knowledge14®

April 27–May 1, 2014 • Moscone Center • San Francisco, CA

Lab Guide

Integration 301:
Flexible Web Services
for Advanced Needs

John Andersen & Jason Petty

Lab instance: https://**<<name>>**.lab.service-now.com

Login / Passwords:

admin / Knowledge14

itil / Knowledge14

employee / Knowledge14



®2014 ServiceNow, Inc. All rights reserved.

What is a WSDL

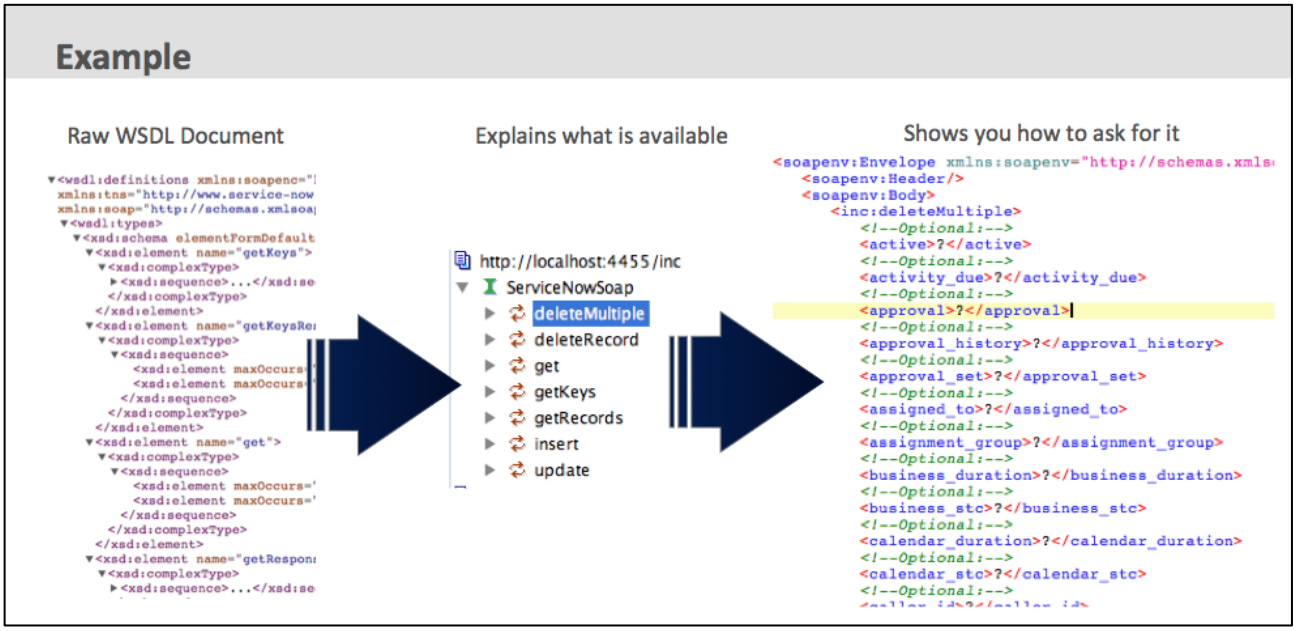
- Web Services Description Language
- Just and XML Document
- Catalog of available web service functions
- Tells you where the SOAP endpoint is located



A WSDL is another XML document that is used in the soap protocol. A WSDL is a catalog of available functions provided by the web service.

A SOAP web service consuming client should be able to read the WSDL and generate a template of each of the available service requests provided by the web service.

©2014 ServiceNow, Inc. All rights reserved.

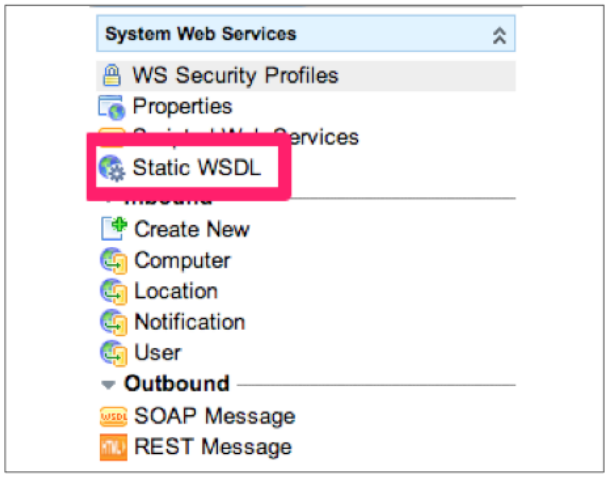


The WSDL XML document will be translated by the soap client into soap web service functions that are available to the client. The WSDL also explains what should be expected in the envelope for each of the different functions.

©2014 ServiceNow, Inc. All rights reserved.

Custom or Static WSDLs

- Plugin: "Web Service Provider – Custom WSDL"
- Navigate to it through the App Navigation Pane
  - System Web Services > Static WSDL



ServiceNow has a plug-in that can be enabled for free that allows you to create your own custom WSDL for web service needs to go beyond the basic out of box create, read, update, delete functions that exist on the various tables in the system.

Once the plug-in is enabled, you can create these WSDL's by using the "static WSDL" module within the "system Web services" application inside ServiceNow.

©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

We are going to take a WSDL document provided by another service and try to replicate that service within ServiceNow. The current WSDL describes a Currency Converter service. Our goal is to load that WSDL into the instance and prep it for our own uses.

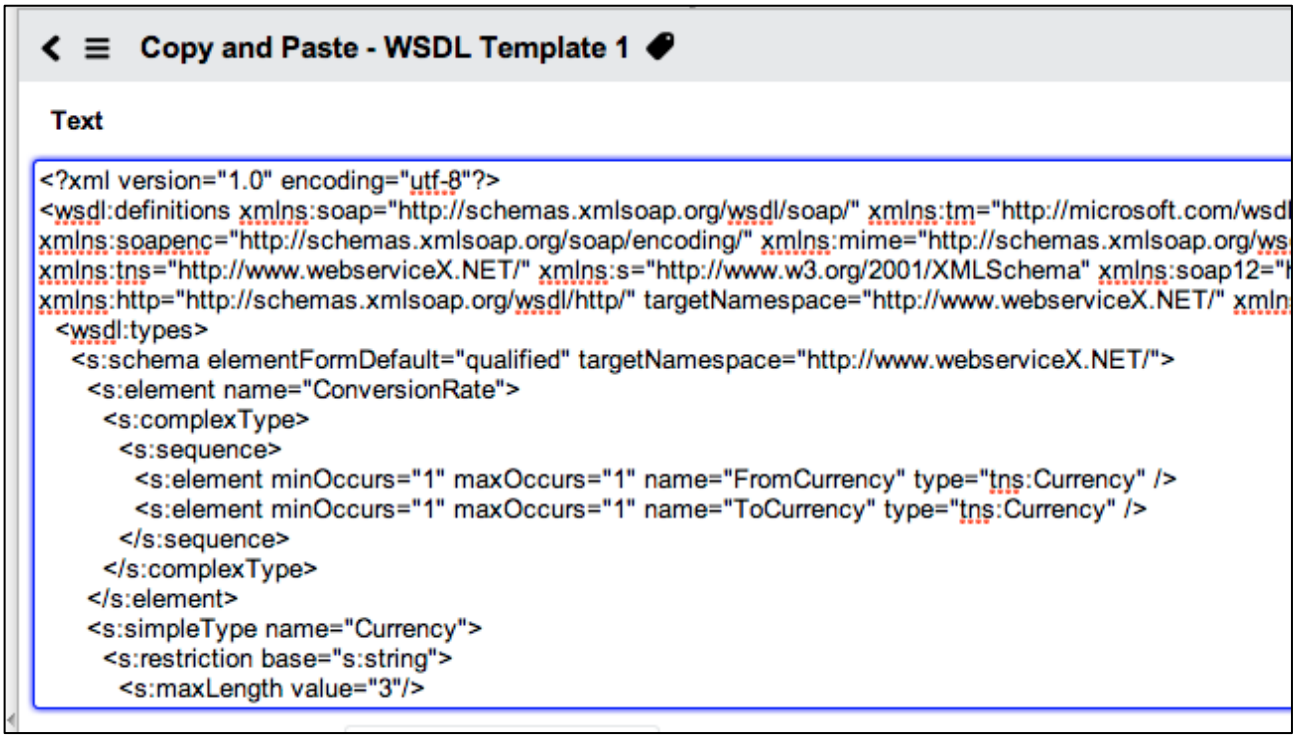
Lab 1
Setup a Custom WSDL

Log on to Your Training Instance

1. Login to the instance URL on the cover page using the provided **admin** credentials.

Create a Custom/Static WSDL

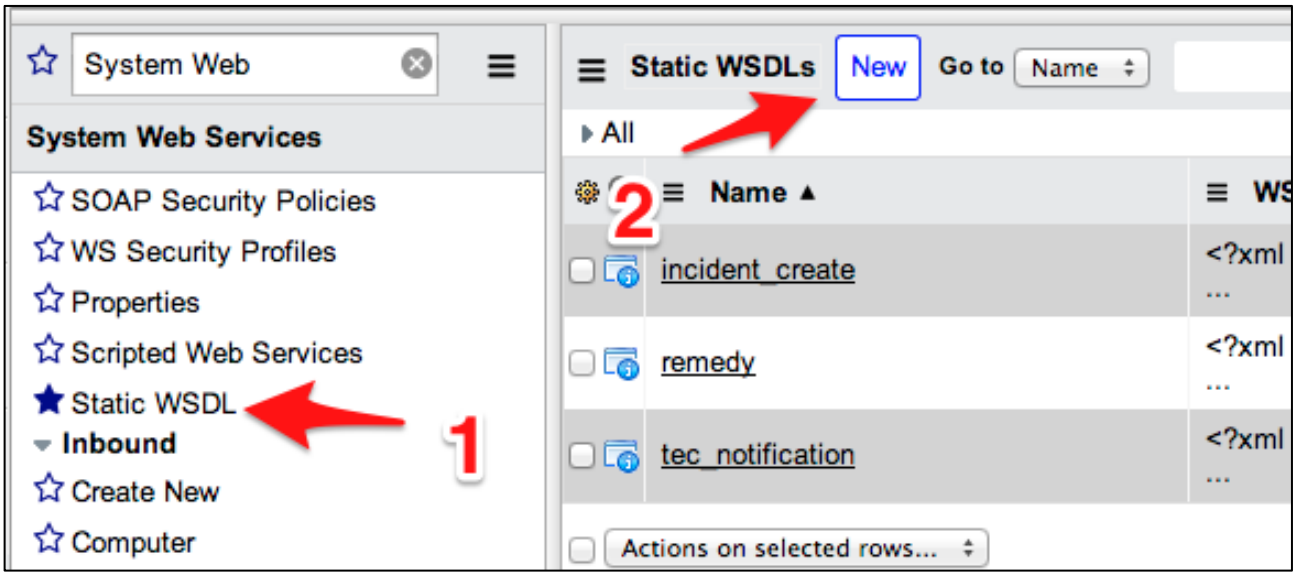
1. Navigate to **Integration Lab 301 > Copy & Paste > WSDL Template #**1.
2. Copy the WSDL from the **Text** field.



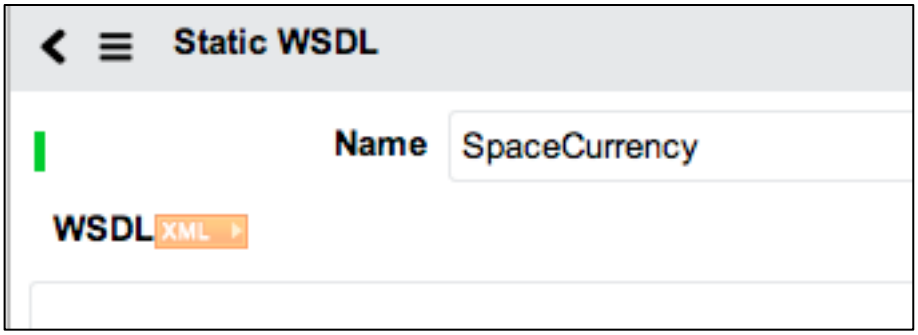
©2014 ServiceNow, Inc. All rights reserved.

3. Navigate to **System Web Services > Static WSDL**.

4. Click **New**.

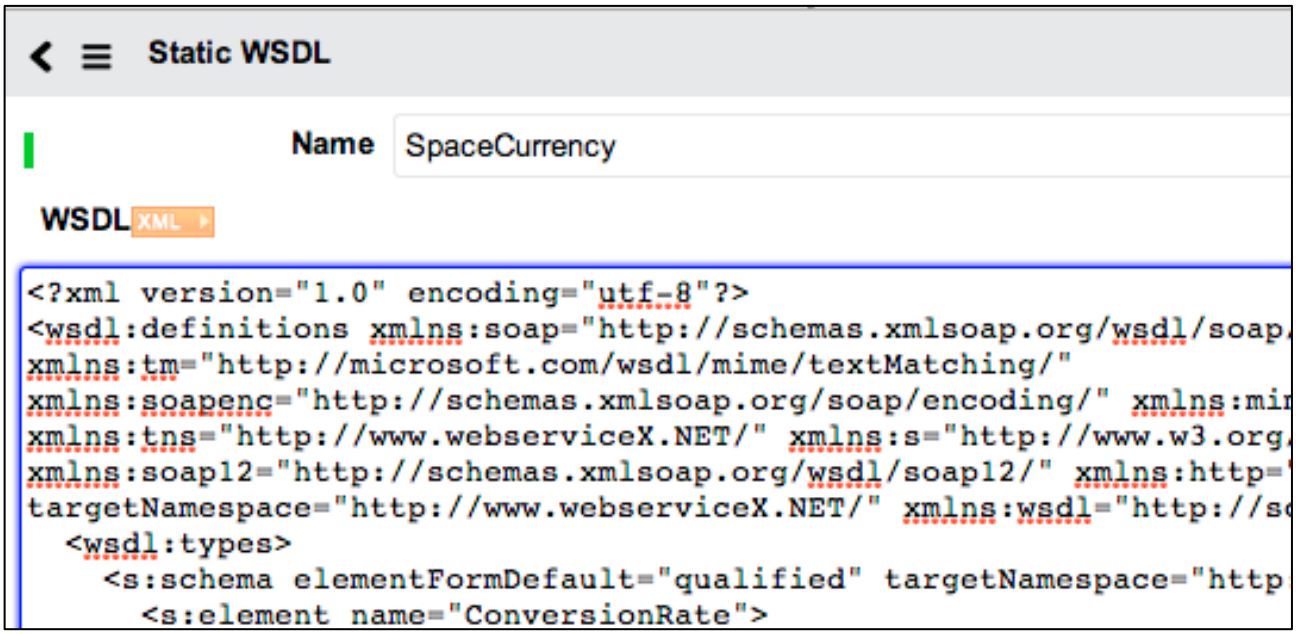


5. Set the **Name** field to be "**SpaceCurrency**".

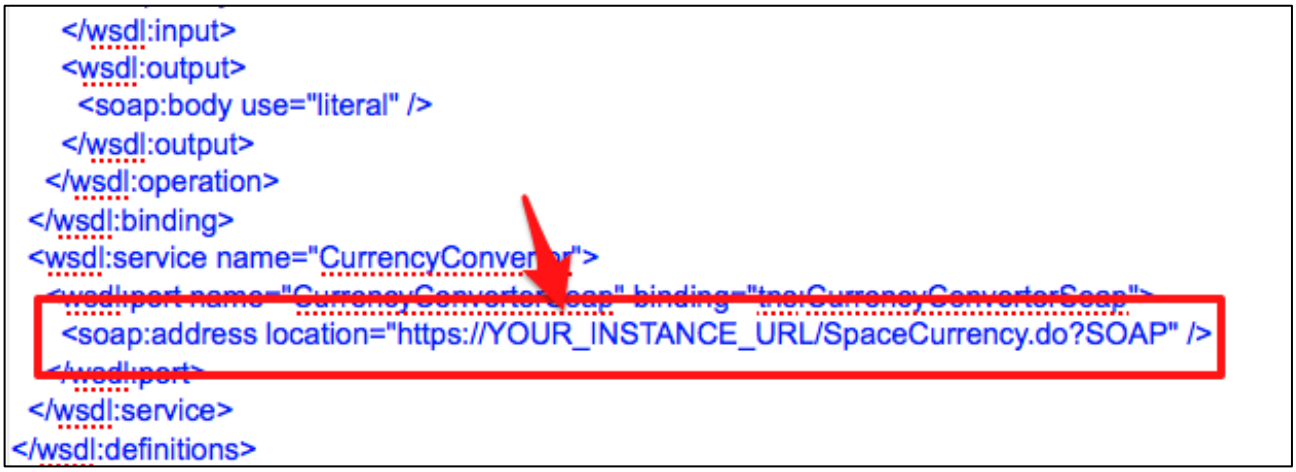


©2014 ServiceNow, Inc. All rights reserved.

