knowledge14®

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

Lab Guide

Integrations 101:

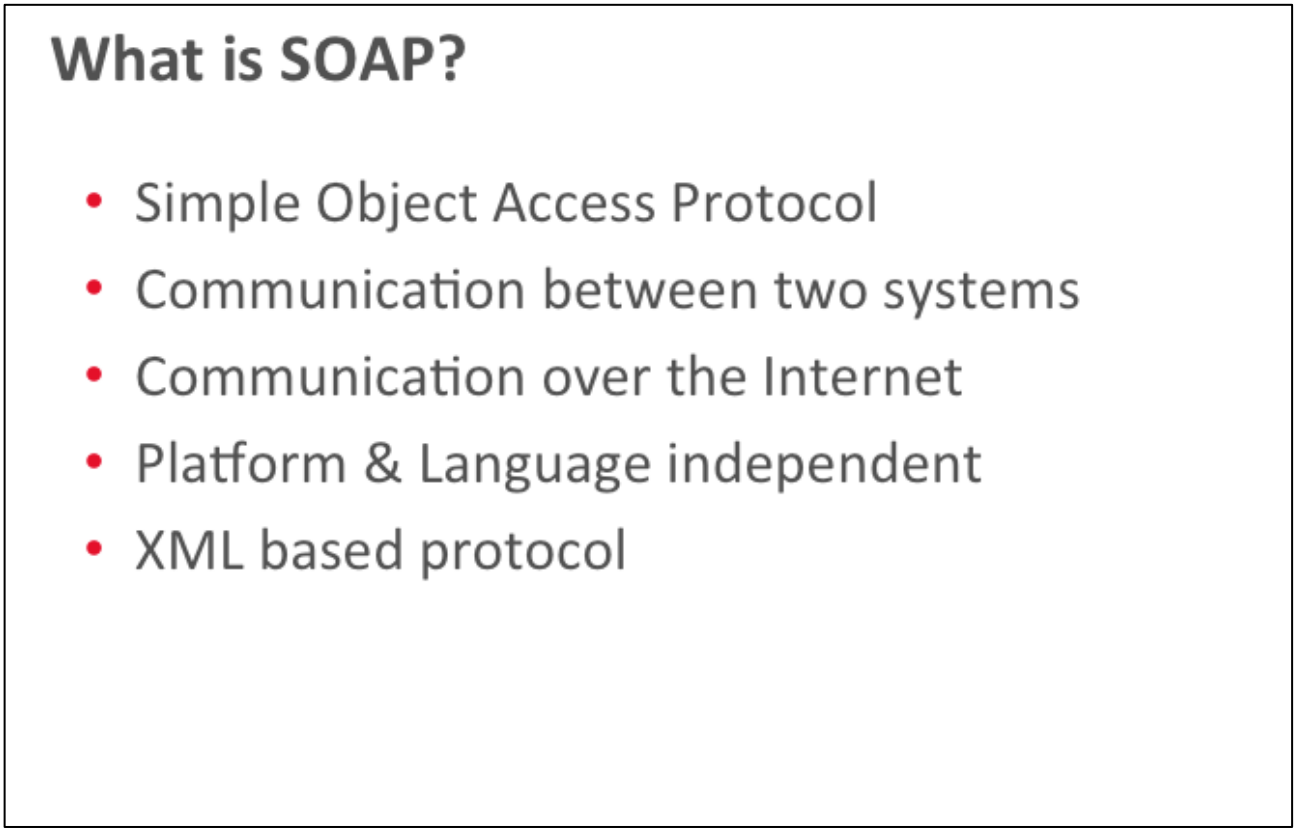
Mastering Outbound SOAP Web Services

John Andersen & Chris Maloy

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

Login / Passwords:

admin / Knowledge14
itil / Knowledge14
employee / Knowledge14



The soap protocol was designed to allow communication between multiple, and unrelated systems. It was primarily designed for communication over a network such as the Internet.

When working with soap, you do not have to worry about what language or platform if third-party service is utilizing. As long as it can speak the soap protocol, you can interact with that third-party system without any knowledge of its implementation details.

The soap protocol is based on the XML markup language. It uses a series of XML documents to both describe the request, and provide a response.

There are four main parts to a SOAP message transportation.

®2014 ServiceNow, Inc. All rights reserved.

SOAP Transport

- Endpoint
- Action
- Headers/Attributes
- Envelope



The first is the endpoint. This is the URL or address where the soap request will be delivered. It must be able to reach that end point over the network in order for the communication to succeed.

The SOAP action is a string that is often optional with the soap web service. However, some services do require this to be set. The string is often descriptive of what action should be performed by the web service.

The headers and attributes of a message are found within the HTTP request. They contain information regarding the protocol being used as well as other custom information such as session IDs, authentication tokens, etc.

The envelope is the actual document that describes the details of the SOAP request. The envelope is written in XML format.

®2014 ServiceNow, Inc. All rights reserved.

SOAP Envelope

- Main Document
- Header (optional)
  - containing processing instructions for the envelope
- Body
  - contains the actual message to be delivered
- Fault element (optional)
  - providers error information



The SOAP envelope has a header and a body similar to an HTML webpage. The header and body details are dictated by the web service and explained in a corresponding WSDL document.

The header will often contain processing instructions for the envelope, but is not used in all web services. The web service will document any need for a header.

The body is the main part of the envelope, and is required for SOAP requests. It contains the actual message or request that is to be delivered to the web service.

A third and final part of a SOAP envelope is the optional fault element. This element is often present if there was an error processing the web service request. The web service is responsible for populating this element with descriptive information that helps the consumer understand what went wrong, and/or what they should do next.

©2014 ServiceNow, Inc. All rights reserved.

SOAP WSDL

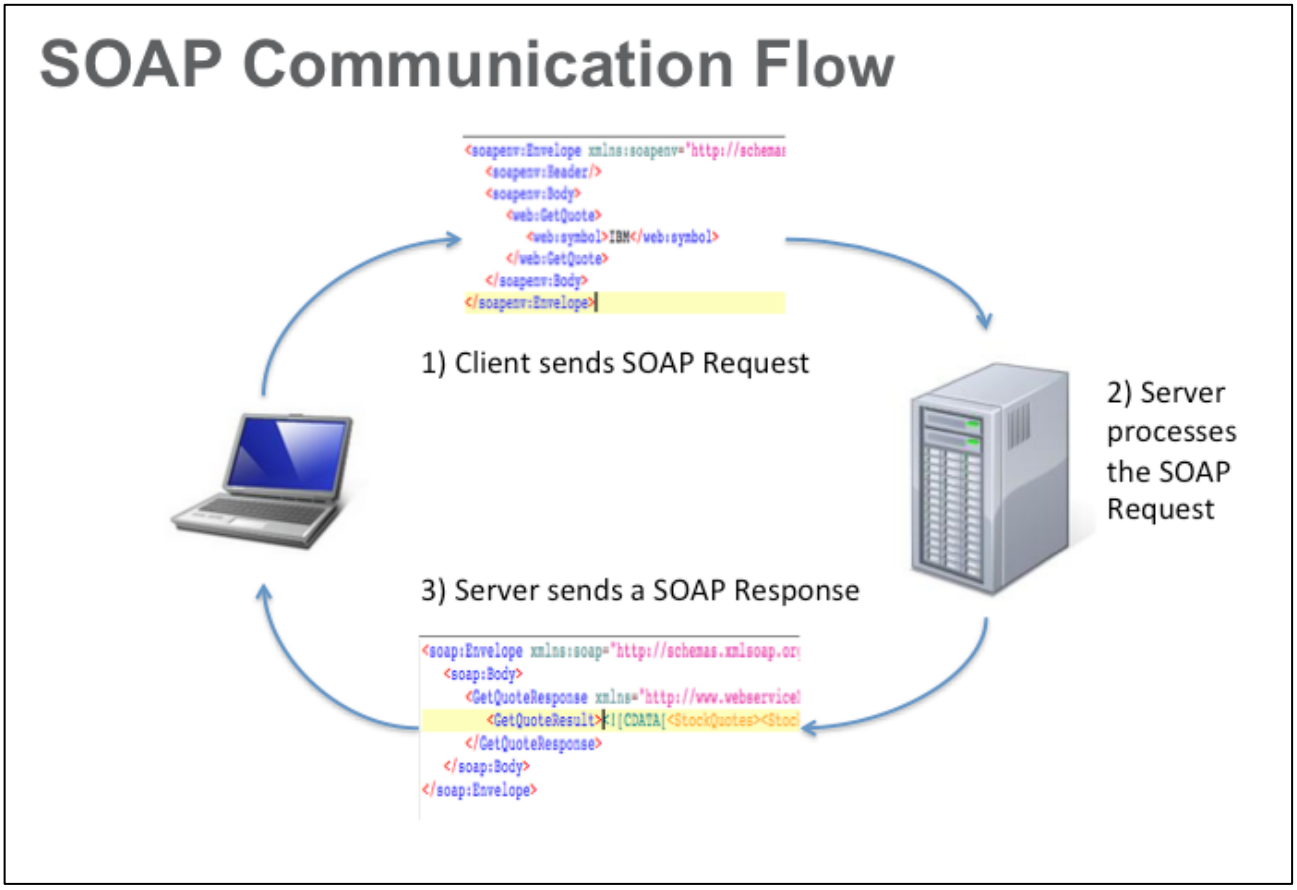
- WSDL
  - Catalog of available functions
  - Templates of requests



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.



There are three main parts of the standard SOAP communication flow.

The first step consists of a client that generates an XML envelope document and sends it to the service as an SOAP request.

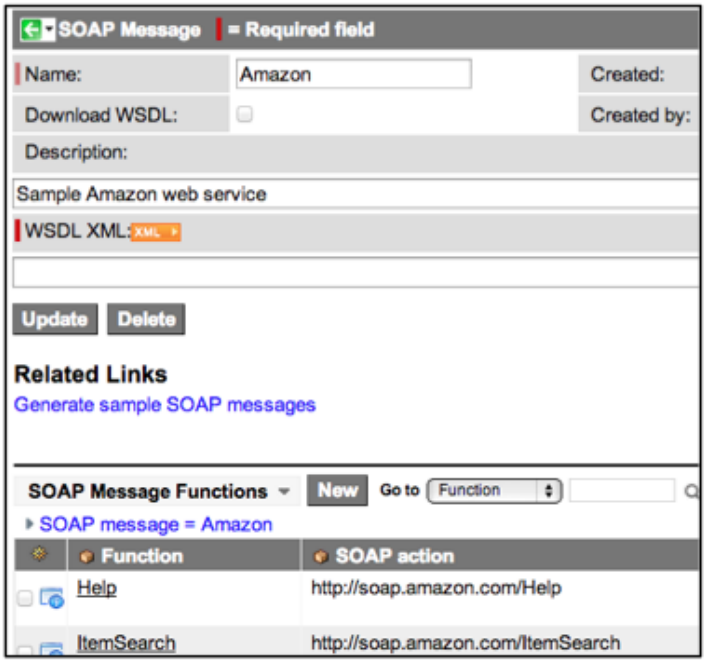
Step 2 takes place when a server receives the SOAP request and processes that request in the manner in which it was coded.

Finally, the server sends a SOAP response back to the client with an answer or response to the request that was submitted.

©2014 ServiceNow, Inc. All rights reserved.

Demo: SOAP Message UI

- WSDL
  - Download
  - Paste
- Generate



ServiceNow provides an easy to use web interface that allows you to build SOAP requests from a WSDL document.

With this tool you can easily specify the URL to a WSDL document and import it to generate the SOAP functions that are available in the web service. Alternatively, you can take a WSDL document and paste it directly into the tool to generate those functions if the WSDL is not available via a reachable URL.

Once you have the WSDL URL or WSDL document associated with your SOAP message record, you can click the generate sample SOAP messages link on the form to have ServiceNow generate templates for all of the available services provided by that SOAP web service.

©2014 ServiceNow, Inc. All rights reserved.

Demo: SOAP Message Function UI

- Lock
- Security
- SOAP
  - Action
  - Endpoint
- Envelope
- Variables
- Test



Once you have generated some SOAP functions from the SOAP message record, you can open up a function record to view the details on the template for the request.

The Lock field can be checked to ensure that future regeneration of SOAP functions do not overwrite the changes that you make to this function template.

ServiceNow supports the use of either basic authentication, or WS-security with outbound SOAP requests. You can check those checkboxes to see how to set up those types of security.

You can customize the SOAP action and/or endpoint manually by changing the text. With the SOAP endpoint, it may be necessary to click the padlock icon to change the link into a text box that can be edited.

©2014 ServiceNow, Inc. All rights reserved.

The envelope document contains the XML template for the request. ServiceNow automatically detects possible elements where data will reside and creates variable strings in the form of ${VARIABLENAME}. These can be used by scripts to make the SOAP request dynamic.

When using variables, you can create test values for those variables in the OPEN message parameters section of the form.

Once you have the template set up the way that you want it, you can click the "test" link to submit the request to the web service. The tool will show you the response that is provided by the web service once the communication flow has finished.

©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

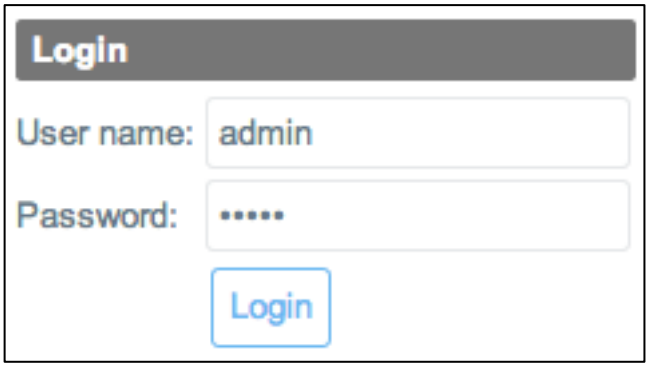
Get a good understanding of how the SOAP Message GUI setup utility works within ServiceNow.

Lab 1
SOAP Message

Log on to Your Training Instance

1. Your instance URL is on the cover page.

2. Login using the **admin** credentials provided on the cover.



Create a new SOAP Message Record

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

2. Click **New**.

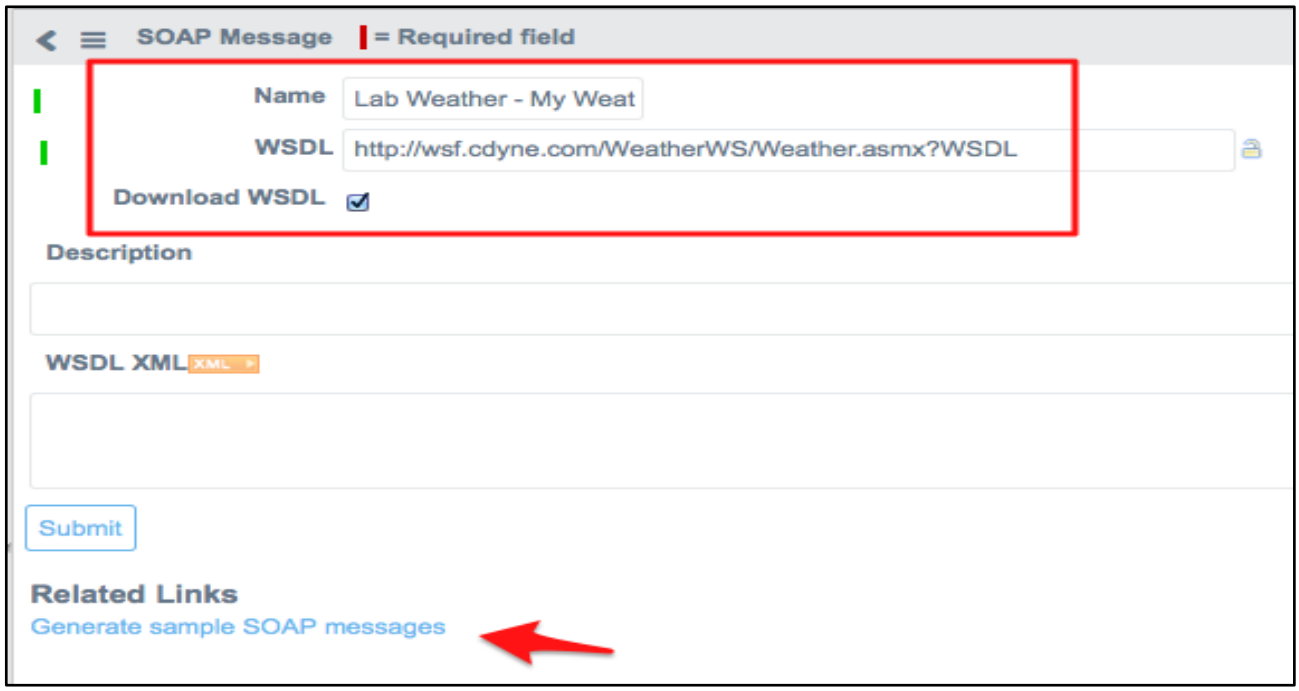
3. Name the record: **Lab Weather – My Weather.**

©2014 ServiceNow, Inc. All rights reserved.

4. Make sure the **Download WSDL** checkbox is checked.

5. Enter the following **WSDL** URL (You may have to click the padlock icon first): http://wsf.cdyne.com/WeatherWS/Weather.asmx?WSDL.

6. Click **Generate the sample SOAP messages.**



7. If there is a Warning, Click **OK**.

Set up a SOAP Message Function

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

2. Open the **Lab Weather – My Weather** SOAP Message record.

©2014 ServiceNow, Inc. All rights reserved.

3. Open the following SOAP function: **WeatherSoap.GetCityWeatherByZIP.**



4. NOTE: Notice the variable name from the auto generated SOAP Envelope. SOAP function variables are in the format of **${VAR_NAME}**.



