Creating, Configuring and Testing a Web Service Based on a Function Module

Applies to:
SAP EC6 6.0/7.0. For more information, visit the Web Services homepage.

Summary
The article describes how to create a web service from function module.

Author: Ajantha Ratnakumar
Company: Larsen & Toubro Infotech Limited
Created on: 10 September 2009

Author Bio
Ajantha Ratnakumar is a Software Consultant in Larsen & Toubro Infotech Limited. She has three years of SAP experience. She has worked extensively in ABAP, in APO modules such TPVS and SNP and in SD module.
### Table of Contents

- Concepts .................................................................................................................. 3
- Web Service ............................................................................................................... 3
- Creating Function Group ......................................................................................... 3
  - Transaction SE80 .................................................................................................... 3
- Creating Function module ......................................................................................... 5
  - Transaction SE37 .................................................................................................... 5
- Code .......................................................................................................................... 8
- Create Web Service of Function Module ................................................................. 9
- Executing Web Service from Web Service Navigator ............................................ 14
  - Transaction SOAMANAGER .................................................................................. 14
- Summary .................................................................................................................. 24
- Related Content ....................................................................................................... 24
- Disclaimer and Liability Notice ................................................................................ 25
Concepts

Web Service

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. Web services are frequently just Web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.

Or

Web service can be defined as any service/functionality available over the internet and related transport protocols which uses a standardized XML messaging system (i.e. SOAP) is not tied to any operating system or programming language.

Creating Function Group

Transaction SE80

Select the “Function Group” in the Drop down box on left side plane in the Object Navigator Screen and then enter the Function Group name to be created in the input box below the drop down box and then press ENTER and click on “Yes” button on the pop up window for creating Function Group.

Object Navigator

You will get pop up asking for entering short text for Function group, provide a meaningful short text and then “SAVE”.

Create Function Group

Function group ZV_ATP_ES
Short text Function group for web service
Person Responsible DEVELOPER
While saving it will ask the package under you want to save, you can provide $TMP$ (for Local Object) or the any package name as per your requirement. Then click on “SAVE”.

You will get a message saying Function Group has been created as per below screen shot
Creating Function module

Transaction SE37
Enter the Function Module name to be created and click on “Create”

Function Builder: Initial Screen

Enter the Function group name and Short text for function module in the pop up window as per below screen and click on “SAVE”, then you will get a pop up saying “Function module name is reserved for SAP”, click on “Continue”.

Create Function Module

Function Module ZV_CREATE_SALESORDER_ES_POC
Function group ZV_ATP_ES
Short text Function module to create sales order for ES POC

Function module name is reserved for SAP.
Then it will take you to “Function Builder” screen as in below screen shot and enter meaningful values. And a very important thing is to make the function module “Remote-Enabled Module”.

**Function Builder: Change ZV_CREATE_SALESORDER_ES_POC**

- **Function Group**: ZY_ATP_ES  
- **Function Module**: ZV_CREATE_SALESORDER_ES_POC

**Import parameters for Function Module**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Associated Type</th>
<th>Default Value</th>
<th>Opt</th>
<th>Pos</th>
<th>Short Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUSTOMER_NAME</td>
<td>BNAME_V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Name of order</td>
</tr>
<tr>
<td>REQUIRED_DATE</td>
<td>EDATU_VBAK</td>
<td>SV-DATUM</td>
<td></td>
<td></td>
<td></td>
<td>Requested delivery date</td>
</tr>
<tr>
<td>REQUIRED_TIME</td>
<td>EZEIT_VSEP</td>
<td>SV-UZEIT</td>
<td></td>
<td></td>
<td></td>
<td>Amount time</td>
</tr>
<tr>
<td>PLANT</td>
<td>WERKS_D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Plant</td>
</tr>
<tr>
<td>MATERIAL</td>
<td>MATNR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Material Number</td>
</tr>
<tr>
<td>QUANTITY</td>
<td>ZMENG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Target quantity in sales units</td>
</tr>
<tr>
<td>UOM</td>
<td>DZIEME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Target quantity UoM</td>
</tr>
</tbody>
</table>
Define the **Export parameters** for Function module