6. Paste the WSDL text into the **WSDL** field.



7. At the bottom of the WSDL, change the **YOUR_INSTANCE_URL** string to be the full domain name for your instance. For example, if my instance were “k14integration.service-now.com”, that is the string I would use to replace the placeholder string



8. Click **Submit**.

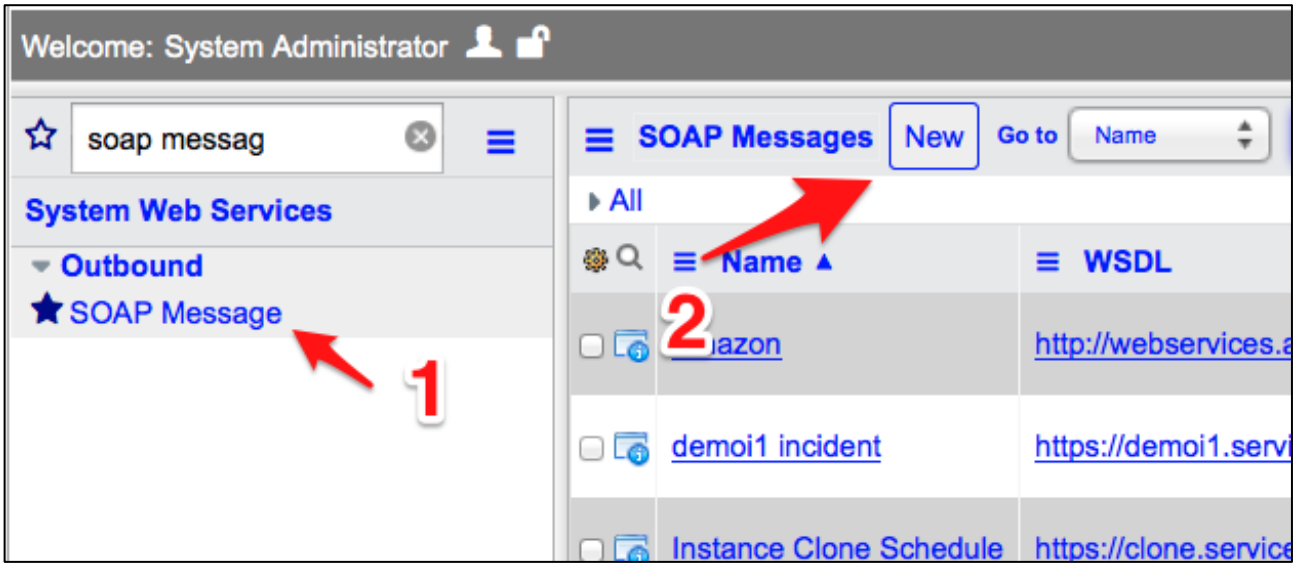
©2014 ServiceNow, Inc. All rights reserved.

Browse the WSDL Catalog

1. You have now stored a third party WSDL into ServiceNow. We can consume this WSDL with any type of SOAP Client. For this example, we will use ServiceNow's built in GUI SOAP client to consume your web service.

2. Navigate to **System Web Services > Outbound > SOAP** Message.

3. Click **New** to start a new SOAP client project.



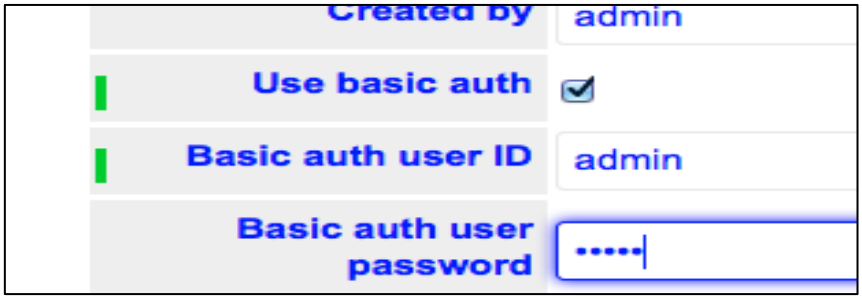
4. Set the **Name** field to be: "**SpaceCurrency**".

5. Unlock the **WSDL** field and the address to your WSDL. It will look something like: **https://MYINSTANCE.lab.service-now.com/SpaceCurrency.do?WSDL**.

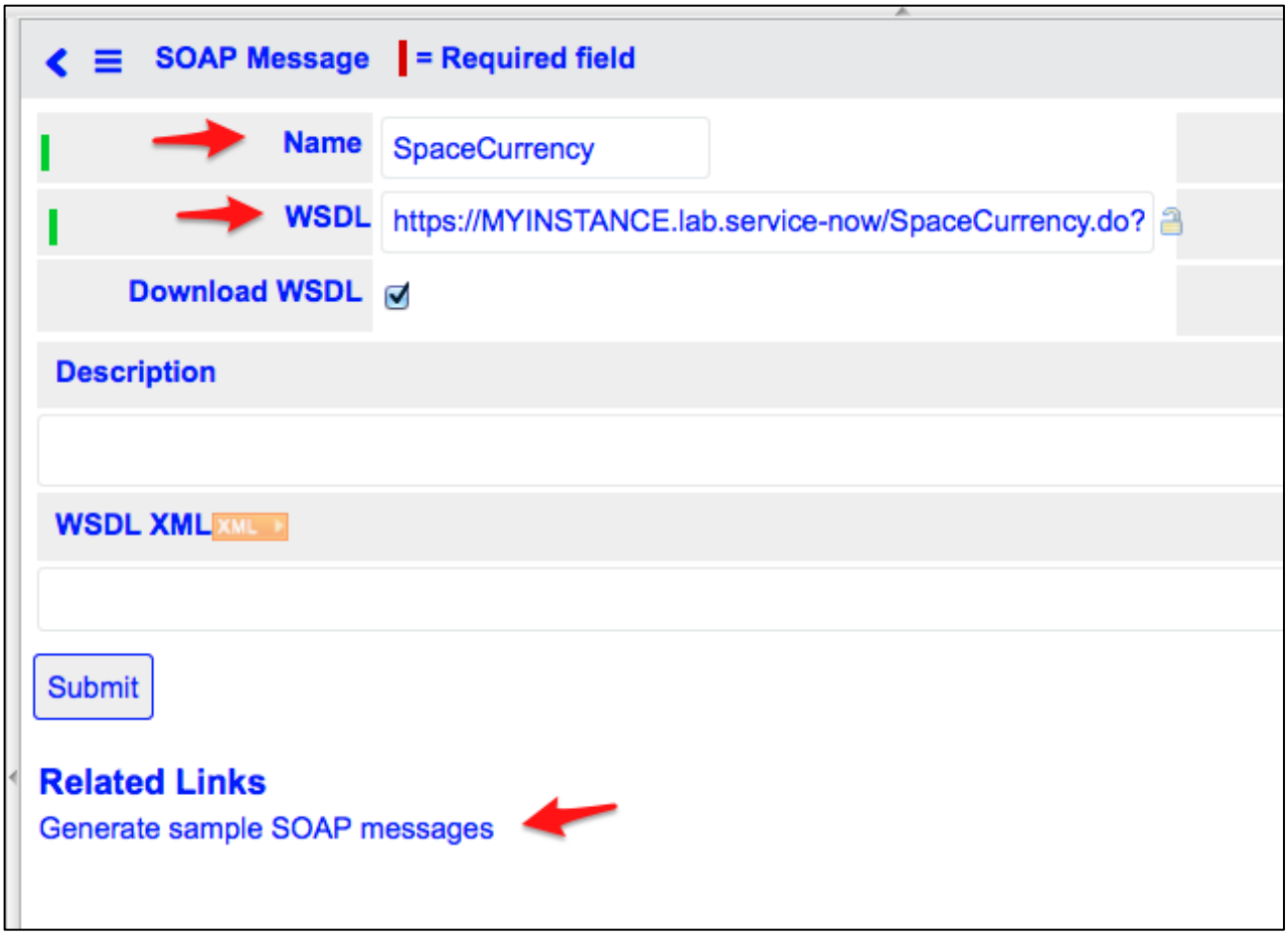
6. This WSDL is protected by default, so we will need to authenticate to access it. On the form, check the **Use basic auth** checkbox

©2014 ServiceNow, Inc. All rights reserved.

7. Enter in your username and password in the **Basic auth user ID**, and **Basic auth user password** fields respectively.



8. Click the "**Generate sample SOAP messages"** link.

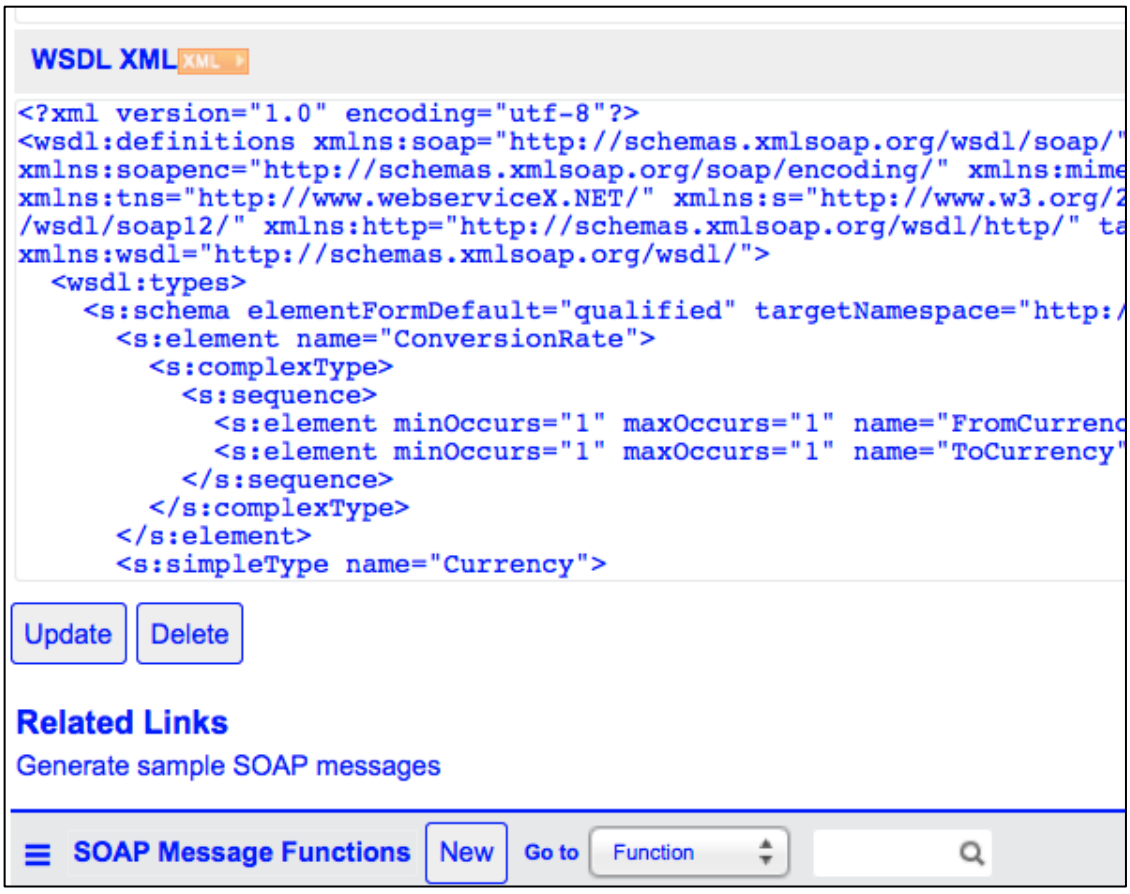


©2014 ServiceNow, Inc. All rights reserved.

9. If a dialog box pops up with a warning, just click **OK**.



10. When the page refreshes, you will notice that the **WSDL XML** field is populated with your Static WSDL document



©2014 ServiceNow, Inc. All rights reserved.

11. Verify that a new **SOAP Message Function** record is created and is named **ConversionRate**.



12. Click on the **ConversionRate** function. You should see a SOAP Envelope template that looks something similar to the following:



13. We won't call this function yet because this is only a Description of the web service. We haven't built the web service to support the function yet. This will be done in the next exercise.

Report your Completion of this Lab Exercise

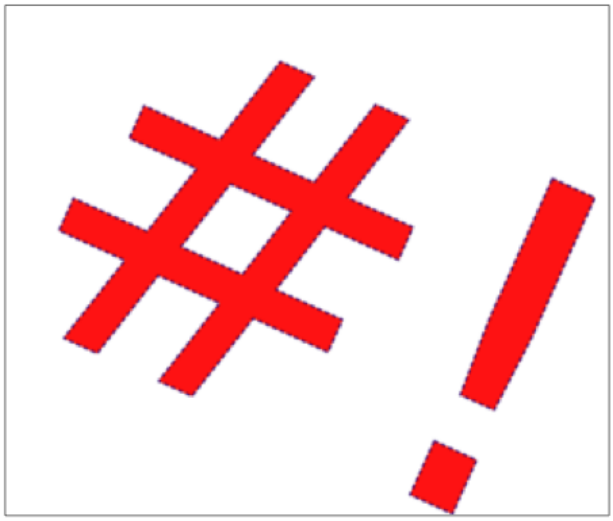
1. Navigate to **Lab Management > Report Lab Progress**.

2. Enter your progress status

©2014 ServiceNow, Inc. All rights reserved.

Scripted Web Services

- Leverage two different WSDL Options
  - Custom/Static WSDL
  - Autogenerated WSDL
- WSDL vs. Scripted Web Service
  - WSDL describes the custom service
  - Scripted Web Service implements the service
- When to use Static WSDL
  - Third part is dictating the SOAP format
  - If you want to handle multiple functions in a single service



When creating your own Scripted Web Service within ServiceNow, you can provide your custom WSDL to a third party in two main ways: first, you can auto generate this WSDL based on the settings within your Scripted Web Service. Second, you can use a custom/static WSDL that you create from scratch.

The purpose of the Scripted Web Service is to implement a service that is not offered out-of-the-box within ServiceNow. The custom, or auto generated WSDL should describe your Scripted Web Service and what is expected as input as well as what should be expected for the response.

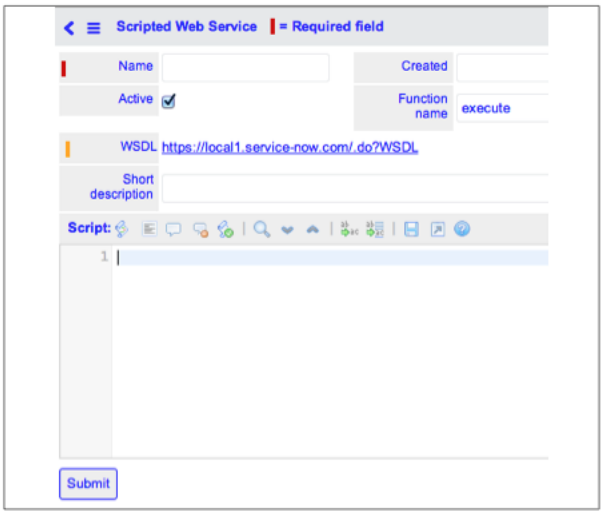
Typically, it is sufficient to use the auto generated WSDL that is provided by the system based on the configuration settings that you have on your web service. However, there are times when a third party product has very little flexibility in the way in which it needs to consume another web service. In these cases it will often dictate what the soap envelope should look like in the request. In that case, you will probably have to use a static WSDL to build out the WSDL in the format that the third-party can consume.

Another occasion where you might need to use a static WSDL instead of the auto generated one, is if you want to add more than one function to your script web service. In this case, you can take the auto generated WSDL and modify it to present the client with the multiple functions that are available.

®2014 ServiceNow, Inc. All rights reserved.

Implementing the Scripted Web Service

- Name (this will be the access point)
  - https://myinstance/NAME.do?SOAP
  - If using a Static WSDL the name will need to be the same as that of the Static WSDL
- Function Name
  - Only useful if using auto generated WSDL
- Short Description
  - Optional and Arbitrary
  - Used to help describe the web service
- Script
  - The code that implements the web service



When creating a Scripted Web Service, the text that you store in the name field will serve as the endpoint for the web service. Please use great care to choose a name for the Scripted Web Service that will not compete against the table name in the system. Otherwise you will have a conflict if a web service tries to consume the table's endpoint because it will have the same SOAP endpoint as your Scripted Web Service.

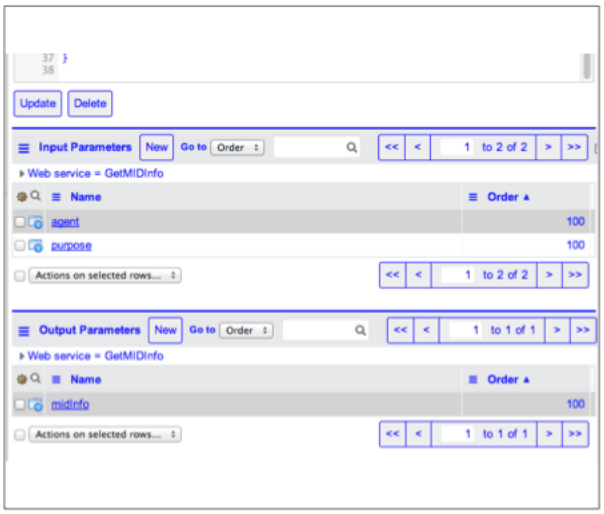
If you are using a static WSDL, the name of that static WSDL will have to be the same as the name of your Scripted Web Service.

