Creating Multiple Methods/Operations and Exposing BAPI as a Webservice

Applies to:

SAP Netweaver 7.0 SP14. For more information, visit the SOA Management homepage.

Summary

This article discuss about how to create multiple operations and exposing BAPI as a webservice and consuming the webservice using different ways.

SOAMANAGER t-code will be available in the SAP application system only from SP>=14 (service pack) and remaining t-codes related to webservice administration like WSADMIN and WSCONFIG are obsolete.

Author: Pavan Kumar Nukala

Company: Intelligroup Asia Pvt. Ltd.

Created on: 20 January 2011

Author Bio

Pavan Kumar Nukala is an SAP Netweaver Consultant in Intelligroup Asia Pvt. Ltd. published several articles and blogs.

Table of Contents

Introduction:

We had a requirement to create multiple function modules which are remote enabled and exposing them as webservices all together. So that the customer can choose the particular webservice which they want to consume from the available services (webservices).For example we will consider an example to know the details of the flight availability and list of flights and details of the flight from the remote location at a time. To fulfill this requirement follow the below steps:

Step 1:

Go to t-code **SE37** and check for the Remote enabled function modules

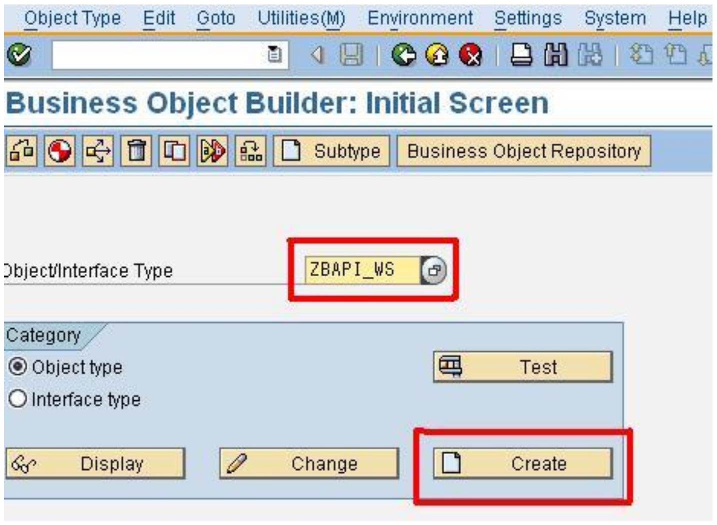
BAPI_FLIGHT_CHECKAVAILIBILITY

BAPI_FLIGHT_GETDETAIL

BAPI_FLIGHT_GETLIST

Step 2:

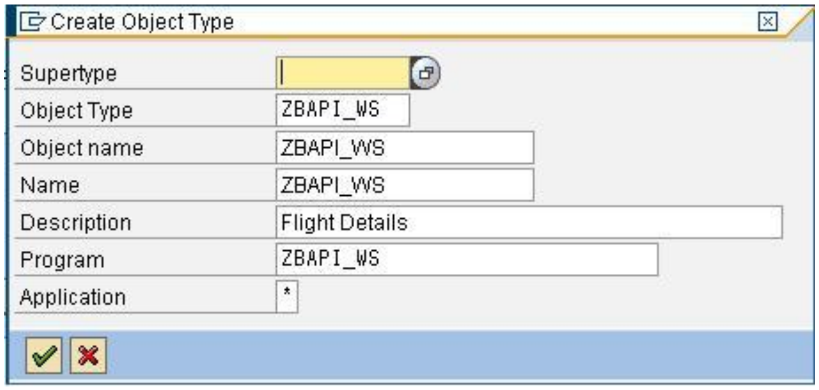
Hit the transaction **SWO1 (not zero it is alphabet 'O')** and provide the object/interface name as ZBAPI_WS and click on create button as shown in below figure.



Object/Interface Type: ZBAPI_WS

Category: [✓ Object type] Interface type

A popup appears as below and provide the required mandatory fields as below.

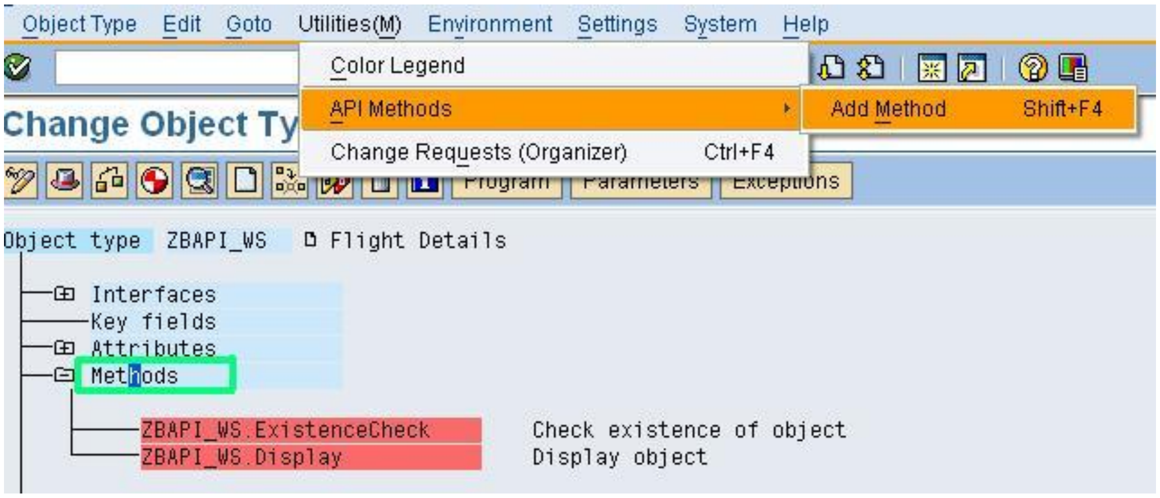


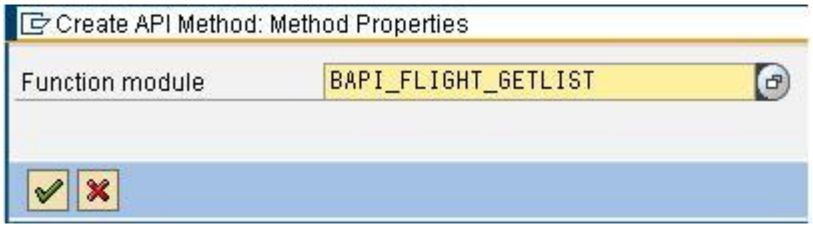
Supertype	
Object Type	ZBAPI_WS
Object name	ZBAPI_WS
Name	ZBAPI_WS
Description	Flight Details
Program	ZBAPI_WS
Application	*

Click on continue and save it to a package.

Step 3:

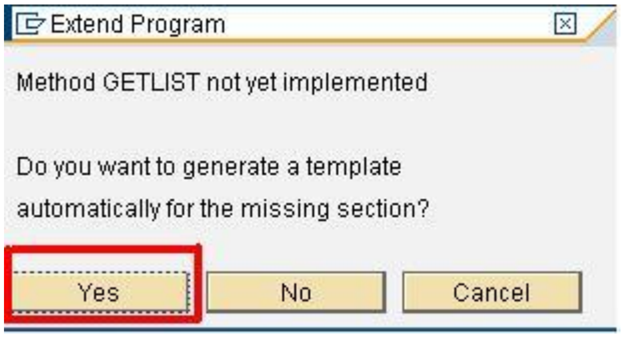
Select the methods and click on from Menu→ Utilities→ API methods→ Add Method and provide the name of the function module.


