Integration Framework for SAP Business One

Web Service Call Inbound Adapter

PUBLIC

Global Roll-Out

January 2019, Vinita Yadav

Note: The example templates in this document are not officially supported by SAP.

THE BEST RUN SAP

TABLE OF CONTENTS

Web services describes a standardized way of integrating Web based applications using the XML, SOAP, and WSDL over an Internet protocol . XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available.

Web Service Inbound

With Web service inbound, the integration framework handles incoming Web service calls. Applications can send a Web service request to the integration framework server. For Web service inbound, the integration framework acts as a Web service provider. The integration framework provides functions to generate a WSDL (Web Service Description Language) document. You can generate and hand over the WSDL document to the service consumer.

Integration framework supports synchronous Web service calls and the following Web services standards:

- SOAP 1.1
- WSDL 1.1

The integration framework controls Web service inbound processing using the System Landscape Directory (SLD) entry and with parameters of the inbound definition of the scenario step.

1. PREREQUISITES

1.1. Create a Scenario Package

To create a scenario package in the integration framework for SAP Business One, please refer to the openSAP course In Action - Integration Framework for SAP Business One*.*

In the *Week 2* and *Unit 1* chapter, you learn how to create a scenario package.

1.2. Create a Scenario Package

To create a scenario package in the integration framework for SAP Business One, please refer to the openSAP course In Action - Integration Framework for SAP Business One*.*

In the *Week 2* and *Unit 1* chapter, you learn how to create a scenario package.

2. INBOUND USING THE ROOT TAG AS IDENTIFICATION METHOD

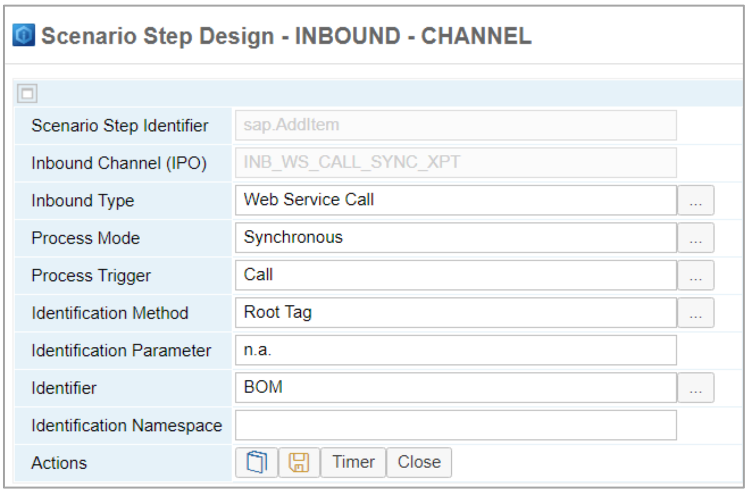
This chapter shows you how to use the **Root Tag** identification method to identify the incoming Web service request.

In this example, we handle the *Web Service request with the specified root tag (BOM).* When you use the root tag as the identification method, you do not need to enter an identification parameter. The identifier is the root tag name of the sender message.

The function inserts item master data to the SAP Business One and provides the response.

Please refer to the sample **sap.AddItem** scenario step available in the sap.in.WebService scenario package.

To define the inbound definition of the scenario step, select *Scenarios → Step Design → [Inbound] → [Channel]*.



The integration framework uses the following parameters to control inbound processing:

- Inbound Type = Web Service Call
- Process Mode = Synchronous
- Process Trigger = Call
- Identification Method = Select how the integration framework identifies the incoming message, either by the root tag or by XPath.
- Identification Parameter = For XPath, enter the XPath expression and use $msg for the incoming message. For root tag identification method, you do not enter anything.
- Identifier = For root tag, enter the name of the root tag. For XPath enter the value of the XPath
- Identification Namespace = We can additionally provide a namespace.

2.1. Outbound Channel

For a synchronous inbound call, the outbound channel is always VOID, since the integration framework sends back the response to the caller.

To define the outbound definition of the scenario step, select *Scenarios → Step Design → [Outbound] → Select* ***Void*** *at [Outbound channel]*.

2.2. Insert Data to SAP Business One

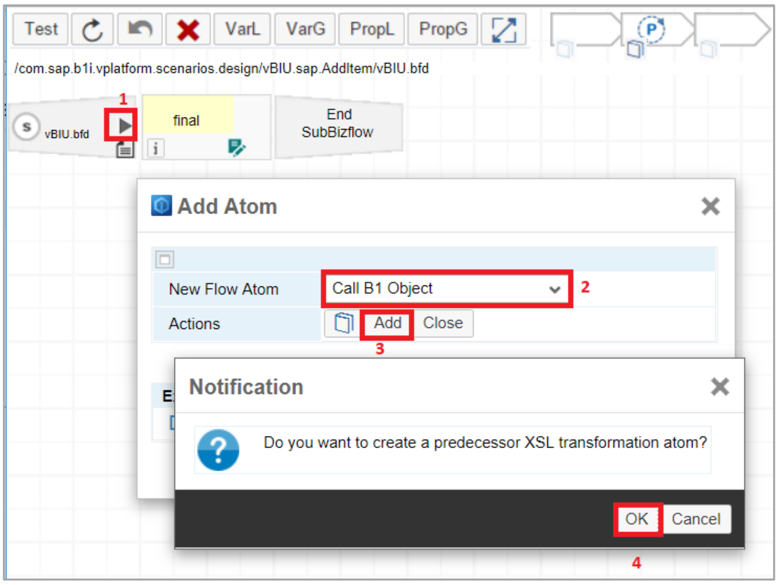
The chapter guides you through the procedure to insert item master data into SAP Business One using Web service post.

2.2.1. Add an Atom to Scenario Step Processing

In this example we have taken B1 Object atom. To add the B1 Object to the process flow:

- (1) Press the ▶ [Add] button on the *Start* atom to insert a new functional processing atom.
- (2) In the *New Flow Atom* field, select the **Call B1 Object** value.

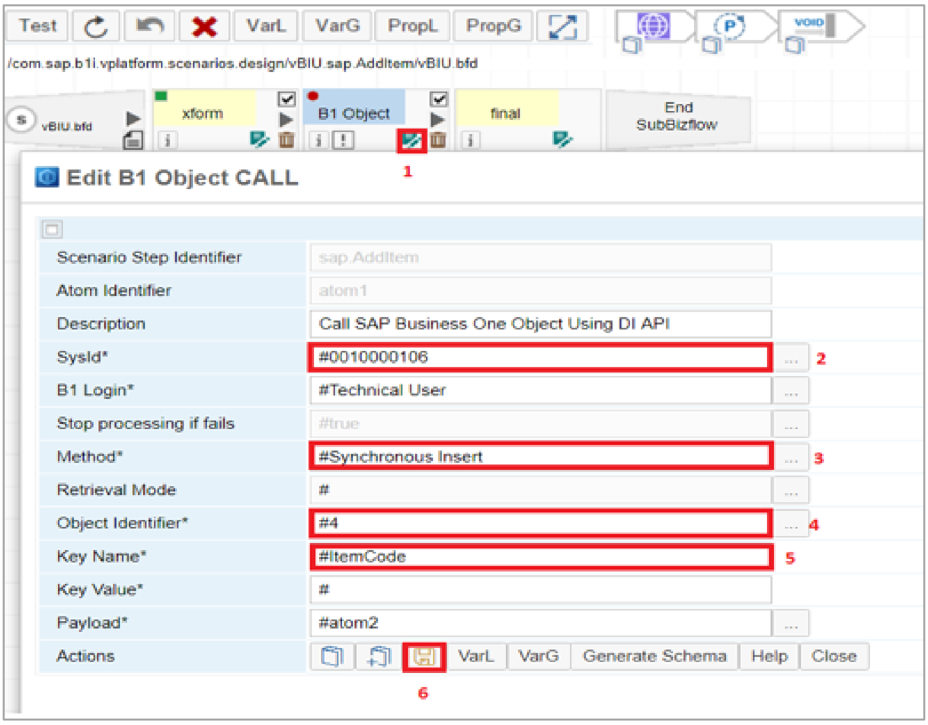
- (3) Click **Add**.
- (4) Finally, press **OK** to generate the *predecessor XSL transformation* atom.