The function name on the Scripted Web Service is optional. The system will auto generate a function name for you. If you're only using one function for your Scripted Web Service, then it is usually just fine to use the default auto generated function name.

©2014 ServiceNow, Inc. All rights reserved.

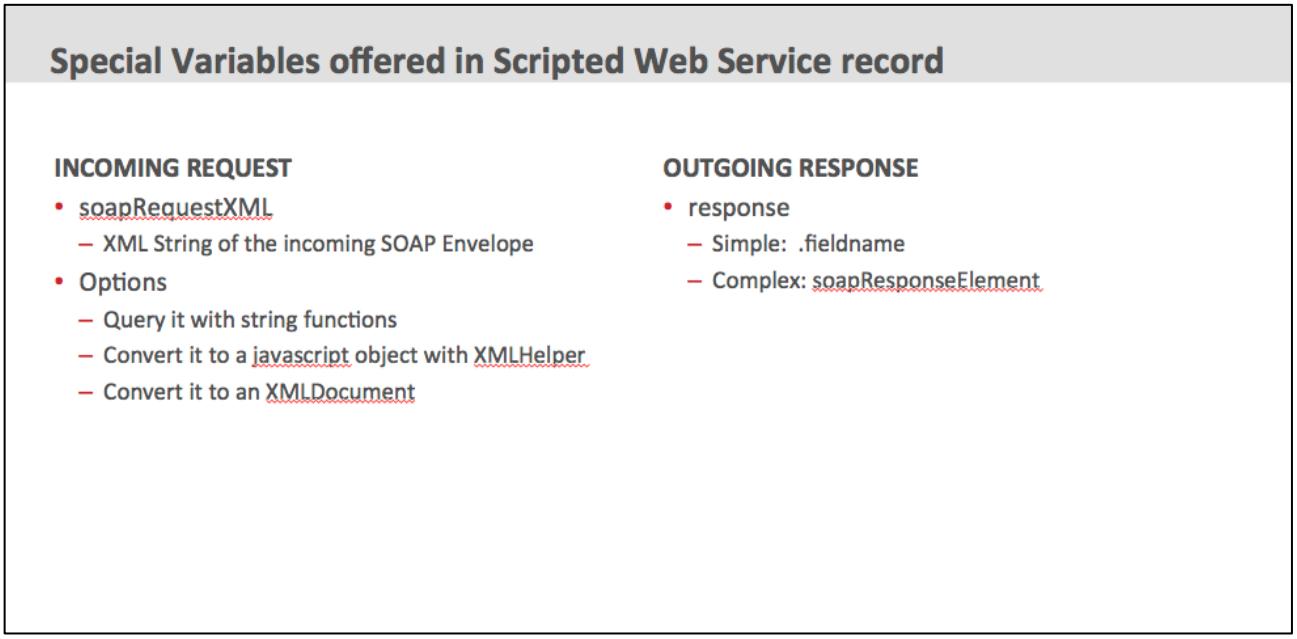
Scripted Web Service with Auto WSDL Generation

- Input Parameters
  - XML elements that will be expected in the SOAP Body
- Output Parameters
  - XML Fields that the client should expect in the response



If you're going to use an auto generated WSDL for your Scripted Web Service, you need to make sure that you create "input parameters" and "output parameters" records on the Scripted Web Service. These records are key in helping the auto generation script to create the proper input and output expectations for the web service.

©2014 ServiceNow, Inc. All rights reserved.



When working with a Scripted Web Service, there are a couple of global variables that are very important for you to utilize in order to successfully interpret the request as well as to provide a detailed response back to the client when you are completed with your script.

The "soapRequestXML" variable is an XML string of the SOAP envelope. You can take that string and convert it to an XMLDocument object to perform XPATH queries on it.

The "response" variable is the global variable that stores the response elements that will be provided back to the client. Adding member variables to the “response” object will add flat level fields to the soap response with that name. Using the soapResponseElement variable on the "response" object, you can set an XML node containing any level of data that you feel is important to return in the web service response.

©2014 ServiceNow, Inc. All rights reserved.

Grabbing incoming SOAP parameters with XMLDocument

```
<soapenv:Envelope>
   <soapenv:Header/>
   <soapenv:Body>
      <rtes:execute>
         <firstname>John</firstname>
         <lastname>Andersen</lastname>
      </rtes:execute>
   </soapenv:Body>
</soapenv:Envelope>
```

```
1 var xml = new XMLDocument(soapRequestXML);
2 firstname = xml.getNodeText("//execute/firstname");
3 lastname = xml.getNodeText("//execute/lastname");
```

Grabbing incoming SOAP parameters in Auto-Generated WSDL mode

```
<soapenv:Envelope>
   <soapenv:Header/>
   <soapenv:Body>
      <rtes:execute>
         <firstname>John</firstname>
         <lastname>Andersen</lastname>
      </rtes:execute>
   </soapenv:Body>
</soapenv:Envelope>
```

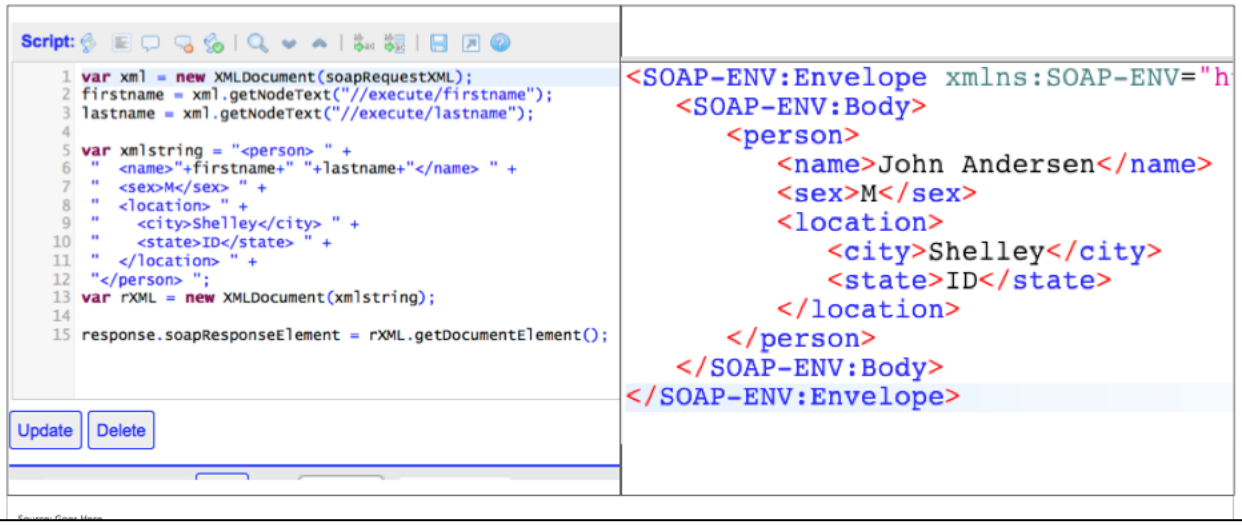
```
1 var firstname = request.firstname;
2 var lastname = request.lastname;
```

©2014 ServiceNow, Inc. All rights reserved.

Building a Simple Response in Scripted Web Service



Building a Complex Response in Scripted Web Service



©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

In this exercise, you will implement the code required to support the Static WSDL that you loaded in the prior exercise.

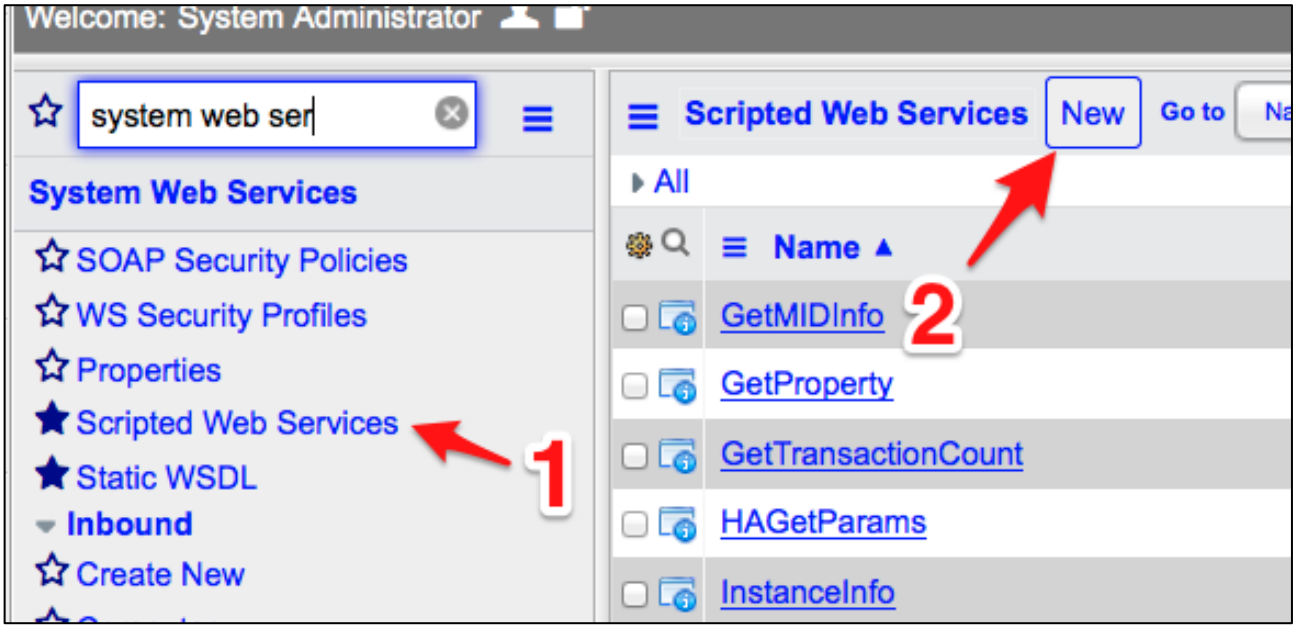
You will write code that will read the SOAP request and pull out a FROM and TO currency code. You will take these codes and use them to determine the multiplier used to convert from the one currency to the other. You will then send that multiplier back in the SOAP Response.

Lab 2 Implement a Scripted Web Service

Create the Scripted Web Service Record

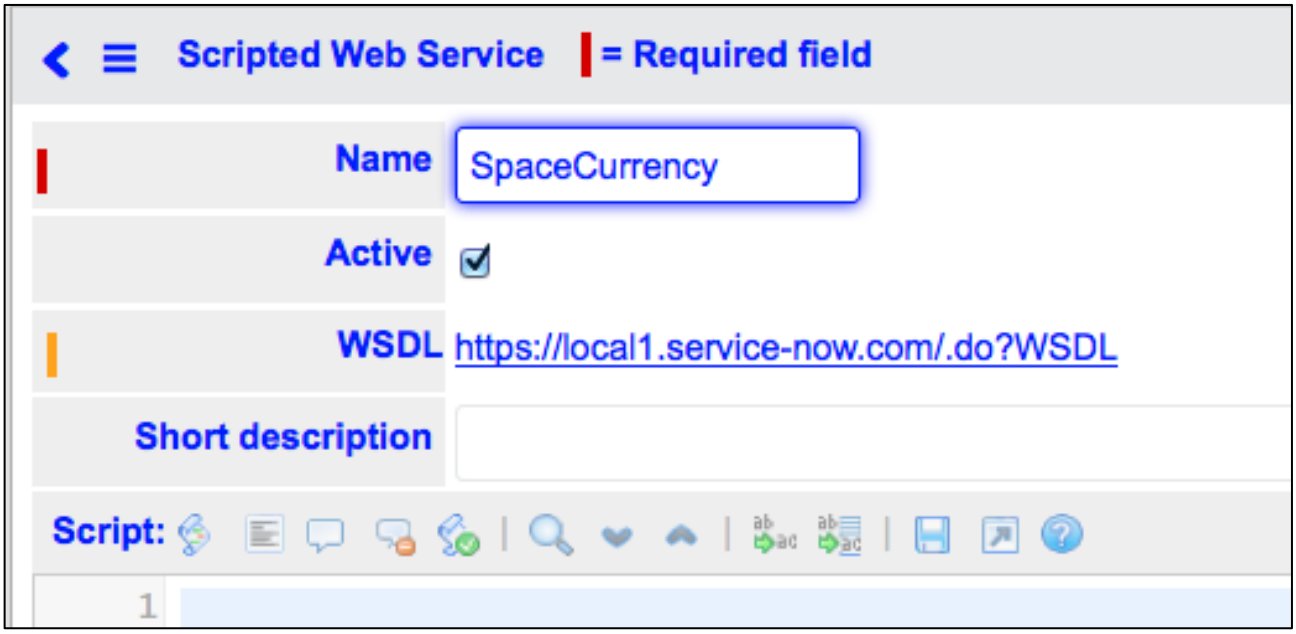
1. Navigate to **System Web Services > Scripted Web Service**.

2. Click the **New** button to go to a blank Scripted Web Service form.

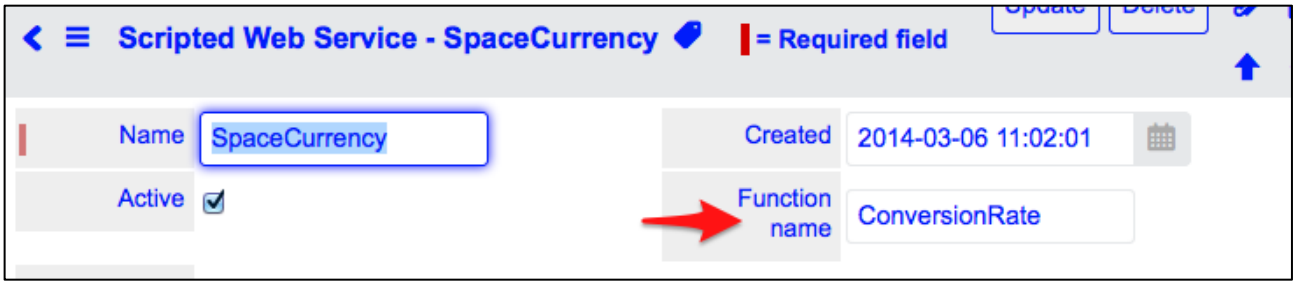


©2014 ServiceNow, Inc. All rights reserved.

3. Set the **Name** field to be the same as the name of the Static WSDL. In this case, it is "**SpaceCurrency**".



4. Set the **Function name** to be "**ConversionRate**".



5. Set the Short Description field to be: "**Convert Inter-galactic currencies**".

©2014 ServiceNow, Inc. All rights reserved.

Set up the script in the Scripted Web Service

Within the same Scripted Web Service record, we will build the script that reads the incoming response, calculates the rate, and returns it in a response

1. In the **Script** field, enter the portion of script that takes the incoming SOAP Envelope string and converts it to an XMLDocument object.

```
Script:
1 //Convert incoming SOAP request string to an XML Doc object
2 var xmlDoc = new XMLDocument(""+soapRequestXML);
3
```

2. Continue building the script by pulling out the **FromCurrency** and **ToCurrency** elements from the incoming SOAP request.

```
Script:
1 //Convert incoming SOAP request string to an XML Doc object
2 var xmlDoc = new XMLDocument(""+soapRequestXML);
3
4 //Pull out XML elements using XPATH
5 var fromC = xmlDoc.getNodeText("//ConversionRate/FromCurrency");
6 var toC = xmlDoc.getNodeText("//ConversionRate/ToCurrency");
7
```

3. Have the script call a special function that has been preloaded for this lab. This function will calculate the inter-galactic rate for you.

```
Script:
1 //Convert incoming SOAP request string to an XML Doc object
2 var xmlDoc = new XMLDocument(""+soapRequestXML);
3
4 //Pull out XML elements using XPATH
5 var fromC = xmlDoc.getNodeText("//ConversionRate/FromCurrency");
6 var toC = xmlDoc.getNodeText("//ConversionRate/ToCurrency");
7
8 //Calculate the currency conversion rates
9 var rate = space_getConversionRate(fromC, toC);
10
```

©2014 ServiceNow, Inc. All rights reserved.

4. Have your script handle errors by checking to see if the rate comes back with a value. If it doesn't, then write out a complex XML Fault element to the response for the client.

Script:
```
//Convert incoming SOAP request string to an XML Doc object
var xmlDoc = new XMLDocument(""+soapRequestXML);

//Pull out XML elements using XPATH
var fromC = xmlDoc.getNodeText("//ConversionRate/FromCurrency");
var toC = xmlDoc.getNodeText("//ConversionRate/ToCurrency");

//Calculate the currency conversion rates
var rate = space_getConversionRate(fromC, toC);

if(!rate){
    //Error condition occurred - return a FAULT message
    //This will require the complex method since we are nesting elements
    var f = "<Fault><faultstring>Invalid FROM or TO value</faultstring></Fault>";
    var s = new XMLDocument(f);
    response.soapResponseElement = s.getDocumentElement();
}
```

5. If the rate comes back with a value, we will return a simple, flat "**rate**" element back to the client in the response.

Script:
```
//Calculate the currency conversion rates
var rate = space_getConversionRate(fromC, toC);

if(!rate){
    //Error condition occurred - return a FAULT message
    //This will require the complex method since we are nesting elements
    var f = "<Fault><faultstring>Invalid FROM or TO value</faultstring></Fault>";
    var s = new XMLDocument(f);
    response.soapResponseElement = s.getDocumentElement();
}

if(rate){
    //Success - send the conversion rate
    //We can use the simple method since we just want a flat response on success
    response.rate = rate;
}
```

6. NOTE: if necessary, you can copy this entire script by going to **Integration Lab 301 > Copy & Paste > Space Currency Script**.

