. For the operation input parameters select businesspartnerlistdc → external → sap_com_document_sap_rfc_functions_bapi_bupa_searc → Data Types → Simple Types → CHAR_6

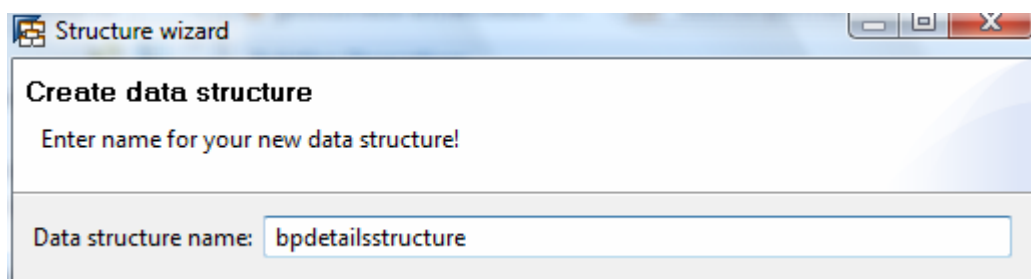


29. Choose Add to Input.
30. Rename the input parameter to bprole.
31. For the operation output parameters select businesspartnerlistdc → external → sap_com_document_sap_rfc_functions_bapi_bupa_searc → Data Types → Complex Types → TbISEARCHRESULT.
32. Choose Add to Output
33. Rename the output parameter to bpidlist.

34. For our second operation `getBusinessPartnerDetails` input parameters select `businesspartnerlistdc` → `external` → `sap_com_document_sap_rfc_functions_bapi_bupa_cent` → `Data Types` → `Simple Types` → `CHAR_10`.
35. Choose add to input.
36. Rename the input parameter to `bpid`.
37. For the operation output parameters select `businesspartnerlistdc` → `external` → `sap_com_document_sap_rfc_functions_bapi_bupa_cent` → `Data Types` → `Complex Types` → `BAPIBUS1006_CENTRAL_PERSON`.
38. Rename the parameter to `bpdetails`.
39. For our third operation `getBusinessPartnerList` input parameters select `businesspartnerlistdc` → `external` → `sap_com_document_sap_rfc_functions_bapi_bupa_search` → `Data Types` → `Simple Types` → `CHAR_6`
40. Rename the input parameter to `bprole`.
41. For the operation output parameter we need a list of type `BAPIBUS1006_CENTRAL_PERSON`, therefore we need to create a new data type.
42. Go to the Composite Application Perspective and under your application select modeled context menu then select create new structure.



43. Name the structure `bpdetailsstructure`



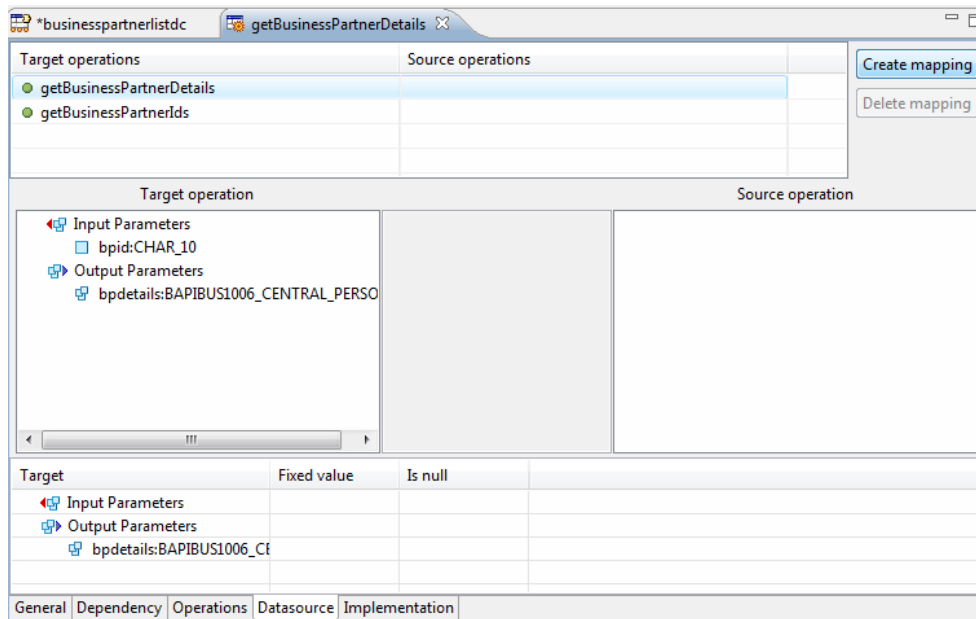
44. In the edit tab go to `businesspartnerlistdc` → `external` → `sap_com_document_sap_rfc_functions_bapi_bupa_cent` → `Data Types` → `Complex Types` → `BAPIBUS1006_CENTRAL_PERSON` from the Existing Types window and move it to your structure list.