2.2.2.Configure the Functional Processing Atom

To configure the processing atom:

- (1) Press the [Change Configuration] button.
- (2) In the *SysId* field, select the SAP Business One system entry.
- (3) In the *Method* field, select the **Synchronous Insert** value.
- (4) In the *Object Identifier* field, select **Items**.
- (5) The *Key Name* property is filled automatically based on the *Object Identifier*.
- (6) Finally, press the [Save] button.



2.2.3. Customize the XSL Transformation

- Click the XSL Transformation Atom (xform). The integration framework opens the *Embedded XML Editor* to edit the XSL file.
- In the XSL file, change the **transform** template accordingly:

```
<xsl:template name="transform">
  <BOM>
        <BO>
              <AdmInfo>
                    <Object>
                          <xsl:value-of select="$msg/BOM/BO/AdmInfo/Object"/>
                    </Object>
                    <Version>
                          <xsl:value-of select="$msg/BOM/BO/AdmInfo/Version"/>
                    </Version>
              </AdmInfo>
              <Items>
                    <row>
                          <ItemCode>
                                <xsl:value-of select="$msg/BOM/BO/Items/row/ItemCode"/>
                          </ItemCode>
                          <ItemName>
                                <xsl:value-of select="$msg/BOM/BO/Items/row/ItemName"/>
                          </ItemName>
                    </row>
              </Items>
        </BO>
  </BOM>
</xsl:template>
```

2.2.4. Customize the final atom XSL Transformation

- Click the XSL Transformation Atom (xform). The integration framework opens the *Embedded XML Editor* to edit the XSL file.
- In the XSL file, change the **transform** template accordingly:

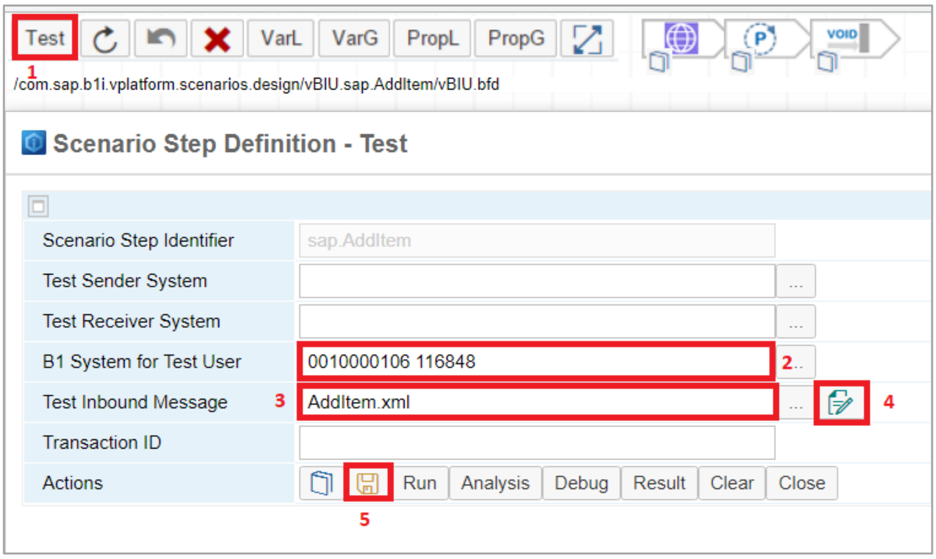
```
<xsl:template name="transform">
   <xsl:attribute name="pltype">xml</xsl:attribute>
   <xsl:variable name="B1CallResult"
select="/vpf:Msg/vpf:Body/vpf:Payload[./@id=&apos;atom1&apos;]"/>
   <AddItemResponse>
         <Result>
               <xsl:value-of select="$B1CallResult/@DIresult"/>
         </Result>
         <DIMsg>
               <xsl:value-of select="$B1CallResult/@DImsg"/>
         </DIMsg>
   </AddItemResponse>
</xsl:template>
```

The final atom returns the DI result and the DI message.

2.3. Scenario Step Definition – Test Configuration

Follow the procedure for Scenario step test:

- (1) Click *Test* button to open *Scenario Step Definition -Test*.
- (2) Select a B1 System for Test User.
- (3) Define the *Test Inbound Message* (AddItem.xml).



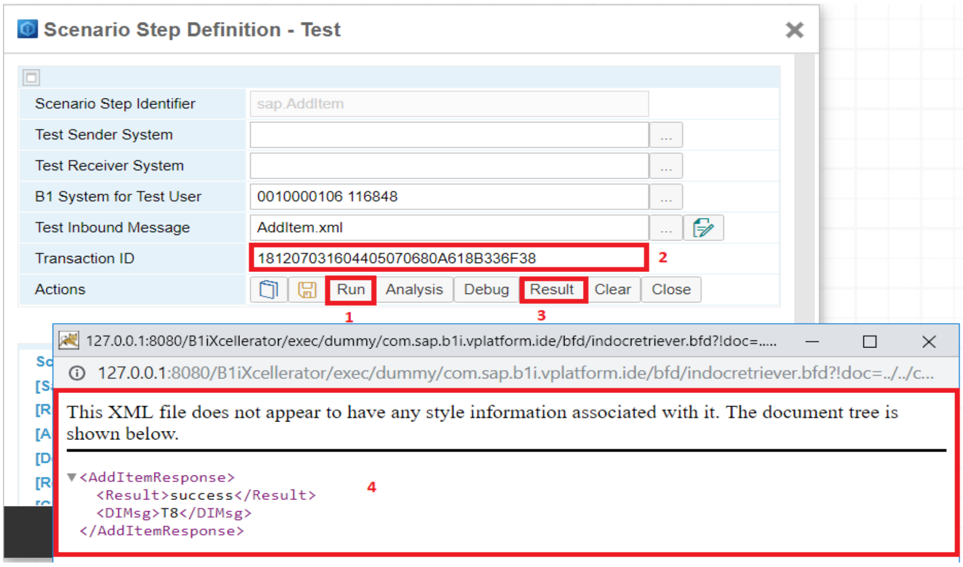
- (4) Press the button.
- In the XSL file, change the **transform** template accordingly:

```
<?xml version="1.0" encoding="UTF-8"?>
<BOM>
    <BO>
        <AdmInfo>
            <Object>4</Object>
            <Version>2</Version>
        </AdmInfo>
        <Items>
            <row>
                <ItemCode>Test5</ItemCode>
                <ItemName>Test5</ItemName>
            </row>
        </Items>
    </BO>
</BOM>
```

- (5) Finally, press the [Save] button.

2.4. Scenario Step Definition Test - Result

- (1) Click the **Run** button.
- (2) The transaction ID is generated
- (3) Click the **Result** button.
- (4) We can see the result as per final atom XSL definition.



2.5. Creating a WSDL Document

You can generate a WSDL document in the integration framework that you can hand over to the Web service consumer. The integration framework creates the WSDL for the scenario package. For generation, there must be at least one inbound message available in the integration framework message log.

Prerequisites

- For each scenario step with Web service inbound, you have run a successful test in scenario step processing.
- You have defined and configured the Web service system or systems in SLD.

