

LAB- Consuming Soap Webservice

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.

NOTE: We need JDK 11 to run this rest service with out any problem. If u have any other version also, it is a problem. Even mule server and studio needs the same version. So, make sure that u have JDK 11 installed.

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

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

1) Create a project with name consumesoap.

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

According to Wsdl, there are 3 operation.

- a) getAllProducts
- b) addProduct
- c) getProductsByName

Save this products.wsdl into src/main/resources of the current project. Make sure that the extension of the file is .wsdl and not .wsdl.xml

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

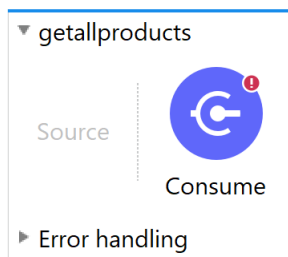
You get added product details in the response. You should see the generated productId.

STEP 2 --- Consuming SOAP Webservice using webservice Consumer

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

Add Webservice Consumer module in the mule palette on the right side

Now drag Consume component . A new Flow will be created . Rename it as getAllproducts



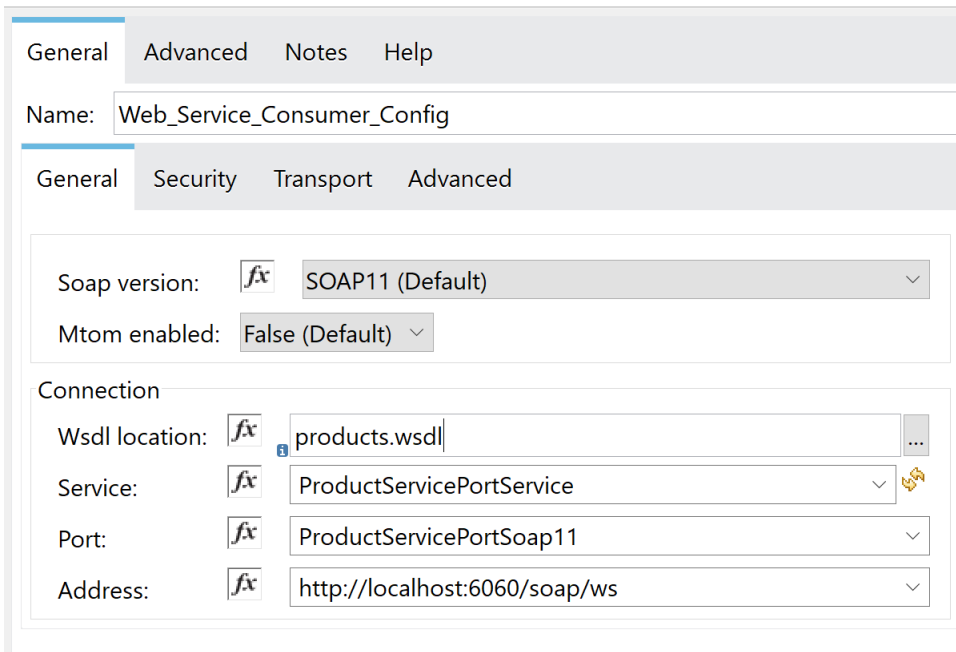
Observe that we have already copied the wsdl in src/main/resources/products.wsdl

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

In the window, Just give the wsdl location as products.wsdl and press tab. it should look like below :

Web Service Consumer Config


Default configuration



General Advanced Notes Help


Name: Web_Service_Consumer_Config


General Security Transport Advanced


Soap version:  SOAP11 (Default) ▼


Mtom enabled: False (Default) ▼

Connection

Wsdll location:  products.wsdl ...

Service:  ProductServicePortService ▼

Port:  ProductServicePortSoap11 ▼

Address:  http://localhost:6060/soap/ws ▼

Click Ok

Now select the operation as getAllProducts.

When you were consuming getAllProducts using Soap UI, soap body was the below xml:

```
<dom:getAllProductsRequest xmlns:dom="http://way2LearnOnline.com/domain"/>
```

I have created a file getallproducts.xml with above content in src/main/resources .

We need to set the contents of this file as payload.