45. Name it bpdetaillist.
46. Now to make this a real list under properties of the bpdetaillist structure, change the Cardinality of the structure to 0..n.
47. Now back to the output parameters of method getBusinessPartnerList select businesspartnerlistdc → modeled → data Types → Complex types → bpdetaillist and move it to the output parameters name it bpdetaillist.

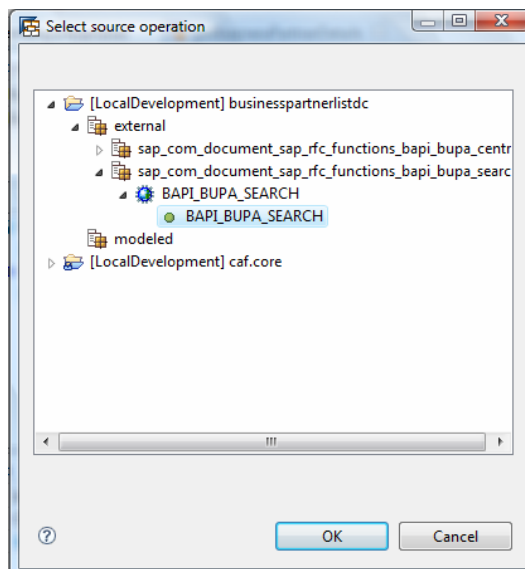
Operation Data Source

Now we map our operations to the standard BAPIs, this avoids us implementing the code for those methods.

48. Click on the Data Source tab.

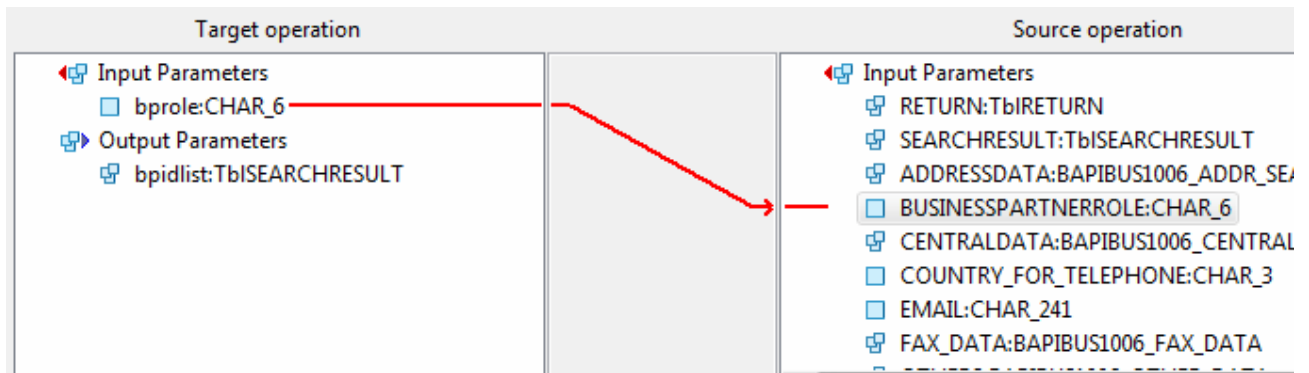


49. Select the operation getBusinessPartnerIds
50. Press the Create mapping button.
51. Drill down to external and choose the BAPI_BUPA_SEARCH