Object Type Edit Goto Utilities(M) Environment Settings System Help
Color Legend
API Methods
Add Method Shift+F4
Change Requests (Organizer) Ctrl+F4
Change Object Ty
Program Parameters Exceptions
Object type ZBAPI_WS Flight Details
Interfaces
Key fields
Attributes
Methods
ZBAPI_WS.ExistenceCheck Check existence of object
ZBAPI_WS.Display Display object


Create API Method: Method Properties

Function module BAPI_FLIGHT_GETLIST

After giving function module name click on continue and follow the ***wizard process***. In this wizard process we can select the fields for request and response based on our requirement and finally clicks on Yes button as shown in below.


Extend Program

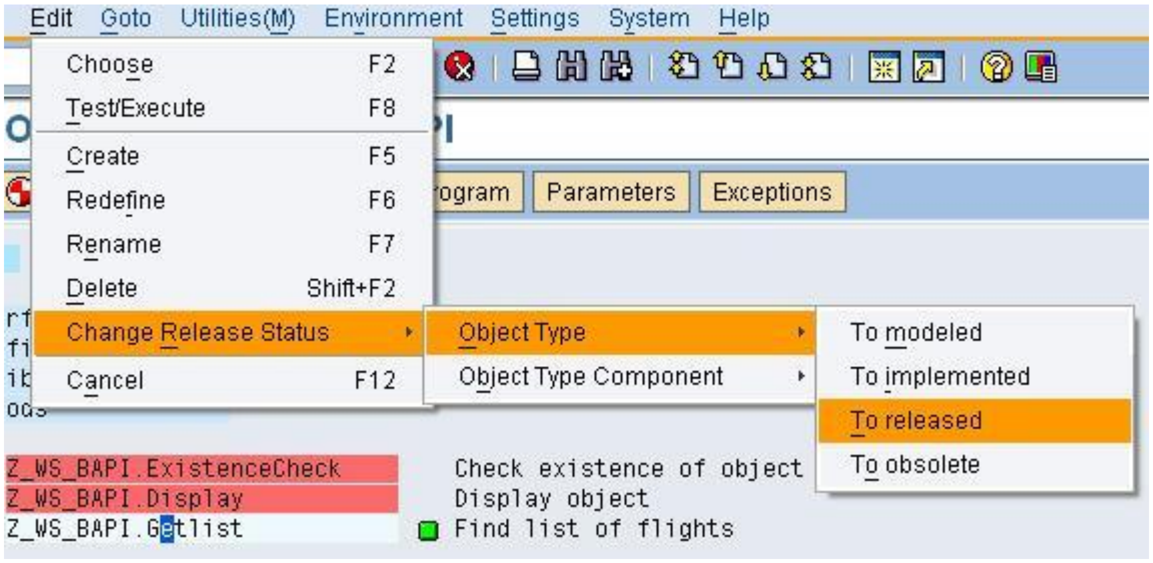
Method GETLIST not yet implemented

Do you want to generate a template automatically for the missing section?

Now the RFC will be under Methods node.

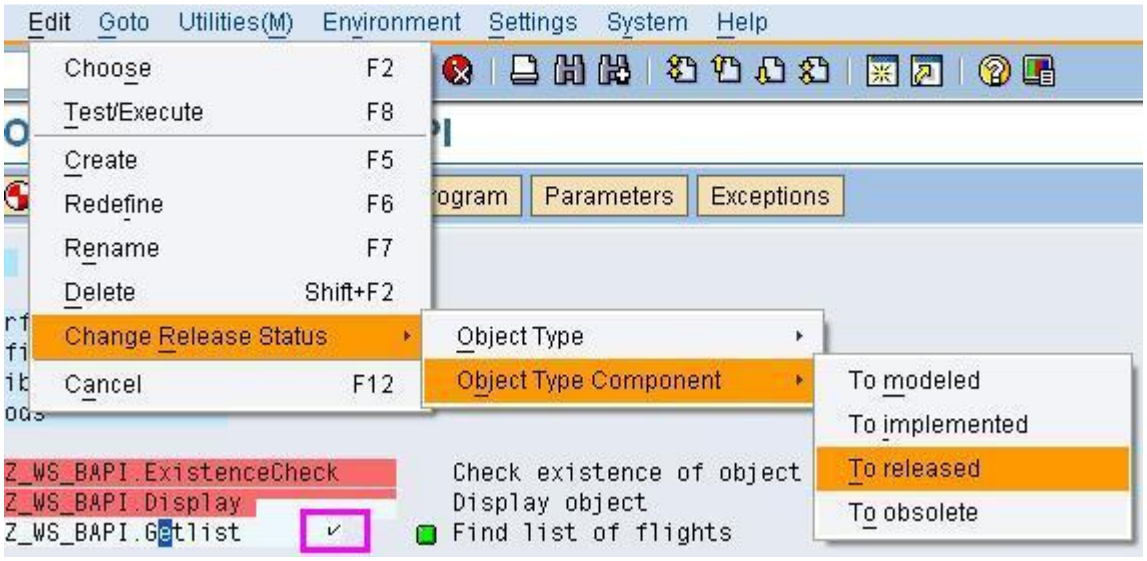
Step 4:

In this step we have to release the object and method. Before releasing we need to ***model*** and ***implement***. Menu→Edit→Change Release Status→Object Type→To Implemented and To Released as below



```
Z_WS_BAPI.ExistenceCheck     Check existence of object
Z_WS_BAPI.Display            Display object
Z_WS_BAPI.Getlist            Find list of flights
```

Repeat the above step for object type component also. Once you released you can see an icon next to method in the box which shows the release status as below.

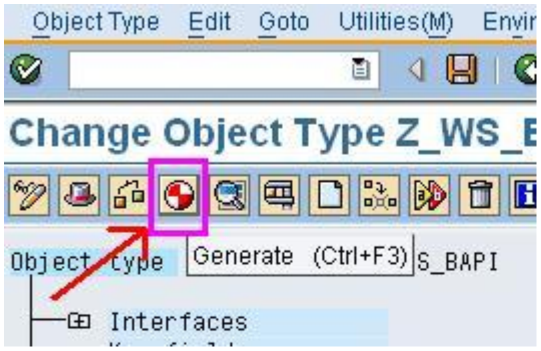


```
Z_WS_BAPI.ExistenceCheck       Check existence of object
Z_WS_BAPI.Display              Display object
Z_WS_BAPI.Getlist        ✓     Find list of flights
```