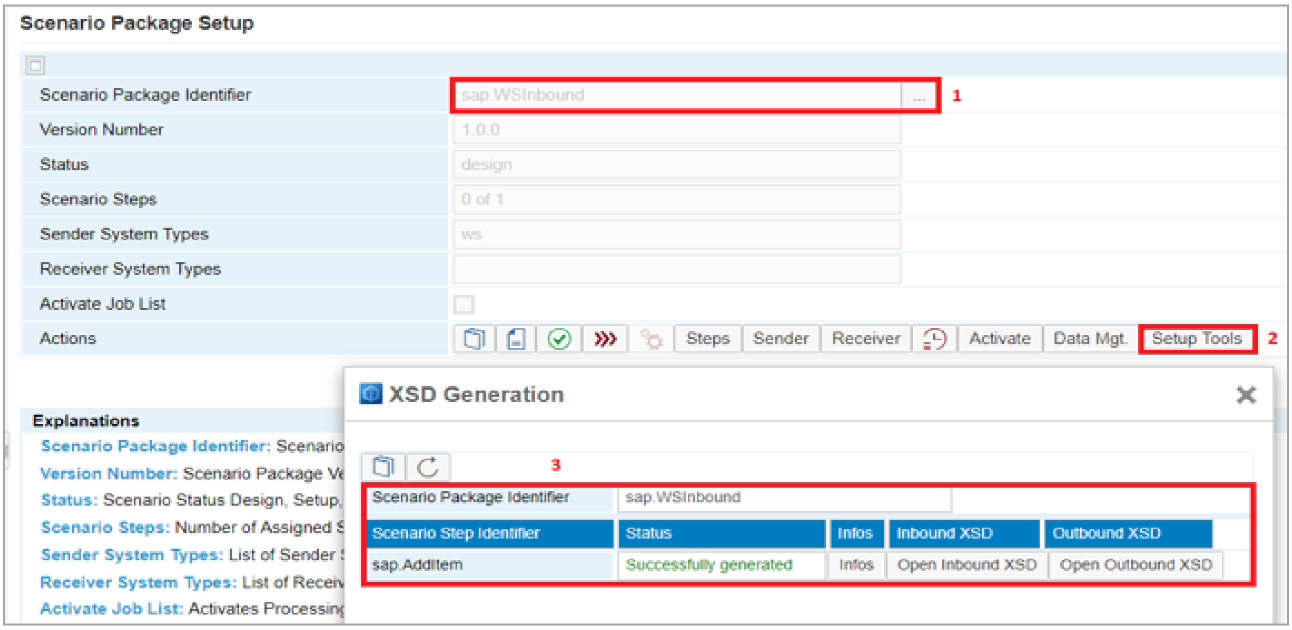
Procedure

1. To create the WSDL document, choose Scenarios → Setup.

2. Select your scenario package

3. Click [Setup Tools] and select *XSD Generation*. The integration framework displays the result of the XSD generation. All inbound and outbound XSD documents must be available. The integration framework displays *Successfully generated* in the *Status* field. Alternatively, you can provide the XSD documents manually in the base directory of the scenario package in the following way:

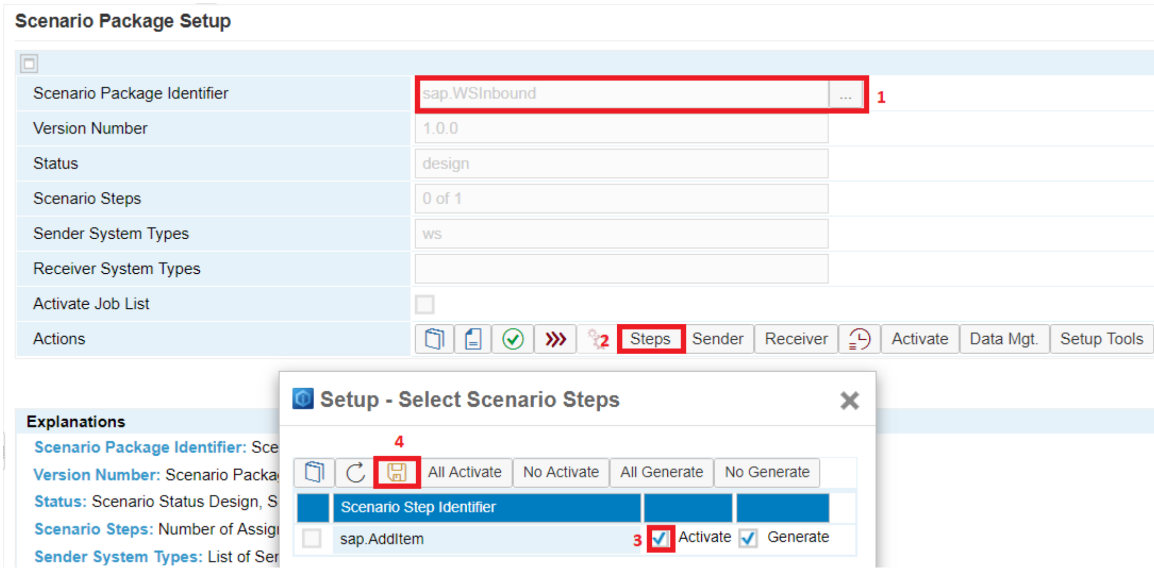
/com.sap.b1i.vplatform.scenarios.design/vBIU./_in.xsd
/com.sap.b1i.vplatform.scenarios.design/vBIU./_out.xsd



2.6. Activate the Scenario Package

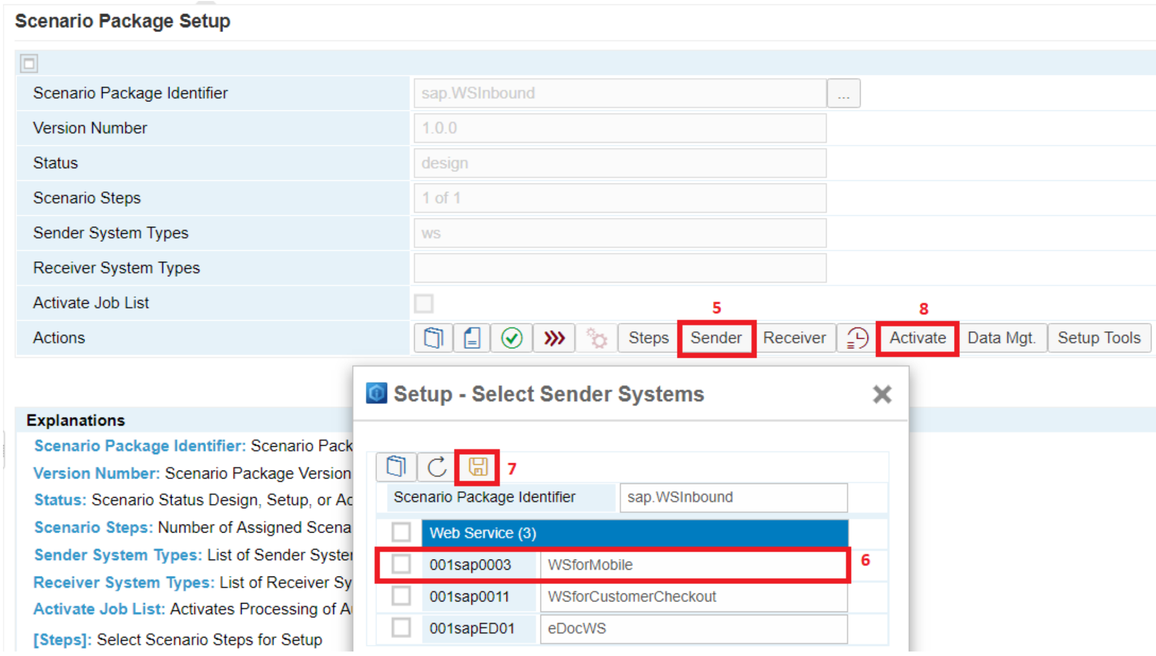
Follow the steps to activate the scenario package.