7. Click **Submit**.

©2014 ServiceNow, Inc. All rights reserved.

Test the Scripted Web Service

1. To test the web service, we will use the built-in SOAP Client in your ServiceNow instance to make the request. You could use any type of SOAP Client you wish, but for the purposes of the lab, we will leverage the SOAP Message client.

2. Navigate to **System Web Services > Outbound > SOAP Message**

3. Open the **SpaceCurrency** record that you created earlier to test the Static WSDL.

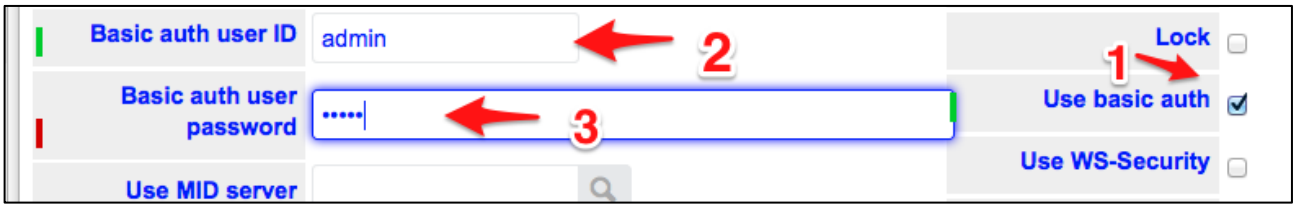
4. Find to the **ConversionRate** record in the **SOAP Message Functions** list on the form. Click on the record to open the function.



©2014 ServiceNow, Inc. All rights reserved.

5. Once on the **ConversionRate** form, check the **Use basic auth** checkbox.

6. Enter your credentials in the **Basic auth user ID** and **Basic auth user password** fields.



7. In the **Envelope** field, change **${FromCurrency}** to "**USD**" (for US Dollar).

8. In the **Envelope** field, change **${ToCurrency}** to "**FED**" (for United Federation of Planets – Federation Credits)



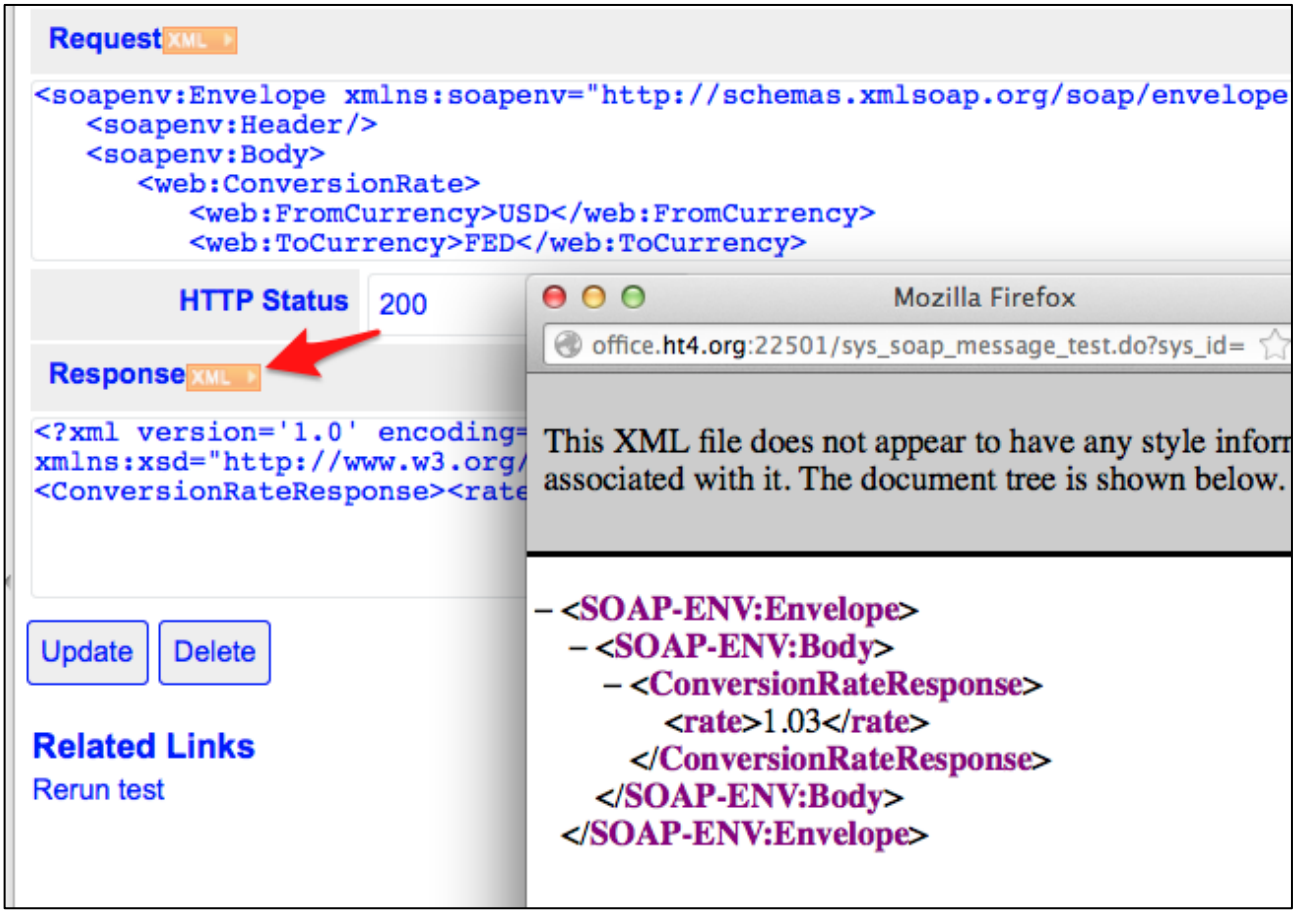
©2014 ServiceNow, Inc. All rights reserved.

9. Click the **Test** link to submit the request to your web service.



©2014 ServiceNow, Inc. All rights reserved.

10. Your page should refresh and show the **Request** and **Response** strings. Click the **XML** button on the **Response** field to see a formatted view of the response.



Report your Completion of this Lab Exercise

1. Navigate to **Lab Management > Report Lab Progress**.

2. Enter your progress

©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

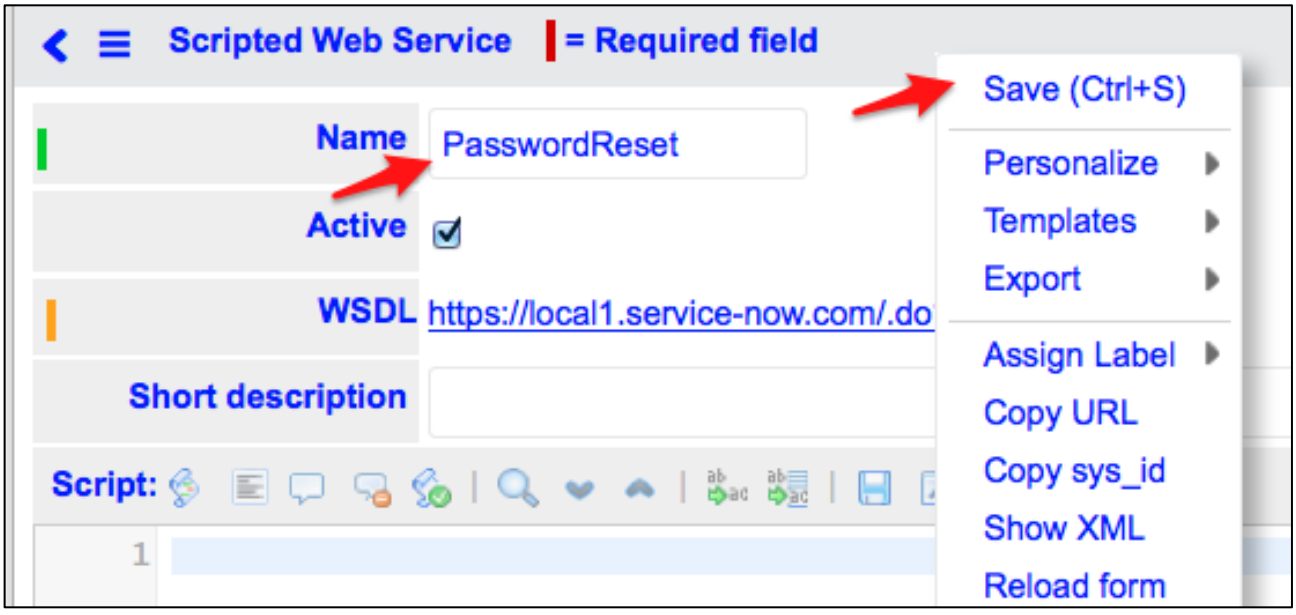
Learn how to create a Scripted Web Service in a manner that automatically generates a custom WSDL for you.

In this exercise you will create a Scripted Web Service that allows a third party to submit a SOAP request that specifies a username and password. If that username matches a user in the instance, that user's password will be set to the specified value. Also, that user's account will be flagged to require a password change on the next login.

Lab 3
Leverage Auto Generated WSDLs

Create the Scripted Web Service Record

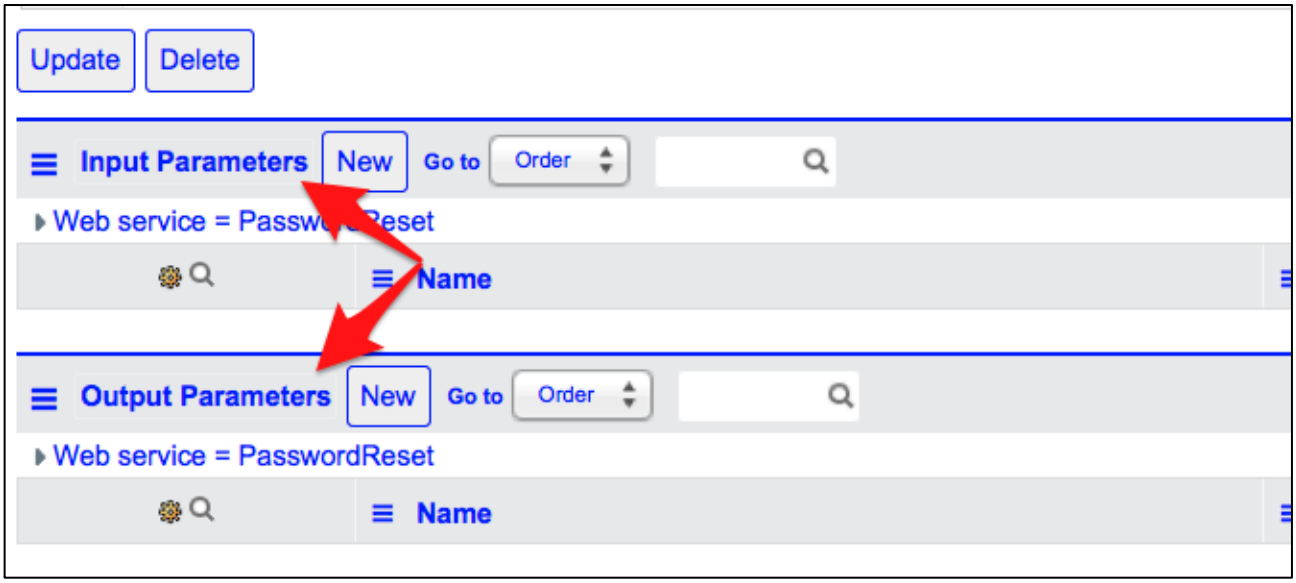
1. Navigate to **System Web Services > Scripted Web Services**.
2. Click **New**.
3. Set the **Name** field to be "**PasswordReset**".
4. Right click the title bar of the form and click **Save**.



©2014 ServiceNow, Inc. All rights reserved.

Create the Input Parameters for the Web Service

1. After the save takes place, you should have two tabs or lists associated with the Scripted Web Service. They are "**Input Parameters**" and "**Output Parameters**".

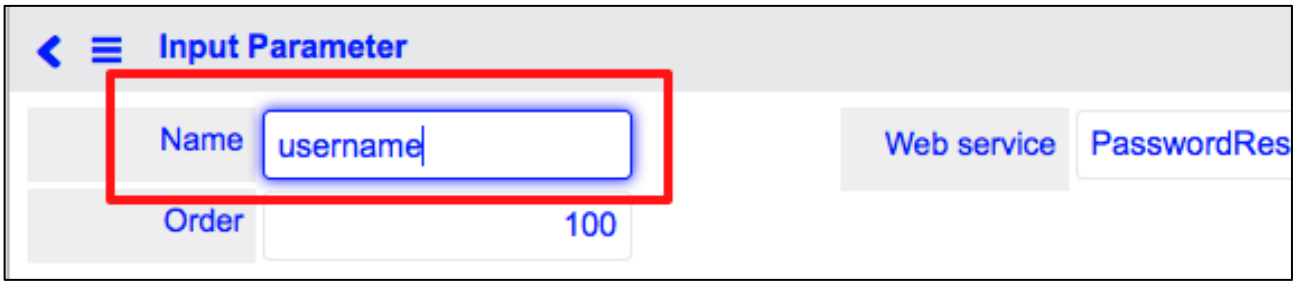


2. We want to create two input parameters. One for the username, and the second for the new password value. Start by clicking **New** on the **Input Parameter** list.



©2014 ServiceNow, Inc. All rights reserved.

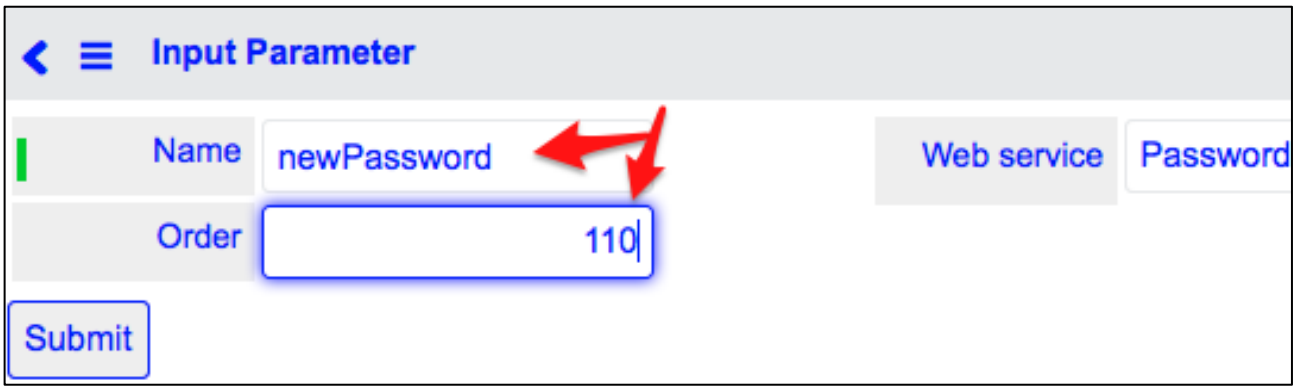
3. Set the **Name** field to be "**username**".



4. Click **Submit**.

5. Click **New** once again on the **Input Parameter** list.

6. Set the **Name** field to be "**newPassword**". Also, set the **Order** field to be "**110**".



7. Click **Submit**.

©2014 ServiceNow, Inc. All rights reserved.

8. Your **Input Parameters** list should now look like this:



Create the Web Service's Output Parameter

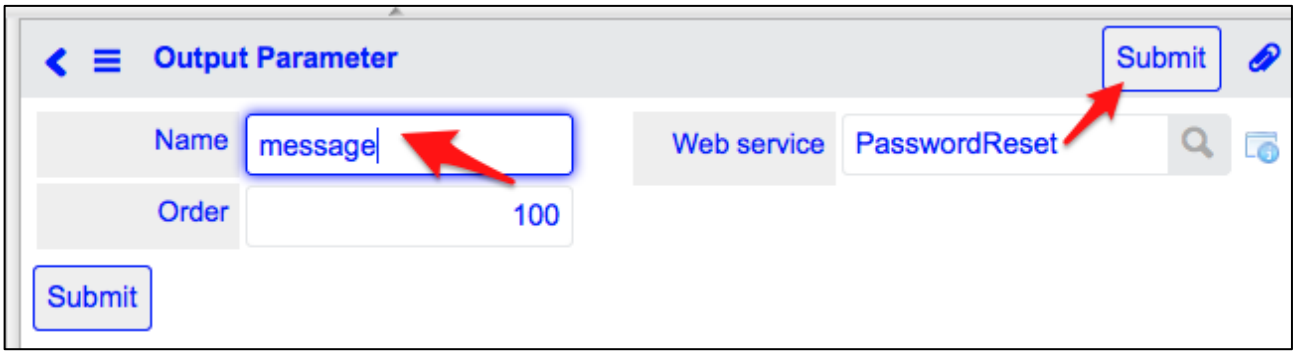
1. Click on the **Output Parameter** tab (if using the tab view, otherwise scroll down to the **Output Parameter** list).