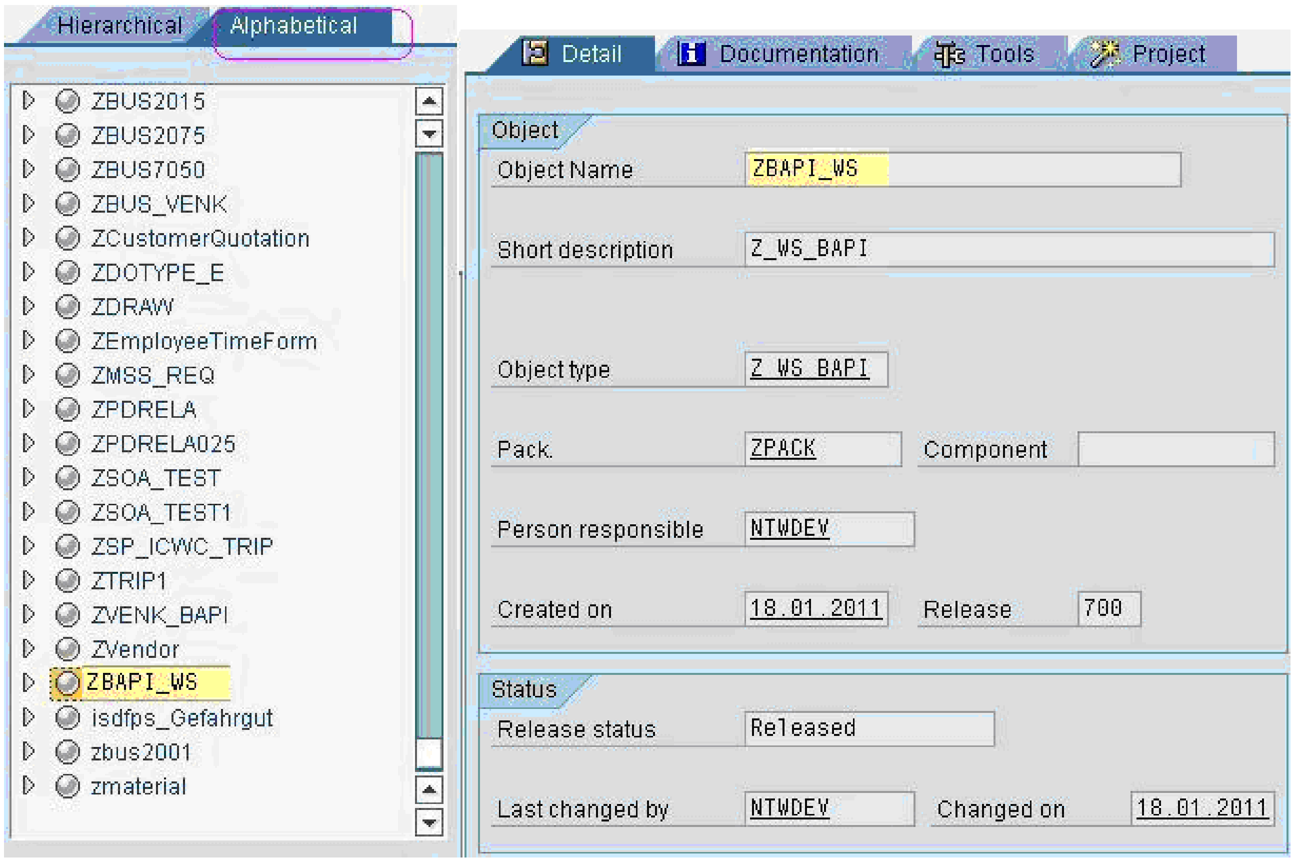
Repeat the same above steps for the remaining function modules as discussed in above steps.

Step 5:

Click on generate button as per the below screen.



Hit the transaction **BAPI** and search for your object name and the properties of the object. Once the object is released and generated we can view it in **BAPI** t-code. Click on alphabetical tab to search the object.



Object

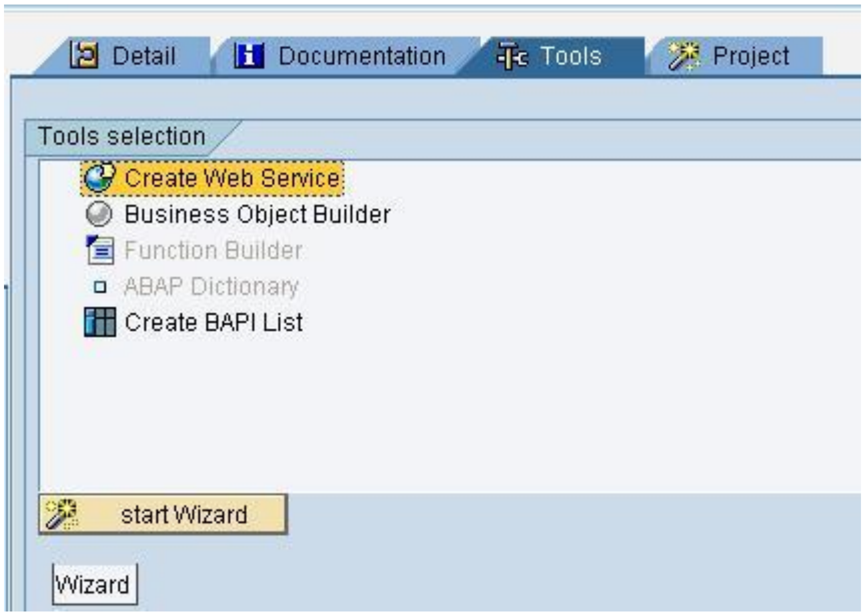
Object Name	ZBAPI_WS		
Short description	Z_WS_BAPI		
Object type	Z_WS_BAPI		
Pack.	ZPACK	Component	
Person responsible	NTWDEV		
Created on	18.01.2011	Release	700