- (1) Select your scenario package at Scenario Package Identifier.
- (2) Click the **Steps** button.
- (3) Select the *Activate* checkbox *Select Scenario Steps*.
- Press the [Save] button.



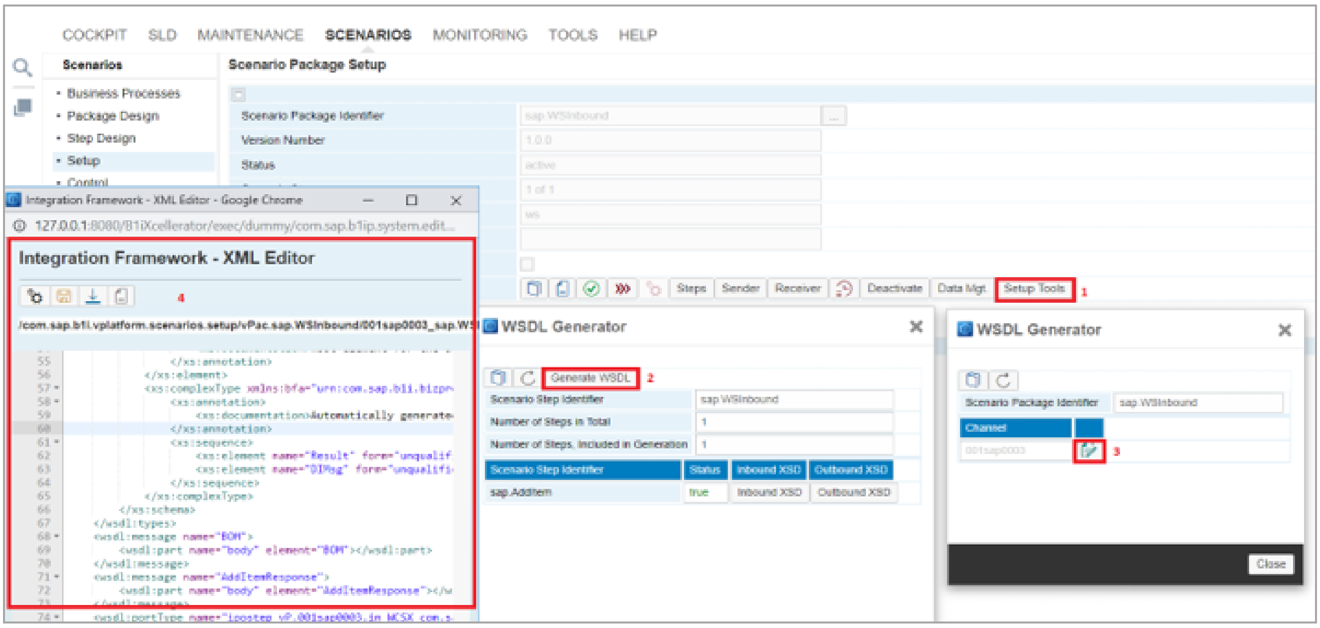
- (5) Click the **Sender** button.
- (6) Select Sender System.

- (7) Press the [Save] button.
- (8) Click the **Activate** button to activate the scenario.



2.7. WSDL Generation

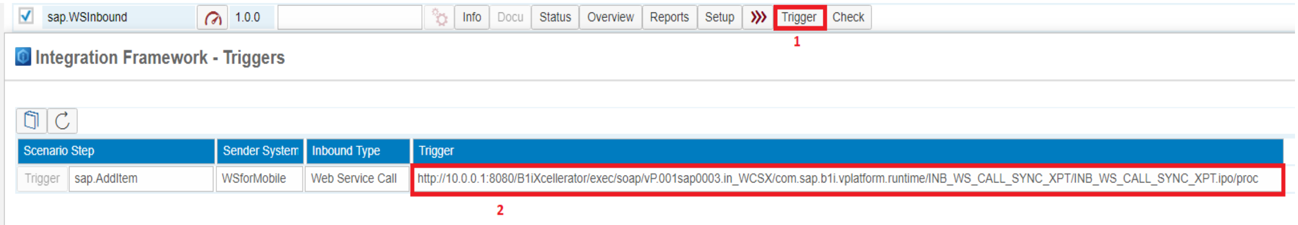
- (1) Click [Setup Tools] and then select WSDL Generation. The integration framework displays the scenario steps, the status of XSD generation and provides buttons to open the inbound and outbound XSD documents.
- (2) To generate the WSDL document, click the Generate WSDL button.
- (3) In the *WSDL Generator* user interface, press the [Open] button.
- (4) The WSDL document opens in a Web browser window.



2.8. Trigger of the Scenario Step

- Once the scenario package is activated, the integration framework provides the trigger of the scenario steps. To display the trigger of the scenario steps, choose *Scenarios → Control*, and for your scenario package, click the [Trigger] button.
- Just copy the URL under **Trigger** for a standard Web service call.

Example:
http://10.0.0.1:8080/B1iXcellerator/exec/soap/vP.001sap0003.in_WCSX/com.sap.b1i.vplatform.runtime/INB_WS_CALL_SYNC_XPT/INB_WS_CALL_SYNC_XPT.ipo/proc



2.9. Simplified Web Service Call

Once you have tested your scenario step, you can use the simplified call. It is faster than the standard call, because it no longer checks

To distinguish between a simplified Web service call and the standard Web service call, the integration framework provides additional URL parameters. To switch to a simplified Web service call, use the ***biu*** parameter. Enter the name of the scenario step you want to call as the parameter value.

EXAMPLE

Example of a standard Web service call:

http://10.0.0.1:8080/B1iXcellerator/exec/soap/vP.001sap0003.in_WCSX/com.sap.b1i.vplatform.runtime/INB_WS_CALL_SYNC_XPT/INB_WS_CALL_SYNC_XPT.ipo/proc

Example of a simplified Web service call

http://10.0.0.1:8080/B1iXcellerator/exec/soap/vP.001sap0003.in_WCSX/com.sap.b1i.vplatform.runtime/INB_WS_CALL_SYNC_XPT/INB_WS_CALL_SYNC_XPT.ipo/proc?biu= GSC.AddItem

msglog=true, false

Use the parameter to switch the message log on or off for the scenario step. If you set the parameter, the integration framework ignores any settings for the message log in the integration framework server.

setvar=true, false

To bypass the generation of local and global variables and properties and properties into the message, if you do not use any local or global variables and properties in your scenario step, set the parameter to false. By default, the integration framework generates the variables and properties into the message.

http://10.0.0.1:8080/B1iXcellerator/exec/soap/vP.001sap0003.in_WCSX/com.sap.b1i.vplatform.runtime/INB_WS_CALL_SYNC_XPT/INB_WS_CALL_SYNC_XPT.ipo/proc?biu= GSC.AddItem**&msglog=false&setvar=false**