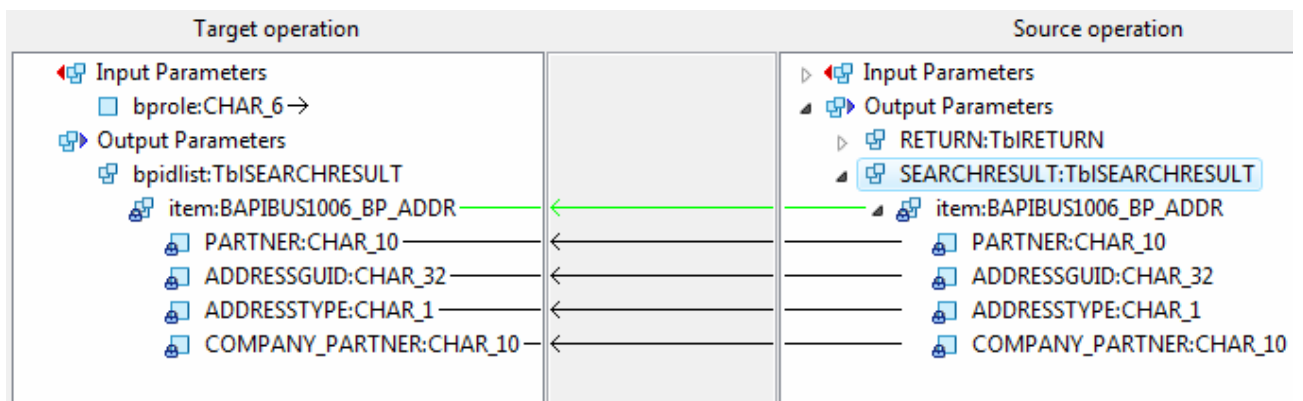


52. Click OK.

53. Drag the parameter BUSINESSPARTNERROLE from the input parameters of the source operation to the input parameter bprole of the source operation and then release it.

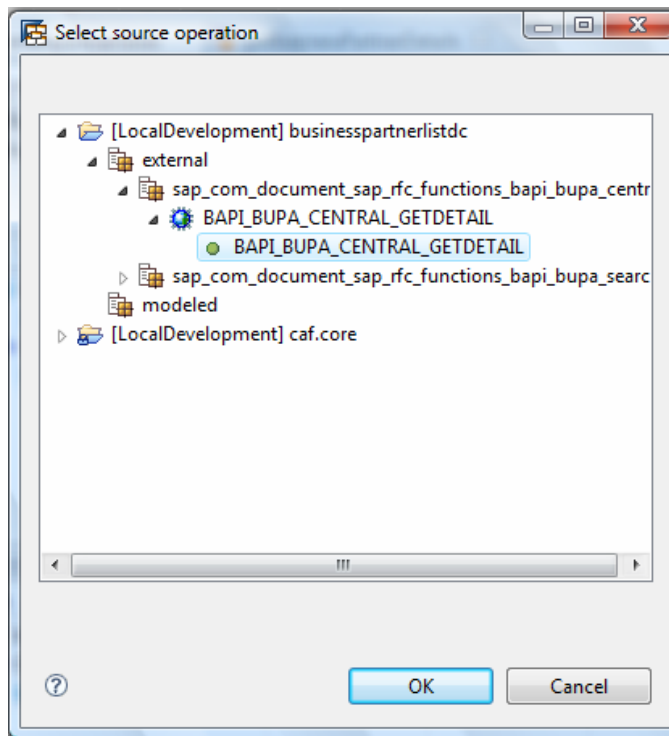


54. Drag the parameter TbISEARCHRESULT from the output parameter of the source operation to the output parameter bpidlist of the source operation and then release it.