2. Click **New**.



©2014 ServiceNow, Inc. All rights reserved.

3. Set the **Name** field to be "**message**" and click **Submit**.



4. Your **Output Parameters** section should now look like this:



View the Auto Generated WSDL

1. Click on the **WSDL** link for the **Scripted Web Service**. (Please note, your WSDL will have a different base domain name than what is listed in this lab guide).



©2014 ServiceNow, Inc. All rights reserved.

2. This should open up a new tab or window in your browser and display an XML Document that is your auto-generated WSDL. Note the input and output parameter definitions appear within the Type section. Also note the function this will be called. Finally, notice the SOAP Endpoint that the client will be directed to call

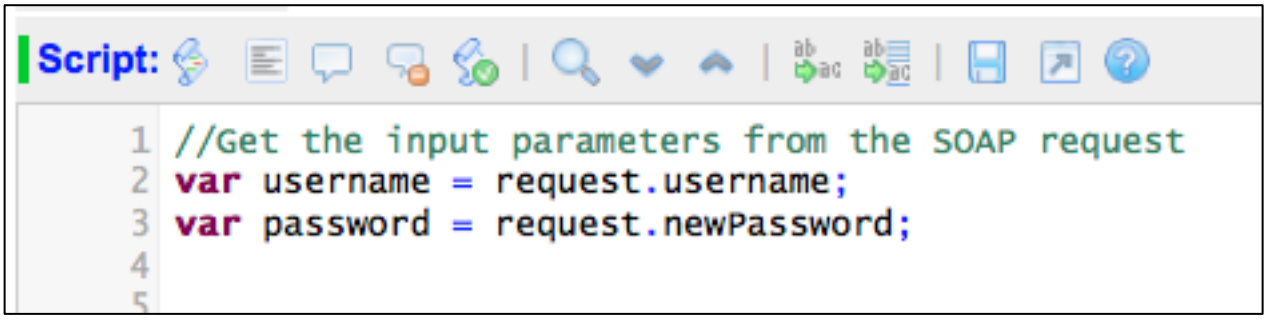
```
- <wsdl:definitions targetNamespace="http://www.service-now.com/PasswordReset">
  - <wsdl:types>
    - <xsd:schema elementFormDefault="unqualified" targetNamespace="http://www.service-now.com/PasswordReset">
      - <xsd:element name="execute">
        - <xsd:complexType>
          - <xsd:sequence>
              <xsd:element maxOccurs="1" minOccurs="0" name="username" type="xsd:string"/>
              <xsd:element maxOccurs="1" minOccurs="0" name="newPassword" type="xsd:string"/>
          </xsd:complexType>
        </xsd:element>
      - <xsd:element name="executeResponse">
        - <xsd:complexType>
          - <xsd:sequence>
              <xsd:element maxOccurs="1" minOccurs="0" name="message" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:schema>
    </wsdl:types>
  - <wsdl:message name="executeSoapOut">
      <wsdl:part name="PasswordReset" element="tns:executeResponse"> </wsdl:part>
    </wsdl:message>
  - <wsdl:message name="executeSoapIn">
      <wsdl:part name="PasswordReset" element="tns:execute"> </wsdl:part>
    </wsdl:message>
  - <wsdl:portType name="ServiceNowSoap">
    - <wsdl:operation name="execute">
        <wsdl:input message="tns:executeSoapIn"> </wsdl:input>
        <wsdl:output message="tns:executeSoapOut"> </wsdl:output>
      </wsdl:operation>
    </wsdl:portType>
  - <wsdl:binding name="ServiceNowSoap" type="tns:ServiceNowSoap">
      rt="http://schemas.xmlsoap.org/soap/http"/>
    - <wsdl:operation name="execute">
        ww.service-now.com/PasswordReset/execute" style="document"/>
      - <wsdl:input>
          <soap:body use="literal"/>
        </wsdl:input>
      - <wsdl:output>
          <soap:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  - <wsdl:service name="ServiceNow_PasswordReset">
      <soap:address location="http://arbonboy.homedns.org:4455/PasswordReset.do?SOAP"/>
    </wsdl:service>
</wsdl:definitions>
```

©2014 ServiceNow, Inc. All rights reserved.

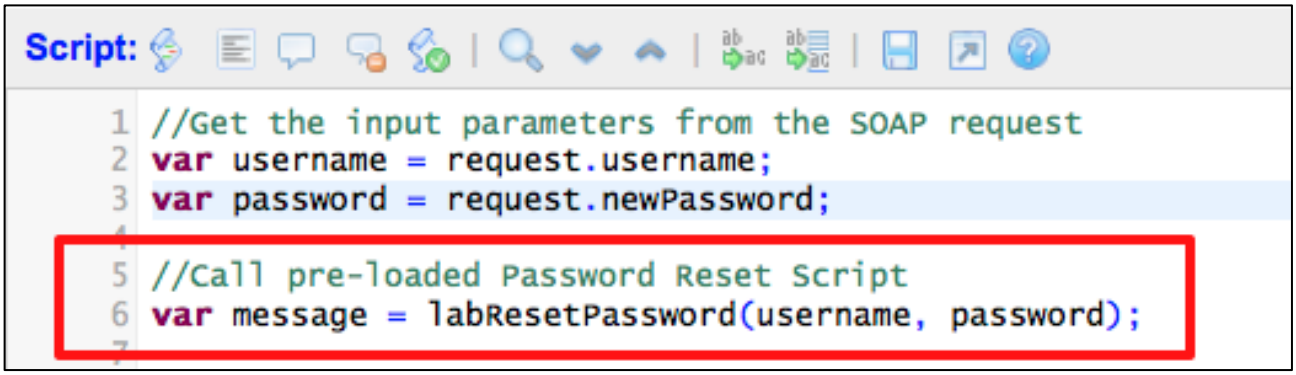
Create the Scripted Web Service

1. Browse to **System Web Services > Scripted Web Services** and open up your **PasswordReset** web service (if you are not already there). The next steps will take place within the **Script** field of your Scripted Web Service record.

2. At the top of the **Script** field, add code that sets a **username** and **password** variable from the request.

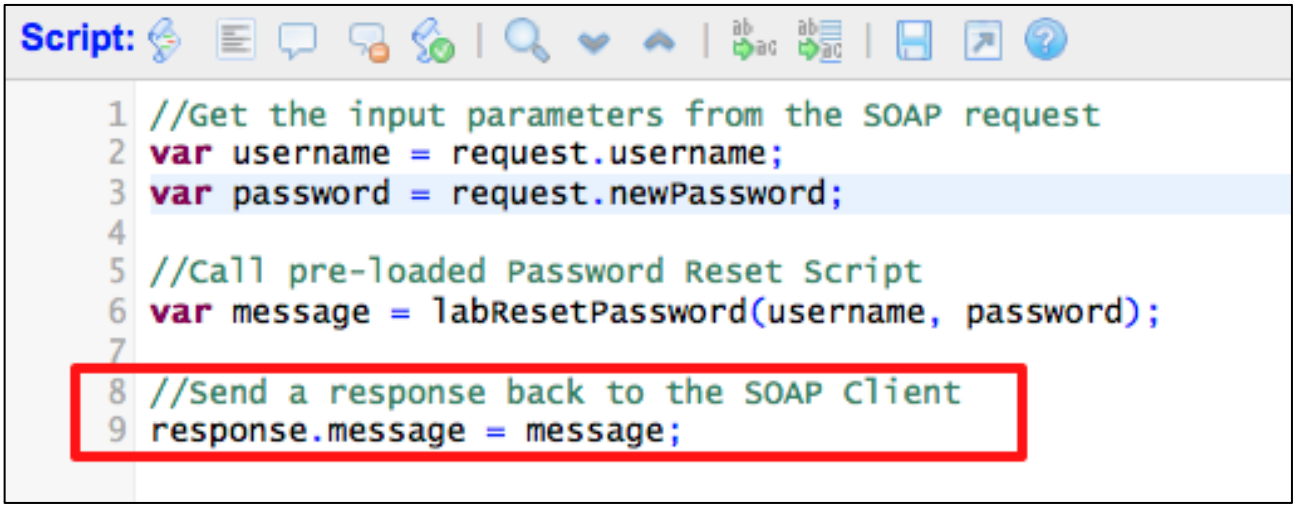


3. Use the next bit of code to call a pre-loaded Password Reset function that was created for this lab. The function is "**labResetPassword**" and it takes the username and the new password as parameters.



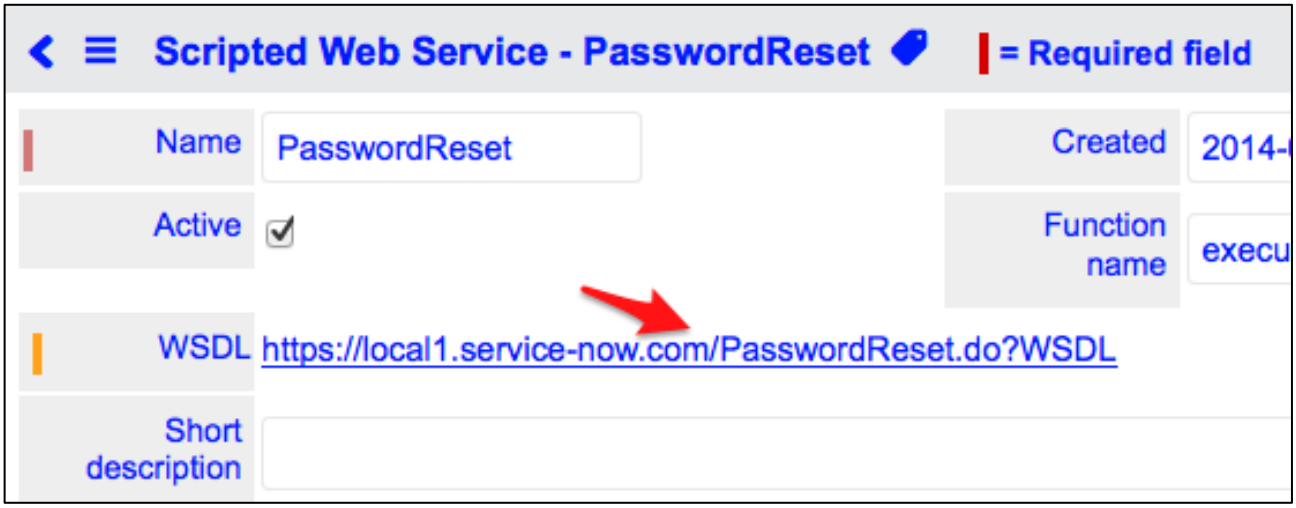
©2014 ServiceNow, Inc. All rights reserved.

4. Now add the following script that will take the reset status message and set it in our response that will be sent to the SOAP client.



5. If you need it, the entire script can be copied from **Integration Lab 301 > Copy & Paste > PasswordReset WS Script**.

6. In the **Scripted Web Service** form, copy the **WSDL** field and have it handy for the next steps.



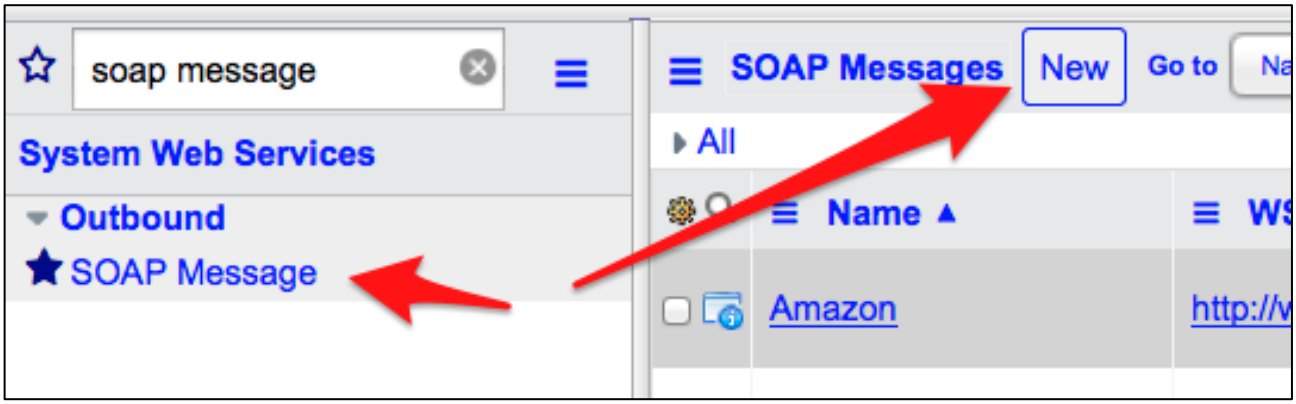
7. Click the **Update** button on your **Scripted Web Service** form.

©2014 ServiceNow, Inc. All rights reserved.

Test your Scripted Web Service

1. Now that your Scripted Web Service has been coded up, we will use the ServiceNow SOAP Message GUI utility to post to the web service. You could use SOAP-UI or any other SOAP client to perform this test as well.

2. Browse to **System Web Services > Outbound > SOAP Message** and click the **New** button to create a new SOAP Message record.



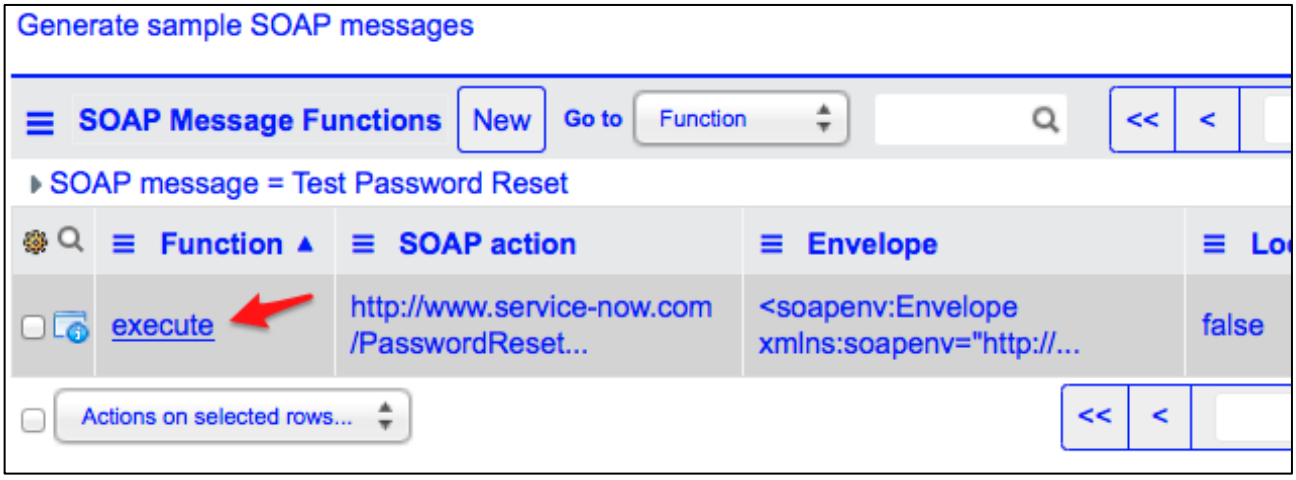
3. Set the following fields:
   - **Name**: Test Password Reset
   - **WSDL**: Paste the WSDL you copied from the Scripted Web Service. If you need to enter it manually, it will look something like: **https://YOURINSTANCE.lab.service-now.com/PasswordReset.do?WSDL**.
   - **Download WSDL**: checked
   - **Use Basic Auth**: checked
   - **Basic auth user ID**: <your admin user>
   - **Basic auth user password:** <your admin user password>

4. Click the **Generate sample SOAP messages** link on the form.

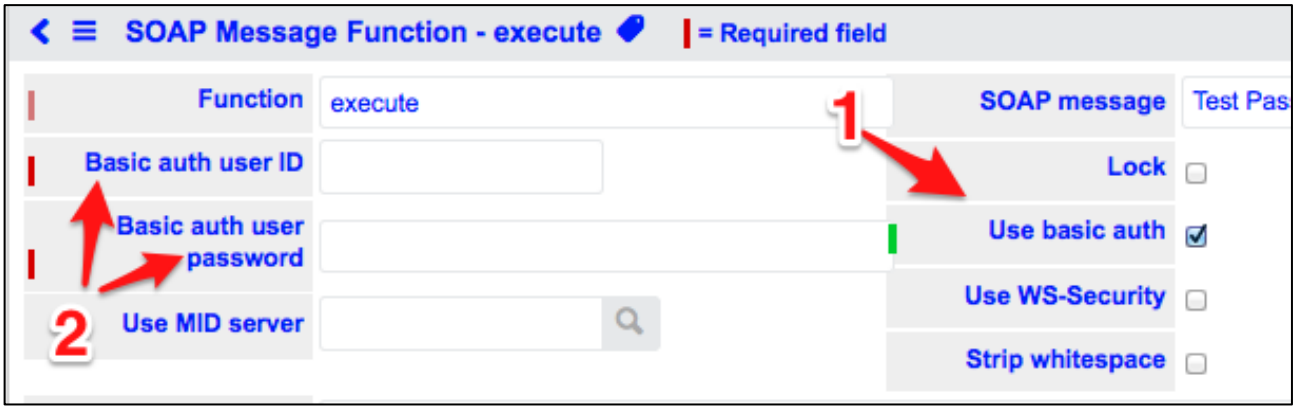
5. Click **OK** if any warning box appears.