Status

Release status	Released		
Last changed by	NTWDEV	Changed on	18.01.2011

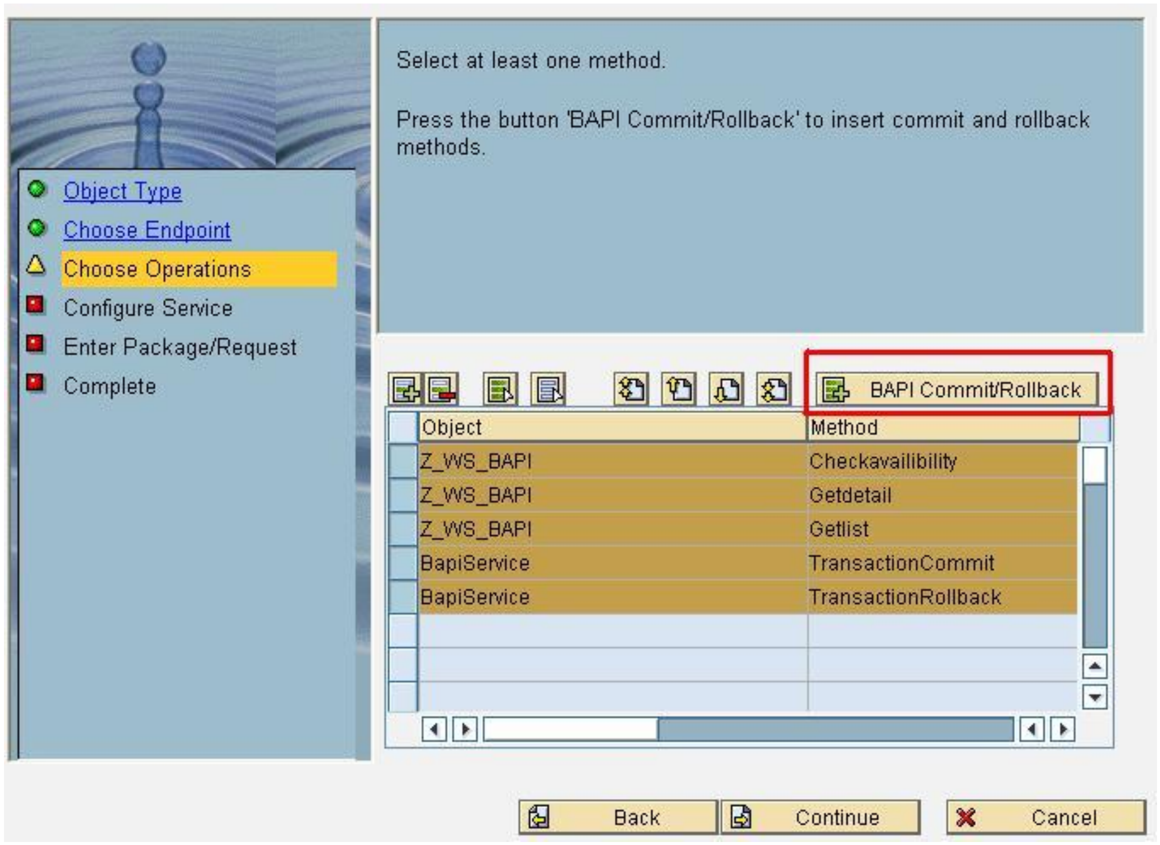
Step 6:

Click on Tools tab and select create webservice and click on start wizard as shown below.



Follow the wizard process to create webservice as below and click on continue button.

As we all are familiar in generating webservice using wizard process I am not showing the steps for this.



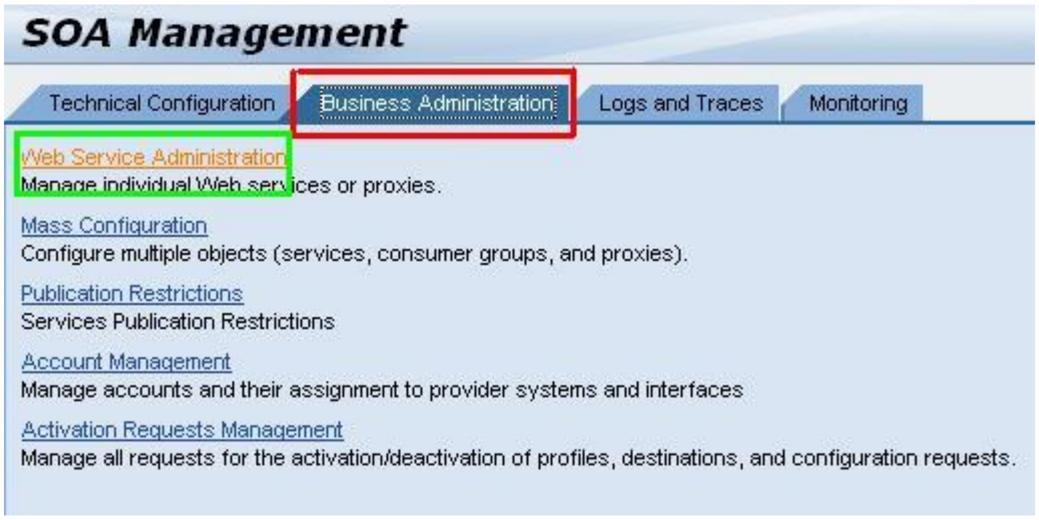
Select at least one method.

Press the button 'BAPI Commit/Rollback' to insert commit and rollback methods.

Object	Method
Z_WS_BAPI	Checkavailibility
Z_WS_BAPI	Getdetail
Z_WS_BAPI	Getlist
BapiService	TransactionCommit
BapiService	TransactionRollback

Step 7:

Hit the t-code: SOAMANAGER and log on to it. Click on Business Administration tab and click on Web Service Administration as shown below.



Web Service Administration
Manage individual Web services or proxies.

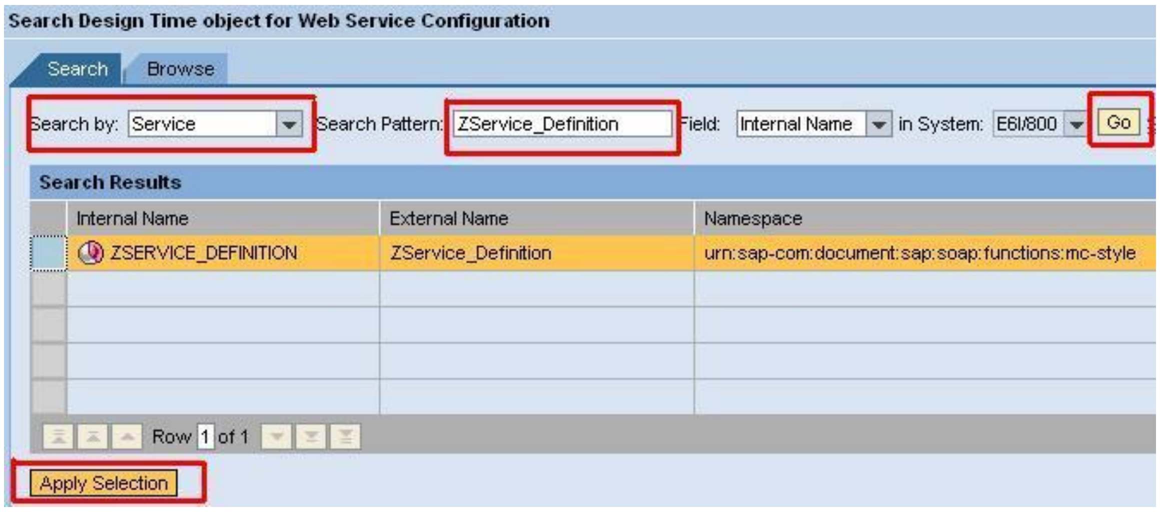
Mass Configuration
Configure multiple objects (services, consumer groups, and proxies).

Publication Restrictions
Services Publication Restrictions

Account Management
Manage accounts and their assignment to provider systems and interfaces

Activation Requests Management
Manage all requests for the activation/deactivation of profiles, destinations, and configuration requests.

Select the search by category as service and provide the name of the service definition (if remember or search as Z*) and click on Go button. Our webservice will be displayed, select the service and click on apply selection button as below.