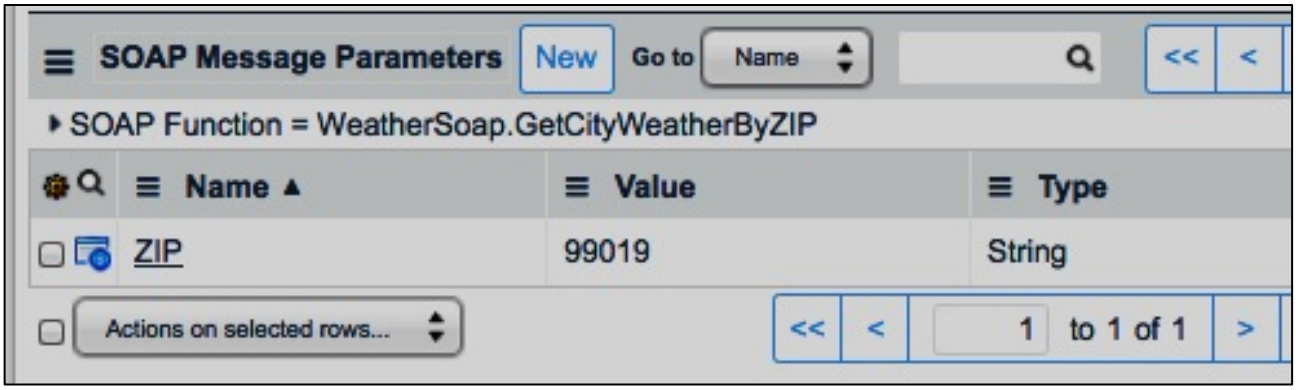
5. Scroll down the form to the **SOAP Message Parameters** section. Click the **New** button and set the following fields:

   Name: **ZIP**
   Value: **99019**
   Type: **String**

©2014 ServiceNow, Inc. All rights reserved.

6. Click **Submit.**

7. Verify that the new variable and value are listed.



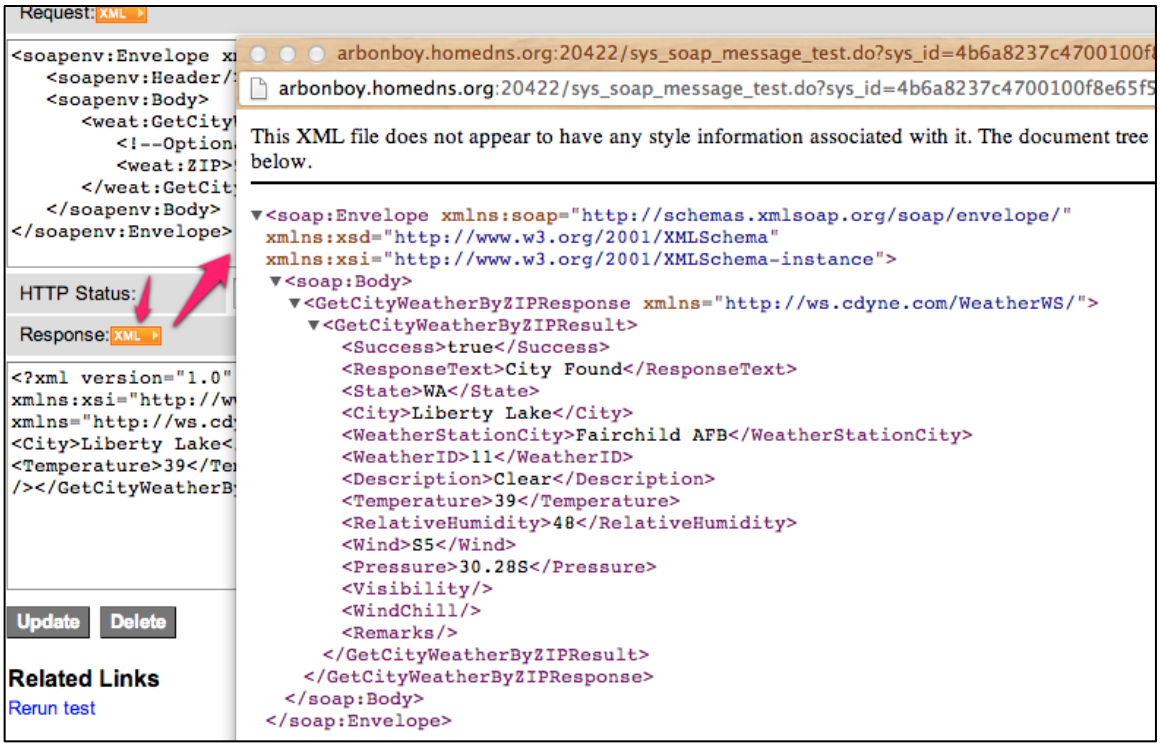
8. Click the **Test** link to test the web service.

9. You should see a screen that lists the actual **Request** that was sent to the server. Notice that the ${ZIP} string no longer exists in the XML payload, but has been replaced by the value you set in the prior step.



©2014 ServiceNow, Inc. All rights reserved.

10. You can click the XML icon on the **Response** field to get a formatted view of the SOAP Response.



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.



Once you have set up your soap message function templates, you can leverage them within ServiceNow scripts.

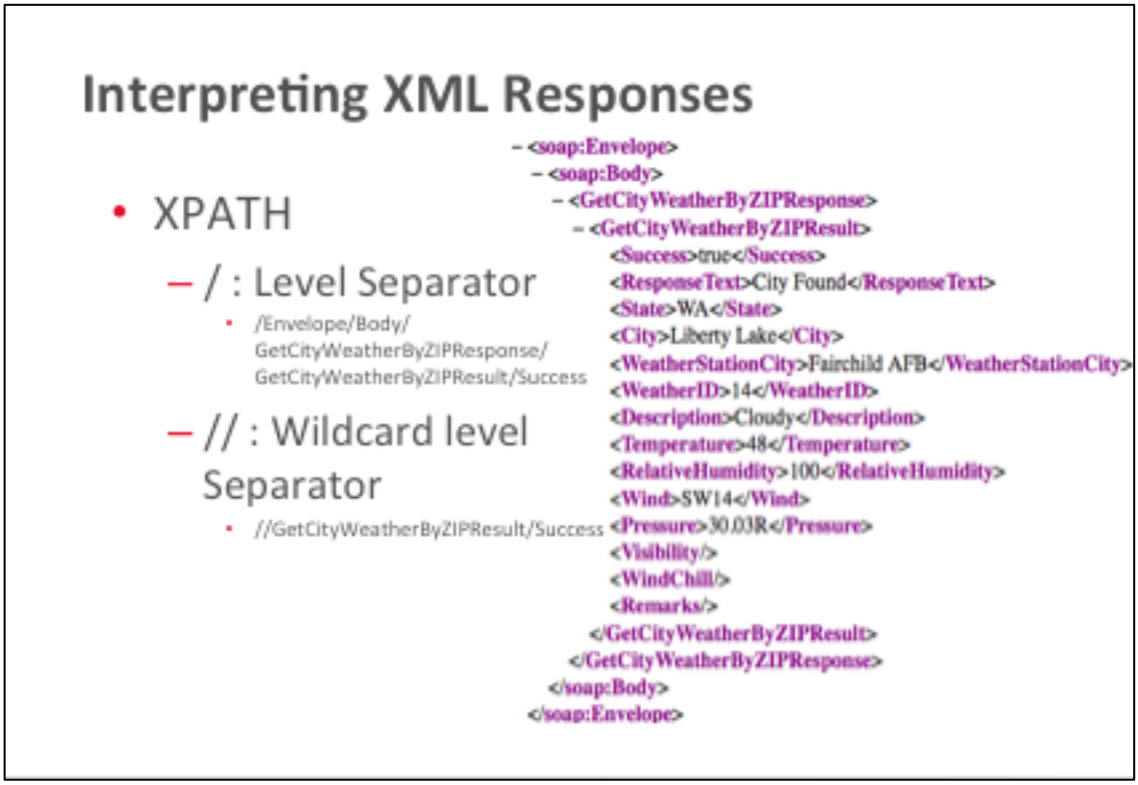
Some of the most common scripting areas where SOAP messages are called from include business rules, script includes, workflows, and other server-side scripts.

There are a few script include libraries that support a user in sending out SOAP requests.

The library, SOAPMessage, is the API around the GUI tool that we have just demonstrated. It can be used to set and point values, set variables, submit the request, etc.

The SOAPEnvelope and SOAPRequest libraries are leveraged under the covers by the SOAP Message tool but can also be used by an administrator in creating customs SOAP messages, bypassing the GUI SOAP Message utility, if needed.

©2014 ServiceNow, Inc. All rights reserved.



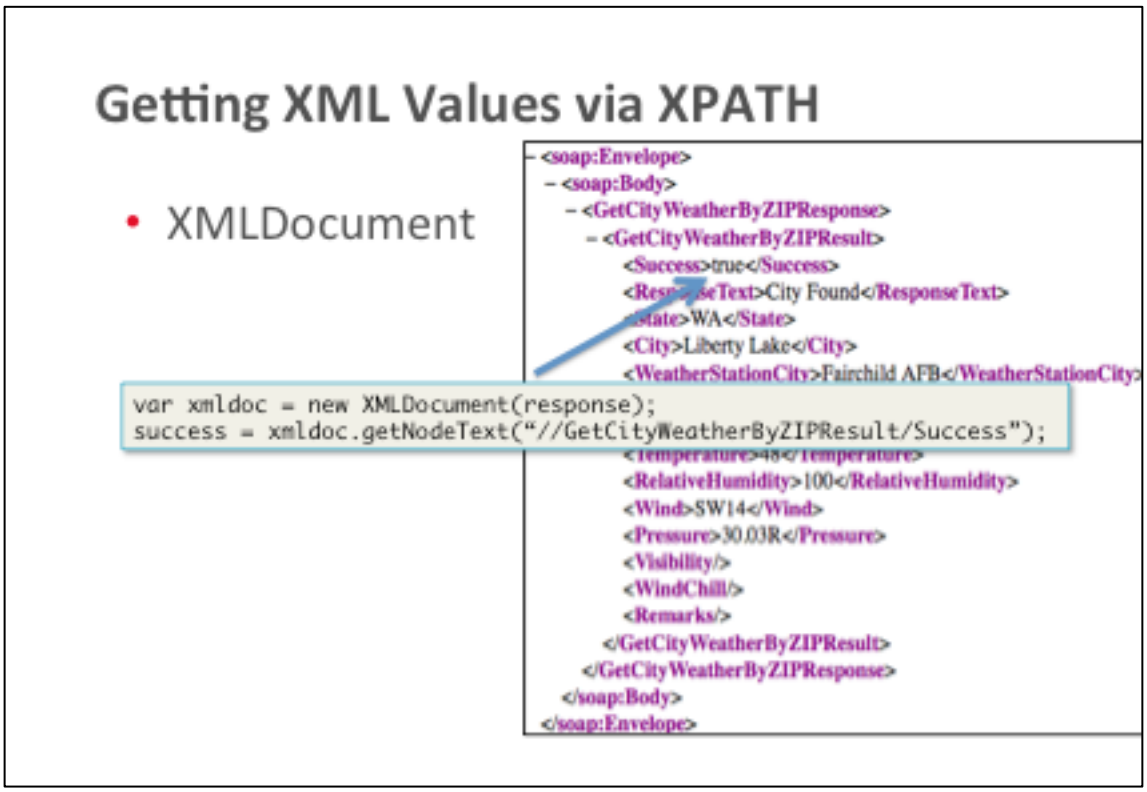
XPATH is a standard mechanism used to parse XML documents. ServiceNow supports the use of XPATH in interpreting the SOAP responses.

While XPATH contains a number of syntax and functions available to it to make it a very powerful tool in interpreting XML documents, the most basic and primary elements that you will typically have to be familiar with is the "/" character. This character is a level separator similar to Linux/UNIX paths on a file system.

The other common character that is used is a "//" character. This character denotes a relative path, rather than a strict path.

Another widely used character is the "*" character. It is denoted as a wildcard character in a string.

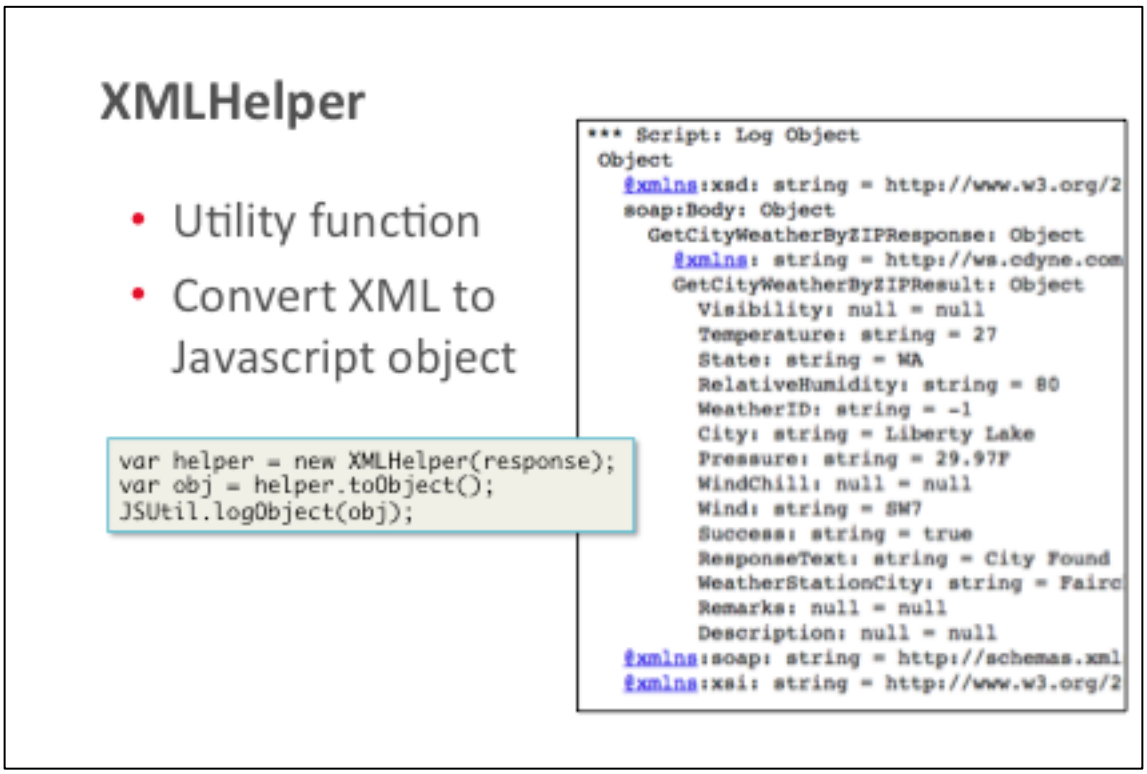
©2014 ServiceNow, Inc. All rights reserved.



ServiceNow has a script include that facilitates operations that you can perform on XML documents. This library is called, "XMLDocument".

With this library, you can load an XML string and perform operations on it. A common operation that you will perform is the "getNodeText" function that takes an XPATH string in and attempts to parse the XML document with the XPATH to find the element that you are interested in and return the contents of that element node.

©2014 ServiceNow, Inc. All rights reserved.



Another handy function library that is provided by ServiceNow is the"XMLHelper" library. This library has a number of functions, however, a very commonly used function is the one that will take an XML string and convert it to a JavaScript object.

®2014 ServiceNow, Inc. All rights reserved.

Using XMLHelper to get Element Value

```
*** Script: Log Object
 Object
   @xmlns:xsd: string = http://www.w3.org/2
   soap:Body: Object
     GetCityWeatherByZIPResponse: Object
       @xmlns: string = http://ws.cdyne.com
       GetCityWeatherByZIPResult: Object
         Visibility: null = null
         Temperature: string = 27
         State: string = WA
         RelativeHumidity: string = 80
         WeatherStationCity: string = Fairc
         Remarks: null = null
         Description: null = null
   @xmlns:soap: string = http://schemas.xml
   @xmlns:xsi: string = http://www.w3.org/2
```

```
var helper = new XMLHelper(response);
var obj = helper.toObject();
var weatherResponse = obj['soap:Body']['GetCityWeatherByZIPResponse'];
var state = weatherResponse['GetCityWeatherByZIPResult']['State'];
gs.log(state);
```

XMLHelper allows you to access the parts of the XML document as if the various elements were member variables to an object. Each level down the XML document, or represented by member variables that are, themselves, new objects.

©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

Learn how to leverage SOAP Message through Server-side Scripts.

In this exercise, you will create a business rule on the "ZIPCodeWeather" table to update the weather for a ZIPCode any time it is created or modified.

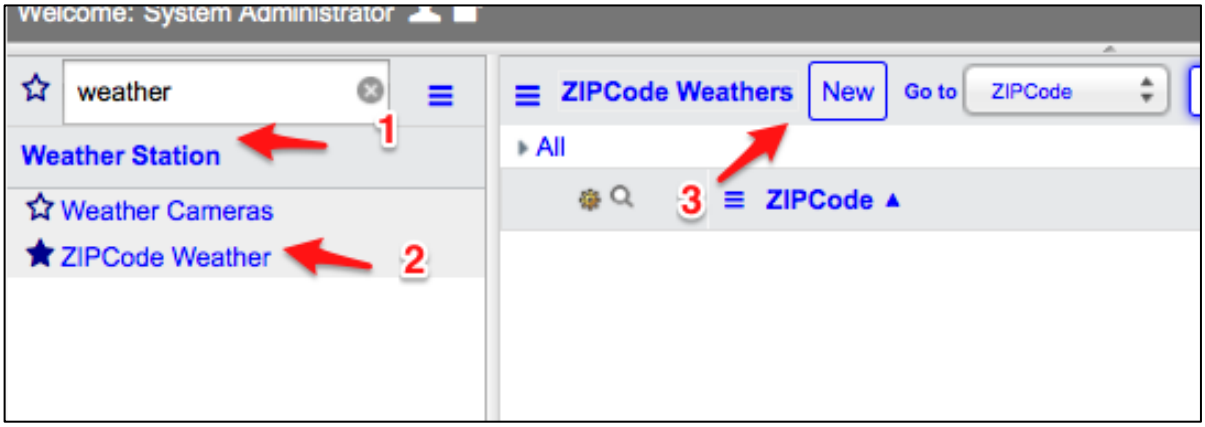
Lab 2 Scripted SOAP Message

Browse to the ZIPCode Weather module

1. Browse to the **Weather Station** application in ServiceNow.

2. Click on the **ZIPCode Weather** module.

3. Click the New button to create a new record.

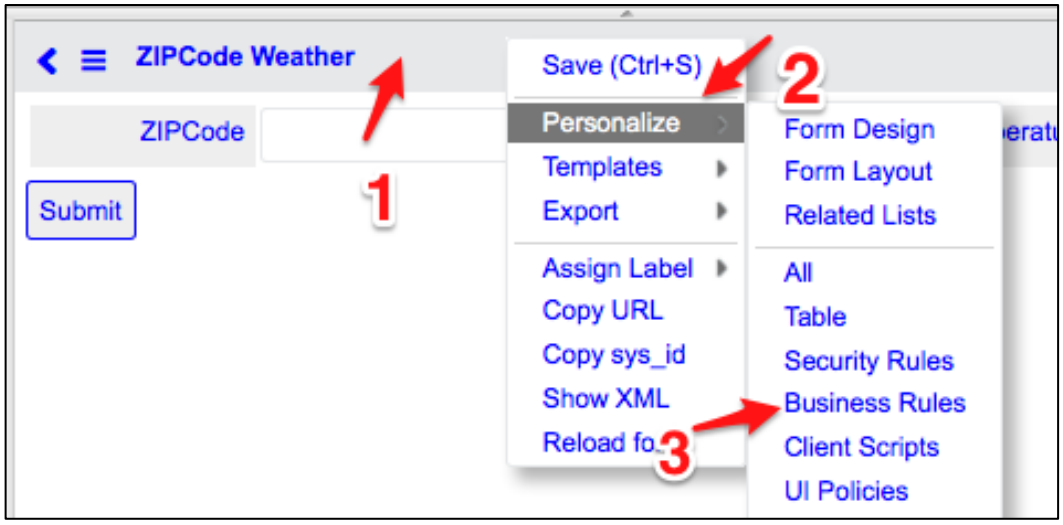


©2014 ServiceNow, Inc. All rights reserved.