©2014 ServiceNow, Inc. All rights reserved.

6. When the page refreshes, you should see one **SOAP Message Function** by the name of **execute**. Open that function.

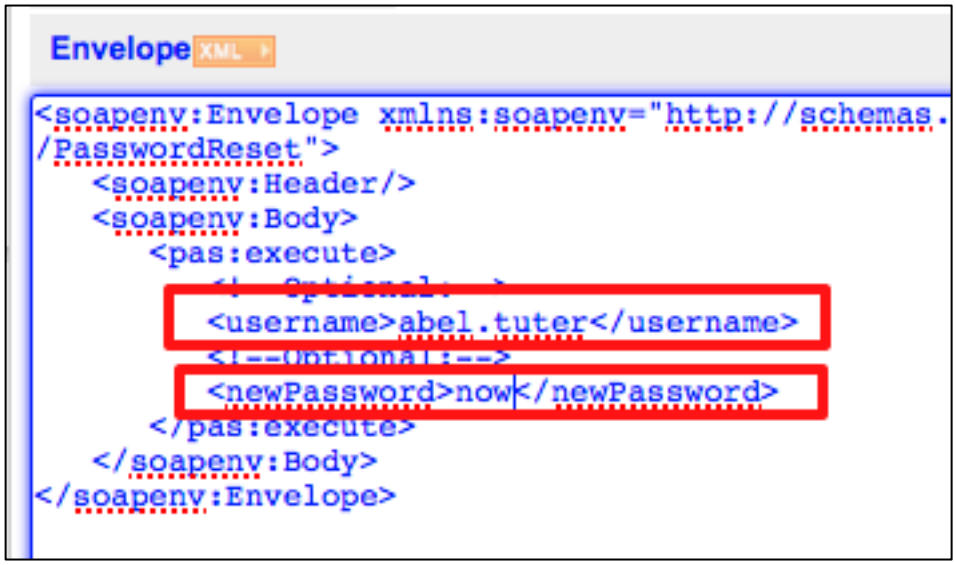


7. By default, ServiceNow web services are protected by Basic Authentication. Check the **Use basic auth** checkbox and enter in your admin credentials in the **Basic auth user ID** and **Basic auth user password** fields that appear on the form.



©2014 ServiceNow, Inc. All rights reserved.

8. The **Envelope** field will have the template XML that is provided through the information in the auto-generated WSDL. You will want to put in real values for the username and newPassword elements.
   - Replace ${username} with “abel.tuter”
   - Replace ${newPassword} with “now”

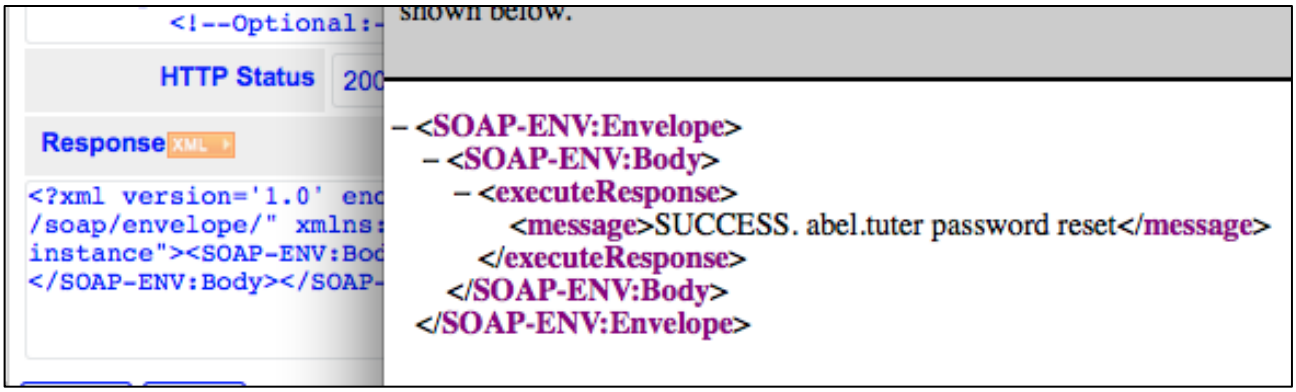


9. Now that you have entered in some test values, click the **Test** link on the bottom of the form.



©2014 ServiceNow, Inc. All rights reserved.

10. When the page loads, you should see the **Request** and the **Response** fields. The Response should contain a message of SUCCESS for user “abel.tuter”. Click the **XML** button next to the **Response** field to see a formatted view of the XML.



Report your Completion of this Lab Exercise

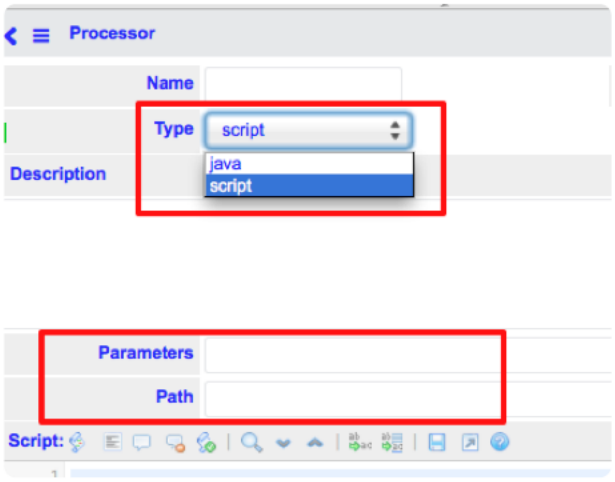
1. Navigate to **Lab Management > Report Lab Progress**.

2. Enter your progress status

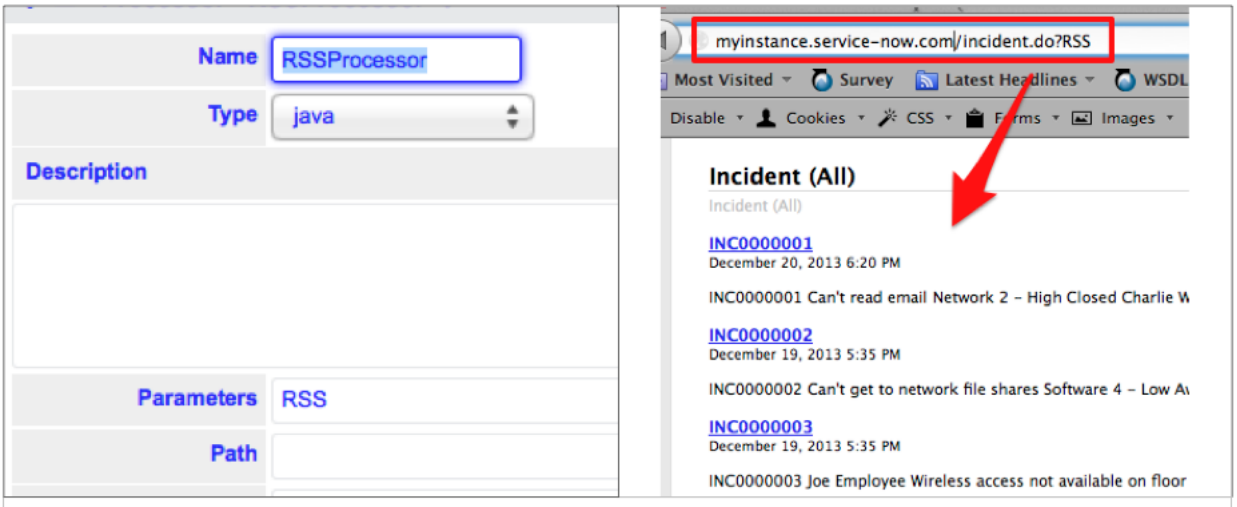
©2014 ServiceNow, Inc. All rights reserved.

Introduction to Processors

- Processors are to REST as Scripted Web Services are to SOAP…almost
- Configurable records that can trigger a script or action
- Scripted vs. Java Processors
  - Scripted are our only options
  - Java is for ServiceNow developers only
- Parameter Processors
  - Triggered by a special parameter on the request
  - Great for table specific actions
- Path Processors
  - Triggered by a specified url page
  - Good for specific or global service

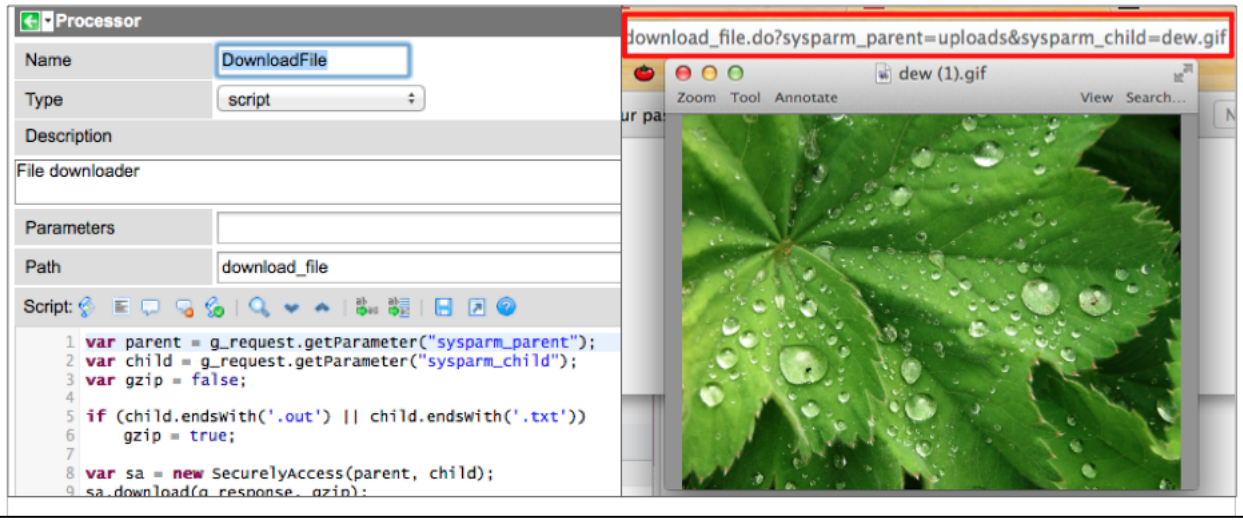


Parameter Processor Example



©2014 ServiceNow, Inc. All rights reserved.

Path Processor Example



Available Functions in Processor Scripts

- g_request – the HTTP Request Object
  - getMethod() – PUT, POST, GET, DELETE, etc
  - getHeaderNames() – returns HTTP Headers
  - getHeader(headerName) – returns header value
  - getParameterNames() – returns HTTP Params
  - getParameter(paramName) – returns HTTP param
  - getInputStream() – streaming data from request
- g_response – the HTTP Response Object
  - setHeader(name, val) – set return headers to client
- g_processor – the processor object
  - writeOutput(output) – the string to be written back to the client

©2014 ServiceNow, Inc. All rights reserved.

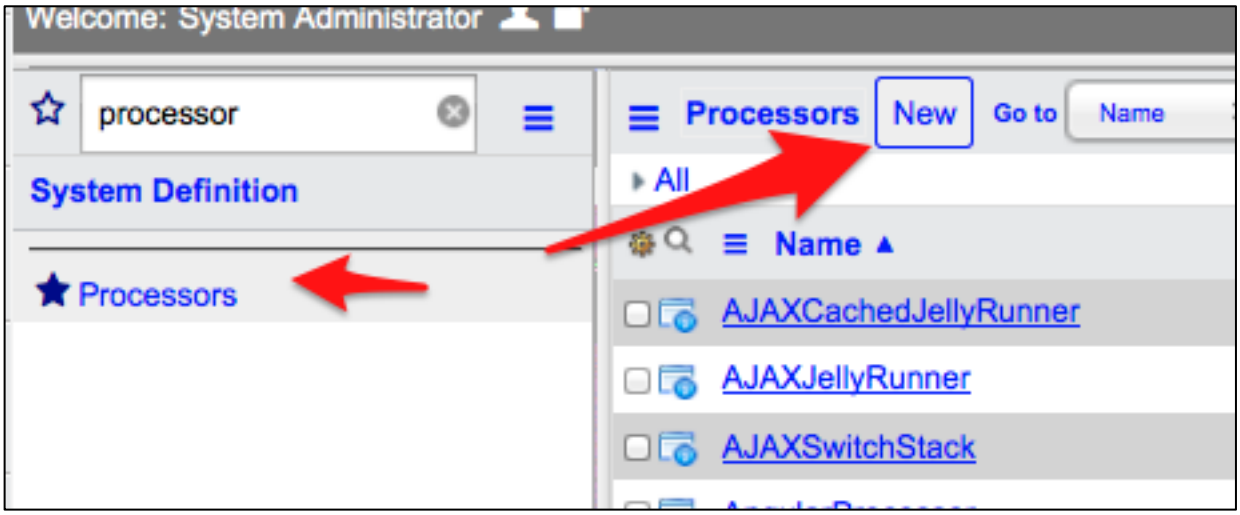
Lab Goal

Learn how to write a path processor by creating a RESTful Space Currency Conversion web service similar to what was done with the Scripted SOAP Web Service in the prior labs.

Lab 4
Create a Path Processor

Create the Scripted Path Processor

1. Navigate to **System Definition > Processors** and click the **New** button to start a new processor record.

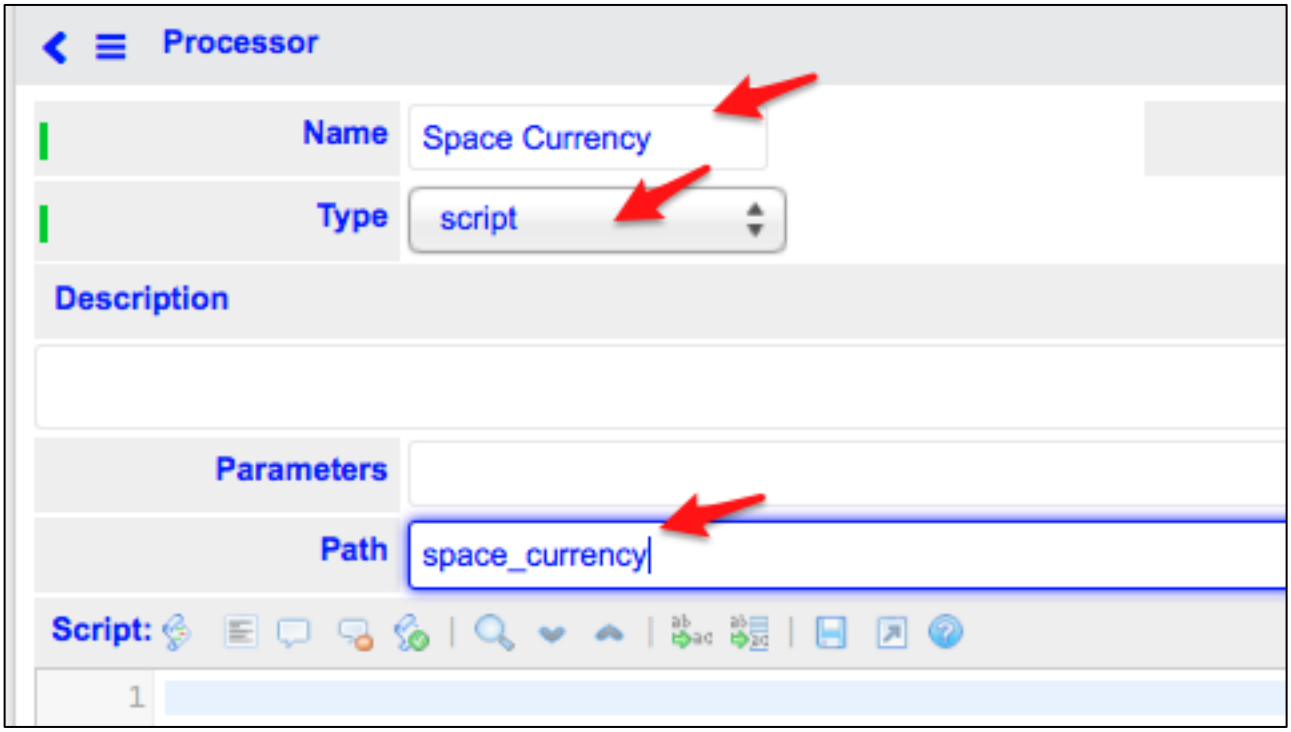


2. Set the **Name** field to be "**Space Currency**".

3. Set the **Type** field to be "**script**".

©2014 ServiceNow, Inc. All rights reserved.