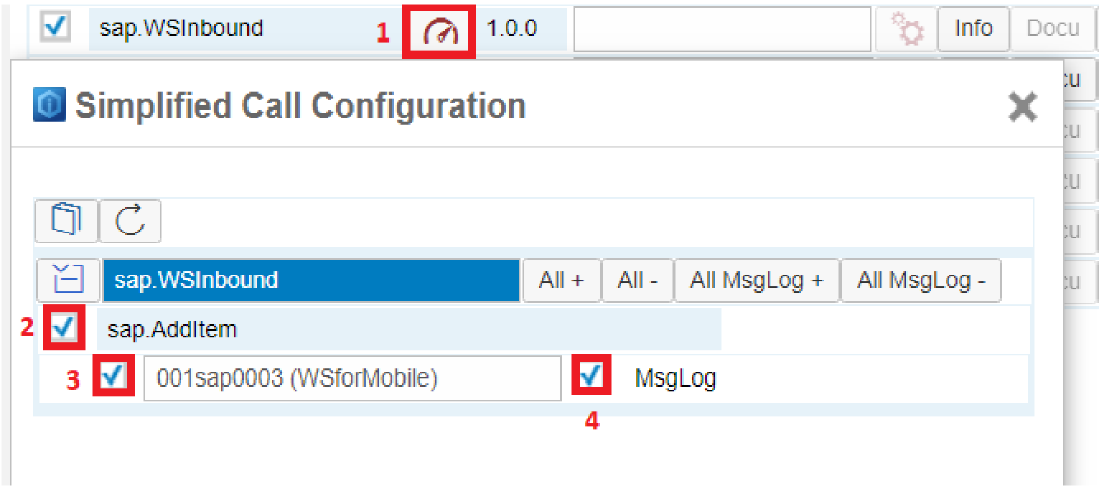
2.9.1.Enabling the Simplified Web Service Call

In case if you do not want to change the URLs of your scenario steps, you can enable simplified calls in the integration framework. Enabling simplified calls from integration framework is available for the Root Tag not for the XPath identification method. To enable simplified calls, choose Scenarios → Control.

If the integration framework displays the icon in the package name field, the package contains at least one Web service or HTTP call step.

Process flow:

- To enable simplified calls, click the icon.
- To enable simplified calls for the step, select the step.
- To enable simplified calls for the step and a specific sender system, select the system.
- To enable the message log for the call, select MsgLog.



If simplified calls are enabled for scenario package, the integration framework displays the icon.

2.10. Testing Web Service SOAP request with Postman

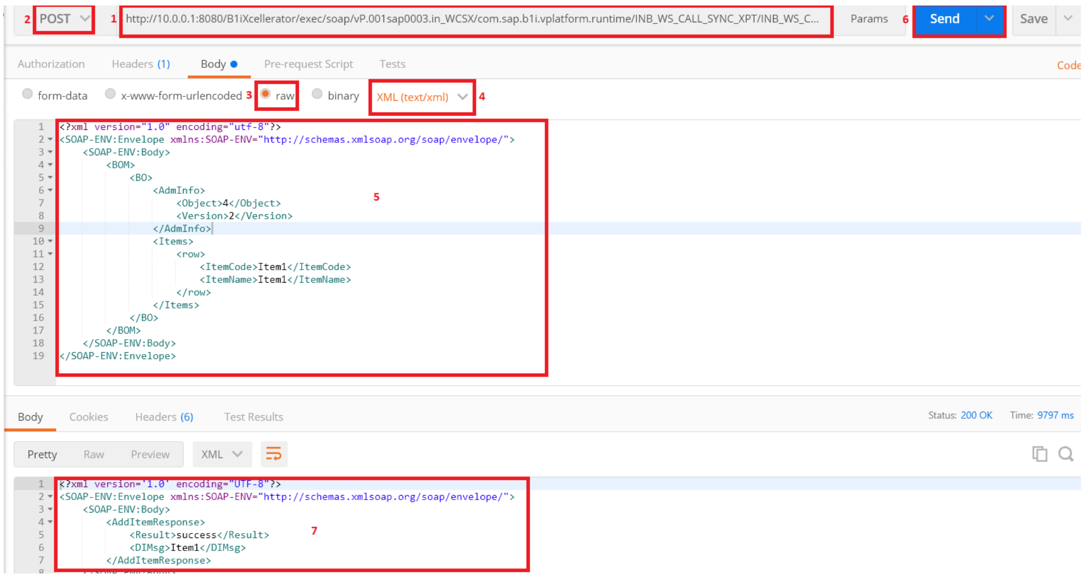
SOAP is a basic XML based protocol used for the communication.

Process flow:

- Enter the SOAP endpoint as the request **URL** in Postman [You can retrieve the URL as mentioned in step 8].
- Set the Request method to **POST**.
- Under the Body tab, set the body type to **raw**.
- Select XML **(text/xml)** from the dropdown. This will automatically add the correct Content-Type header as can be seen under the Headers tab.
- In the request body, define the SOAP envelope, body, and header tags as bellow:

```xml
<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <BOM>
      <BO>
        <AdmInfo>
          <Object>4</Object>
          <Version>2</Version>
        </AdmInfo>
        <Items>
          <row>
            <ItemCode>Item1</ItemCode>
            <ItemName>Item1</ItemName>
          </row>
        </Items>
      </BO>
    </BOM>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- Click on **Send** button to send the request.
- You can see the result in response.



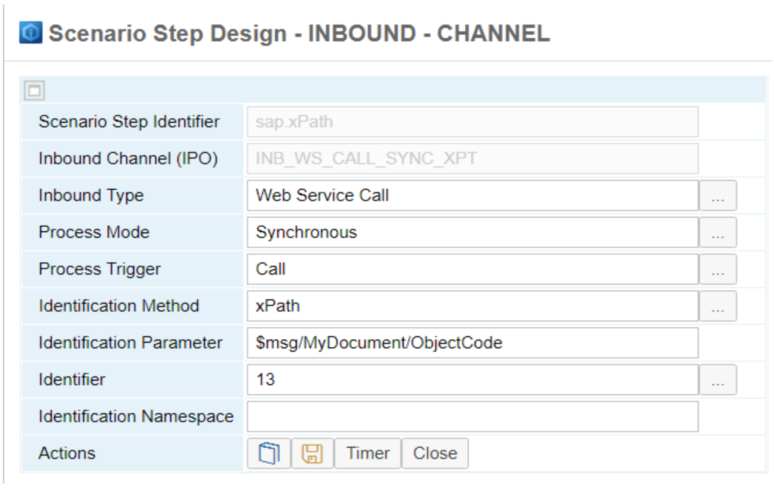
3. INBOUND USING XPATH AS IDENTIFICATION METHOD

This chapter shows you how to use the **xPath** identification method to identify the incoming web service request.

The Web service call supports the XPath expression as the identification method. When you use this identification method, enter the XPath as the identification parameter. The integration framework supports any XPath language

Please refer to the sample **sap.xPath** scenario step available in the sap.in.WebService scenario package.

To define the inbound definition of the scenario step, select *Scenarios → Step Design → [Inbound] → [Channel]*.



The integration framework uses the following parameters to control inbound processing:

- Inbound Type = Web Service Call
- Process Mode = Synchronous
- Process Trigger = Call
- Identification Method = Select how the integration framework identifies the incoming message, either by the root tag or by XPath.
- Identification Parameter = For XPath, enter the XPath expression and use $msg for the incoming message.
- Identifier = For XPath enter the value of the XPath
- Identification Namespace = We can additionally provide a namespace.

3.1. Outbound Channel

For a synchronous inbound call, the outbound channel is always VOID, since the integration framework sends back the response to the caller.

To define the outbound definition of the scenario step, select *Scenarios → Step Design → [Outbound] → Select **Void** at [Outbound channel]*.

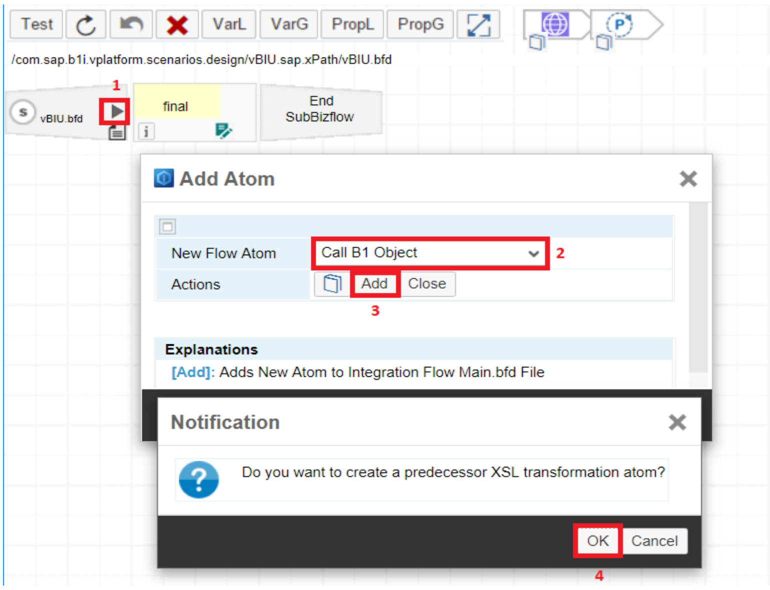
3.2. Insert Data to SAP Business One

The chapter guides you through the procedure to insert Invoice data into SAP Business One using the web service post.

3.2.1. Add an Atom to Scenario Step Processing

In this example we have taken B1 Object atom. To add the B1 Object to the process flow:

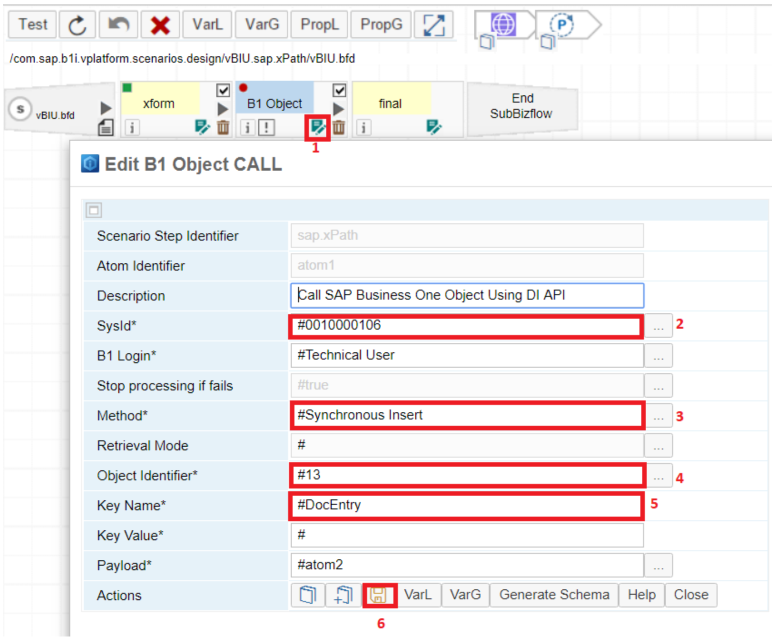
- Press the ▶ [Add] button on the *Start* atom to insert a new functional processing atom.
- In the *New Flow Atom* field, select the **Call B1 Object** value.
- Click **Add**.
- Finally, press **OK** to generate the *predecessor XSL transformation* atom.



3.2.2. Configure the Functional Processing Atom

To configure the processing atom:

- Press the [Change Configuration] button.
- In the *SysId* field, select the SAP Business One system entry.
- In the *Method* field, select the **Synchronous Insert** value.
- In the *Object Identifier* field, select **Invoices**.
- The *Key Name* property is filled automatically based on the *Object Identifier*.
- Finally, press the [Save] button.



3.2.3. Customize the XSL Transformation

- Click the XSL Transformation Atom (xform). The integration framework opens the *Embedded XML Editor* to edit the XSL file.
- In the XSL file, change the **transform** template accordingly:

```
<xsl:template name="transform">
        <xsl:copy-of select="$msg/MyDocument/Documents"/>
        <xsl:copy-of select="$msg/MyDocument/Document_Lines"/>
</xsl:template>
```

3.2.4. Customize the final atom XSL Transformation

- Click the XSL Transformation Atom (xform). The integration framework opens the *Embedded XML Editor* to edit the XSL file.
- In the XSL file, change the **transform** template accordingly:

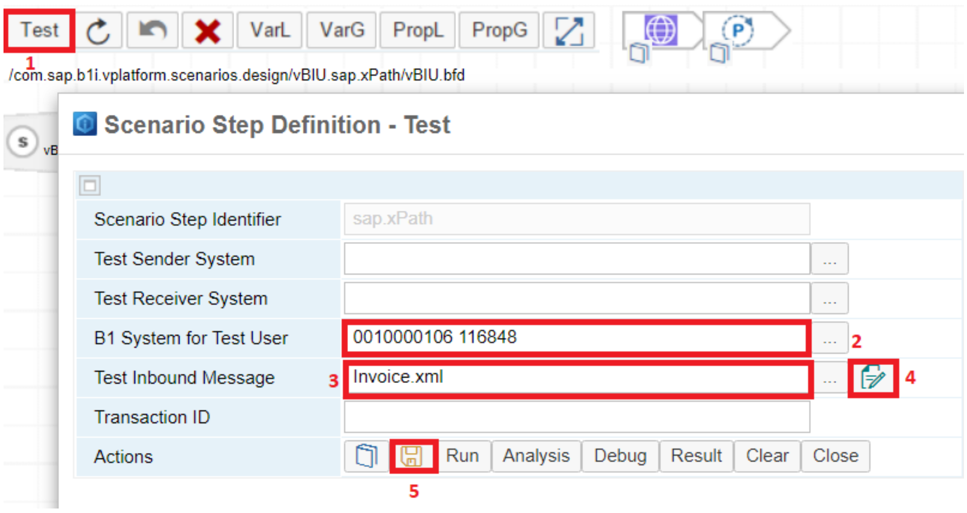
```
<xsl:template name="transform">
  <xsl:attribute name="pltype">xml</xsl:attribute>
  <xsl:variable name="b1CallResult"
select="/vpf:Msg/vpf:Body/vpf:Payload[./@id=&apos;atom1&apos;]"/>
  <Response>
        <System>
              <xsl:value-of select="$b1CallResult/@system"/>
        </System>
        <Status>
              <xsl:value-of select="$b1CallResult/@status"/>
        </Status>
        <B1Object>
              <xsl:value-of select="$b1CallResult/@objectid"/>
        </B1Object>
        <B1Method>
```