Create a New Business Rule on the form

1. Right click on the **ZIPCode Weather** title bar.

2. Select **Personalize > Business Rules** in the popup menu.



3. Click **New** to create a new Business Rule.

Set up the Business Rule Settings

1. Fill out the business rule form with the following information.

Name	Get Current Temperature
Table	ZIPCode Weather
When	before
Insert	Checked
Update	Checked
Condition	Filter: ZIPCode : Changes

©2014 ServiceNow, Inc. All rights reserved.

2. Review the form as it should look like the following:



3. Click **Submit.**

©2014 ServiceNow, Inc. All rights reserved.

Browse to the SOAP function

1. Browse to **System Web Services > Outbound > SOAP Message.**

2. Select the **Lab Weather – My Weather** SOAP Message record.

3. In the **SOAP Message Functions** list, select the **WeatherSoap.GetCityWeatherByZIP** function.

Generate a sample SOAP script

1. Ensure that there is a “ZIP” variable setup in the SOAP envelope.

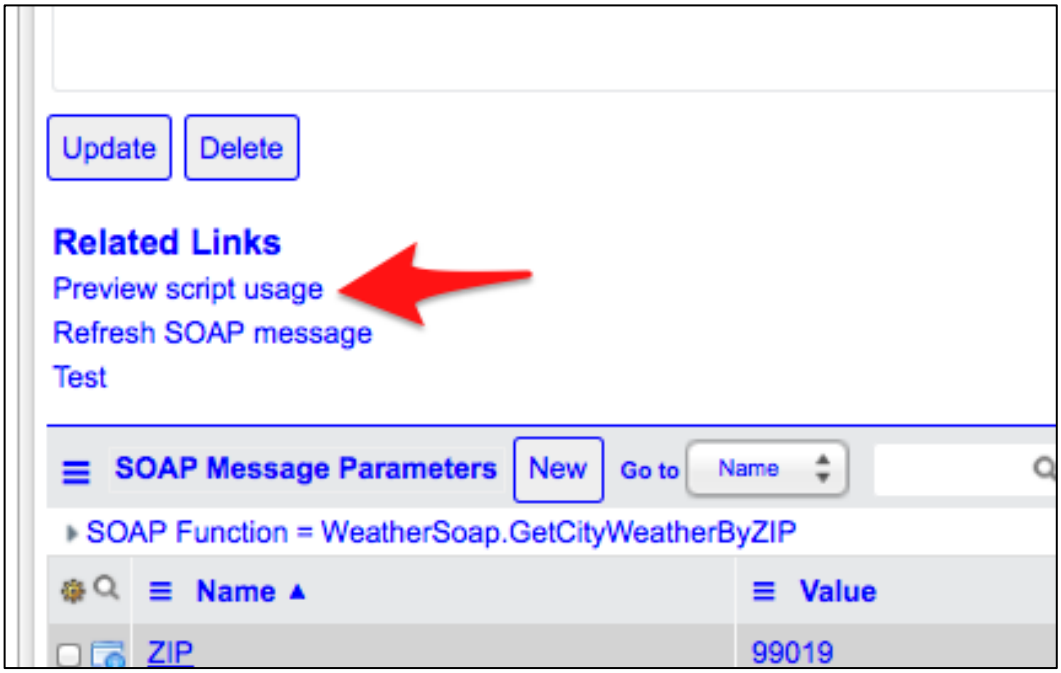


©2014 ServiceNow, Inc. All rights reserved.

2. Ensure that there is a **ZIP** variable created in the **SOAP Message Parameters** list.

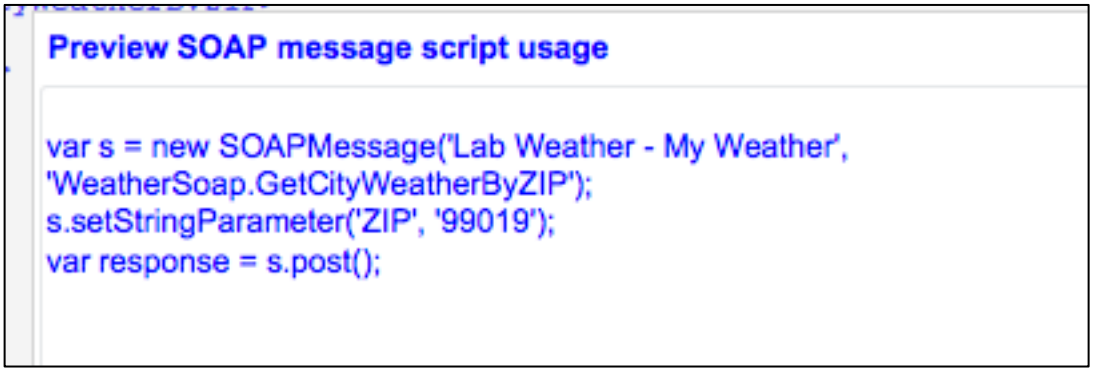


3. Click **Preview script usage.**



®2014 ServiceNow, Inc. All rights reserved.

4. Copy the code that is generated in the popup dialog.



5. Close the preview dialog window.

Paste and modify the script in the business rule

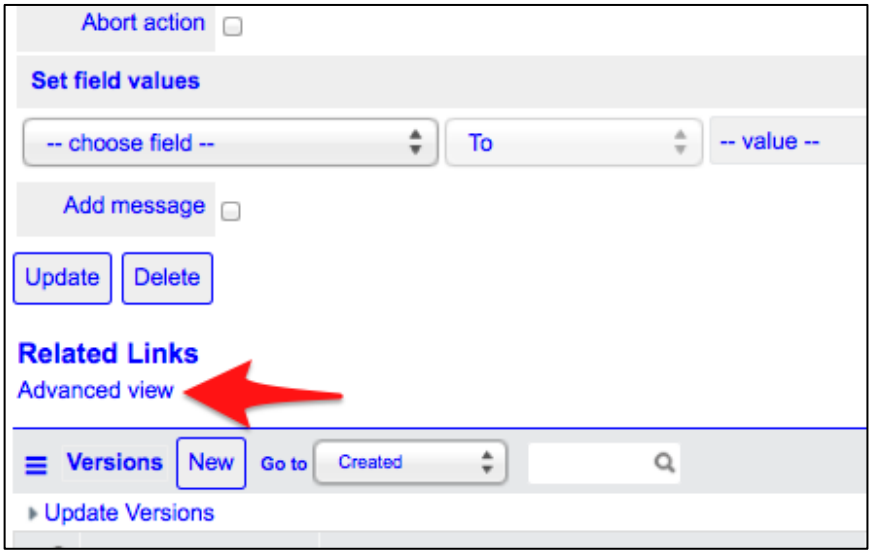
1. Navigate to **Weather Station > ZIPCode Weather.**

2. Right click on the list title bar and choose **Personalize > Business Rules** from the menu options.

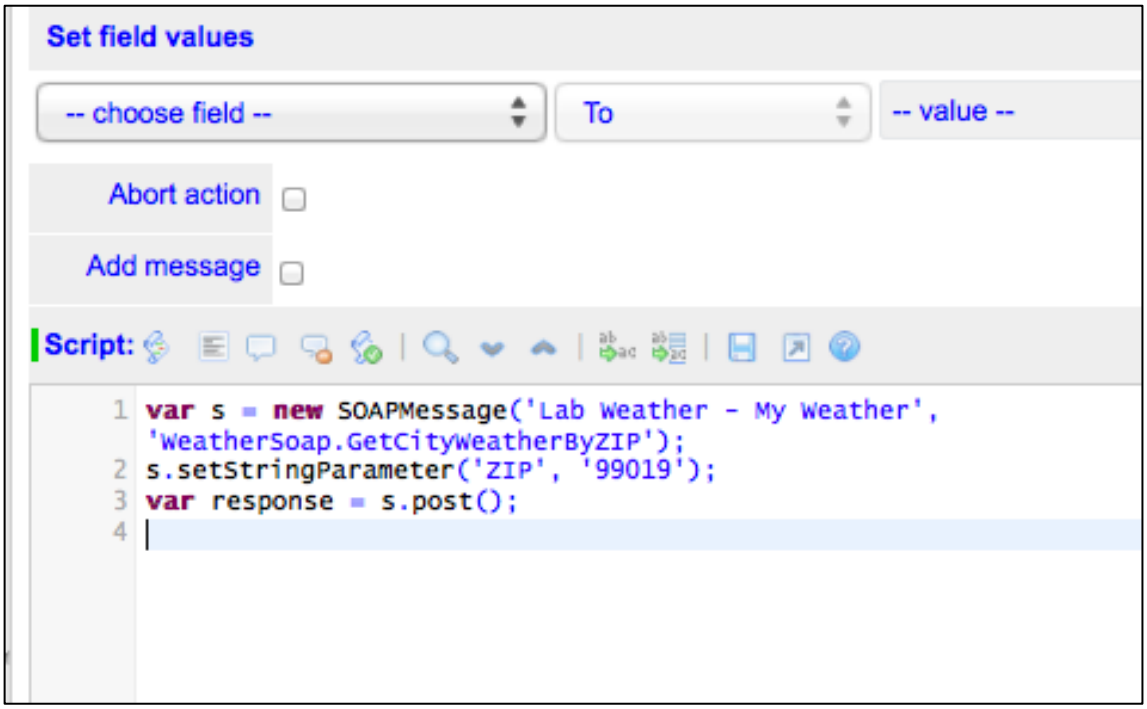
3. Open the **Get Current Temperature** business rule record.

©2014 ServiceNow, Inc. All rights reserved.

4. Click the **Advanced View** link on the business rule form.

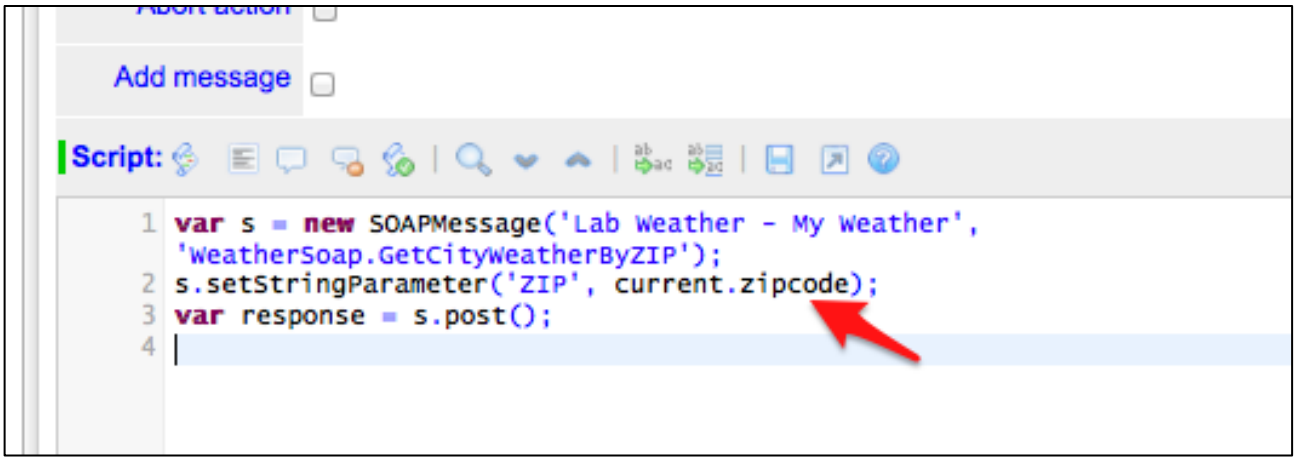


5. Once the form refreshes, paste your SOAP script in the “script” field.



©2014 ServiceNow, Inc. All rights reserved.

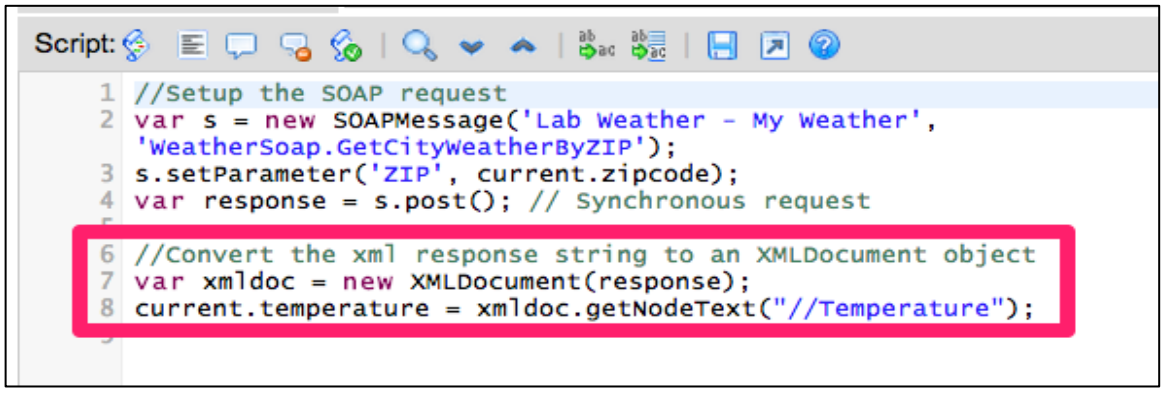
6. Instead of hard coding the zipcode in the **setStringParameter** function, change the **99019** to be the variable: **current.zipcode.**



Set the temperature value on the form

1. Add the following lines to your script.

```
//Convert the xml response string to an XMLDocument object
var xmldoc = new XMLDocument(response);
current.temperature = xmldoc.getNodeText("//Temperature");
```



2. Click **Update**.

©2014 ServiceNow, Inc. All rights reserved.

Test the scenario

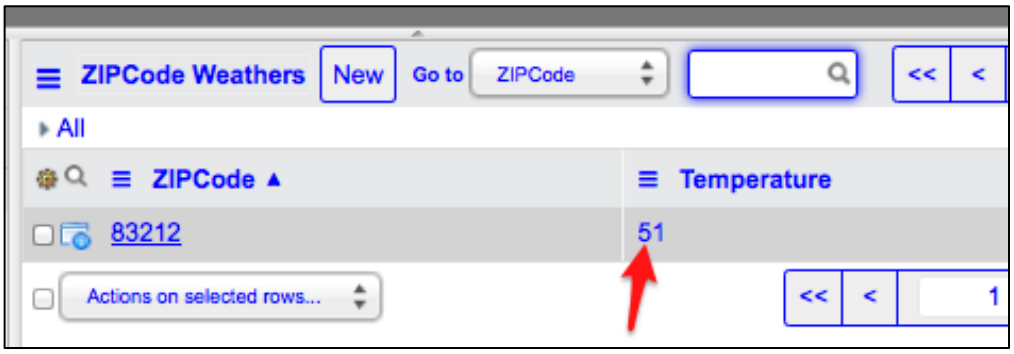
1. Navigate to **Weather Station > ZIPCode Weather.**

2. Click **New.**

3. Enter a **ZIPCode** value of **83212.**

4. Click **Submit.**

5. Verify that the temperature has been set on the record automatically.



Report your Completion of this Lab Exercise

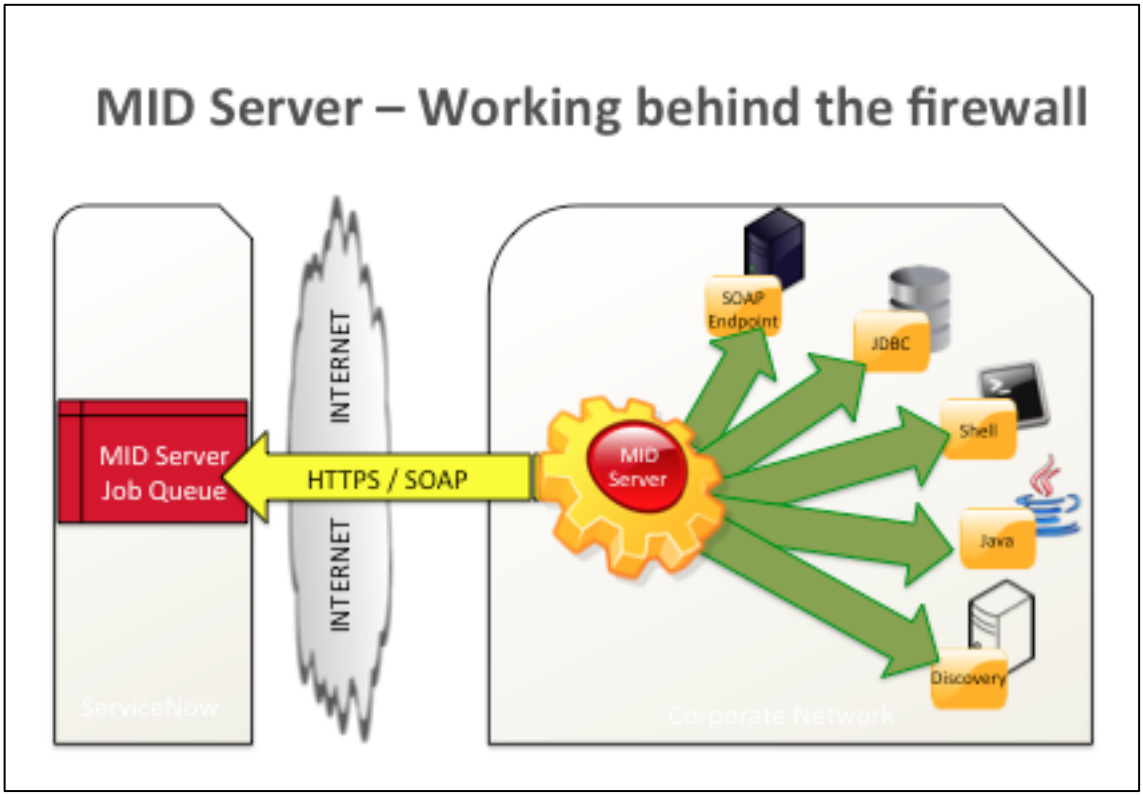
1. Navigate to **Lab Management > Report Lab Progress**.