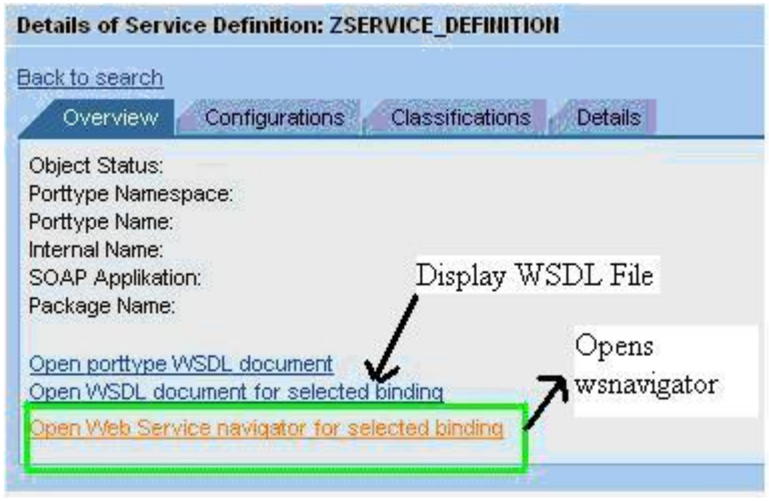
Search by: Service
Search Pattern: ZService_Definition
Field: Internal Name
in System: E6I/800

Search Results

Internal Name	External Name	Namespace
ZSERVICE_DEFINITION	ZService_Definition	urn:sap-com:document:sap:soap:functions:mc-style

Row 1 of 1

After click on apply selection click on the webservice navigator link to test the webservice as below.



In our case we have three methods or operations. We can choose the method or operation which we want to test. In the same way we can display the WSDL by clicking on the URL and provide the WSDL to the client or customer who wants to utilize/consume this kind of application which is exposed as webservice.

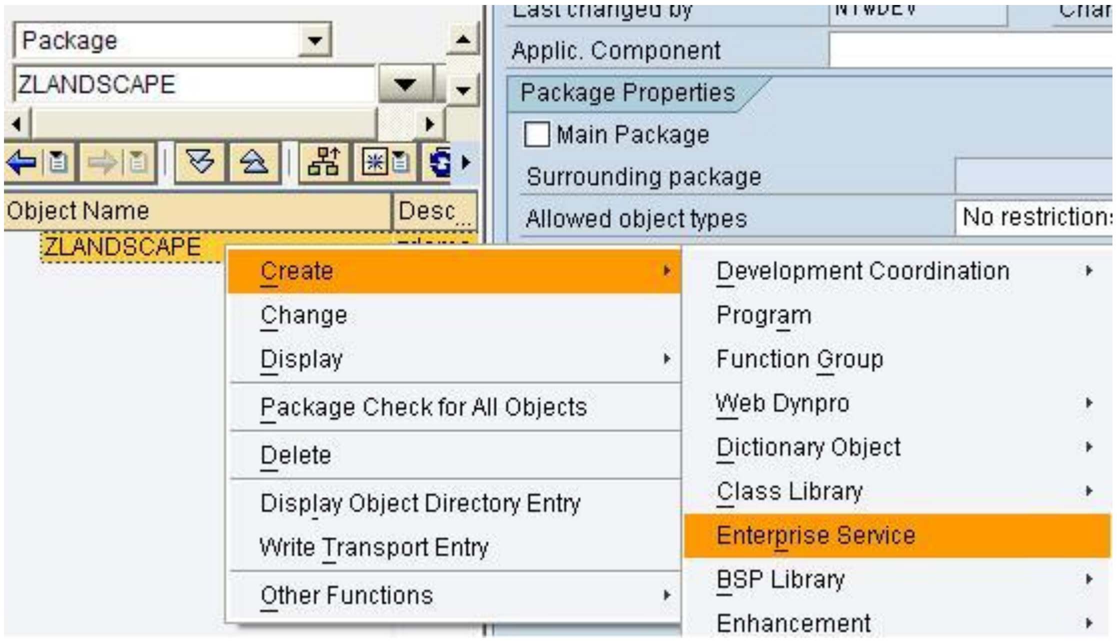
Consuming webservice:

The above created webservice can be consumed using different approaches.

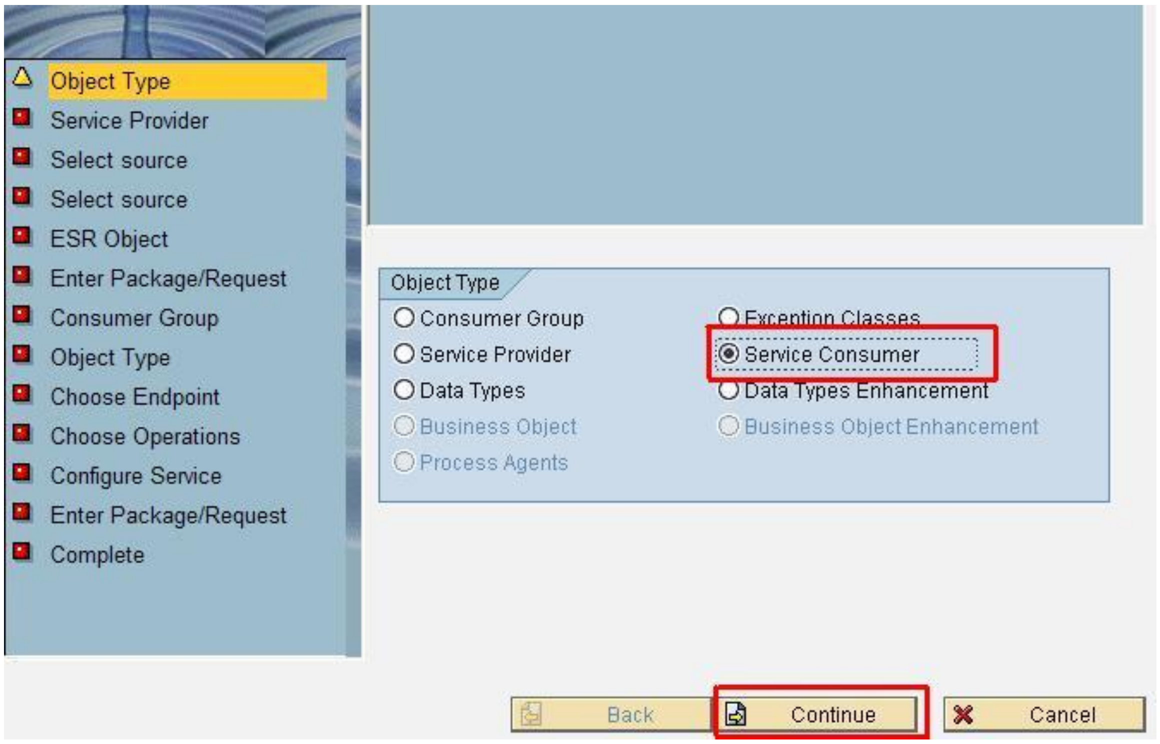
If we want to consume the above generated webservice from an SAP application system which is in different landscape we can create client proxy from object navigator.

Approach 1:

Hit the t-code **SE80** select our own package and right click on package and create → enterprise service as shown below.