**Function Builder: Change ZV_CREATE_SALESORDER_ES_POC**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Associated Type</th>
<th>Pass Val</th>
<th>Short Text</th>
<th>Long Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>SALEDOCUMENT</td>
<td>TYPE</td>
<td>BAPIYBELN-VEELN</td>
<td></td>
<td>Sales Document</td>
<td>Cre...</td>
</tr>
</tbody>
</table>

Define the **tables** for Function module

**Function Builder: Change ZV_CREATE_SALESORDER_ES_POC**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Associated Type</th>
<th>Optional</th>
<th>Short Text</th>
<th>Long Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETURN</td>
<td>LIKE</td>
<td>BAPIRET2</td>
<td></td>
<td>Return Parameter</td>
<td>Cre...</td>
</tr>
</tbody>
</table>

One of the important things to be done is **commenting and maintaining header information** for the function module which gives information about the function module

```
FUNCTION ZV_CREATE_SALESORDER_ES_POC.
** Local Interface:

** IMPORTING

  VALUE (CUSTOMER_NAME) TYPE RNAME_V
  VALUE (REQUIRED_DATE) TYPE EDATU_VDAK DEFAULT SY-DATUM
  VALUE (REQUIRED_TIME) TYPE EZEIT_VDEB DEFAULT SY-UZEIT
  VALUE (PLANT) TYPE VPLANT
  VALUE (MATERIAL) TYPE MATNR
  VALUE (QUANTITY) TYPE QMNG
  VALUE (ITEM) TYPE OVZWEI

** EXPORTING

  VALUE (SALEDOCUMENT) TYPE BAPIYBELN-VEELN

** TABLES

  RETURN STRUCTURE BAPIRET2
```

---

© 2009 SAP AG

SAP COMMUNITY NETWORK

SDN - sdn.sap.com | BPX - bpx.sap.com | BOC - boc.sap.com
Code

For the example mentioned in document (i.e. creating a sales order) you can use this code or else the code will change as per your requirement

```abap
DATA: order_header_in LIKE bapisdhd1.
  order_partners LIKE bapisopnr OCCURS 0 WITH HEADER LINE.
  order_schedules_in LIKE bapischd1 OCCURS 0 WITH HEADER LINE.
  order_items_in LIKE bapisditm OCCURS 0 WITH HEADER LINE.
  WF_DOC_TYPE TYPE AUART VALUE 'OR'.

CALL FUNCTION 'CONVERSION_EXIT_AUART_INPUT'
  EXPORTING
    INPUT = 'OR'
  IMPORTING
    OUTPUT = WF_DOC_TYPE

* Fill header data
  order_header_in-doc_type = WF_DOC_TYPE.
  order_header_in-sales_org = '5000'.
  order_header_in-distr_channel = '50'.
  order_header_in-division = '50'.
  order_header_in-name = customer_name.
  order_header_in-purch_no_c = '1234'.
  order_header_in-req_date_h = required_date.

* Fill order items data
  order_items_in-itn_number = '000010'.
  order_items_in-material = material.
  order_items_in-plant = plant.
  order_items_in-target_qty = quantity.
  order_items_in-target_uom = uom.
  append order_items_in.

* Fill order partners data
  order_partners-partn_role = 'SP'.
  order_partners-partn_numb = customer_name.
  append order_partners.

* Fill schedule line data
  order_schedules_in-itn_number = '000010'.
  order_schedules_in-sched_line = '000010'.
  order_schedules_in-req_date = required_date.
  order_schedules_in-req_qty = quantity.
  order_schedules_in-date_type = '1'.
  order_schedules_in-sched_type = 'CP'.
  order_schedules_in-req_time = required_time.
  append order_schedules_in.

* Call BAPI to create Salesorder
  CALL FUNCTION 'BAPI_SALESORDER_CREATEFROMDAT2'
    EXPORTING
      order_header_in = order_header_in.
      convert = 'X'.
    IMPORTING
      salesdocument = salesdocument.
    tables
      return = return.
      order_items_in = order_items_in.
      order_partners = order_partners.
      order_schedules_in = order_schedules_in.

* If salesorder is created successfully commit the changes to the database
  IF salesorder IS NOT INITIAL.
    CALL FUNCTION 'BAPI_TRANSACTION_COMMIT'
      EXPORTING
        wait = 'X'.
  ENDIF.
ENDFUNCTION.
```
Create Web Service of Function Module