2. Enter your progress status.

©2014 ServiceNow, Inc. All rights reserved.



®2014 ServiceNow, Inc. All rights reserved.



Oftentimes, and SOAP endpoint is behind a company's firewall and not accessible over the Internet where ServiceNow resides.

In order for us to get around this issue, we are able to use a ServiceNow MID Server.

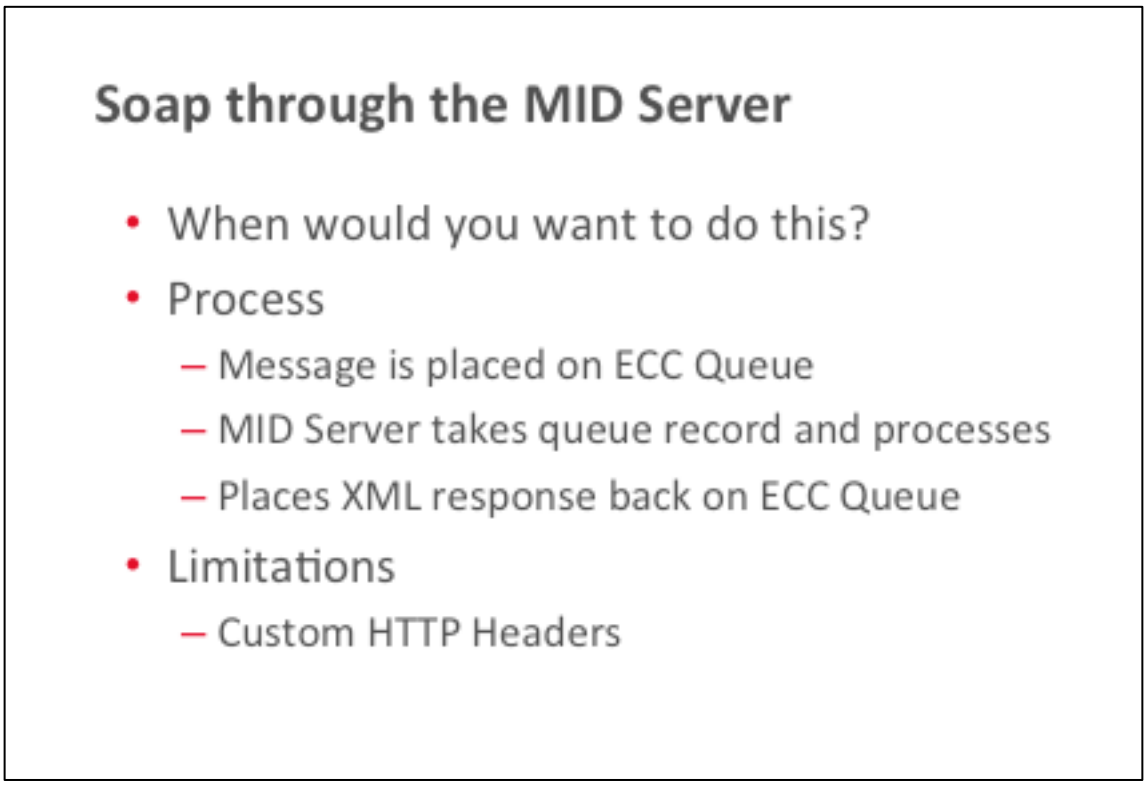
In MID Server is a small Java service that runs on a computer within your network. This Java program will sleep for about 10 to 15 seconds at a time and then wake up to make an outbound SOAP web service call to your ServiceNow instance to see if it has any work for it to do.

If there is work scheduled for it to perform, it will execute that work, and post the result back to ServiceNow through another outbound SOAP web service call.

In the case of requests, ServiceNow can place the SOAP request on the MID Server job queue to be picked up by the MID Server on its next query. When the MID Server is done executing the SOAP request, it will post the response back to the job queue as a response.

This job queue for the MID Server is called the ECC Queue.

©2014 ServiceNow, Inc. All rights reserved.



You typically only need to send a SOAP request through the MID Server when the endpoint is not accessible to your ServiceNow instance. In nearly every other circumstance, it is recommended to go directly to the SOAP endpoint as the communication flow will be faster because you do not have to wait for the MID-server to grab the request on its next polling schedule.

One current limitation with sending a SOAP request through the third party web service involves HTTP Response Headers. If the web service sends any important HTTP header information in its response, that information is not returned to the ECC queue. You can only handle HTTP response headers using direct SOAP calls.

©2014 ServiceNow, Inc. All rights reserved.



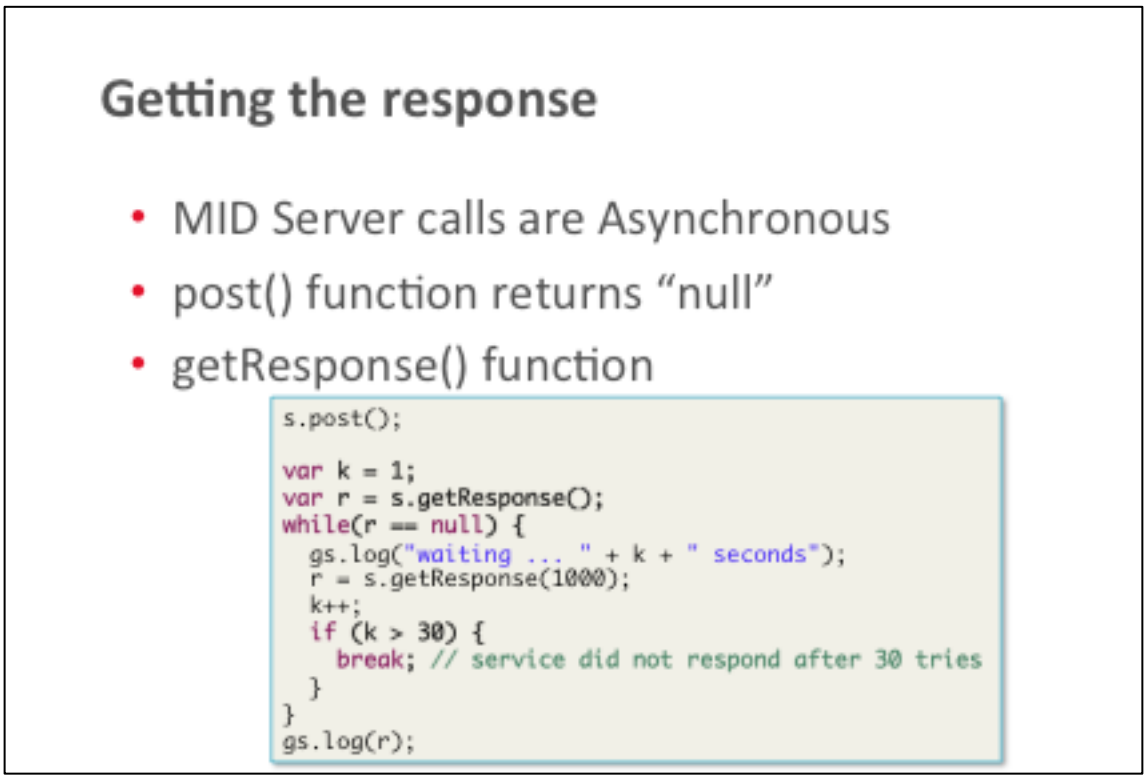
In order to set up the SOAP message record to go through a MID Server, you open up the SOAP message function record and set the "Use MID Server" field to the MID-Server that you wish to use.

©2014 ServiceNow, Inc. All rights reserved.



This can also be done programmatically using a function call, “setMIDServer()” within the SOAPMessage library.

®2014 ServiceNow, Inc. All rights reserved.



One important thing to note is that when you're making web service calls through the MID-server, those calls will be asynchronous.

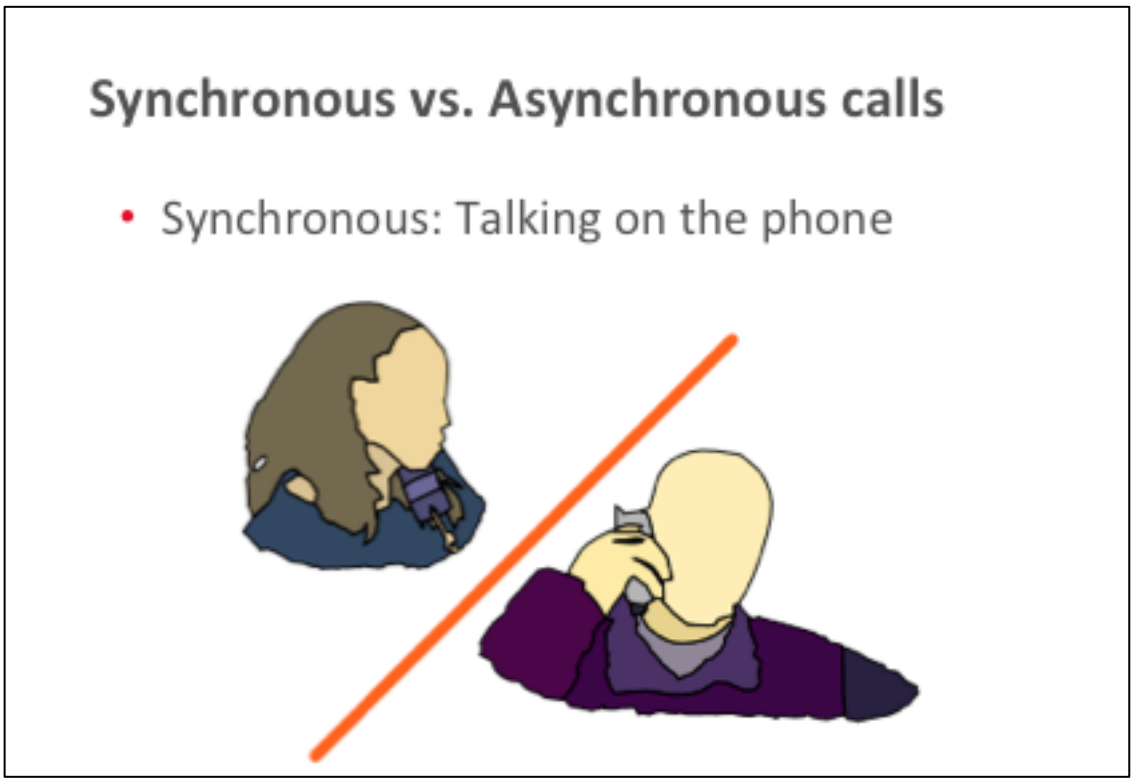
When submitting the request through a script, the "post()" function will return null instead of the actual response.

You must use a function called "getResponse()" that will query the ECC Queue after a set amount of time to see if the response record has been generated by the MID Server.

®2014 ServiceNow, Inc. All rights reserved.



®2014 ServiceNow, Inc. All rights reserved.



Since we mentioned that web service calls going through the MID Server are asynchronous, let us quickly discuss the difference between synchronous and asynchronous web service calls.

You can think of a synchronous call as something similar to a telephone call that you have with a friend.

When you ask your friend a question, you stand the line and expect an answer from your friend nearly immediately.

Similarly, when making synchronous web service call, a connection is established and a request is made. Within that same connection a response will immediately be given to the request and then the request loop is closed.

©2014 ServiceNow, Inc. All rights reserved.



With asynchronous communication you do not get a response immediately.

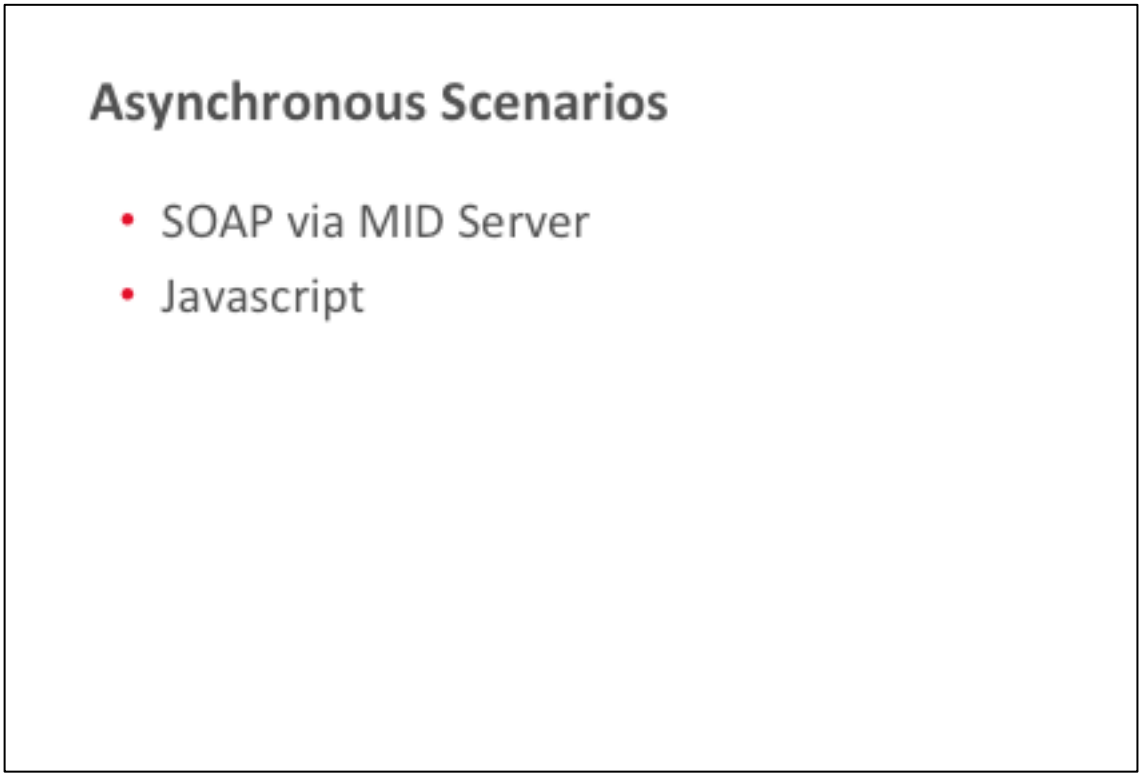
You can compare this to playing phone tag with your friend. You may call your friend's phone number only to have a service pick up the call. You ask your question, and then you hang up the phone and the connection is closed.

When your friend has some time, it will dial your phone and answer the question that you left for them. The answer is provided in a separate phone call.

With asynchronous web service calls, the same holds true. A connection is made to the Web service endpoint and a request is submitted. The client will get a simple response that the connection was successful. No other response information will be provided to the client.

Once the request is processed, a new connection will be created between the responder and the requester and the response will be submitted.

©2014 ServiceNow, Inc. All rights reserved.



There are two ways to set a SOAP request to be asynchronous within ServiceNow.

The first way, is to utilize the MID Server to send the SOAP request, as was previously demonstrated.

ServiceNow also allows you to send direct SOAP communication in asynchronous mode by indicating such in the script when the call is posted the the SOAPMessage "post()" function.

©2014 ServiceNow, Inc. All rights reserved.

Scripting the call to be asynchronous

- Pass 'true' to the POST method

```
var s = new SOAPMessage('Weather', 'GetCityWeatherByZIP');
s.setParameter('ZIP', '99019');
s.post(true);
```

- Use getResponse() method

```
var k = 1;
var r = s.getResponse();
while(r == null) {
  r = s.getResponse(1000);
  k++;
  if (k > 30) break; // no response after 30 tries
}
```

When submitting the post () function, simply pass a value of true in as the parameter to the function. This will tell the function to make this call asynchronously rather than synchronously.

You then get the response in the same method that you used to get a response from sending SOAP through the MID Server.

©2014 ServiceNow, Inc. All rights reserved.

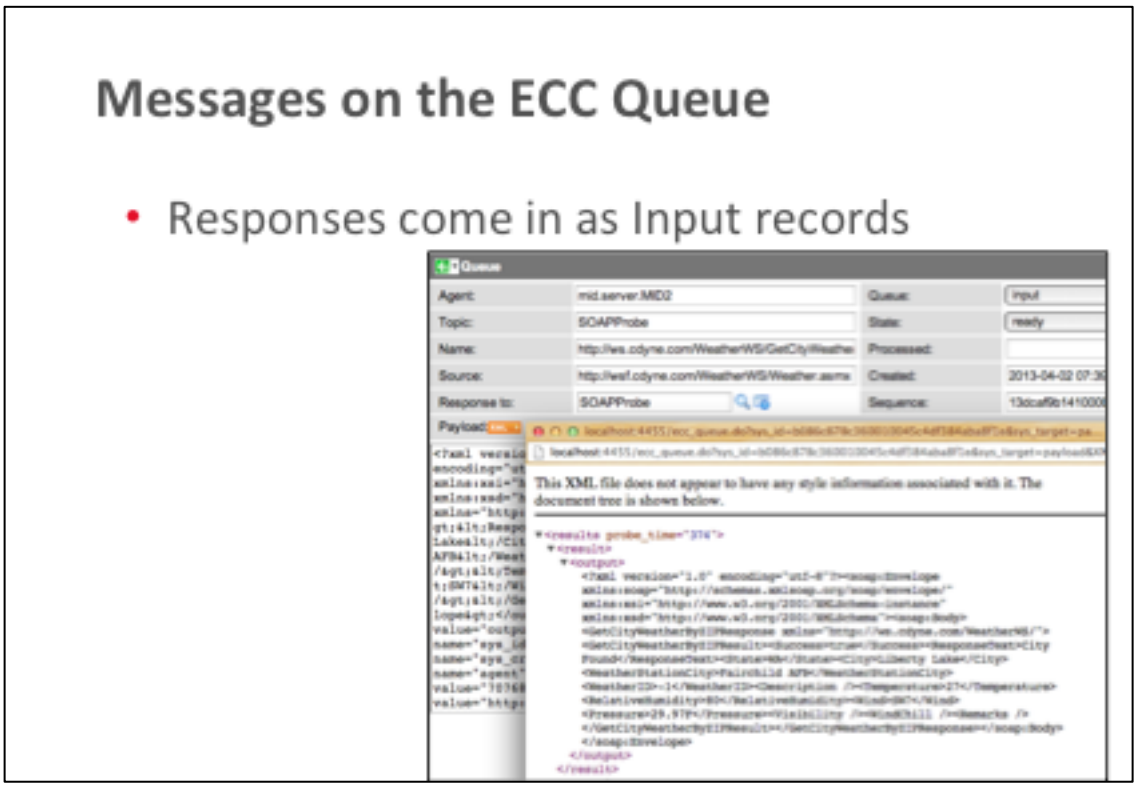
Messages on the ECC Queue

- Asynchronous SOAP Messages go to the ECC Queue



Asynchronous Web service requests go through the ECC Queue just as they did when using a MID Server. However, in these cases they are processed by the instance rather than by the MID Server.