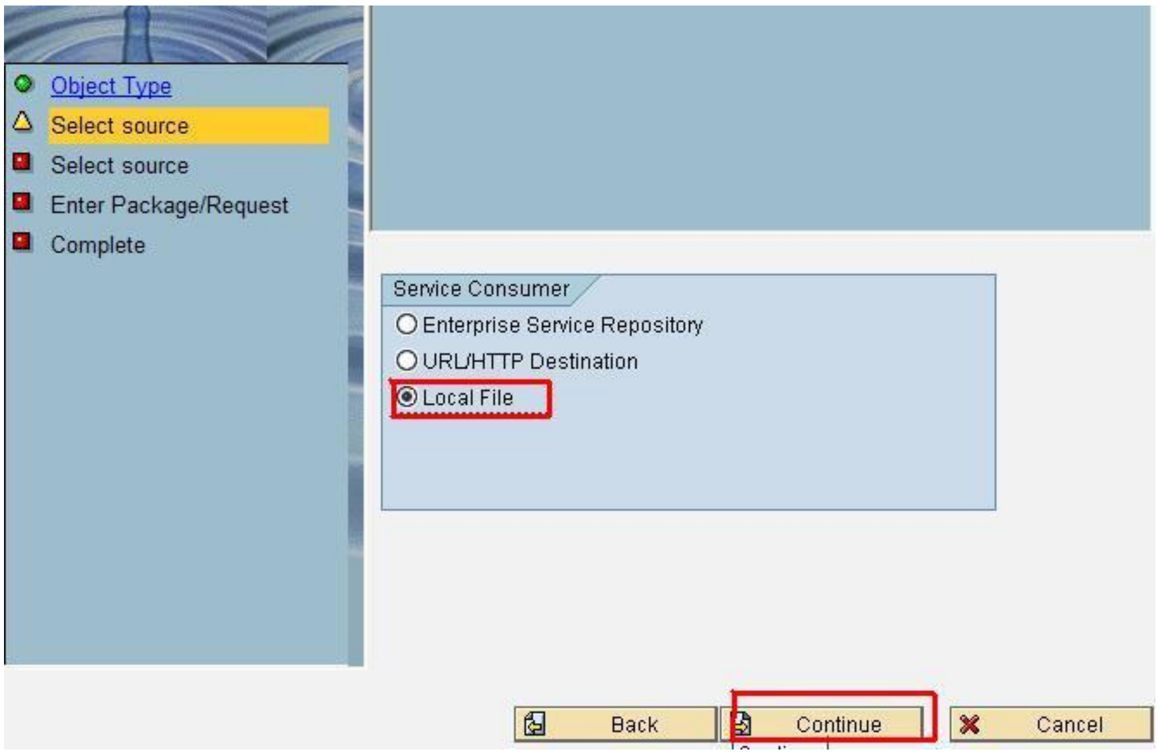


Select the option service consumer and click on continue button



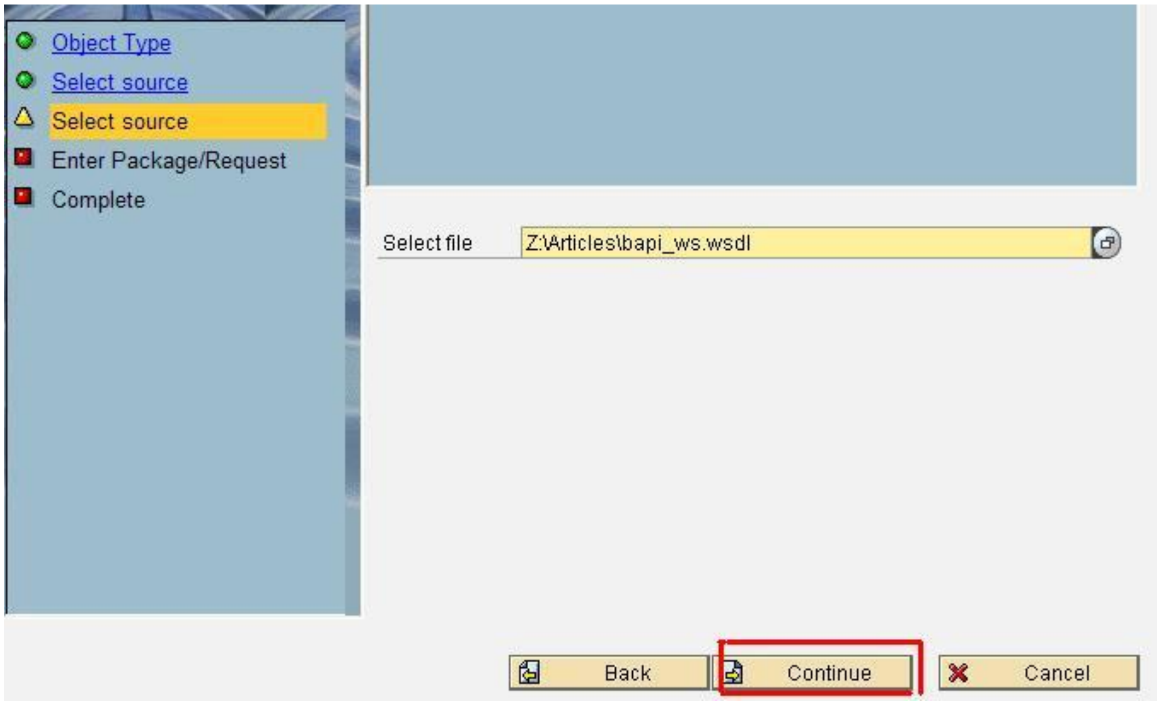
Object Type
- ( ) Consumer Group
- ( ) Service Provider
- ( ) Data Types
- ( ) Business Object
- ( ) Process Agents
- ( ) Exception Classes
- [✓ Service Consumer]
- ( ) Data Types Enhancement
- ( ) Business Object Enhancement

Select the option as local file and click on continue as shown in below.



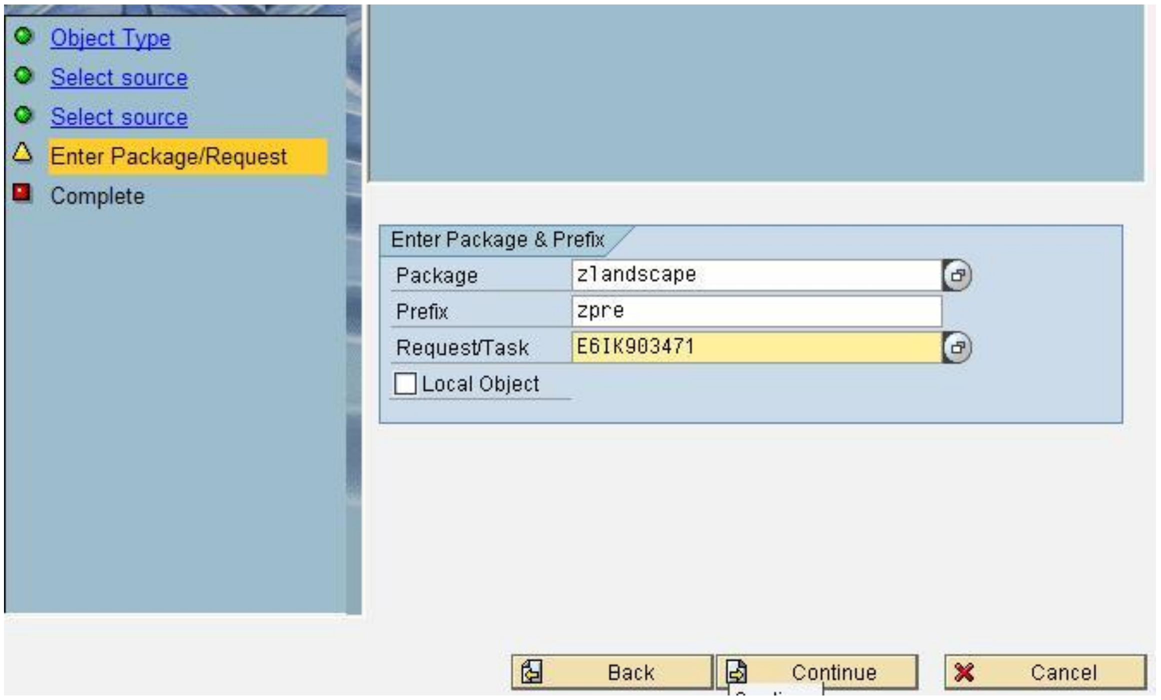
Service Consumer
- ( ) Enterprise Service Repository
- ( ) URL/HTTP Destination
- [✓ Local File]