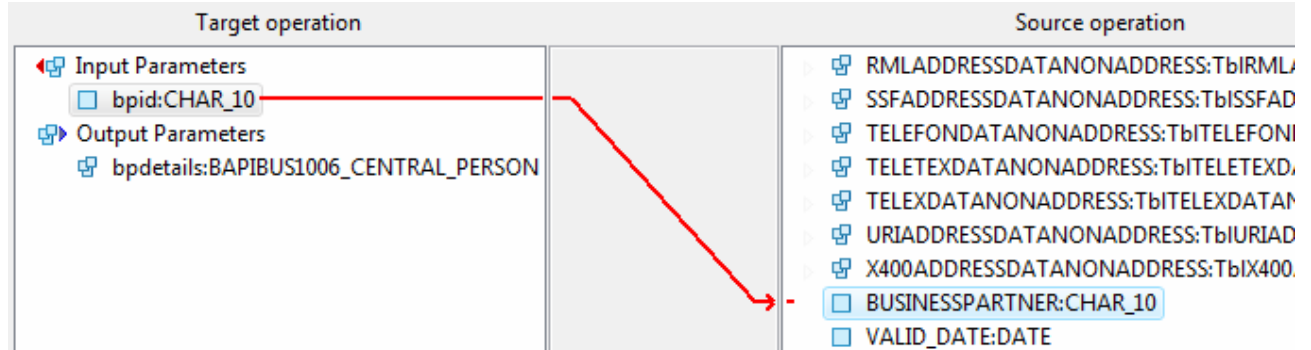
55. Select the operation getBusinessPartnerDetails
56. Press the Create mapping button.

57. Drill down to external and choose the BAPI_BUPA_CENTRAL_GETDETAIL

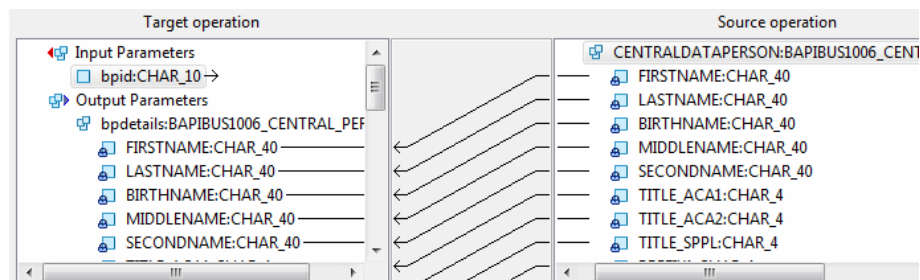


58. Press OK.

59. Drag the parameter BUSINESSPARTNER from the input parameters of the source operation to the input parameter bpid of the source operation and then release it.

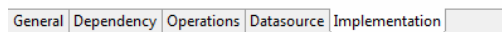
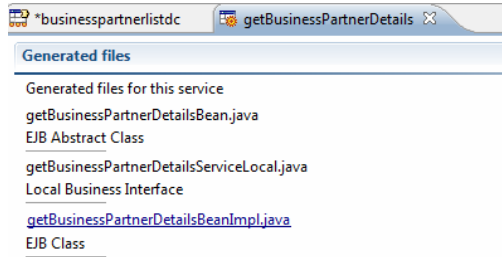


60. Drag the parameter CENTRALDATAPERSON from the output parameter of the source operation to the output parameter bpdetails of the source operation and then release it.



Operation Implementation

61. Now we come to the coding part, remember we never mapped the operation `getBusinessPartnerList` in the data source tab because of the checkbox implemented that we checked.
62. Click on the Implementation Tab



63. Select the `getBusinessPartnerDetailsBeanImpl.java`
Now we see

```
public com.sap.demo.businesspartnerlistdc.types.Bpdetailsstructure
getBusinessPartnerList(java.lang.String bprole)
```

This method `getBusinessPartnerList` accepts a `bprole` as input and returns a `Bpdetailsstructure` List

64. Remove the return null and start typing.
65. We need to initialize :

| Type | Name | Use |
|---|------------|---|
| Bpdetailsstructure | detaillist | This is the list we will fill with the partner details, and should be returned by this method |
| ArrayList< BAPIBUS1006_CENTRAL_PERSON > | details | This will be a temp storage for our details while we loop to get the Partner details |
| TbISEARCHRESULT | Idresults | This is the list of ids that will be retrieved according to the Business Partner role passed to this method |

```

public com.sap.demo.businesspartnerlistdc.types.Bpdetailsstructure
getBusinessPartnerList(java.lang.String bprole)
{
    Bpdetailsstructure detaillist = new Bpdetailsstructure();
    ArrayList< BAPIBUS1006_CENTRAL_PERSON> Names = new ArrayList<
BAPIBUS1006_CENTRAL_PERSON>();
    TblSEARCHRESULT idresults = new TblSEARCHRESULT();
}