©2014 ServiceNow, Inc. All rights reserved.

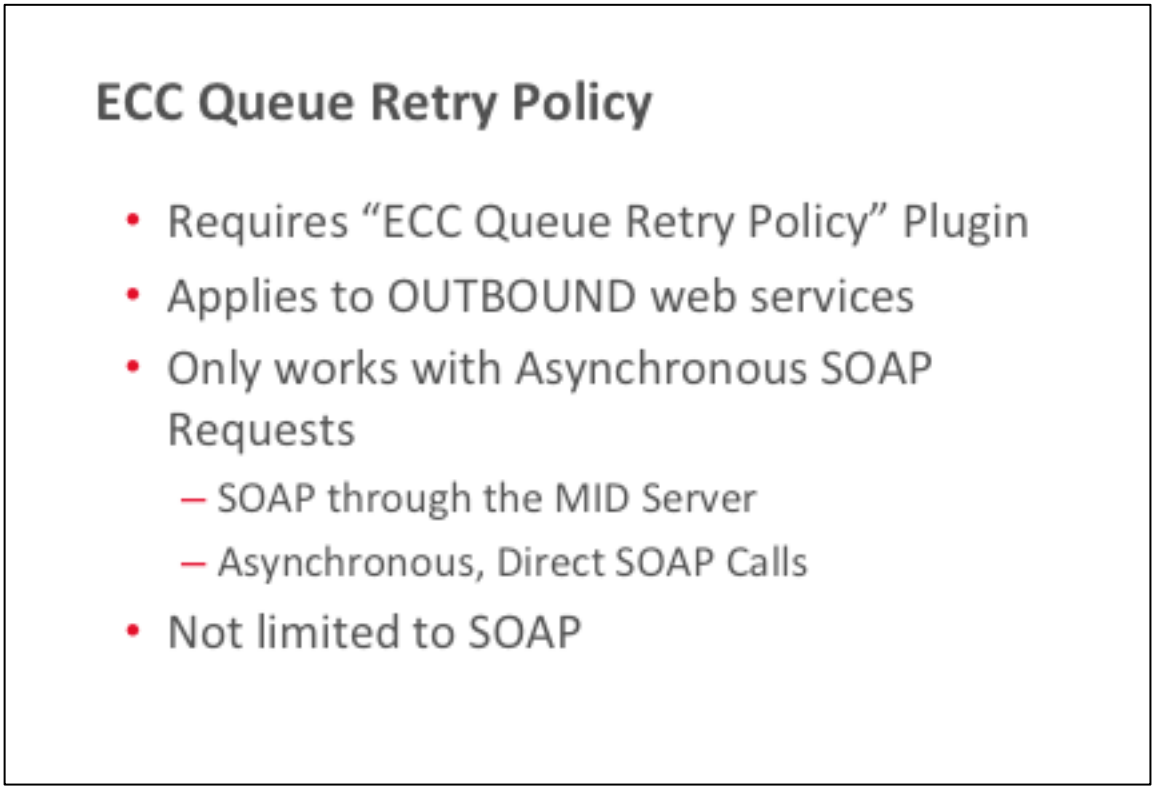


Asynchronous Web service responses come through the ECC Queue as "input" records corresponding to the "output" record that held the request.

®2014 ServiceNow, Inc. All rights reserved.



®2014 ServiceNow, Inc. All rights reserved.



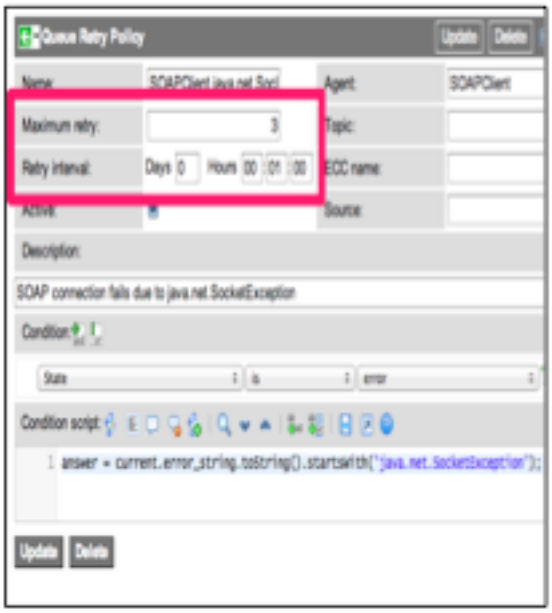
In order to make Web service communication more robust, ServiceNow allows you to create policies around outbound, asynchronous Web services so that you can monitor the types of responses or connection information received from a web service call. If the response or connection information matches certain criteria, it will flag the request as a failure and will wait a predetermined time and then retry the submission of that request.

This is not limited to SOAP requests, but any request that goes through the ECC Queue can leverage this feature as well.

©2014 ServiceNow, Inc. All rights reserved.

Policy Retry Options

1. Maximum retries
2. Interval between retries



These retry policies allow you to set both a maximum number of retries the system will perform as well as the time you want to wait between each retry.

®2014 ServiceNow, Inc. All rights reserved.

Policy Condition Options

1. ECC Queue Field – Exact Matches
2. Dropdown AND/ OR Filtering
3. Server side Javascript logic



There are also three ways to filter the response records and to use logic to determine if the request yielded an error condition in the result.

First, you can do text for text matching on the agent, topic, ECC Name, and source fields that are returned in the ECC Queue "input" record that is a result of the original request.

Second, you can build out powerful filters based on any value condition of all fields associated with the ECC Queue record.

Finally, you have the full power of server-side JavaScript capabilities to enhance your condition logic.

©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

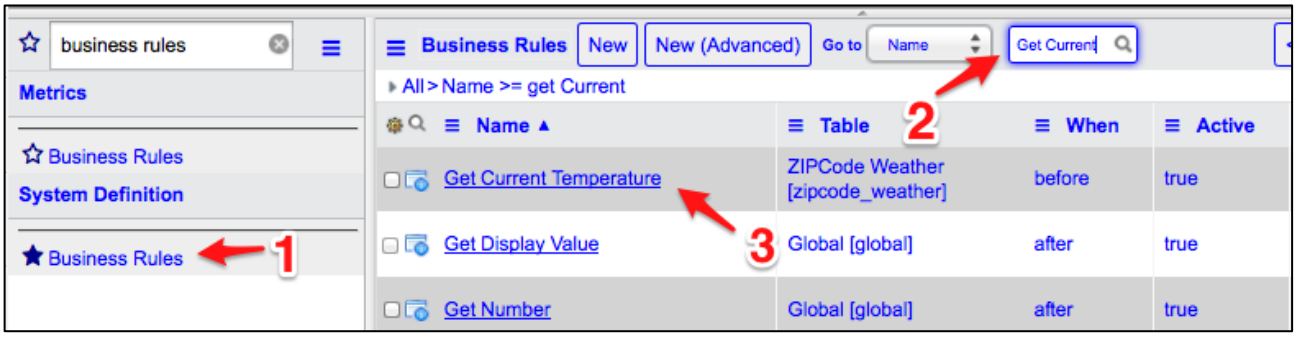
Lab 3
Asynchronous SOAP and Retry Policies

In this exercise, you will learn how to submit and handle Asynchronous SOAP requests. Additionally, you will learn how to create and leverage ECC Queue Retry Policies to make your web service calls more robust.

In this exercise, you will replace the business rule created in the previous exercise on the "ZIPCode Weather" table. This new business rule will update the weather for a ZIP Code asynchronously when a ZIP code is entered or changed. You will then write an ECC Queue Retry policy and simulate a failure scenario for the web service calls. You will verify the Retry Policies are firing and then you will turn off the failure simulation and ensure the retries succeed in getting a response.

Disable your prior Business Rule

1. Navigate to **System Definition > Business Rules** and open the business rule: **Get Current Temperature**.

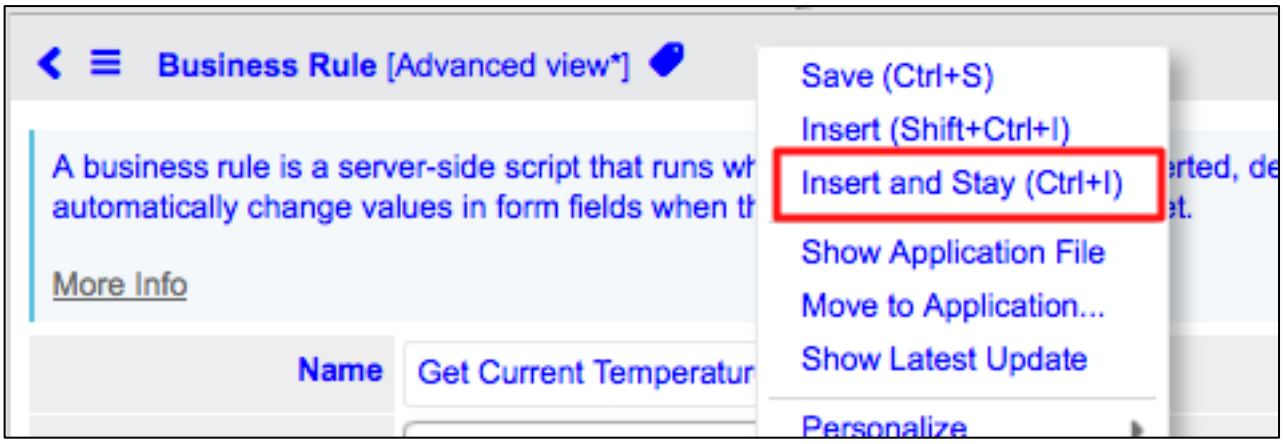


2. Deactivate the rule by unchecking the **Active** checkbox, then right-click the title bar and **Save**.

©2014 ServiceNow, Inc. All rights reserved.

Create a new business rule to run asynchronously

1. Right-click on the title bar of the business rule record you just deactivated and click **Insert and Stay**.



2. Once the form refreshes from the insert, rename the business rule to be: **Get Current Temperature Asynchronously**.

3. Set the **Active** field to be checked.

4. Change the **When** field to be: "**async**". This will execute the business rule whenever there is a free worker process available rather than to hold the User interface until the call is complete.

©2014 ServiceNow, Inc. All rights reserved.

5. Change the script to be the following (lines highlighted are the lines that have had changes from the prior business rule):

```
//Setup the SOAP request
var s = new SOAPMessage('Lab Weather - My Weather', 'WeatherSoap.GetCityWeatherByZIP');
s.setParameter('ZIP', current.zipcode);
s.post(true); // Asynchronous request

//Helpful class from john-james-andersen.com
var ihelper = new IntegrationHelper();
var response = ihelper.getResponse(s, 10000);

//Convert the xml response string to an XMLDocument object
var xmldoc = new XMLDocument(response);
current.temperature = xmldoc.getNodeText("//Temperature");

current.update();
```

6. Right click on the title bar of the business rule and click **Save**.

Verify the business rule runs asynchronously

1. Navigate to **Weather Station > ZIPCode Weather**.

2. Click **New.**

3. Set the **ZIPCode** to be **83404** and click **Submit**.

4. Notice that when the page reloads, there is no temperature listed. This is due to the business rule running asynchronously.

5. Update the list by clicking on **Weather Station > ZIPCode Weather** module again. Do this repeatedly until a temperature appears. (It should take no longer than 30 seconds)

©2014 ServiceNow, Inc. All rights reserved.

Create an ECC Queue Retry Policy

1. Browse to **ECC > Queue Retry Policy.**

2. Click **New**.

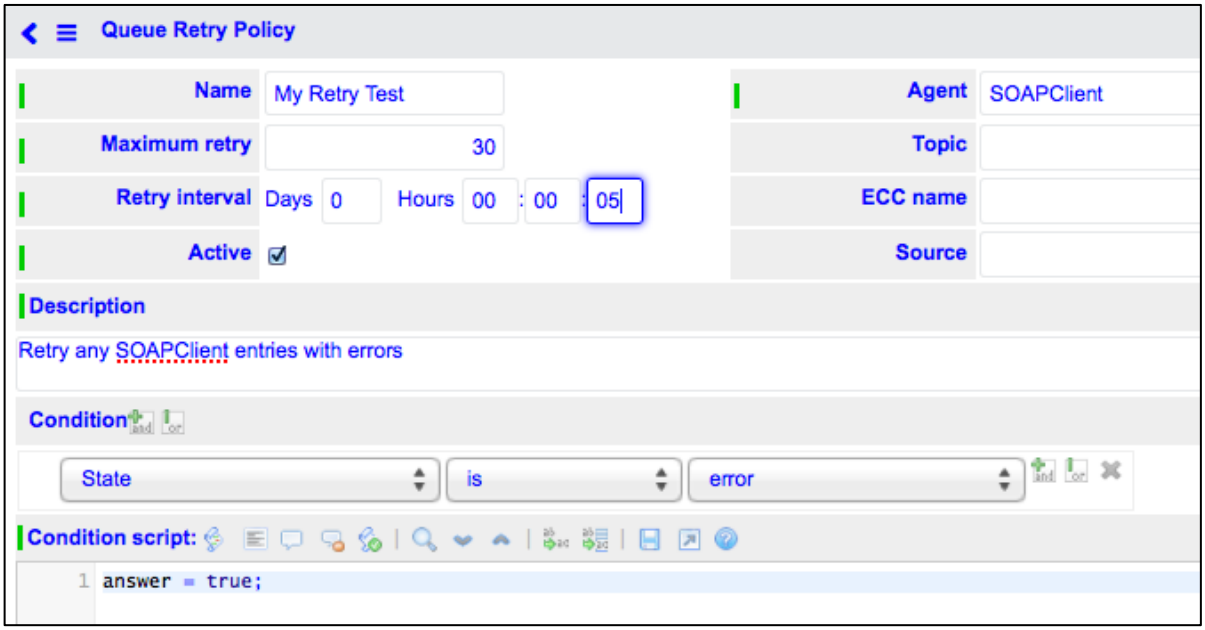
3. Enter the following information:

Name	My Retry Test
Maximum Retry	30
Retry Interval	0 Days 00 Hours 00 Minutes 05 Seconds
Active	True
Agent	SOAPClient
Description	Retry any SOAPClient entries with errors
Condition	State is error

4. Set the **Condition Script** field to be: **answer = true;**

©2014 ServiceNow, Inc. All rights reserved.

5. Your policy should look like the following:



6. Click **Submit**.

Test the Retry Policy

1. In order to show ECC Queue Retry policies, we have included a utility that causes all outbound asynchronous SOAP requests to fail.

2. Browse to **Integration Lab 101 > ECC Queue Retry Helper**.

3. Click the **Automatically Fail SOAP Requests** button.

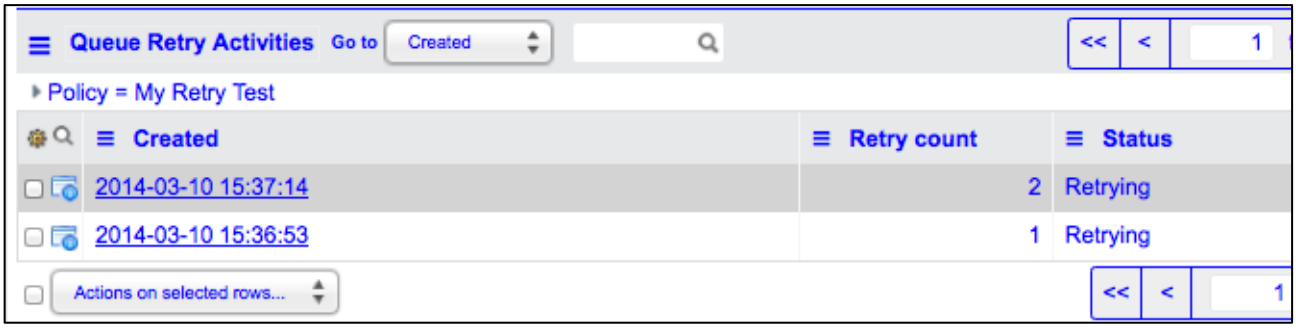
4. Submit a new SOAP request by browsing to the **Weather Station > ZIPCode Weather** module and creating a new record with the **ZIPCode** of **83201**.

5. Wait approximately 30-50 seconds.

©2014 ServiceNow, Inc. All rights reserved.

6. Browse to your retry policy record by going to **ECC > Queue Retry Policy** and opening your **My Retry Test** policy.

7. At the bottom of the record there is a **Queue Retry Activities** list or tab. It should be listing a few retry attempts.

