WAY2LEARN

LAB- Consuming Soap Webservice

In this lab, you will be working on **03-consuming-soap-start** project under **02-consuming-webservices** section

In this lab you will understand

a) how to use a webservice consumer to consume a soap webservice

b) how to use parse template to load and evaluate a template.

STEP 1 --- Running and Understanding the existing SOAP Webservice

Before you proceed, ensure that you have installed SOAP UI.

1) Run soap.bat provided to you. This will start a soap webservice. To view its wsdl, give request to http://localhost:6060/soap/ws/products.wsdl

According to Wsdl, there are 3 operation.

a) getAllProducts

b) addProduct

c)getProductsByName

2) Open Soap UI and create a project using the wsdl location http://localhost:6060/soap/ws/products.wsdl

3) Make request to getAllProducts operation through SOAP UI and observe the result.

Make a request to getProductsByName operation by passing name as Mac or Hp or IBM and observe the result.

Make a request to addProduct by giving some sample values .

remove the productId tag as we want the product id to be generated by the webservice.

You get added product details in the response. You should see the generated productId.

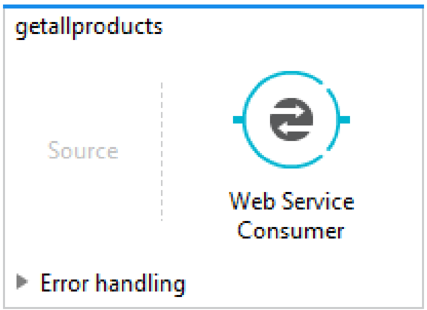
STEP 2 --- Consuming SOAP Webservice using webservice Consumer

1) Create a new Mule Configuration file with name soapdemo.xml.

Drag and drop an empty flow .

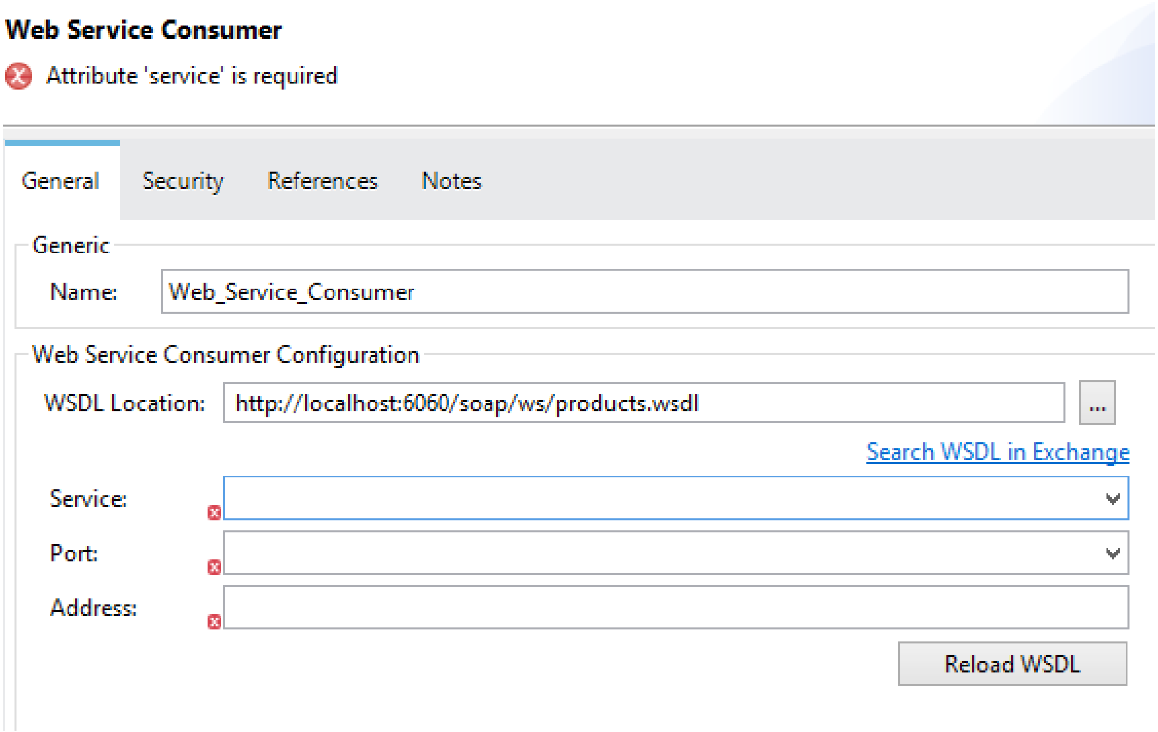
Name it as getallprodcus

Drag Webservice Consumer inside it as shown below :



Now double click on Webservice Consumer and click on + symbol to create connector configuration.

In the window, Just give the wsdl location as http://localhost:6060/soap/ws/products.wsdl and press tab. it should look like below :



Web Service Consumer

Attribute 'service' is required

Generic

Name: Web_Service_Consumer

Web Service Consumer Configuration

WSDL Location: http://localhost:6060/soap/ws/products.wsdl

Service:

Port:

Address:

Now select the service, port from the drop downs and observe that address it automatically populated after selecting port.