Go to utilities -> More Utilities -> Create Web Service -> From the Function Module

Enter **Name** and **Short text** for Service definition and the click on "Continue"
Check the Check box for **Mapping** so that the import and export parameters of Function Module are mapped with that of the Web Service to be created.
Check the **Deployment** check box for the service to be available in SOAMANAGER for further processing.
Here you can select where you want to save the object. In this case, I am saving it as a local object "$TMP".
Click on "Completed" to completed the process of Web Service creation.

The Web Service ZV_ARK_CREATE_SO_NEW will be created.

Transaction Wsadmin (Administration Web Services for SOAP Runtime) supports you with the UDDI registration of the Web Service.

Then you will a message that web service has been created.
Executing Web Service from Web Service Navigator

Transaction SOAMANAGER

Click on the tab “Application and Scenario Communication”

SAP Easy Access

Click on the tab “Application and Scenario Communication”

SOA Management

- Single Service Administration
  - Provides functions for administration and configuration of single web services and web service clients.
  - Business Scenario Communication
    - Configure the communication of a whole business scenario.
  - Publication Rules
    - Create publication rules that define the set of services and consumer groups that has to be published in a Services Registry.
  - User Account Management
    - Manage user 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.
Then click on “Single Service Administration” you will get the below screen where you can enter the service definition name in the Search Pattern. In our case the name will be “ZV_ARC_CREATE_SO_NEW” The click on “GO”.

You will get the service in the search Results column. Select the service and then click on “Apply Selection”.

You can see as per the below screen shots, the service = 0/ Endpoints = 0 for the web service definition selected by us.
Now click on **Configurations** tab and then click on “Create Service”

You will get a pop up asking for service **name, description and binding name** for it. Enter the values and click on “**Apply Settings**”

Then you return to the below mentioned screen where you need to save
Then you can see an entry in the **Configurations** tab. The state of the service should be **Active**.

Now click on the **Overview** tab, you can see the service and Endpoint=

If you click on the "Open WSDL document for selected binding" you can see the WSDL of the Web service in XML format:

```xml
<?xml version='1.0' encoding='utf-8'?>
 xmlns:wdsl="http://schemas.xmlsoap.org/wsdl/"
 xmlns:soap="http://schemas.xmlsoap.org/soap/http"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
 xmlns:tns="urn:sap-com-document-sap:soap:functions:mc-style"
 xmlns:wsa="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
 xmlns:tnsi="urn:sap-com-document-sap:rfc:functions">
  <wsdl:documentation>
  </wsdl:documentation>
</wsdl:definitions>
```
In order to execute the web service through web service navigator click on "Open Web Service navigator for selected binding" it will take you to above screen.

Then click on "Display selected Binding’s WSDL URL" link to get the WSDL URL. You will get a URL displayed on the right and side of the screen.
Copy – paste the link in the “Web service Navigator” screen in the input box below “Enter the WSDL URL of the Web service:” and click on “Next”.

Enter the WSDL URL of the Web service:

38/wSDL11/allinfoe/ws_poiucu/document?sap-client=300 [Next]

Available Web Services - Cluster Node 9353550
- AdobeDocumentServices
- AdobeDocumentServicesSec
- BizCommLayerAuthoring
- BizCommLayerUtilities
- CAFDataService
- CmsAuthorityClAPI
- CMSRTS
- CmsTransportAPIService
- CmsTransportAPIService_DL2
- DeployProxy

Web Services Navigator

Welcome to the Web Services Navigator

Enter WSDL

Web Services Navigator is a tool that gives you a short overview for a specific web service, based on its WSDL and allows you to create and send a client request to the real endpoint. In order to continue specify the location of the WSDL. Note the WSDL URL must be reachable from the server where this Web Services Navigator is installed. If http proxy is needed, be sure it is configured properly from the Visual Administrator. A tree with the local for the server web services is shown too. Expand it in order to get the WSDL in a specific style.
It will take you to the login screen. You need to use the same login as used for SAP logon.