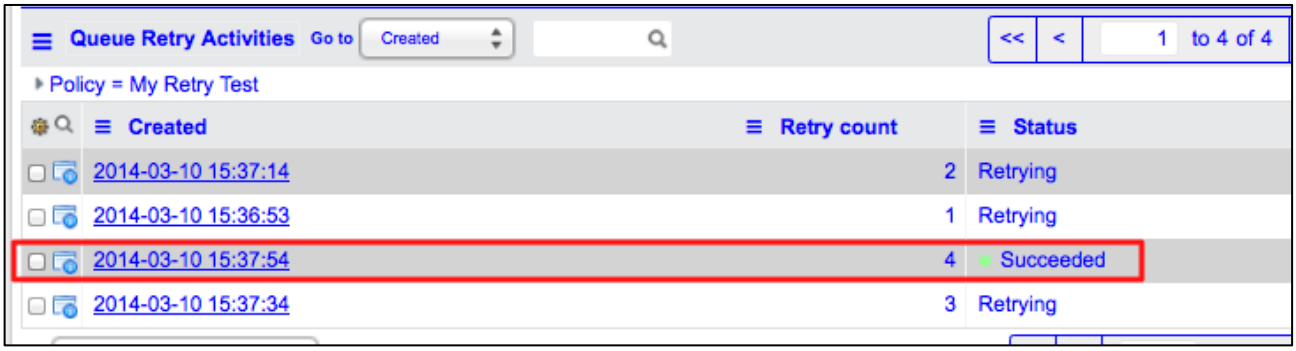


8. Go back to the **Integration Lab 101 > ECC Queue Retry Helper** module and click the **Allow SOAP Requests to Succeed** button.

9. Wait approximately 30-50 seconds for the last retry to fire and register in the system.

10. Browse back to your retry policy record by going to **ECC > Queue Retry Policy** and opening your **My Retry Test** policy.

11. At the bottom of the record, take another look at the **Queue Retry Activities** list or tab. It should be listing a few retry attempts.



©2014 ServiceNow, Inc. All rights reserved.

Report your Completion of this Lab Exercise

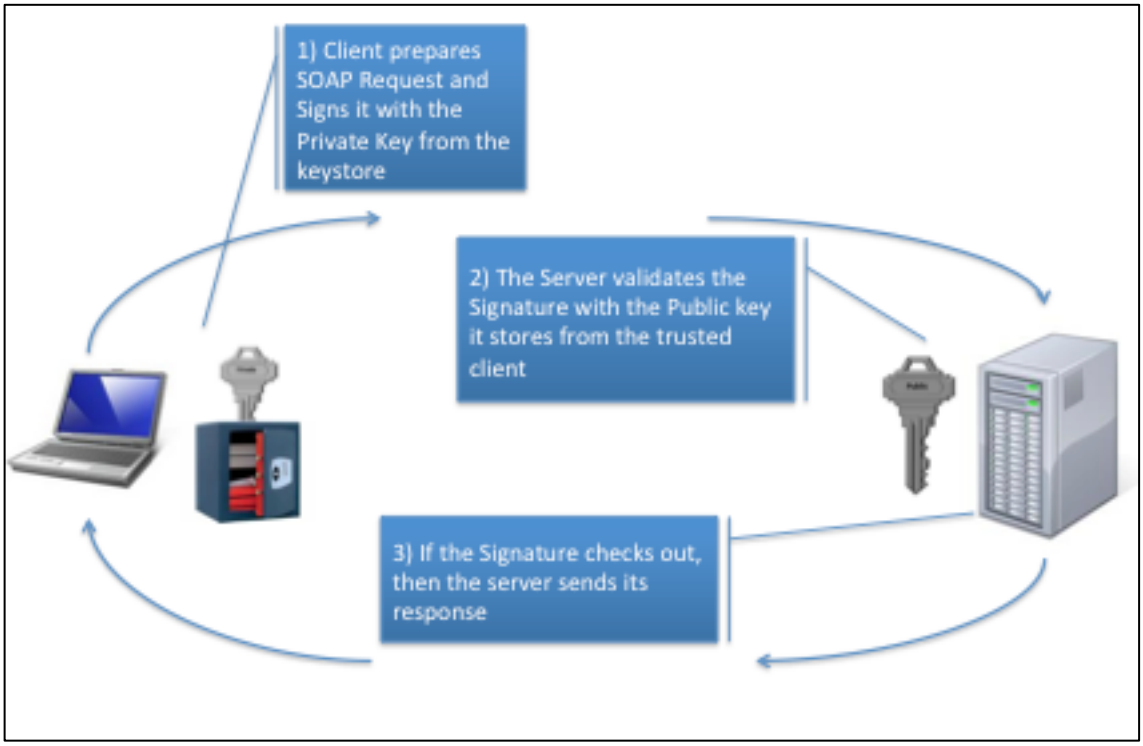
1. Navigate to **Lab Management > Report Lab Progress**.

2. Enter your progress status.

©2014 ServiceNow, Inc. All rights reserved.



®2014 ServiceNow, Inc. All rights reserved.



When signing a request, the client holds a private key within a password protected digital vault, called a key store. The private key is filed away in the key store under the name of an alias.

The server is given a public key from the client. The service provider stores that key with the web service to be used for verifying digital signatures from the client.

When the client generates a SOAP Request, it will get the private key from it's protected key store and use it to digitally sign the SOAP Request payload. It then adds some WS-Security tags to the original SOAP Request that explain that the payload is digitally signed.

When the web service receives this payload, it will see that the message is signed and it will use the client's public key to verify the signature. Once the signature has been verified, the service will process the request and submit the response back to the client.

©2014 ServiceNow, Inc. All rights reserved.

Lab Goal

Learn how to set up a SOAP Request to digitally sign the SOAP payload using the WS-Security profile.

In this exercise, you will consume a SOAP Web Service that requires WS-Security instead of Basic Authentication. You will need to use a certificate and keystore generated for the purpose of signing the SOAP payload.

Lab 4
WS-Security

Download the public signing certificate

1. For this part of the exercise we are going to need to have a signing certificate. We have a certificate already prepared for you in a string format called the "PEM" format.

2. Navigate to **Integration Lab 101 > Copy & Paste > Ex 4 – PEM Cert String.**

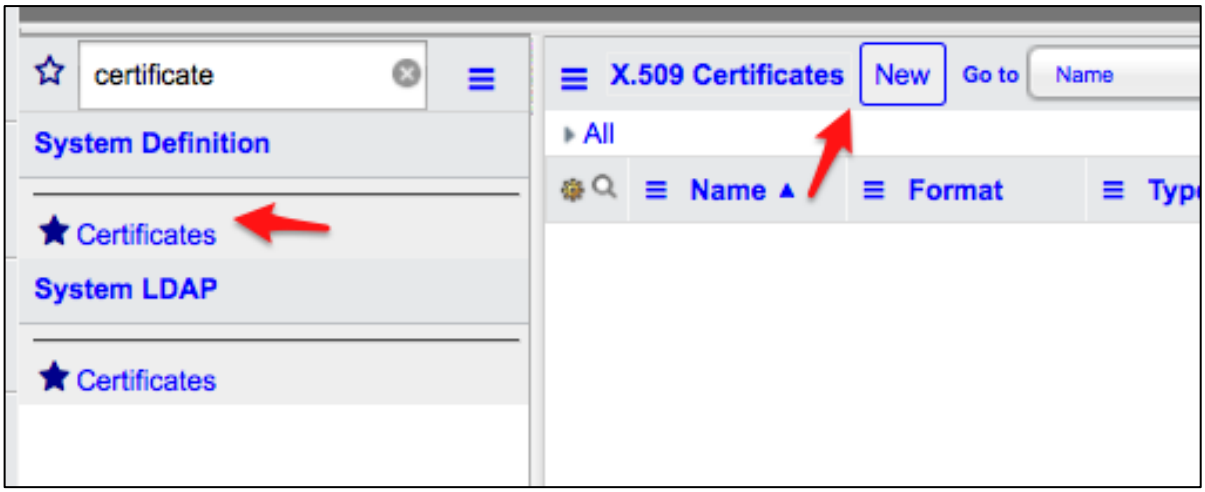


3. Copy the full string to your clipboard.

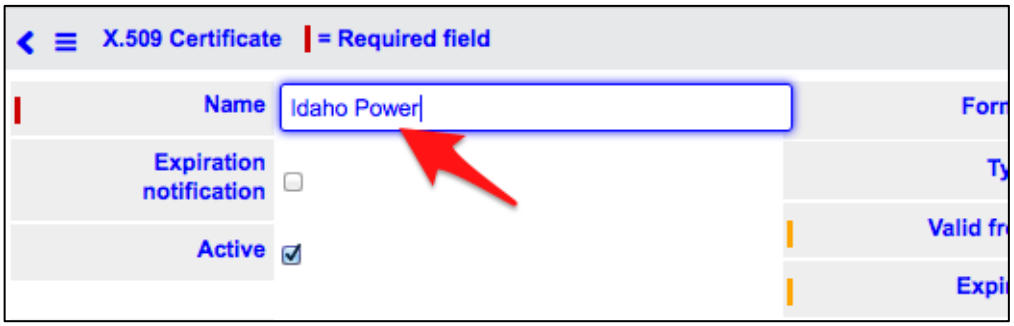
©2014 ServiceNow, Inc. All rights reserved.

Set up the signing certificate in ServiceNow

1. Navigate to **System Definition > Certificates**. Then click **New** to create a new Certificate record.



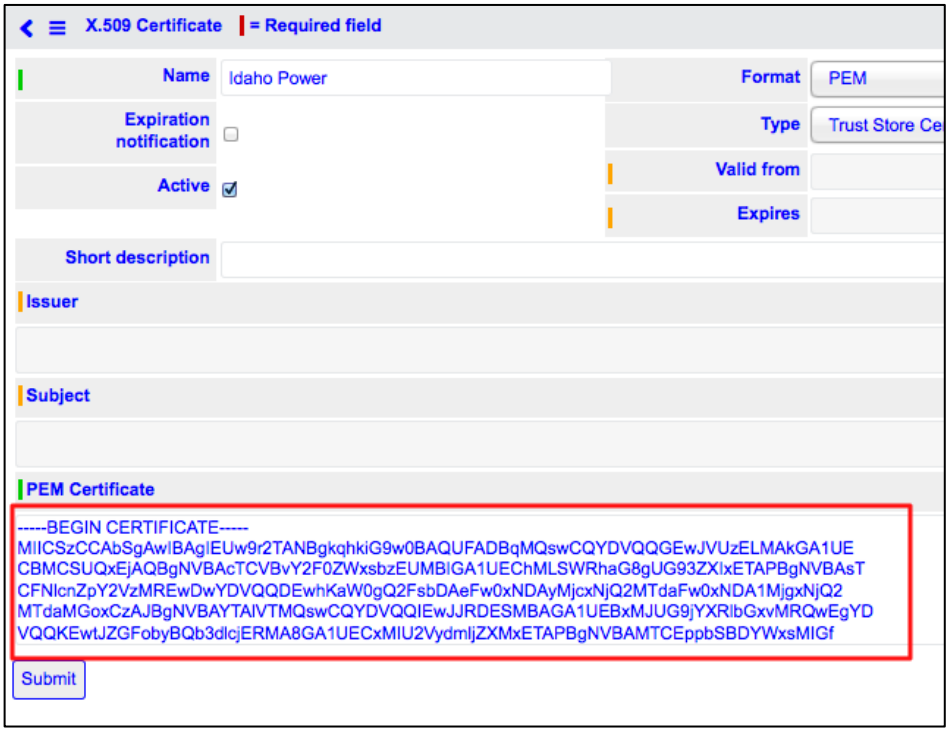
2. Set the **Name** to be **Idaho Power**.



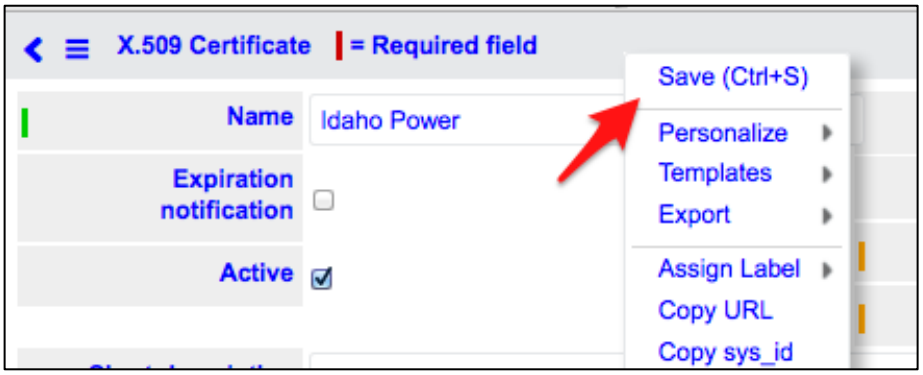
3. Make sure the **Format** is set to **PEM.**

©2014 ServiceNow, Inc. All rights reserved.

4. Paste the certificate string you copied earlier into the **PEM Certificate** field.

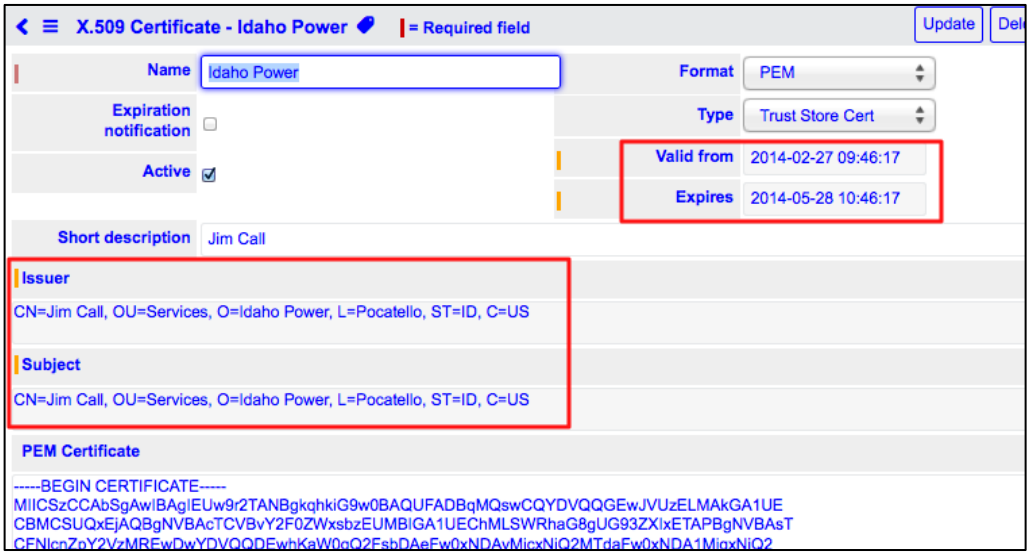


5. Right click on the **x.509 Certificate** title bar and click **Save.**



©2014 ServiceNow, Inc. All rights reserved.

6. The system will decode the PEM string and fill in all certificate meta data for the record.



7. Click **Update.**

Download the Keystore file

1. The keystore file contains the private key that will be used to sign the SOAP Payload. A keystore can contain multiple keys. These keys are identified through “aliases”. The alias for the key in this scenario is “power”. The keystore also has a password used to secure the private key information. The keystore password is “password”.

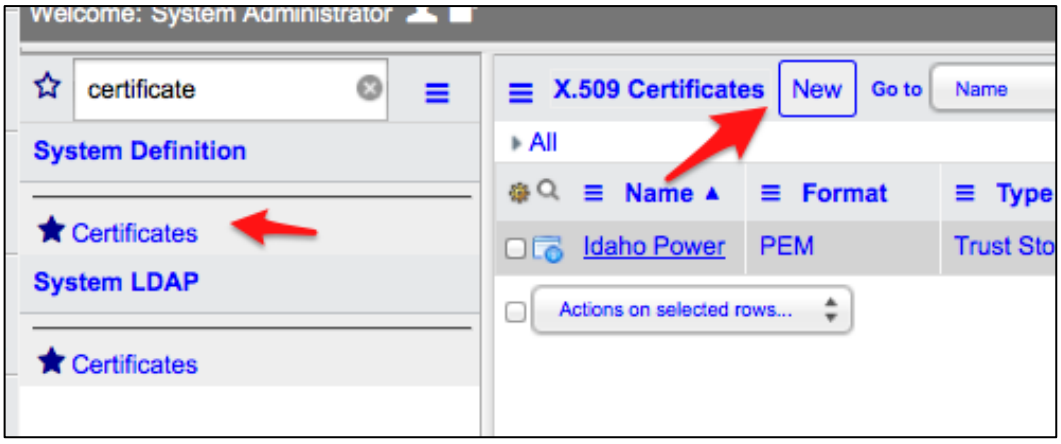
2. To download the keystore file, navigate to **Integration Lab 101 > Download Keystore (ex 4).**

3. Once the page has loaded, click the **Download** button and download the keystore to your computer.

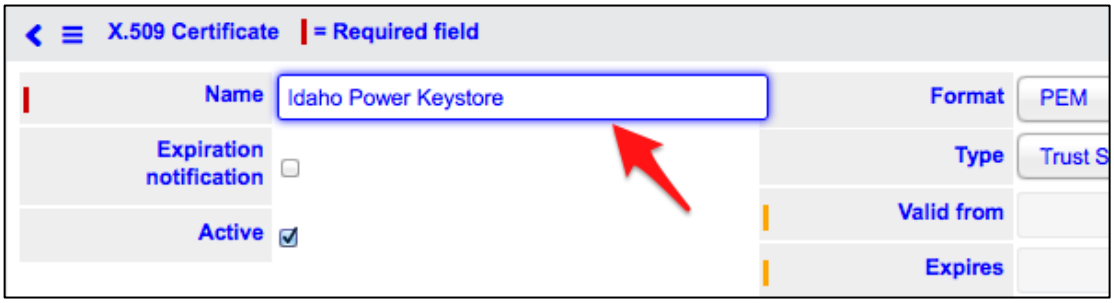
©2014 ServiceNow, Inc. All rights reserved.

Set up the keystore in your ServiceNow Instance

1. In your ServiceNow instance, browse to **System Definition > Certificates**. Then click **New** to create a new Certificate record.