```
                <xsl:value-of select="$b1CallResult/@method"/>
            </B1Method>
            <B1ObjectKeyName>
                <xsl:value-of select="$b1CallResult/@keyname"/>
            </B1ObjectKeyName>
            <B1ObjectKeyValue>
                <xsl:value-of select="$b1CallResult/@keyvalue"/>
            </B1ObjectKeyValue>
            <DIAPIStatus>
                <xsl:value-of select="$b1CallResult/@DIresult"/>
            </DIAPIStatus>
            <DIAPIMessage>
                <xsl:value-of select="$b1CallResult/@DImsg"/>
            </DIAPIMessage>
        </Response>
    </xsl:template>
```

3.3. Scenario Step Definition – Test Configuration

Follow the procedure for Scenario step test:

- Click on Test button to open Scenario Step Definition -Test.
- Select B1 System for Test User.
- Define Test Inbound Message (Invoice.xml).



- Press the button.
- In the XSL file, change the **transform** template accordingly:

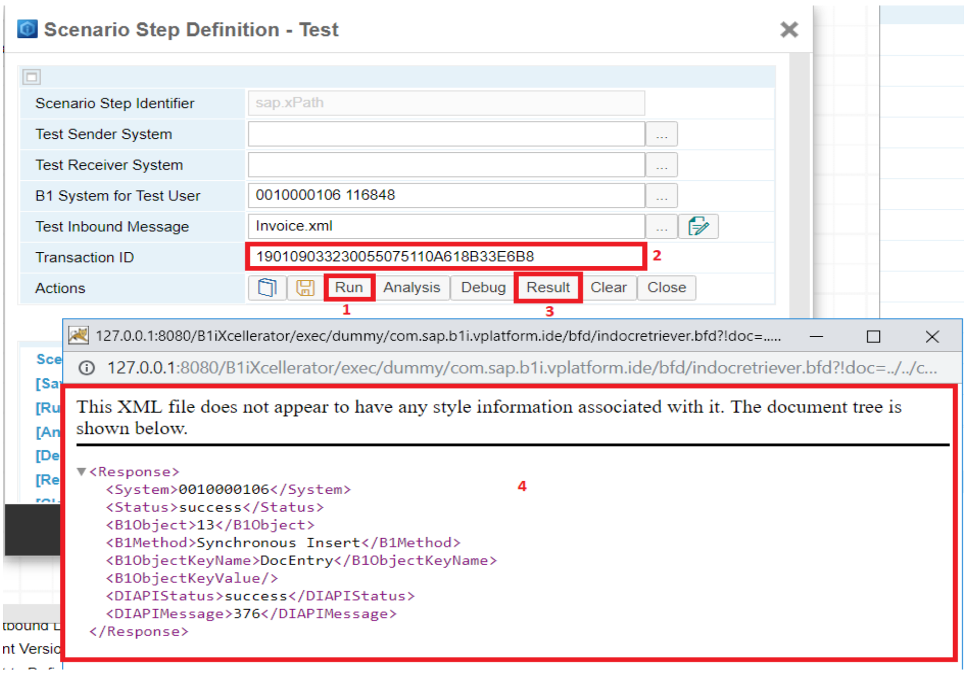
```
<?xml version="1.0" encoding="UTF-8"?>
<MyDocument>
    <ObjectCode>13</ObjectCode>
    <Documents>
        <Row>
            <CardCode>C20000</CardCode>
            <DocDueDate>20190104</DocDueDate>
        </Row>
```