```

66. Now we pass the input parameter bprole to the method getBusinessPartnerIds this should return a the structure type TblSEARCHRESULT we will store it in idresults.

```

try {
    idresults = getBusinessPartnerIds(bprole);
} catch (CAFSERVICEException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}

```

67. Now we retrieve the partner ids from structure TblSEARCHRESULT.

```
List<BAPIBUS1006_BP_ADDR> BpIdList = idresults.getItem();
```

68. Now we shall loop on this list and get each Business Partner Details using the method getBusinessPartnerDetails, on each loop we will pass the Partner id and retrieve his details, hence we will fill the Array List Names.

```

for(BAPIBUS1006_BP_ADDR x :BpIdList)
{
    String BPartnerId = x.getPARTNER();
    try {
        Names.add(getBusinessPartnerDetails(BPartnerId));
    } catch (CAFSERVICEException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

69. Now we have our list so we use the set method to set our return parameter detaillist with the Names List.

```

detaillist.setBpdetaillist(Names);
return detaillist ;

```

70. Here is the full Code.

```
package
```

```
com.sap.demo.businesspartnerlistdc.modeled.appsrv.getbusinesspartnerdetails;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import
```

```
com.sap.caf.rfc.bapi_bupa_central_getdetail.types.BAPIBUS1006_CENTRAL_PERSON;
```

```
import com.sap.caf.rfc.bapi_bupa_search.types.BAPIBUS1006_BP_ADDR;
```

```
import com.sap.caf.rfc.bapi_bupa_search.types.TblSEARCHRESULT;
```

```
import com.sap.caf.rt.exception.CAFServiceException;
```

```
import com.sap.demo.businesspartnerlistdc.types.Bpdetailsstructure;
```

```
@javax.ejb.Stateless(name =
```

```
"com.sap.demo.businesspartnerlistdc.modeled.appsrv.getbusinesspartnerdetails.getBusinessPartnerDetails")
```

```
@javax.ejb.Local(value = {
```

```
com.sap.demo.businesspartnerlistdc.modeled.appsrv.getbusinesspartnerdetails.getBusinessPartnerDetailsServiceLocal.class })
```