2. Set the **Name** to be **Idaho Power Keystore**.

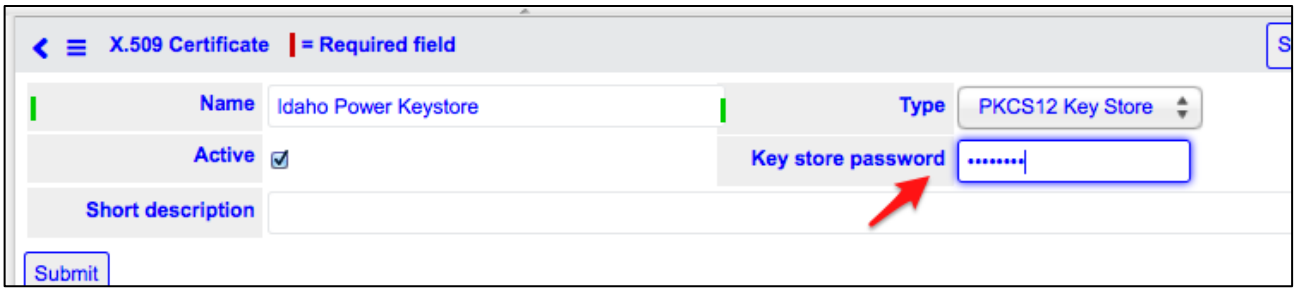


3. Make sure the **Type** is set to **PKCS12 Key Store.**

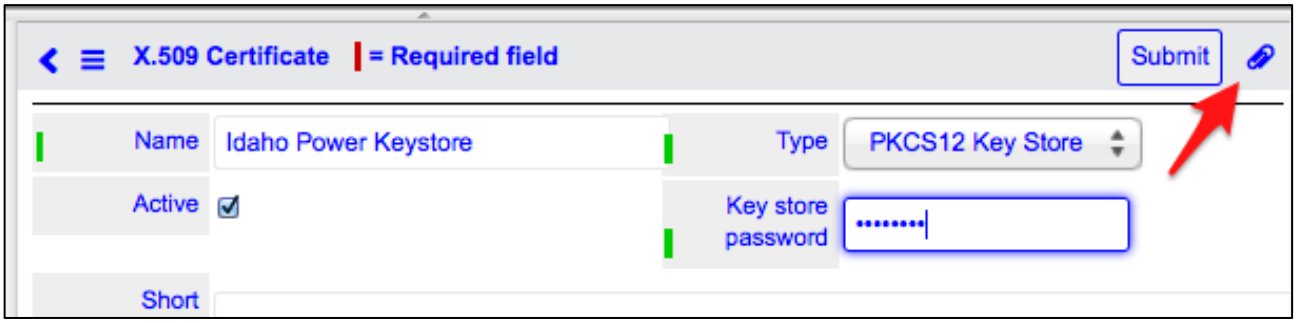


©2014 ServiceNow, Inc. All rights reserved.

4. Set the **Key store password** field to **password** (this is the password used when the keystore was created).

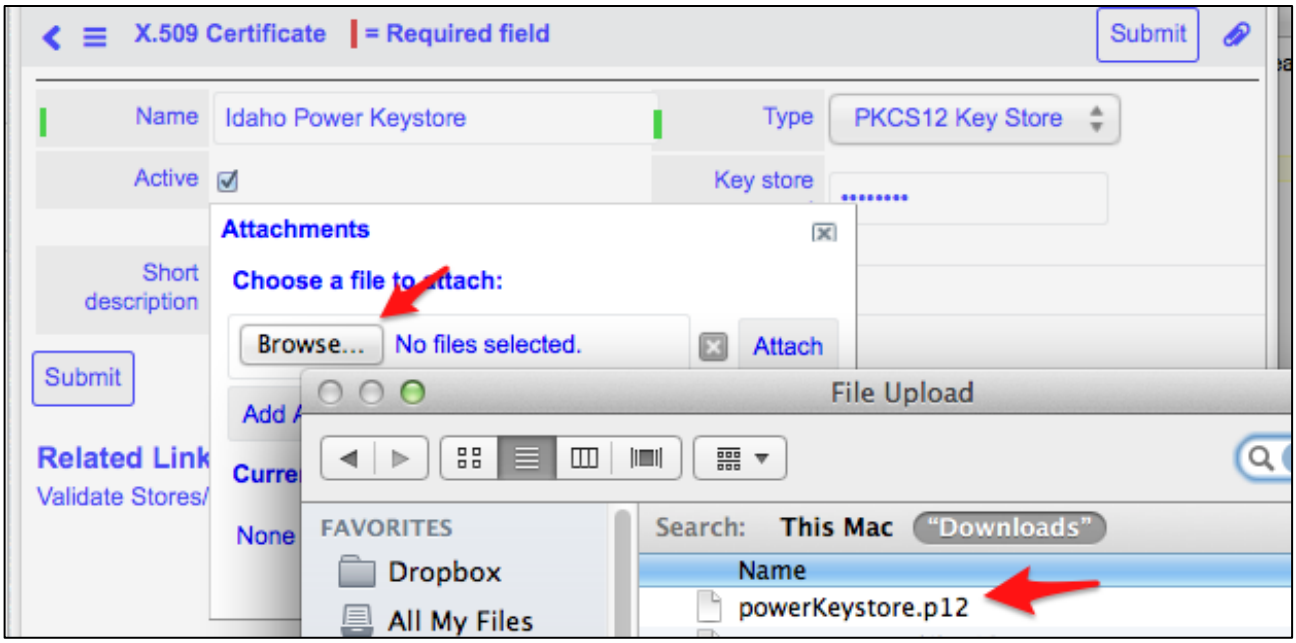


5. Click the paperclip attachment icon on the record.

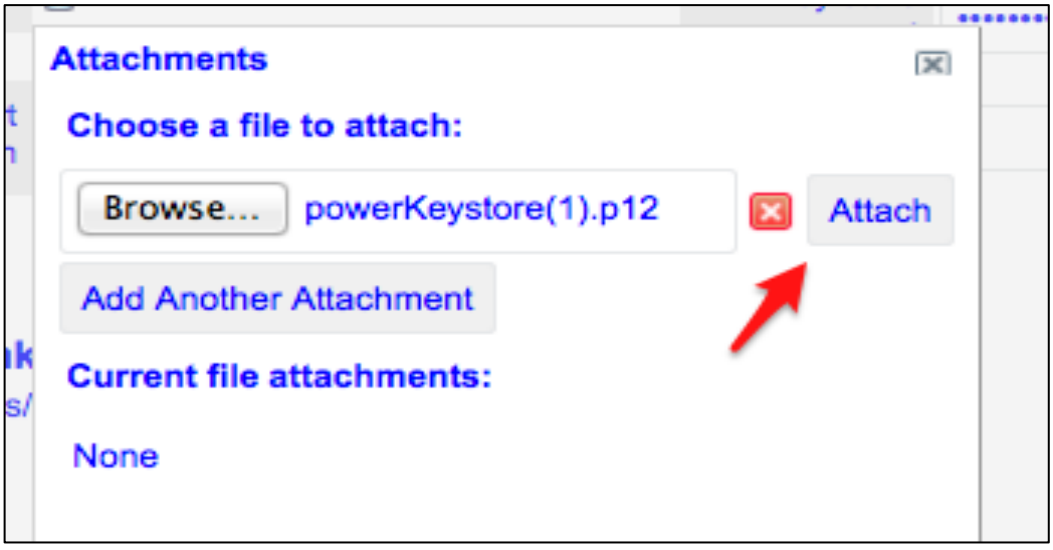


©2014 ServiceNow, Inc. All rights reserved.

6. Click the **Chose Files** button and select the keystore file you downloaded (Your file will be named "**powerKeystore.p12**").

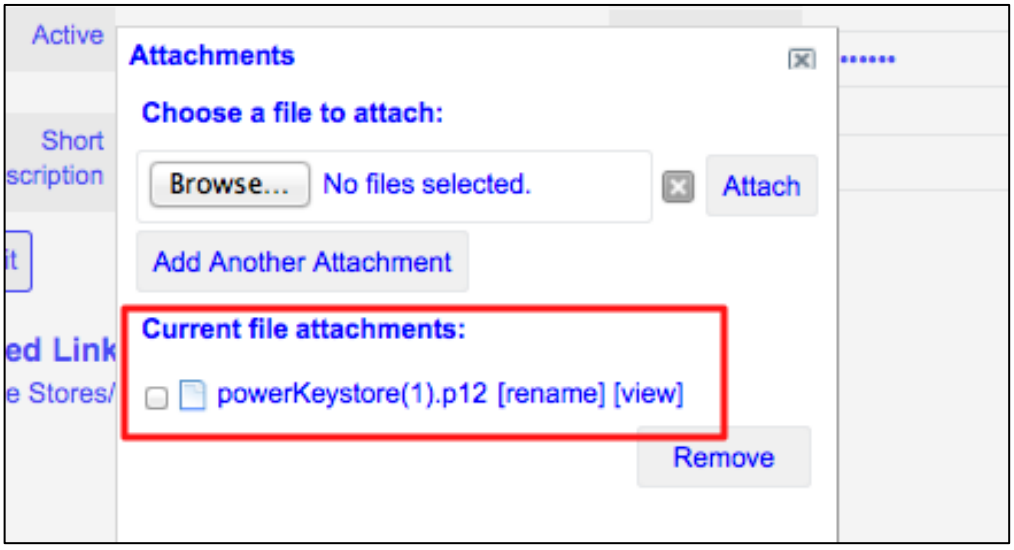


7. Click **Attach**.

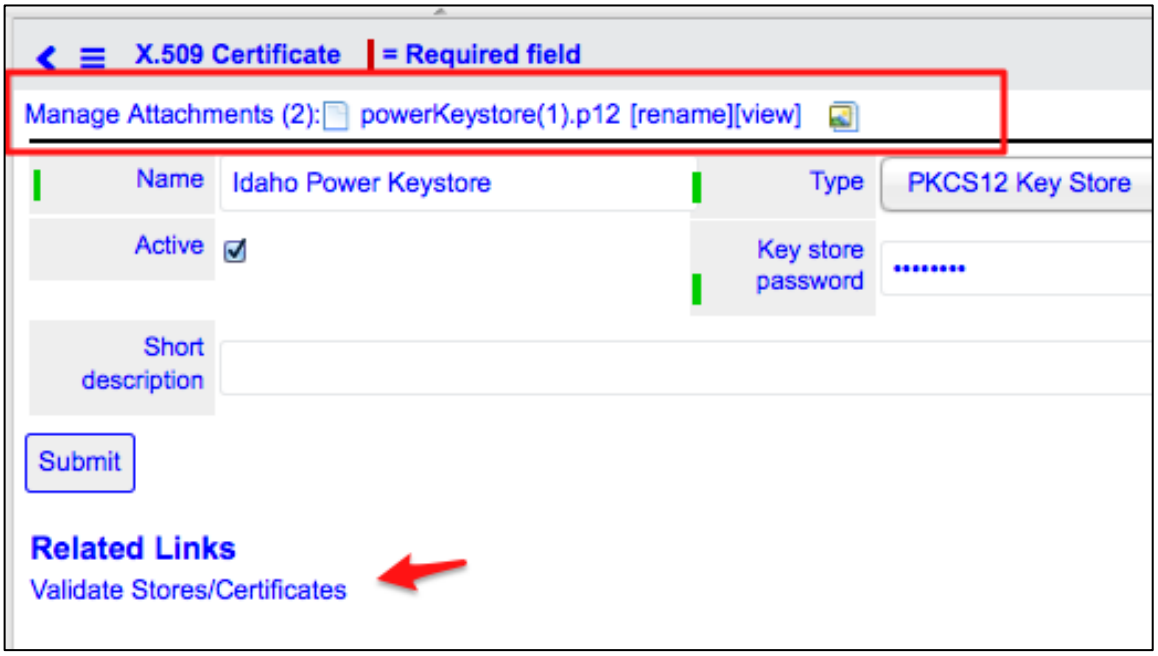


©2014 ServiceNow, Inc. All rights reserved.

8. Once the attachment has loaded, click the “x” in the top right corner of the dialog box.

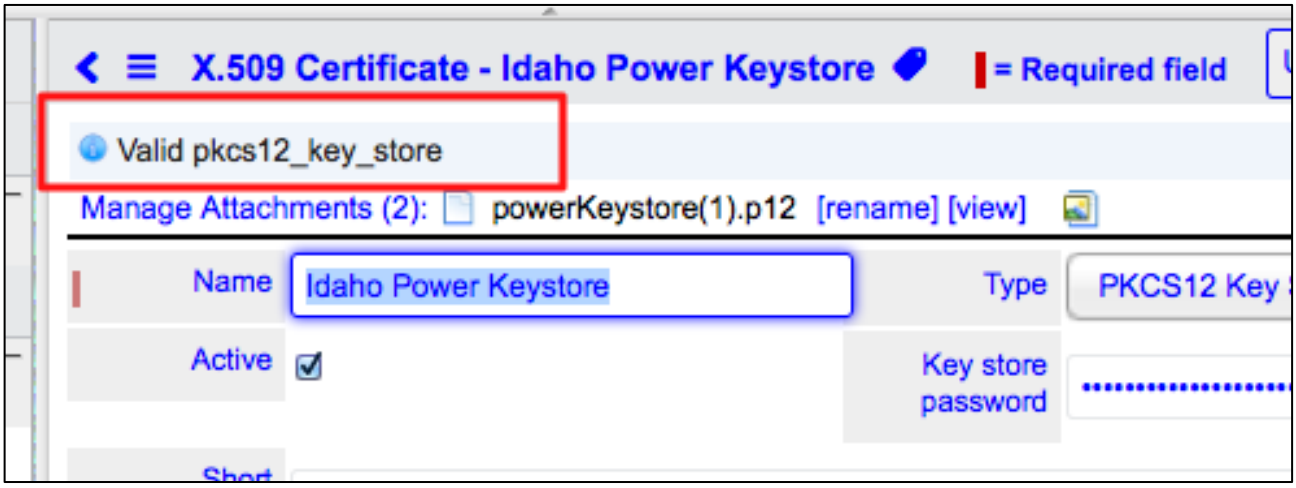


9. Verify that the attachment list listed on the certificate. Then click the **Validate Stores/Certificates** link.



©2014 ServiceNow, Inc. All rights reserved.

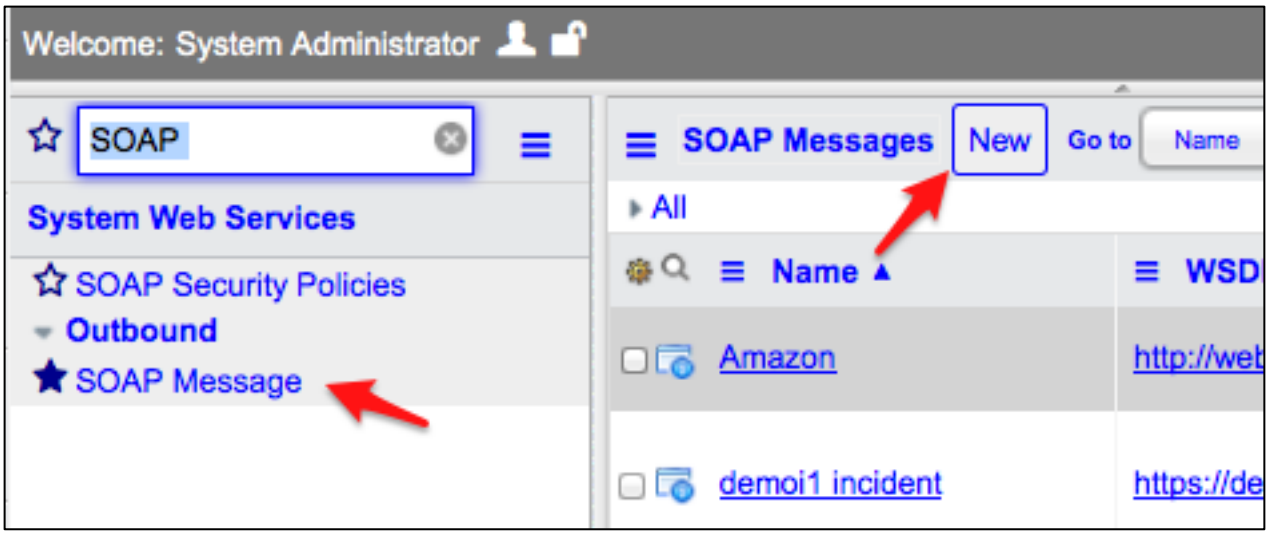
10. Ensure that you get a Validation message on the form.



11. Click **Submit.**

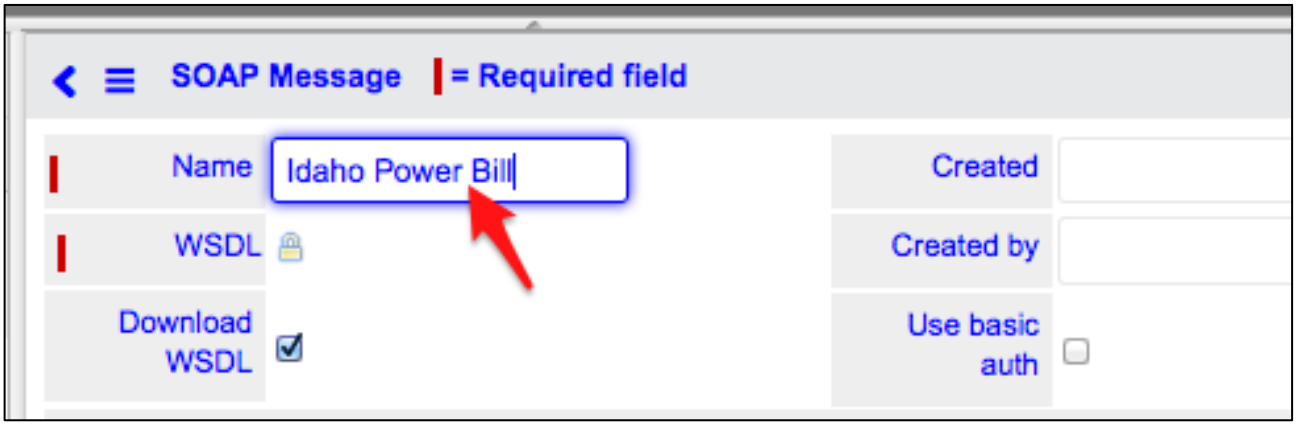
Set up a SOAP Message record

1. Within ServiceNow, navigate to **System Web Services > Outbound > SOAP Message**. Then click **New**.

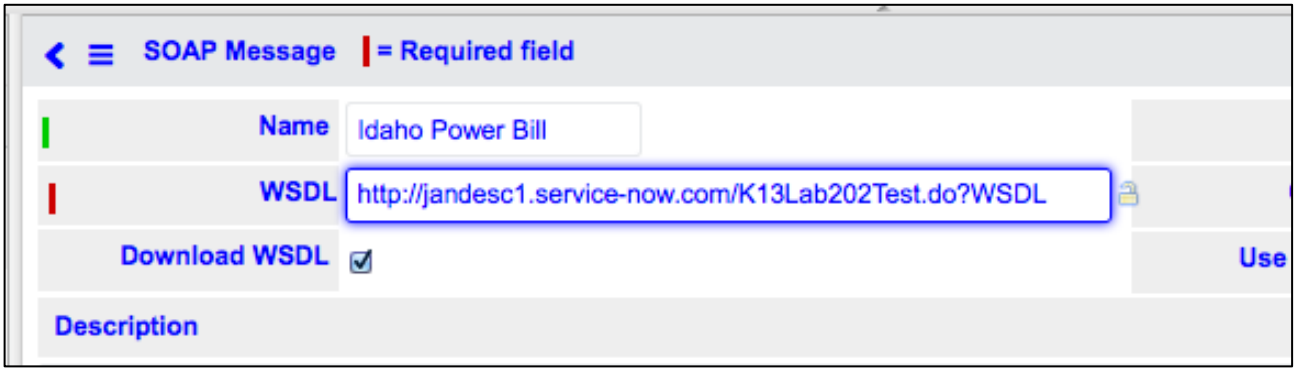


©2014 ServiceNow, Inc. All rights reserved.