```
        </Documents>
        <Document_Lines>
                <Row>
                        <ItemCode>A00001</ItemCode>
                        <Quantity>1</Quantity>
                        <UnitPrice>100</UnitPrice>
                </Row>
                <Row>
                        <ItemCode>A00002</ItemCode>
                        <Quantity>2</Quantity>
                        <UnitPrice>200</UnitPrice>
                </Row>
        </Document_Lines>
</MyDocument>
```

- Finally, press the [Save] button.

3.4. Scenario Step Definition Test - Result

- Click on **Run** button.
- Transaction ID will be generated
- Click on **Result** button.
- We can see the result as per final atom XSL definition.



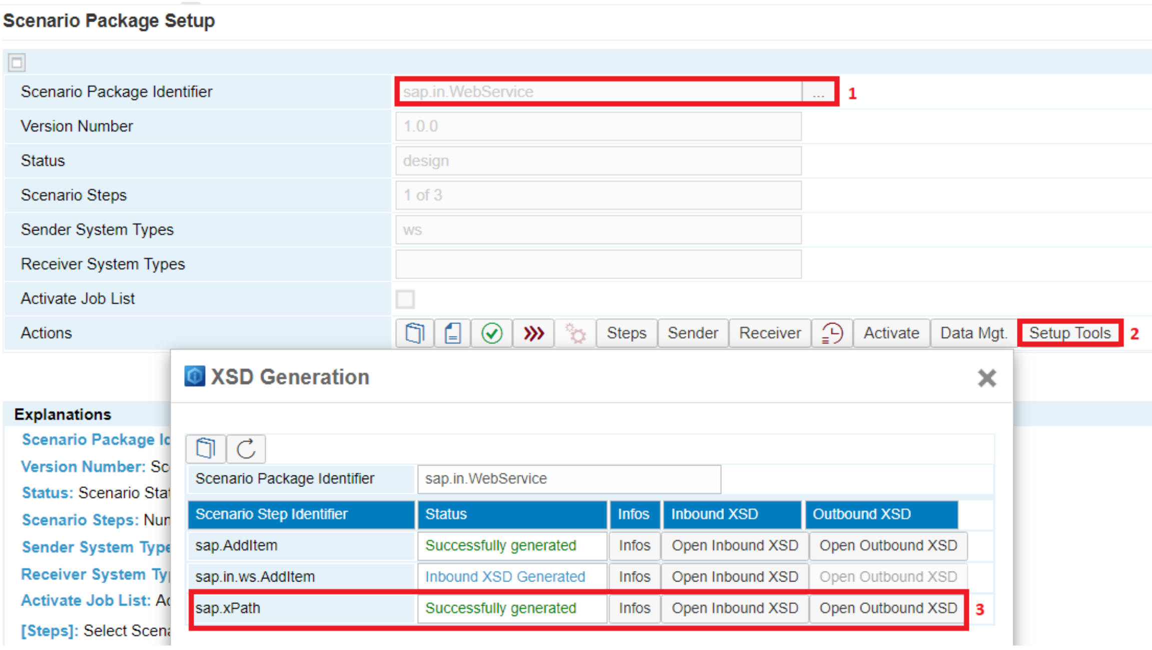
3.5. Creating a WSDL Document

Procedure

1. To create the WSDL document, choose Scenarios → Setup.

2. Select your scenario package

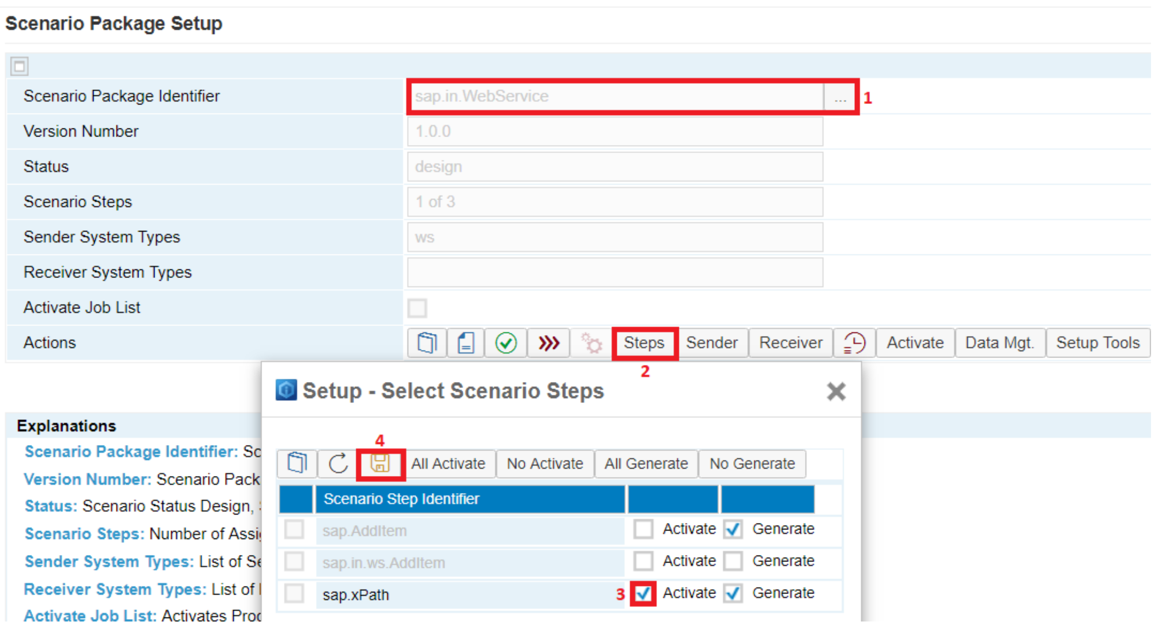
3. Click [Setup Tools] and select XSD Generation. The integration framework displays the result of the XSD generation. All inbound and outbound XSD documents must be available. The integration framework displays Successfully generated in the Status field.



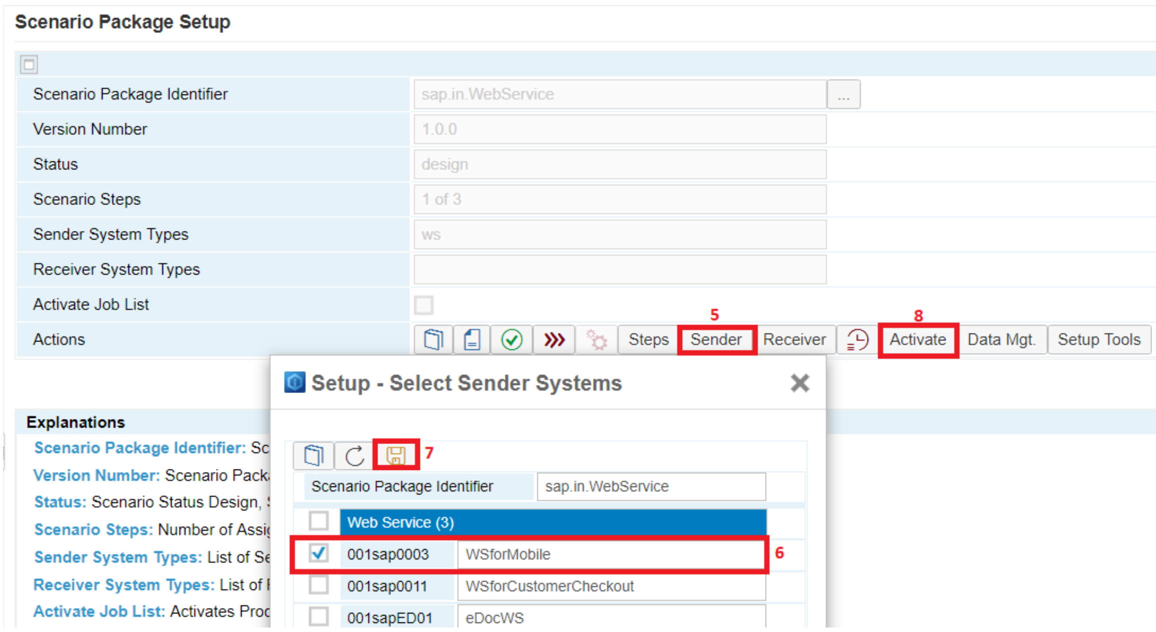
3.6. Activate the Scenario Package

Follow the steps to activate the scenario package.

- Select your scenario package at Scenario Package Identifier.
- Click on **Steps** button.
- Select Activate check box from select scenario steps.
- Press the [Save] button.

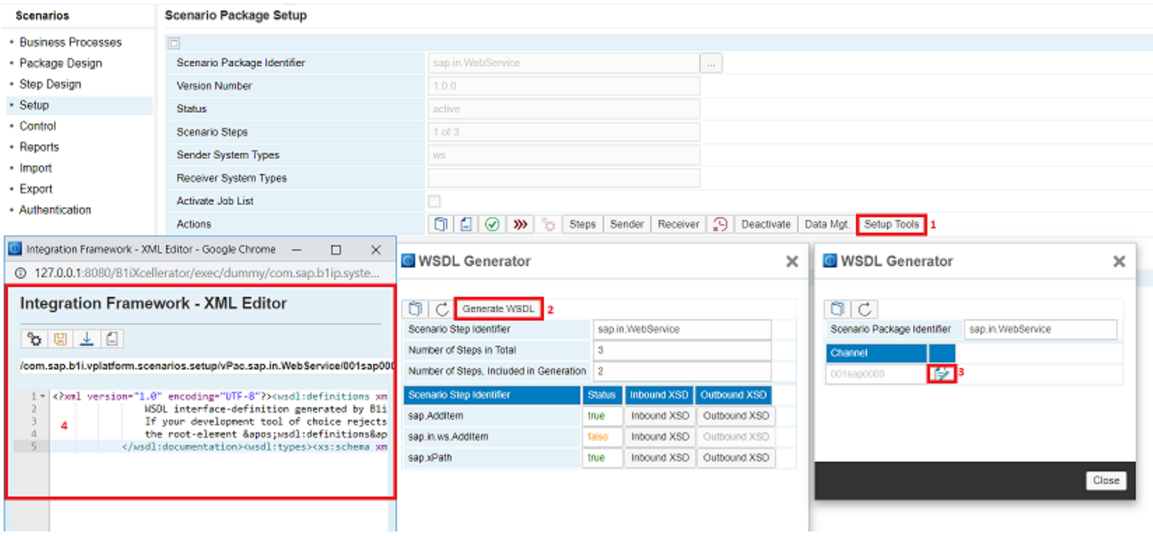


- Click on **Sender** button.
- Select Sender System.
- Press the [Save] button.
- Click on **Activate** button to activate the scenario.



3.7. WSDL Generation

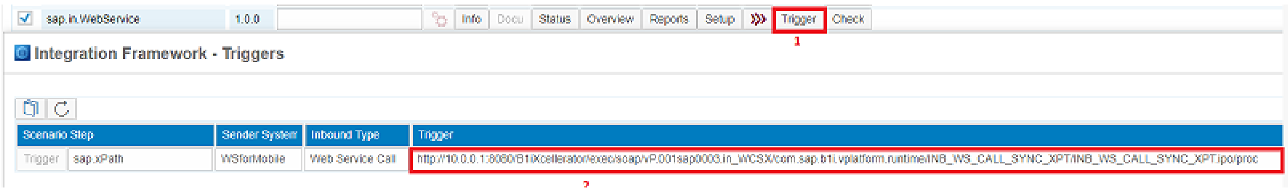
- Click [Setup Tools] and then select WSDL Generation. The integration framework displays the scenario steps, the status of XSD generation and provides buttons to open the inbound and outbound XSD documents.
- To generate the WSDL document, click the Generate WSDL button.
- In the WSDL Generator user interface, press the [Open] button.
- The WSDL document opens in a Web browser window.



3.8. Trigger of the Scenario Step

- To display the trigger of the scenario steps, choose Scenarios → Control, and for your scenario package, click the [Trigger] button.
- Just copy the URL under **Trigger** for a standard web service call.

Example:
http://10.0.0.1:8080/B1iXcellerator/exec/soap/vP.001sap0003.in_WCSX/com.sap.b1i.vplatform.runtime/INB_WS_CALL_SYNC_XPT/INB_WS_CALL_SYNC_XPT.ipo/proc



3.9. Enabling the Simplified Web Service Call

Enabling simplified calls from integration framework is available for the Root Tag not for the XPath identification method.

3.10. Testing Web Service SOAP request with Postman

Process flow:

- Enter the SOAP endpoint as the request **URL** in Postman.
- Set the Request method to **POST**.
- Under the Body tab, set the body type to **raw**.
- Select XML **(text/xml)** from the dropdown. This will automatically add the correct Content-Type header as can be seen under the Headers tab.
- In the request body, define the SOAP envelope, body, and header tags as bellow:

```xml
<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
	<SOAP-ENV:Body>
		<MyDocument>
			<ObjectCode>13</ObjectCode>
			<Documents>
				<Row>
					<CardCode>C20000</CardCode>
					<DocDueDate>20190105</DocDueDate>
				</Row>
			</Documents>
			<Document_Lines>
				<Row>
					<ItemCode>A00001</ItemCode>
					<Quantity>1</Quantity>
					<UnitPrice>100</UnitPrice>
				</Row>
				<Row>
					<ItemCode>A00002</ItemCode>
					<Quantity>2</Quantity>
					<UnitPrice>200</UnitPrice>
				</Row>
			</Document_Lines>
		</MyDocument>
	</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- Click on **Send** button to send the request.
- You can see the result in response.

www.sap.com/contactsap

THE BEST RUN SAP®