```
@javax.interceptor.Interceptors(value = {
```

```
com.sap.caf.rt.interceptors.LogInterceptor.class })
```

```
public class getBusinessPartnerDetailsBeanImpl
```

```
extends
```

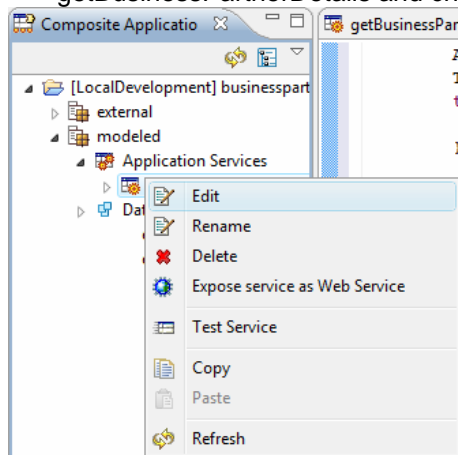


```

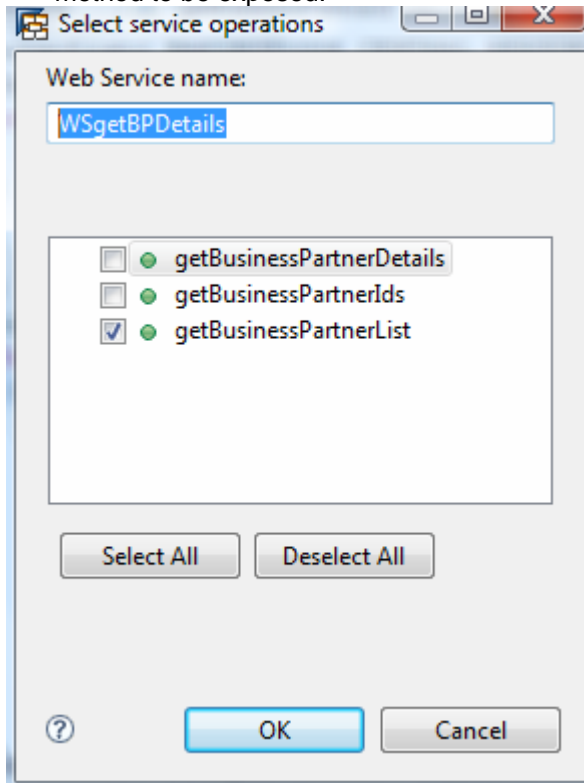
    com.sap.demo.businesspartnerlistdc.modeled.appsrv.getbusinesspartnerdetail
s.getBusinessPartnerDetailsBean {
    @com.sap.caf.dt.CAFOperation(name = "getBusinessPartnerList")
    public com.sap.demo.businesspartnerlistdc.types.Bpdetailsstructure
getBusinessPartnerList(
        java.lang.String bprole) {
        Bpdetailsstructure detaillist = new Bpdetailsstructure();
        ArrayList< BAPIBUS1006_CENTRAL_PERSON> Names = new ArrayList<
BAPIBUS1006_CENTRAL_PERSON>();
        Tb1SEARCHRESULT idresults = new Tb1SEARCHRESULT();
        try {
            idresults = getBusinessPartnerIds(bprole);
        } catch (CAFSERVICEException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        List<BAPIBUS1006_BP_ADDR> BpIdList = idresults.getItem();
        for(BAPIBUS1006_BP_ADDR x :BpIdList)
        {
            String BPartnerId = x.getPARTNER();
            try {
                Names.add(getBusinessPartnerDetails(BPartnerId));
            } catch (CAFSERVICEException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}
detaillist.setBpdetaillist(Names);
return detaillist ;
}}

```

71. Now in the Composite Application Prespective select the Application Service `getBusinessPartnerDetails` and choose `Expose as Web Service`



72. Name the Web Service WSgetBPDetails, and select the method getBusinessPartnerList as the method to be exposed.



73. Click OK.

74. Now GBD your Application (Generate Build and deploy your Application).

Testing the Web Service

75. Go to <http://WASHOST:WASPORT/nwa> (example <http://cicsrvbiw:50000/nwa>)
76. Choose Configuration Management → Destinations
77. Add your Abap system to the systems

Destination Detail

Save Cancel Ping Destination

RFC Destination SAP_ECC

Connection and Transport Logon Data Specific Data

Connection

Load Balancing: Yes No

Local System Connection:

Target Host:

System Number:

System ID:

Message Server:

Logon Group:

Gateway Host:

Gateway Service:

SNC

SNC: Active Inactive

QoP:

SNC Partner Name:

78. Now go <http://washost:wasport/caf>.
79. Choose administrative tools → external service Configuration
80. Choose Service Registry.
81. In External Services column Select demo.sap.com/businesspartnerlistdc → Remote Function Calls → BAPI_BUPA_SEARCH
82. In destination column select remote Function Calls and select the system you created in NWA.
83. Click on MAP button
84. Repeat steps 81-83 for RFC BAPI_BUPA_CENTRAL_GETDETAIL.
85. Save your action and exit.
86. Go to <http://washost:wasport/>
87. Choose Web Service Navigator
88. Choose your web service
89. Select the operation we exposed getBusinessPartnerList

90. Enter the role TR0640 and hit Execute, result should be similar as :

The screenshot displays the SAP NetWeaver Developer Studio interface. On the left, the 'Parameters' tab shows a text field for 'bprole' containing the value 'TR0640'. On the right, the 'Result' tab shows a tree view of the execution results. The tree structure is as follows:

- bpdetailstructure:
 - bpdetaillist []:
 - + bpdetaillist:
 - + bpdetaillist:
 - bpdetaillist:
 - FIRSTNAME: Ina
 - LASTNAME: Imhof

91. You should now be able to use this service in Visual Composer.

Related Content

[Real Word Composites](#)

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.