So, drag ParseTemplate before Consume component and set the location as the path of getallproducts.xml

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

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

Configure the path of http connector as /products.

Rename this flow as “mainflow”

Now drag a flow reference to reference to getallproducts flow.

Deploy the application and give request to <http://localhost:8081/soap/products> in POSTMAN

Did you get a response properly?

Now, drag “Transform Message” component and write dwl to convert the payload to json which displays all flights

Output Payload ▾ ⚙️ ✎ 🗑️

```
1 %dw 2.0
2 output application/json
3 ---
4 payload.body.getAllProductsResponse
```

Deploy the application and give request to <http://localhost:8081/soap/products>.

Observe the output.

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)

Drag a webservice consumer and drop it outside of any flow on the top. It will create a new Flow. Rename the flow to "GetProductsByName" .configure consumer component with same connector configuration which you created in earlier step. Select the Operation as getProductsByName

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

Drag a TransformMessage component before Consumer component and observe the properties of "TransformMessage"

Did u Observe the input Metadata? No.

Now Just Drag Http Listener into Source part of this flow.

Now Observe that input metadata is populated for TransformMessage

Also observe the output metadata. It should look like below :



Now drag queryparams and map it to name on the right side.

The corresponding dwl is generated.

Now delete the Http Listener from this flow. We don't need this here. It is used here just for viewing metadata and drag and drop.

Now modify the dwl as below :

```

1 %dw 2.0
2 output application/xml
3 ns ns0 http://way2learnonline.com/domain
4 ---
5 {
6   ns0#getProductsByNameRequest: {
7     name: attributes.queryParams.productName as String
8   }
9 }
10

```

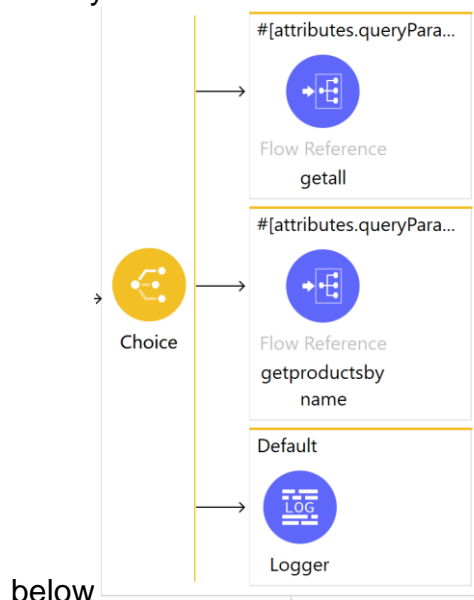
Now, drag “Transform Message” component after Consumer component and write dwl to convert the payload to json which displays all products

3) 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.

if there is queryparameter with name productname, we want to route to getproductsbyname flow.

Modify the main flow to use choice router as shown



When condition for getAllproducts should look like below :

attributes.queryParams.productName ==null

When condition for getproductsbyname should look like below :

attributes.queryParams.productName !=null

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

Note: Is it a good practice to use attributes.queryParams.productName in GetProductsByName flow?

If so, this flow is expecting that it will be called through another flow which has http listener. So, this GetProductsByName flow will not be 100% reusable. It cannot be called from a flow without http listener.

So, Can you modify GetProductsByName flow to expect productname from variables?

Hint : Use vars.productName instead of attributes.queryParams.productName in transform message component.

Also modify the second route of choice router to set a variable with name productName from query parameter product Name.

Now redeploy the application and give request to <http://localhost:8081/soap/products?productName=Mac>

You should get same results as before.

(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 `attributes.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`

(Bonus) STEP 5 --- Consuming public soap apis

Click on the below url to see a public wsdl

<http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?wsdl>

Write a private flow with name “getcapital” to consume an operation which will give Capital City for a Country and

Also write another private flow with name “getcurrency” to consume an operation which will give currency of country

Now write a flow which has HTTP Listener listening on port 9091 and path /countries.

This flow expects 2 query parameters “operation” and “country”

If I pass a query parameter "operation" with value as capital , it should invoke "getcapital" flow . Else, it should invoke "getCurrency" flow. If query parameter country is not passed, I should be response as "invalid request" and reponse status code must be 400

This is the end of the Exercise