Now select the WSDL file which we obtained from the service provider/client. We can see we are importing the WSDL file from our desktop/local directory.



Select file Z:\Articles\bapi_ws.wsdl

Enter the package and request number and finally click on complete button which complete the wizard process in creating the client proxy for consuming webservice.



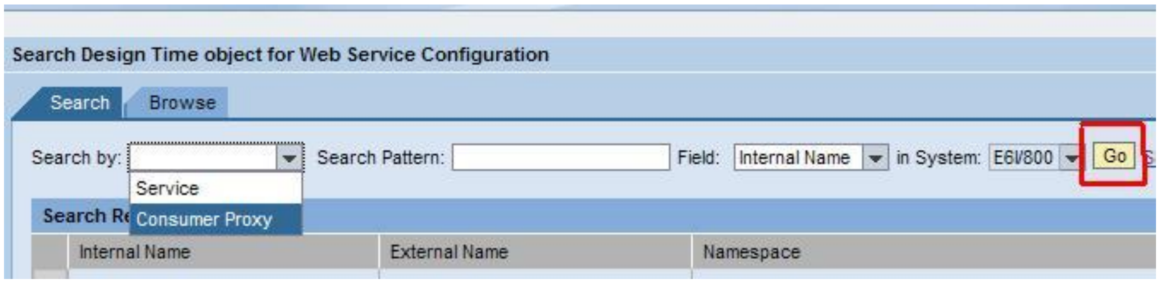
Enter Package & Prefix

Package	zlandscape
Prefix	zpre
Request/Task	E6IK903471

☐ Local Object

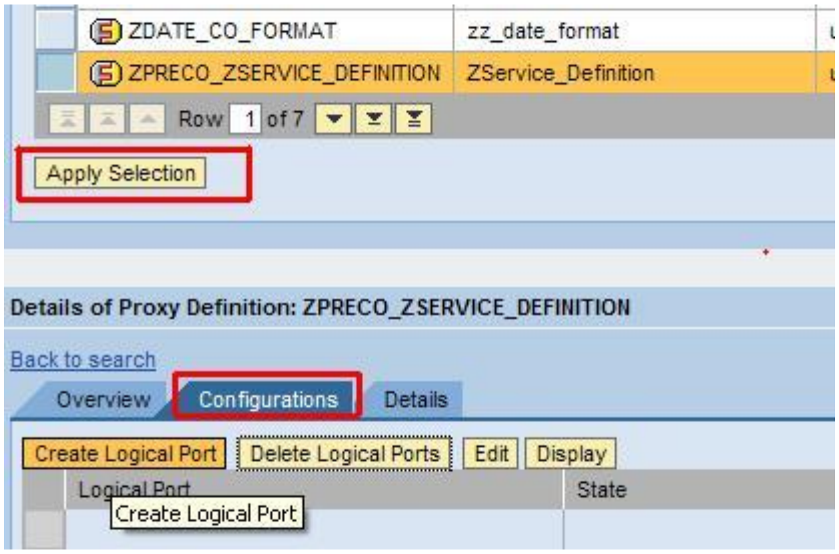
Now our client proxy will be created under the Package→Enterprise Services → Client Proxies →Our Proxy Class.

Once the proxy class is created create the logical port for the generated class in **SOAMANAGER**.Log on to SOAMANAGER and click on business administration → click on webservice administration and from the drop down select the consumer proxy and click on go button.



Search Design Time object for Web Service Configuration

Select our proxy class and click on apply selection and under configuration tab click on create logical port.

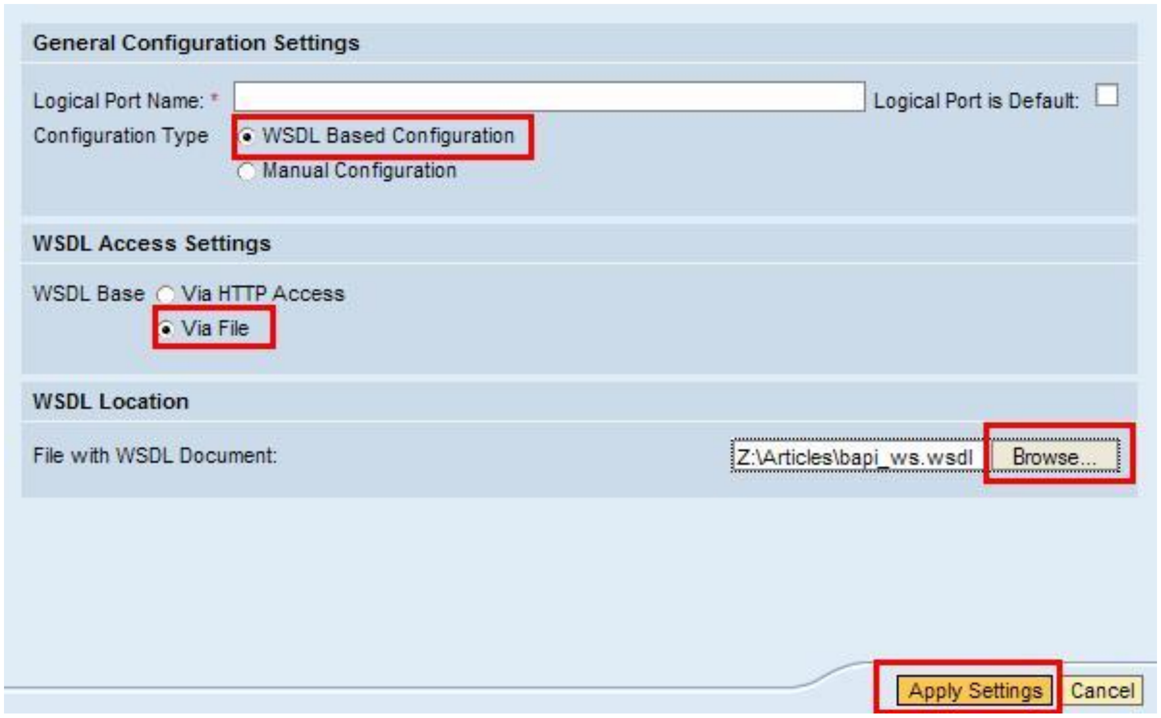


Internal Name	External Name
ZDATE_CO_FORMAT	zz_date_format
ZPRECO_ZSERVICE_DEFINITION	ZService_Definition

Row 1 of 7

Details of Proxy Definition: ZPRECO_ZSERVICE_DEFINITION

When we click on logical port a screen appears and select the options as below and select the WSDL file which we obtained from the client and click on apply settings and save.