On this screen, click on "Test".

Web Services Navigator

**Authorization**

The selected endpoint requires basic authentication. Please, enter the correct username and password:

- **Username:** [Enter]
- **Password:** [Enter]

Submit

On this screen, click on "Test".

Web Services Navigator

**Overview**

**WSDL:**

http://compact2hr.intinfootech.com:8000/sap/bc/srt/wsdl/bndg_DEA13C5C600946F156E20D1F29E7ED93/wsdl11/ällinomyyrs_policy/soap.9?
sap-client=303

**Description:**
The click on the “ZvCreateSalesorderEsPoc (test.types.p1.ZvCreateSalesorderEsPoc parameters)”

Enter the values for creating sales order:

- **CustomerName (String)**
- **Material (String)**
- **Plant (String)**
- **Quantity (java.math.BigDecimal)**
- **RequiredDate (String)**
- **RequiredTime (String)**
- **Return (test.types.p1.Bapiret2[])**
- **element1 (test.types.p1.Bapiret2)**
- **Ucm (String)**

Timeout (seconds): 180

Send
After entering the mandatory values for creating sales order, click on "Send".
Then you will a the output in the same format

Response

```
ZvCreateSalesorderWrap

response (test.types.p1.ZvCreateSalesorderWrapResponse)
- ESalesdocument (String) 0000013357
- ItReturn (test.types.p1.Bapiret2[])
  - test.types.p1.Bapiret2
    - Type (String) S
    - Id (String) V4
    - Number (String) 233
    - Message (String) SALES_HEADER IN h
    - LogNo (String)
    - LogMsgNo (String) 000000
    - MessageV1 (String) VBAKKOM
    - MessageV2 (String)
    - MessageV3 (String)
    - MessageV4 (String)
    - Parameter (String) SALES_HEADER IN
    - Row (Integer) 0
    - Field (String)
    - System (String) EC6IDES75

- test.types.p1.Bapiret2
  - Type (String) S
  - Id (String) V4
  - Number (String) 233
  - Message (String) SALES_ITEM IN has
  - LogNo (String)
  - LogMsgNo (String) 000000
  - MessageV1 (String) VBAKCOM
  - MessageV2 (String) 000010
  - MessageV3 (String)
  - MessageV4 (String)
  - Parameter (String) SALES_ITEM IN
  - Row (Integer) 1
  - Field (String)
  - System (String) EC6IDES75
```
Summary
Web Service can be used in Adobe Flex or any other web-based application to access / communicate with SAP data. ABAP web services can be created when you have custom requirements. There are also standard web services provided by SAP which comes as a part of Enhancement Packages. These standard web service represent standard SAP functionalities like APO ATP check, SO create/change/delete, PO create/change/delete and many more. When more than one web service combined to meet new requirements encapsulating enterprise functionality and exposing it as a reusable business service then that is called as an Enterprise Service.

Related Content
https://wiki.sdn.sap.com/wiki/display/ESpackages/Home
http://www.sdn.sap.com/irj/sdn/soa
http://www.sdn.sap.com/irj/sdn/enterprisesoa;jsessionid=(J2EE3417400)ID1961175350DB10082480365165855639End?rid=/webcontent/uuid/f07d8153-7edd-2910-aeac-fb940ff1ff71
http://www.sdn.sap.com/irj/sdn/enterprisesoa;jsessionid=(J2EE3417400)ID1961175350DB10082480365165855639End?rid=/webcontent/uuid/e044d87c-ebab-2a10-6482-cd4ff5fa4828
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.