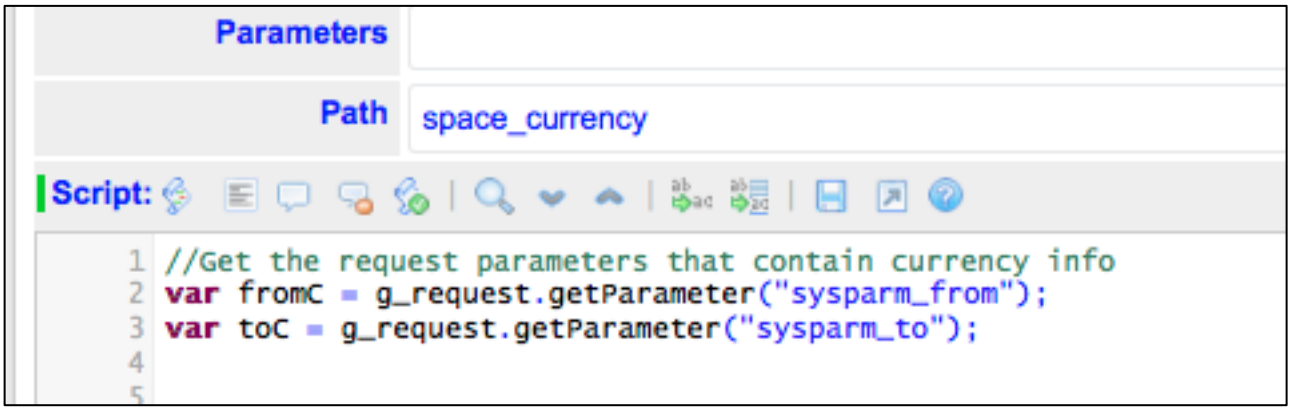
4. Set the **Path** field to be "**space_currency**". This will be the URL page we will browse to. You need to make sure this is not the same name as any UI Page or Table in the instance.



Write the script that handles the incoming request

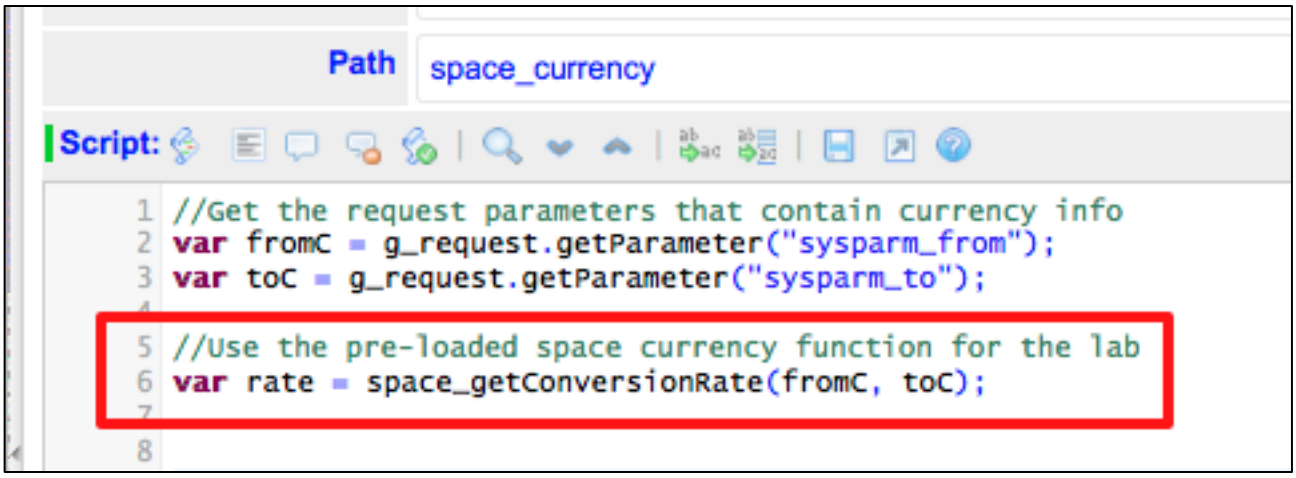
1. While still on the **Processor** form, we will build out the script through the next several steps.

2. Have your script start out by collecting the URL parameters that were passed. The two URL parameters you will support in this web service are **sysparm_from** and **sysparm_to**.

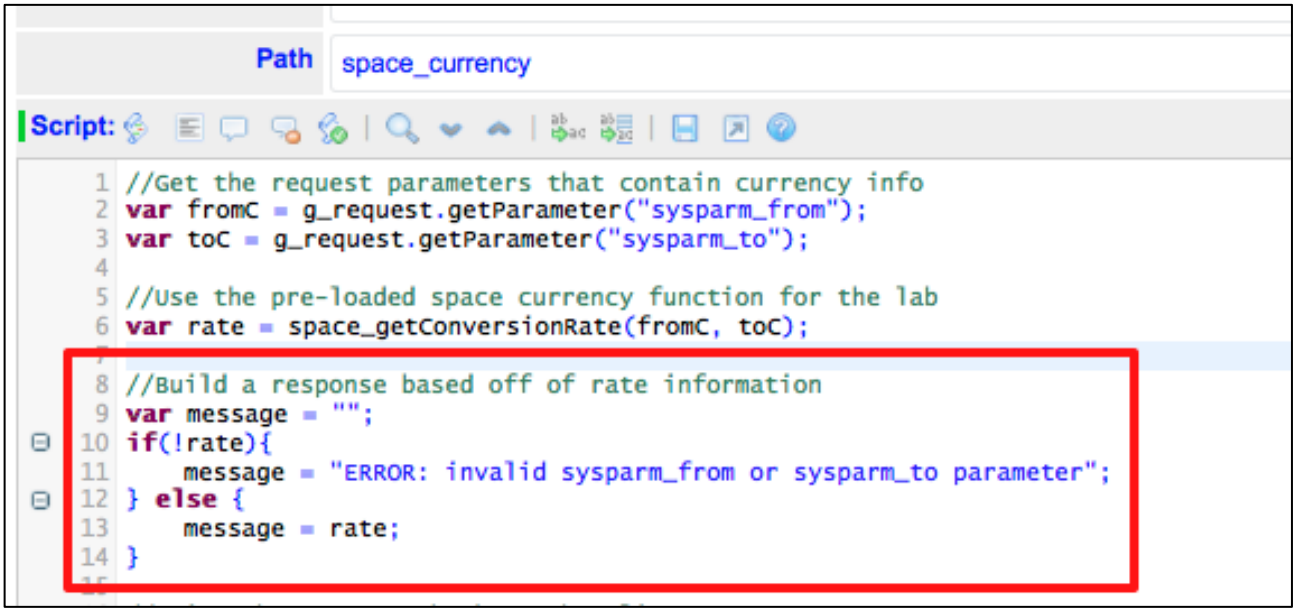


©2014 ServiceNow, Inc. All rights reserved.

3. Now have your script take those values and put them through the pre-loaded script include function that calculates those rates for you. The name of that function is: **space_getConversionRate(from, to)**.

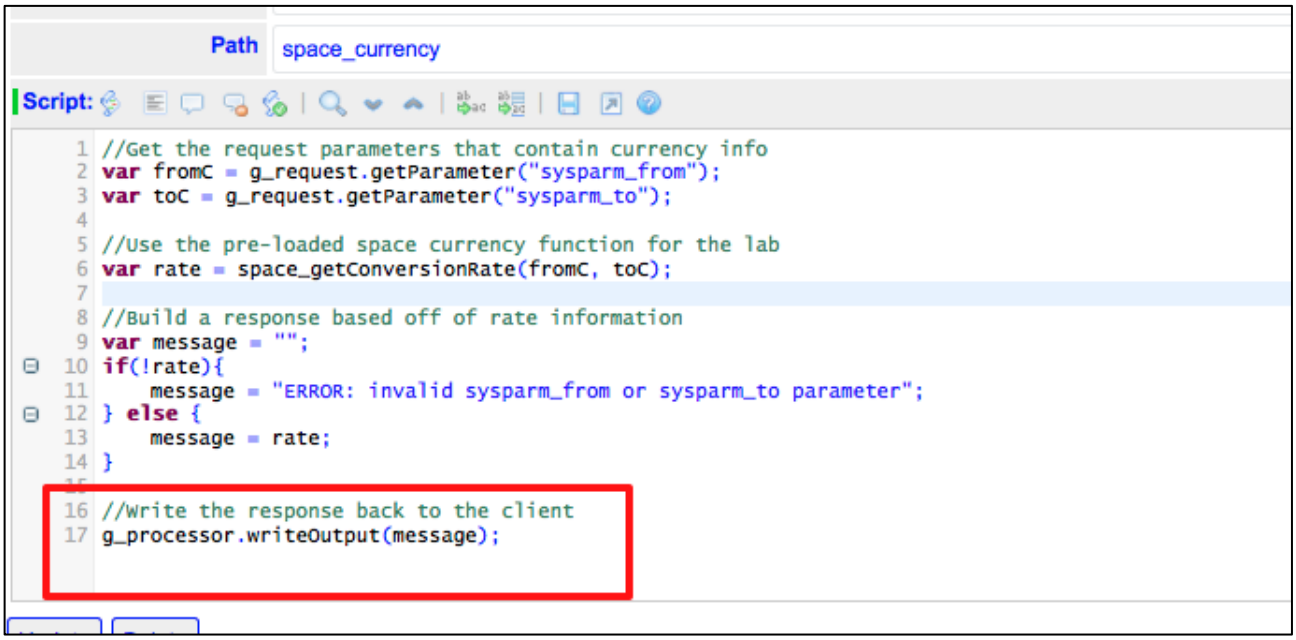


4. Now, based on the response we get from that function, we will build out a message to the client. If no rate information is returned, we will print out an error. If a rate is returned, we will simply display that rate.



©2014 ServiceNow, Inc. All rights reserved.

5. Finally, we will want the script to write the message back to the client.



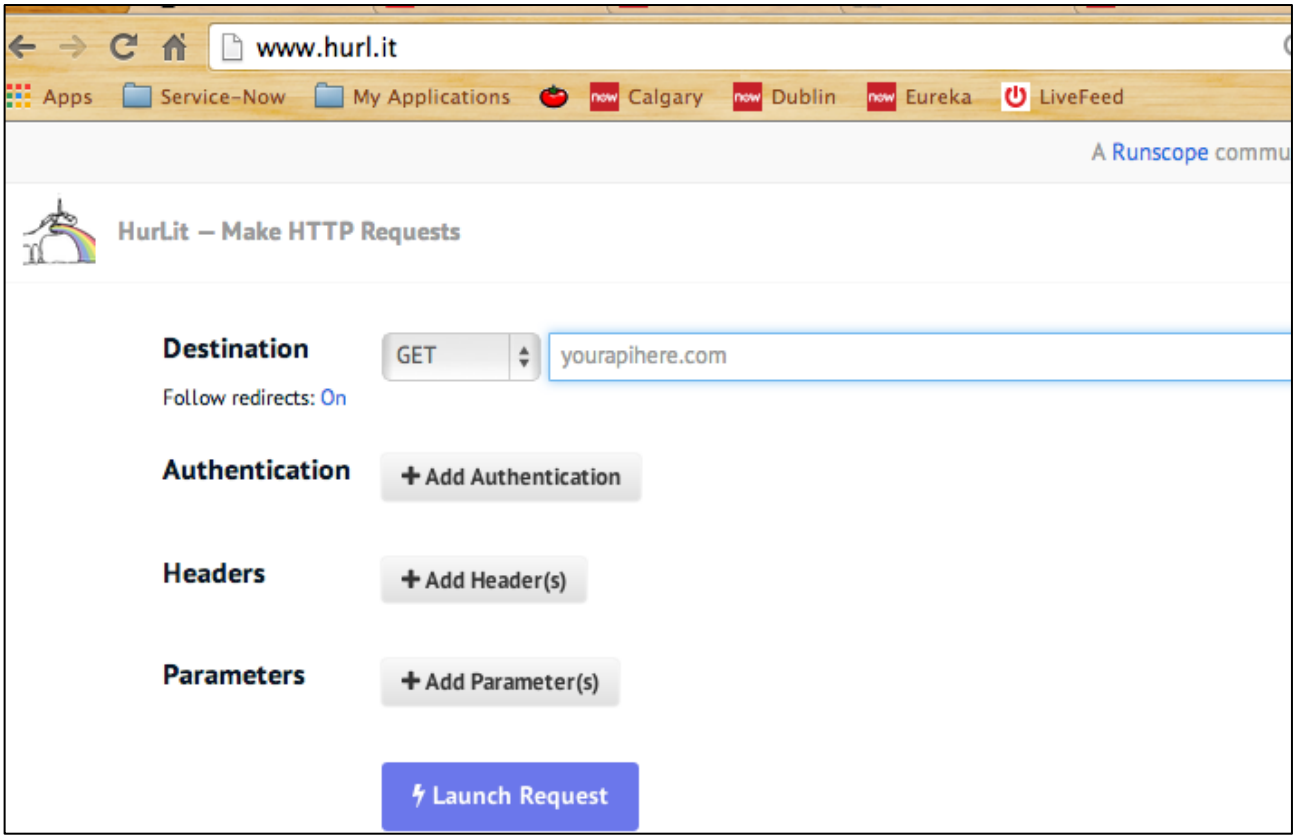
6. We're done…now click **Submit** to save the processor record.

7. NOTE: You can get a copy of the full script by navigating to **Integration Lab 301 > Copy & Paste > Path Processor Script**.

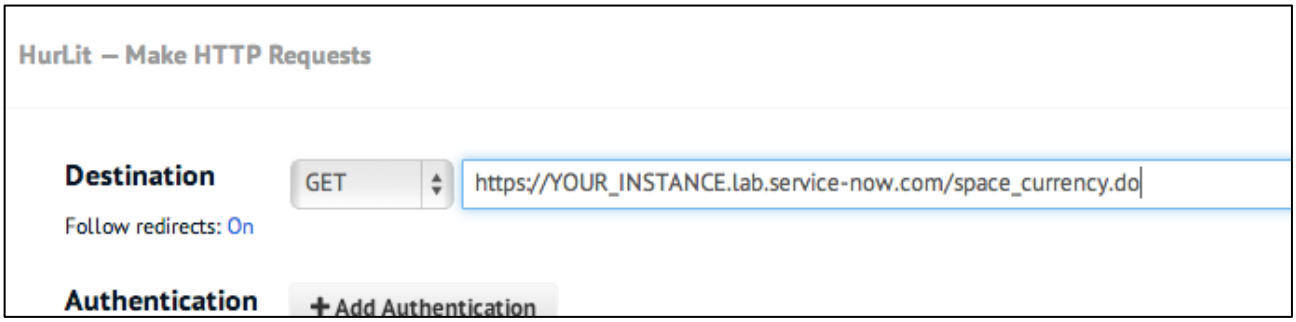
©2014 ServiceNow, Inc. All rights reserved.

Test the scripted path processor

1. To validate this processor, we will use a third party online tool that can easily sent REST requests to the instance. Open a browser window and go to "**http://hurl.it**"

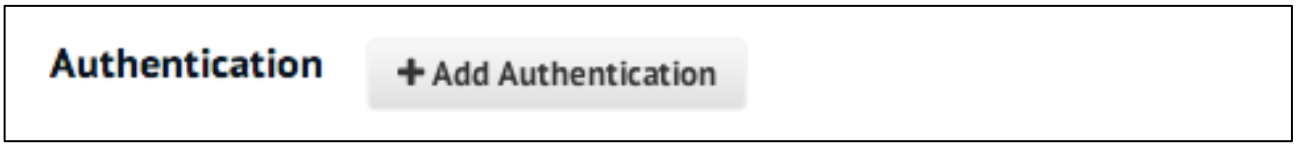


2. In the **Destination** field, type your base URL and then the path to your scripted processor. If you recall, our scripted processor had a **Path** value of "**space_currency**". When you enter the URL, make sure you use the ".do" extension to your processor.



©2014 ServiceNow, Inc. All rights reserved.

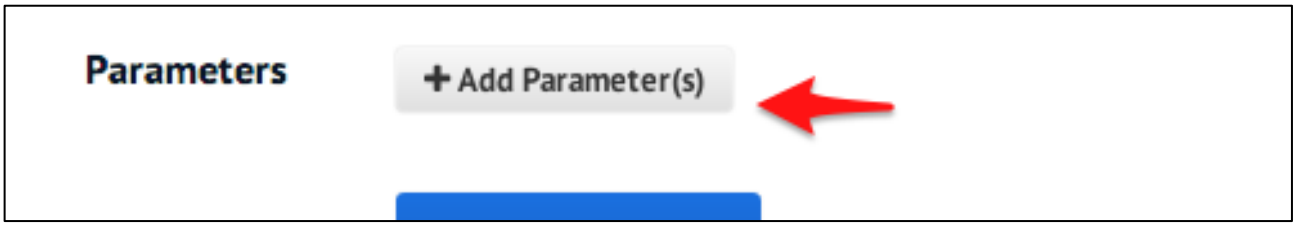
3. Click the **Add Authentication** button.



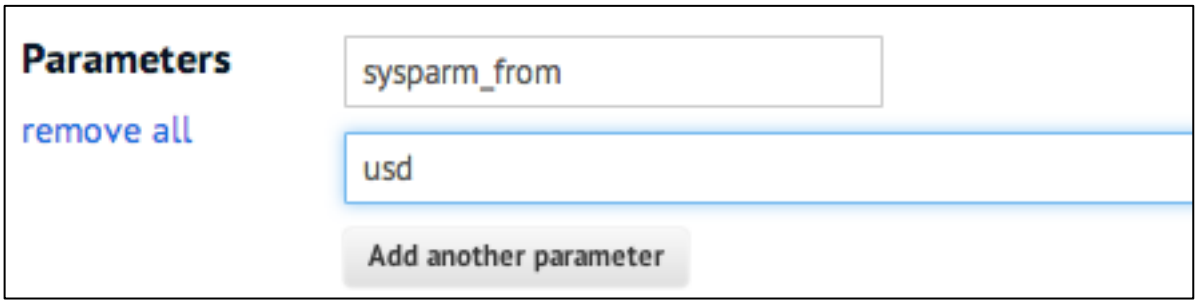
4. Input your admin credentials.