General Configuration Settings

Logical Port Name: * ______ Logical Port is Default: ☐

Configuration Type: [✓ WSDL Based Configuration] Manual Configuration

WSDL Access Settings

WSDL Base: Via HTTP Access [✓ Via File]

WSDL Location

File with WSDL Document: Z:\Articles\bapi_ws.wsdl

Now the generated proxy class will be called into a report program which will be saved in the same package.

Approach 2:

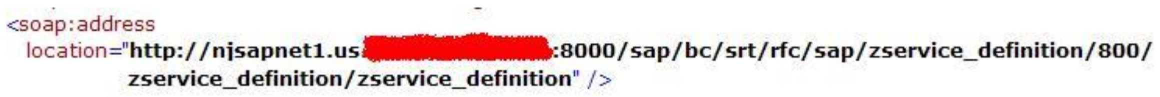
Webservice is independent of the technology on which it is developed. So we can consume the webservice in either the .net application or java application.

Approach 3:

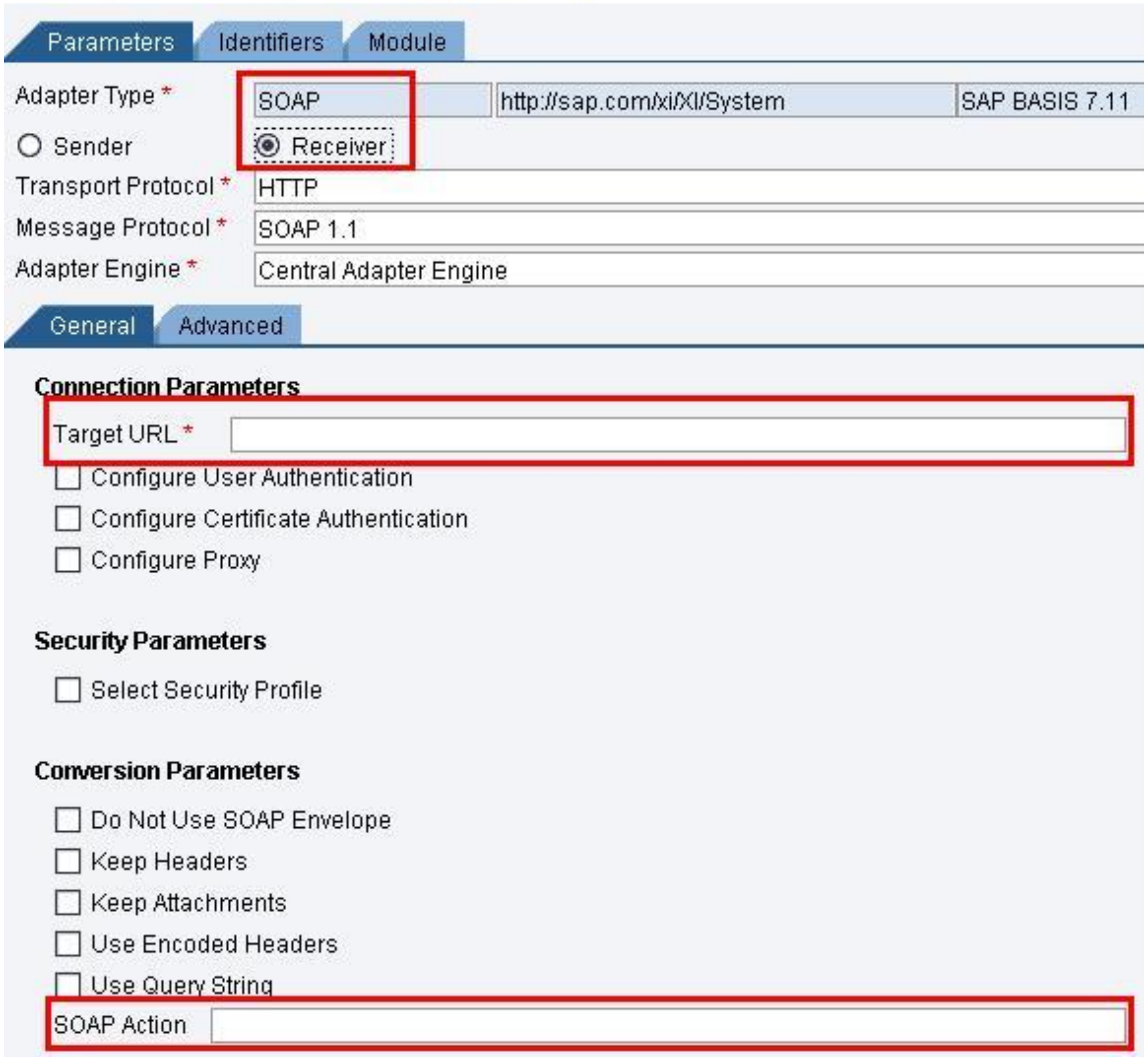
We can also consume the above created webservice by means of SAP PI using receiver SOAP adapter. (Process Integration).Before consuming the webservice using SAP PI we need to import the WSDL file into Enterprise service builder under external definition and mapping program will be created based on source structure and target web service and operation mapping will be done.

Log on to integration directory and configure the receiver SOAP adapter and provide the required details. We can get the Target URL value form WSDL file as soap address location and soap action will be the operation name or method name which can de identified as operation name in WSDL file.

In our case we have three different kinds of operations but at a time we can consume only one operation as of now PI 7.11 does not support for consuming multiple operations at a time.



```
<soap:address
  location="http://njsapnet1.us[redacted]:8000/sap/bc/srt/rfc/sap/zservice_definition/800/
           zservice_definition/zservice_definition" />
```



Adapter Type *	SOAP	http://sap.com/xi/XI/System	SAP BASIS 7.11
○ Sender	[✓ Receiver]		
Transport Protocol *	HTTP		
Message Protocol *	SOAP 1.1		
Adapter Engine *	Central Adapter Engine		

Connection Parameters

Target URL *

☐ Configure User Authentication
☐ Configure Certificate Authentication
☐ Configure Proxy

Security Parameters

☐ Select Security Profile

Conversion Parameters

☐ Do Not Use SOAP Envelope
☐ Keep Headers
☐ Keep Attachments
☐ Use Encoded Headers
☐ Use Query String

SOAP Action

Related Content

SOAMANAGER: How to test service definition using SOAMANAGER transaction

Configuration of Enterprise services using SICF and SAO Manager

SAP Community Network Forums » Service-Oriented Architecture (SOA) » Service-Oriented Architecture

Manual Testing of Enterprise services

http://help.sap.com

For more information, visit the SOA Management homepage

Disclaimer and Liability Notice