Click Ok

Now select the operation as getAllProducts.

That's all . You have configured Webservice Consumer to invoke getAllProducts.

Now Configure another flow with Http Endpoint and make sure that the base path is /soap.

Configure the path of http connector as products.

Now drag a flow reference to reference to getallproducts flow.

Deploy the application and give request to http://localhost:8081/soap/products. You should observe all the products.

STEP 3 --- Consuming   SOAP Webservice operation which requires some input

1) In this step, we want to consume getProductsByName operation.

Try to consume it from soap UI. You will observe that the soap body should look like below :

```
<dom:getProductsByNameRequest xmlns:dom="http//way2learnonline.com/domain">
      <name>Mac</name>
 </dom:getProductsByNameRequest>
```

2) If you want to consume the same operation using webservice consumer, you need to set the payload as above xml before webservice consumer component.

Open getproductsbyname.xml in src/main/resources and observe that we have create a template similar to the required xml.

Observe that instead of hard coding, we used an expression to retirve name from query parameter productname.

ParseTemplate is a component which can load a template file and evaluate expressions inside it.

So, Drag flow with parse template in it. Name the flow as getproductsbyname.

Configure location of parsetemplate as getproductsbyname.xml

Drag a logger and configure it to print payload

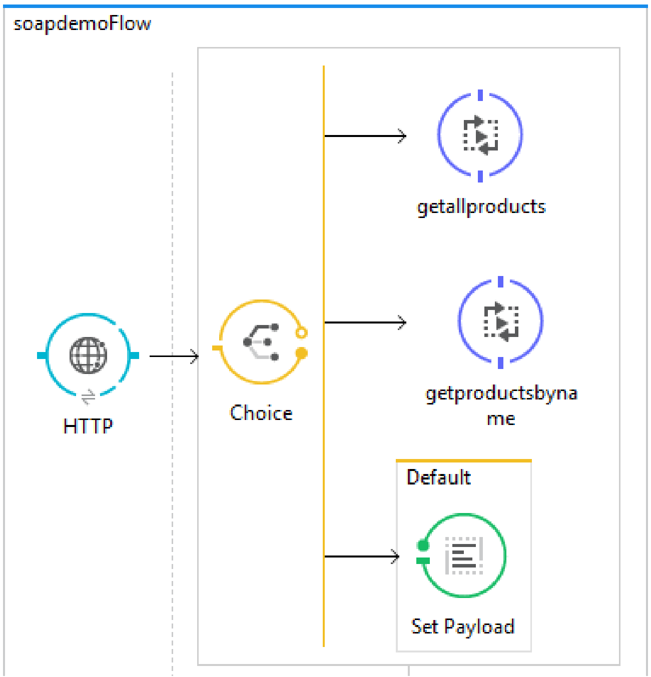
Drag a webservice consumer, configure it with same connector configuration which you created in earlier step. Select the Operation as getProductsByName

You are done with a flow which can consume getProductsByName.

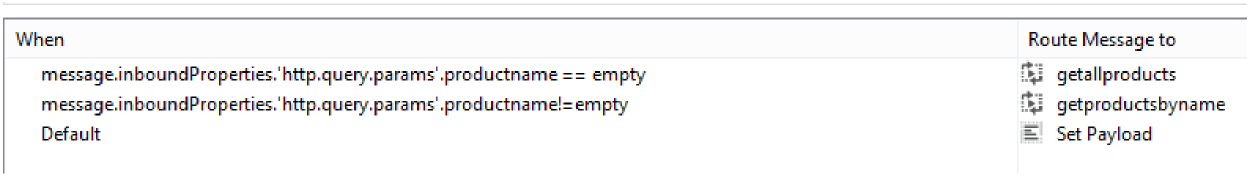
Now we want to link this flow with main flow and conditionally route request to getproductsbyname flow if there is a query parameter with name "productname"

if there is no queryparameter with name productname, we want to route to getallproducts flow.

Modify the main flow which was having Http Endpoint as shown below.



Choice router configuration should look like below:



When	Route Message to
message.inboundProperties.'http.query.params'.productname == empty	getallproducts
message.inboundProperties.'http.query.params'.productname!=empty	getproductsbyname
Default	Set Payload

Deploy the flow and give a request to http://localhost:8081/soap/products .

You should see the result of getallproducts operation

Give request to http://localhost:8081/soap/products?productname=Mac

You should see the result of getproductsbyname operation

(Bonus) STEP 4 --- Consuming addProduct Operation

Can you write a separate flow to consume addProduct Operation?

After writing separate flow, hook this flow in to choice router for condition which checks message.inboundProperties.'http.method'==POST

Try making a POST request to http://localhost:8081/soap/products **using POSTMAN**

The Template xml required for addProduct is given to you in src/main/resources/addproduct.xml

This is the end of the Exercise