5. Click the **Add Parameters** button to start the process of adding our sysparm_from and sysparm_to parameters.



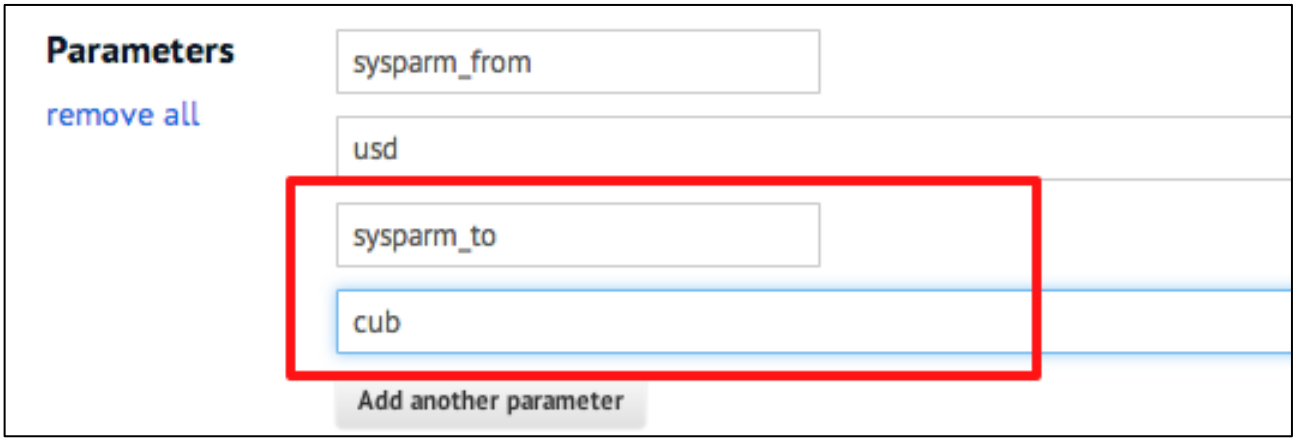
6. For the first set of parameters set the **name** field to "**sysparm_from**" and the value to be **"usd**" for the US Dollar.



7. Now click the **Add another parameter** button

©2014 ServiceNow, Inc. All rights reserved.

8. This time set the **name** field to "**sysparm_to**" and the value to be "**cub**" for Battlestar Galactica's Cubit system.



9. Click the **Launch Request** button to submit the request to your processor

10. You should get a response similar to the following with an exchange multiplier of 1.27:



©2014 ServiceNow, Inc. All rights reserved.

Report your Completion of this Lab Exercise

1. Navigate to **Lab Management > Report Lab Progress**.

2. Enter your progress status

©2014 ServiceNow, Inc. All rights reserved.

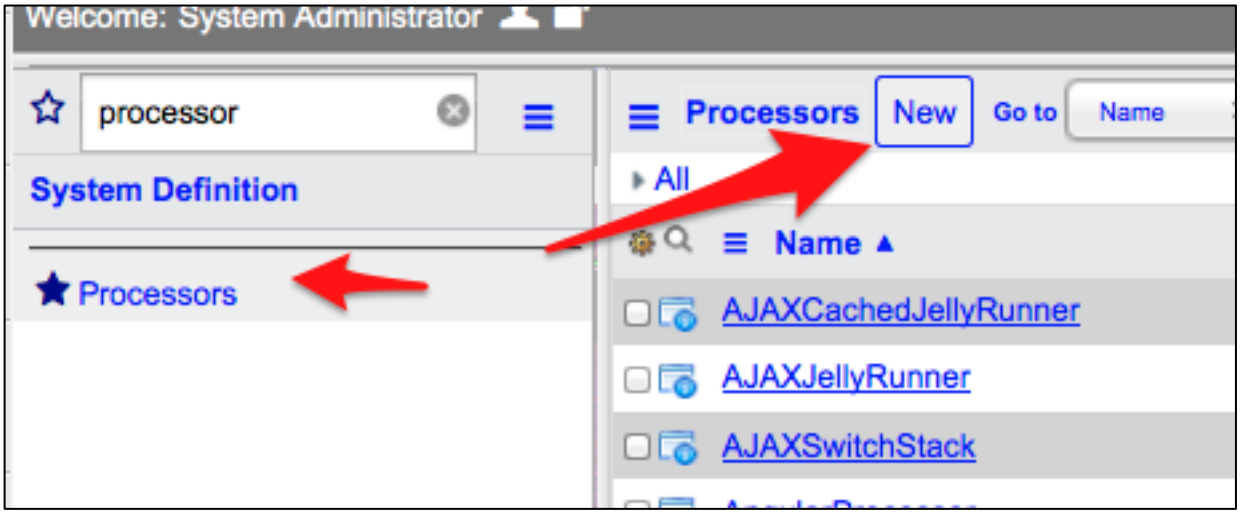
Lab Goal

In this exercise, we will create a Parameter processor that will return the immediate schema of the current table selected in the URL. To activate the processor, you simply have to add a META parameter to your URL.

Lab 5
Parameter Processors

Create the Scripted Path Processor

1. Navigate to **System Definition > Processors** and click the **New** button to start a new processor record.

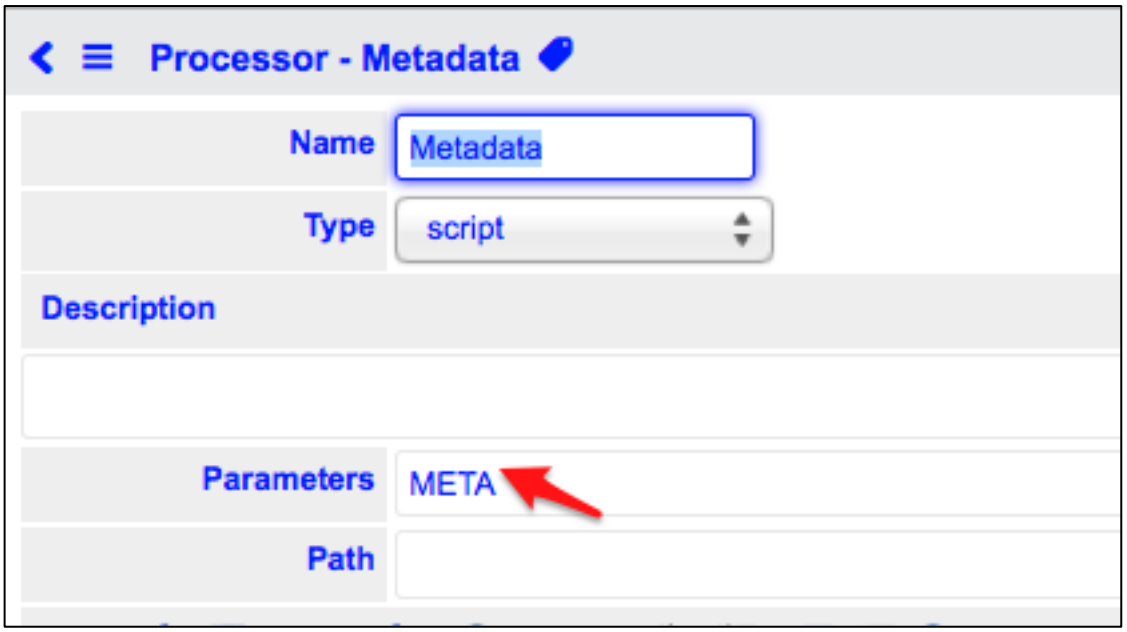


©2014 ServiceNow, Inc. All rights reserved.

2. Set the **Name** field to be "**Metadata**".

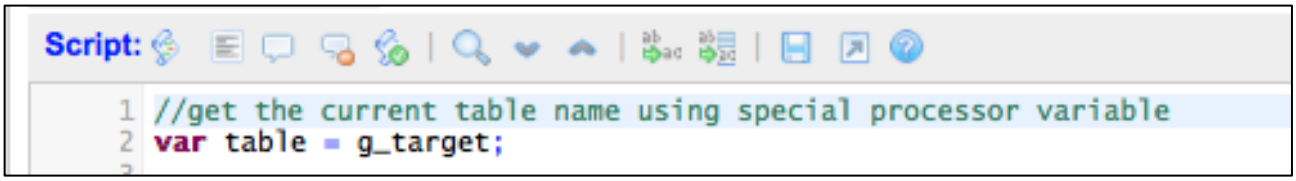
3. Set the **Type** field to be "**script**".

4. Set the **Parameter** field to be "**META**". This will tell the processor to trigger any time the string "META" is a parameter to any table in the instance.



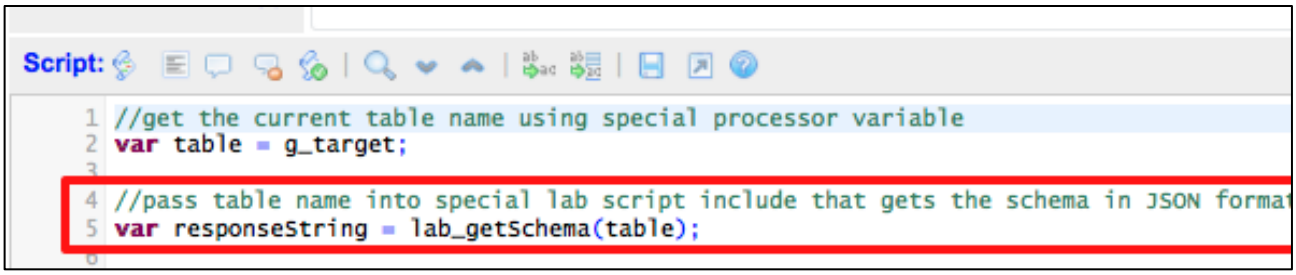
Develop the script that implements the processor

1. Start the script by using the special g_target variable available to parameter processors. This variable tells us what table is currently being accessed in the system.

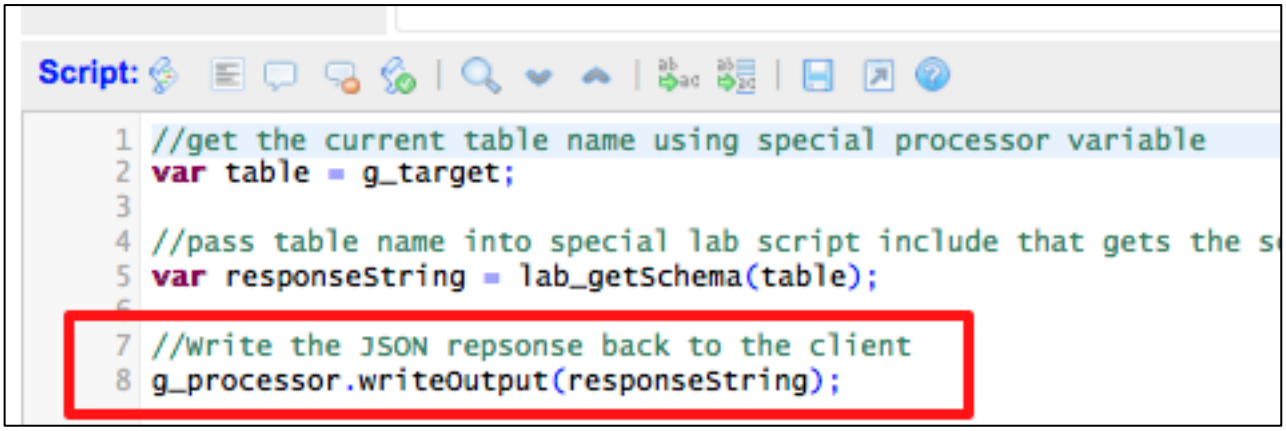


©2014 ServiceNow, Inc. All rights reserved.

2. To make things easy, we have already created and included a script include function that generates a given table's schema and returns that schema in JSON format. The name of the function is **lab_getSchema(table)**. Call this function and pass the table information in to it.



3. Write the output back to the client by calling the g_processor.writeOutput() method.



4. You can download this entire script by navigating to **Integration Lab 301 > Copy & Paste > Parameter Processor Script**.

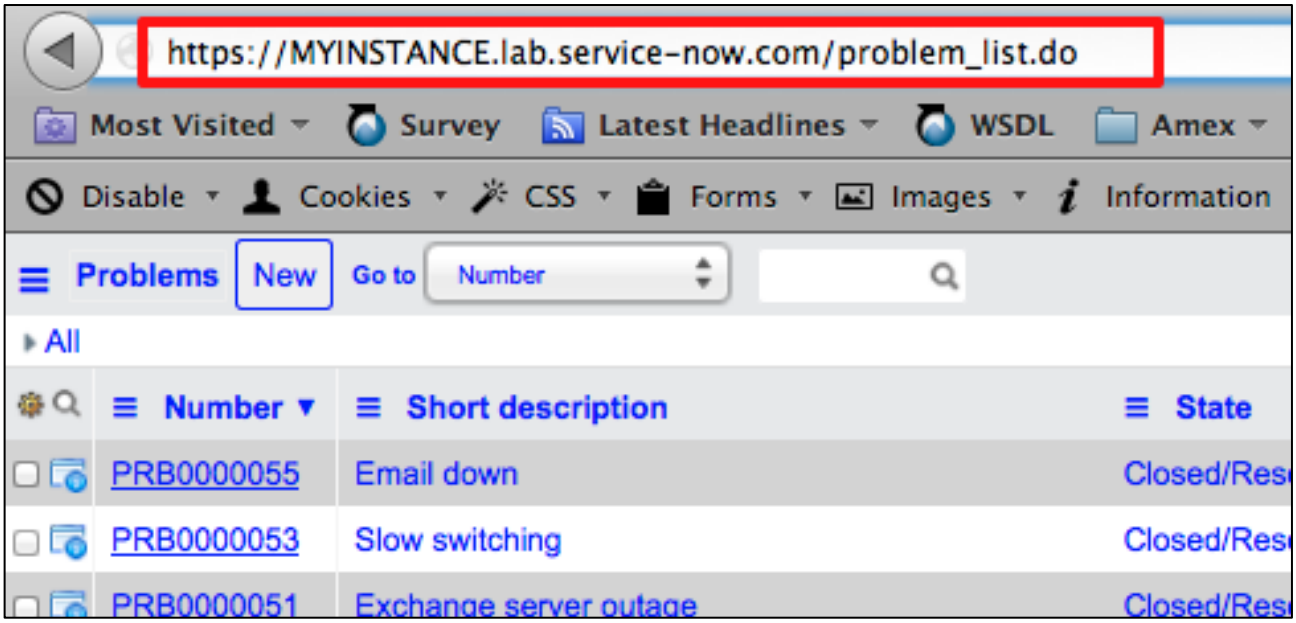
5. Click **Submit**.

6. NOTE: This script is for demo purposes only and is limited to the immediate table being referenced. If the table is a child table of another table, only the elements created on the child table are displayed. Come on, what do you expect for a lab?

©2014 ServiceNow, Inc. All rights reserved.

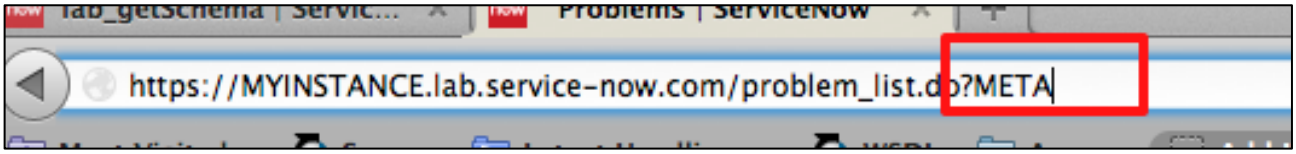
Check out the processor

1. Open up a new browser window and browse to your instance URL and go to the path of "**/problem_list.do**".



2. You should get the **Problem** table with a list of the problem records.

3. Now, modify the URL by adding the **META** url parameter at the end. You will have to tell the browser that it is a parameter by making sure there is a "**?**" character between the URL and the parameter like the following:



©2014 ServiceNow, Inc. All rights reserved.

4. You should then see a JSON representation of the **Problem** table schema.


```
[{"column_label":"Known
error","column_name":"known_error","default_value":"","display":"false",
lse"},{"column_label":"Problem
state","column_name":"problem_state","default_value":"1","display":"fals
ger"},{"column_label":"Related
Incidents","column_name":"related_incidents","default_value":"0","displa
e":"Integer"},{"column_label":"Change
request","column_name":"rfc","default_value":"","display":"false","manda
"Reference"},{"column_label":"Sys
ID","column_name":"sys_id","default_value":"","display":"false","mandat
ID"},{"column_label":"Workaround","column_name":"work_around","default_
reference":"","type":"Journal Input"}]
```


5. NOTE: Since this is a Parameter Processor, you can do this to any table in the system, simply by adding the "**META**" parameter to the URL.

Report your Completion of this Lab Exercise

1. Navigate to **Lab Management > Report Lab Progress**.

2. Enter your progress status

®2014 ServiceNow, Inc. All rights reserved.