2. Set the **Name** to be **Idaho Power Bill**.

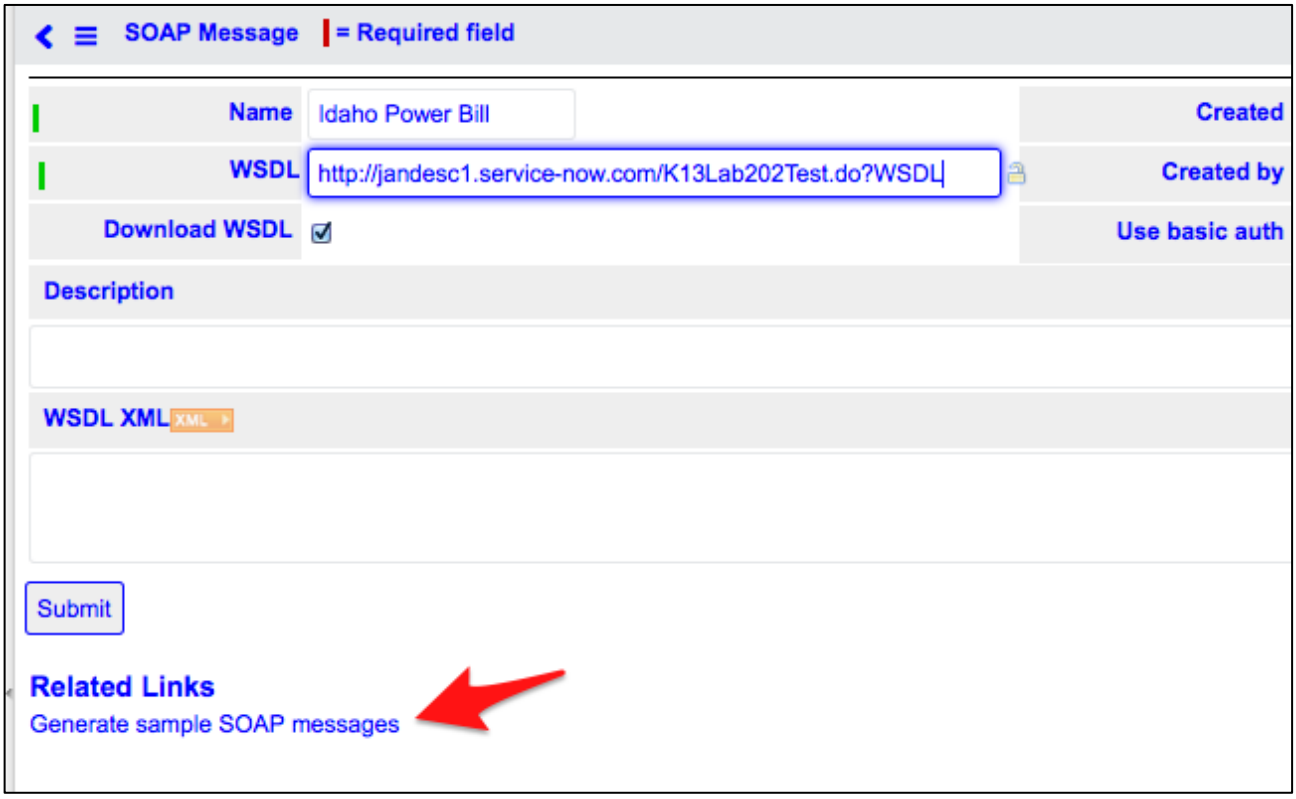


3. Click the padlock icon in the **WSDL** field to make the field editable. Then type the following into the field: **https://jandesc1.service-now.com/K13Lab202Test.do?WSDL**.



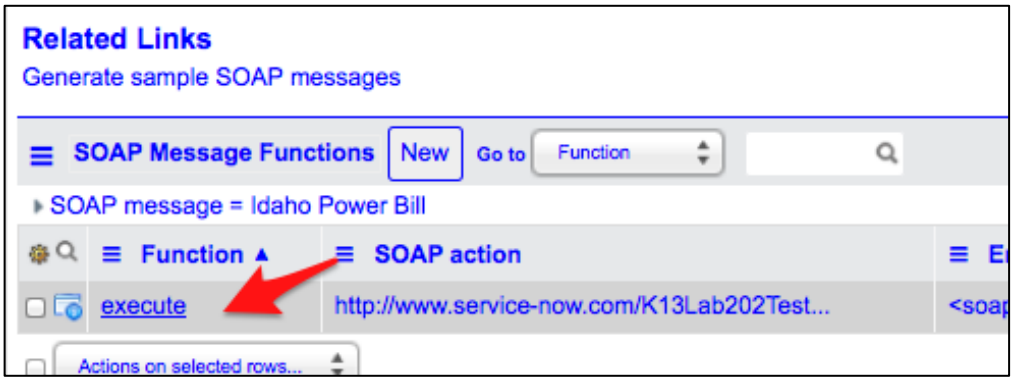
©2014 ServiceNow, Inc. All rights reserved.

4. Click the **Generate sample SOAP Messages** link.



5. Click **OK** if a warning dialog comes up on the screen.

6. Once the function generation has completed, you should see one function named **execute**.



©2014 ServiceNow, Inc. All rights reserved.

Set up the function for standard SOAP submission

1. Inside the **Idaho Power Bill** SOAP Message record, click on the **execute** SOAP Message Function.

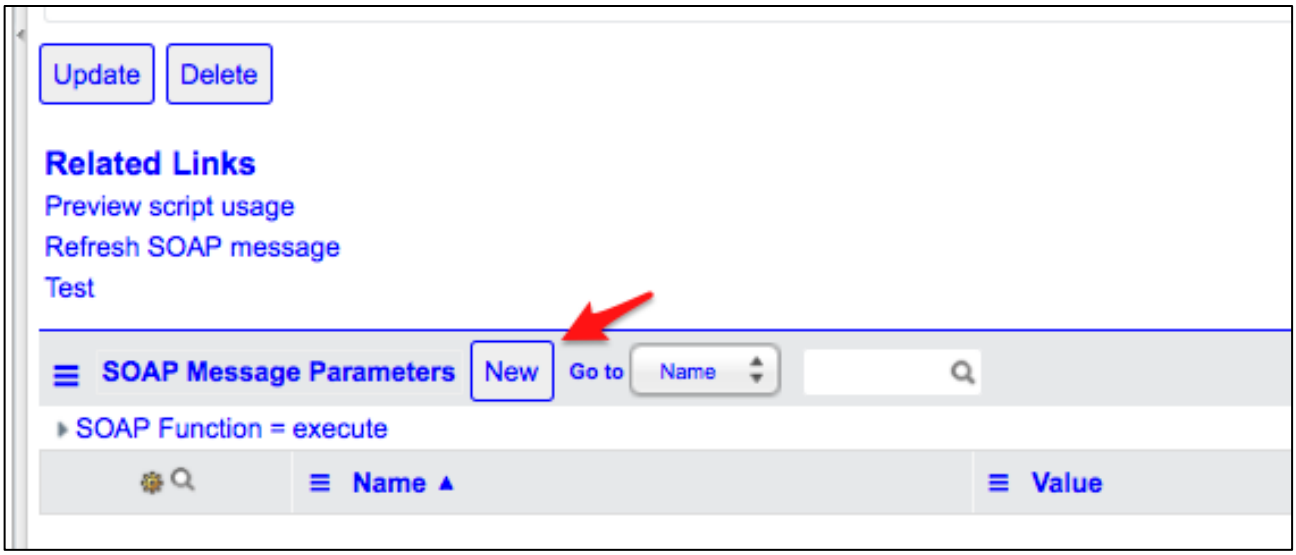


2. You will notice in the **Envelope** field there is a single variable, **${fullName}**.

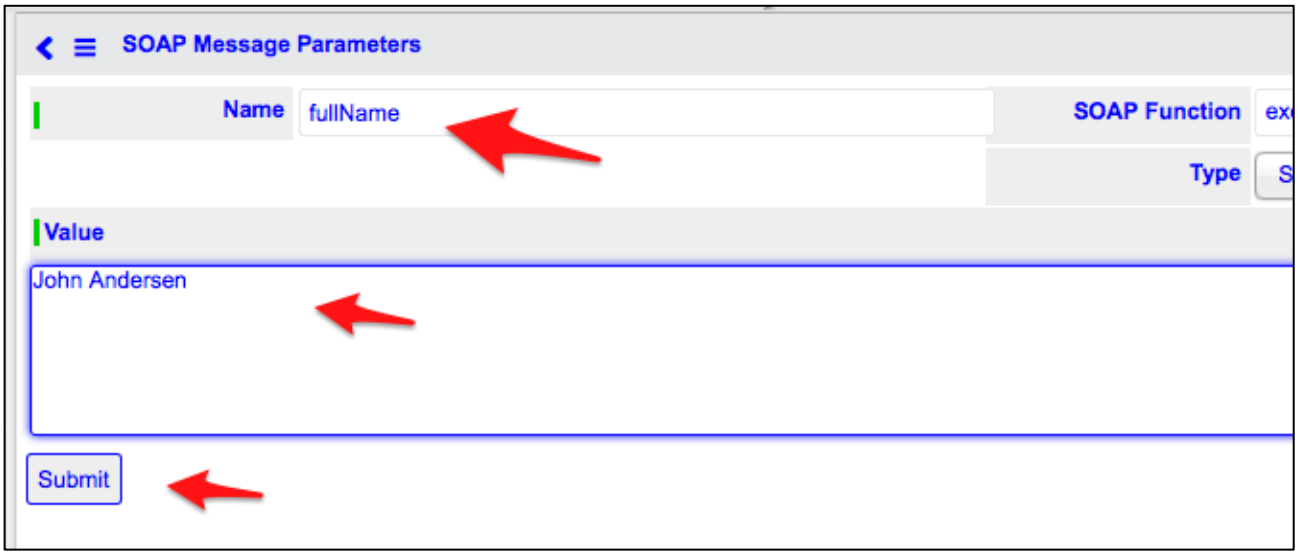


©2014 ServiceNow, Inc. All rights reserved.

3. Create a static value for that variable by browsing to the **SOAP Message Parameters** list on the form and clicking **New**.

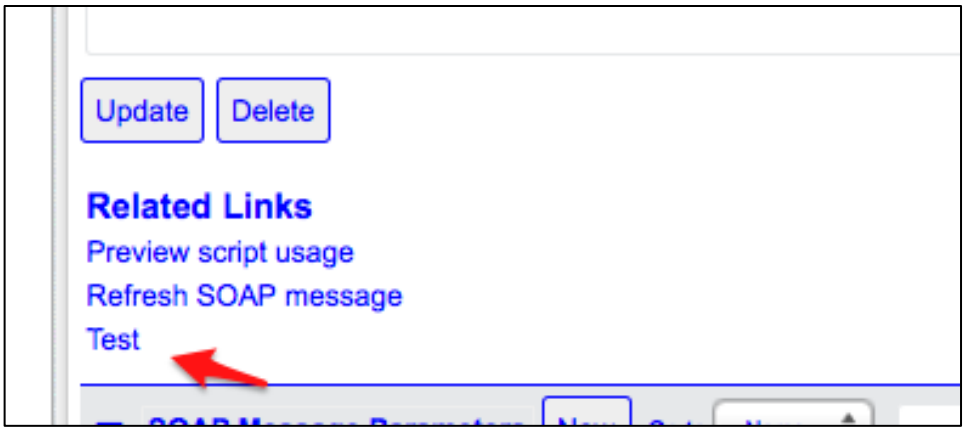


4. Set the **Name** to be "fullName" and the **Value** to be your own name. Then click **Submit.**

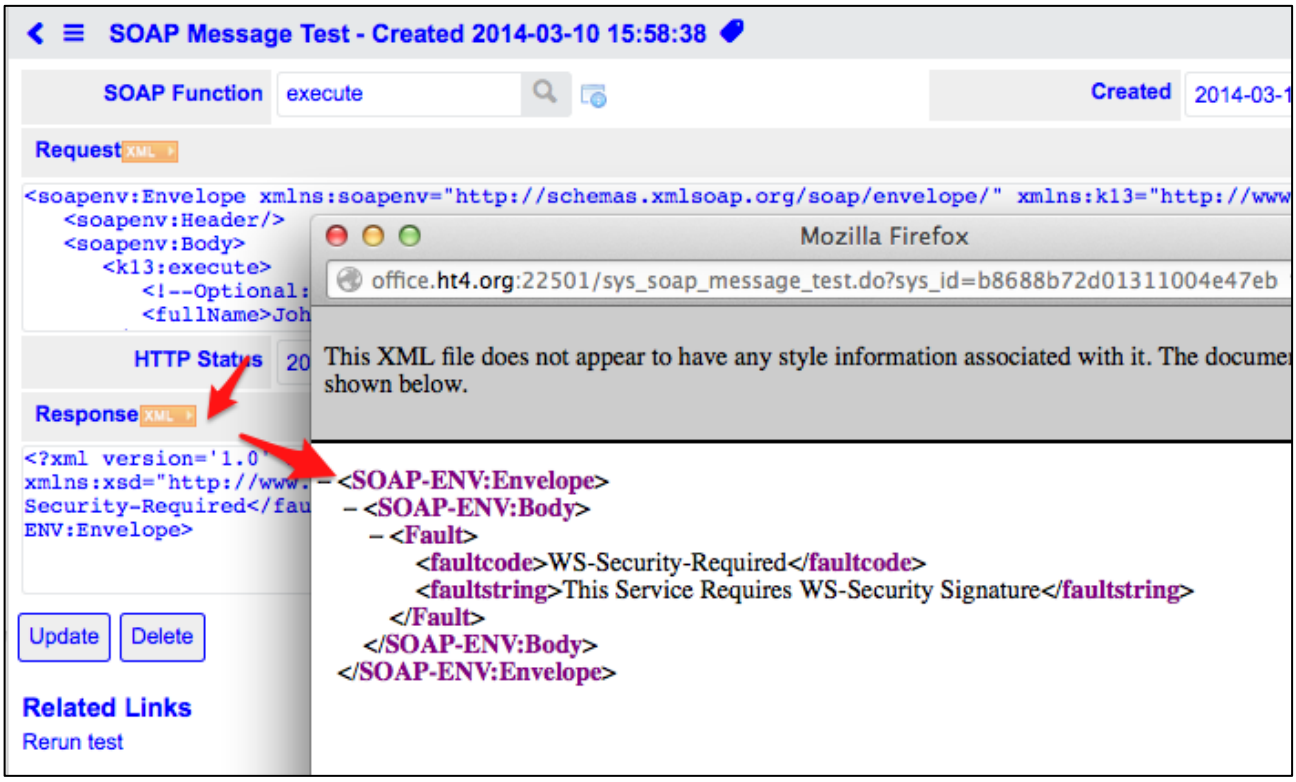


©2014 ServiceNow, Inc. All rights reserved.

5. Now, when the page refreshes, you should be back on the **SOAP Message Function** form. Click the **Test** link to submit the web service call.



6. Once the test result comes back, you can click on the **Response** fields "XML" button to render the response in an easy-to-read format.

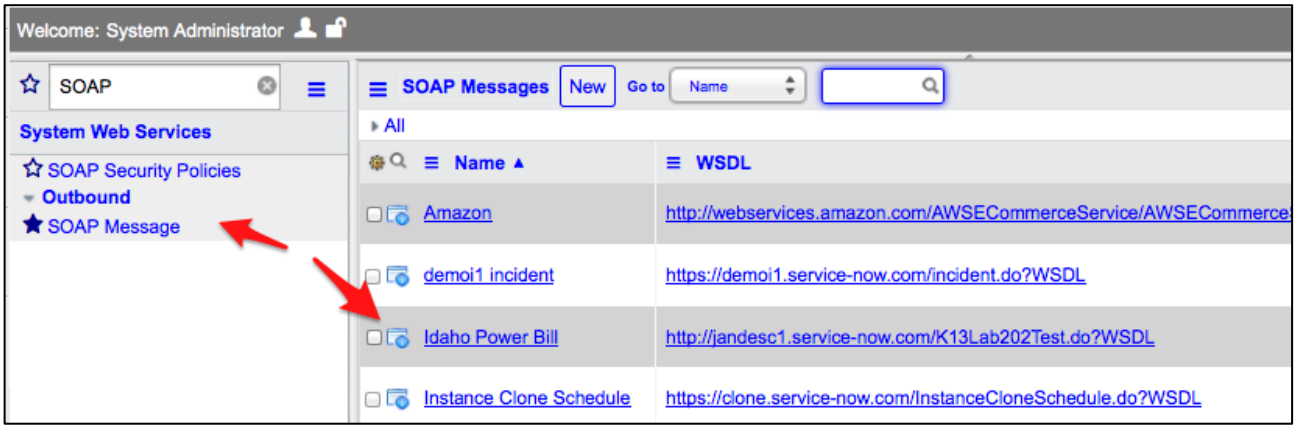


©2014 ServiceNow, Inc. All rights reserved.

7. Notice that we got an error back with the response. The error states that we need to sign our payload using the WS-Security protocol.

Set up the SOAP Message Function to use WS-Security

1. Navigate to **System Web Services > Outbound > SOAP Message** and open the **Idaho Power Bill** record.

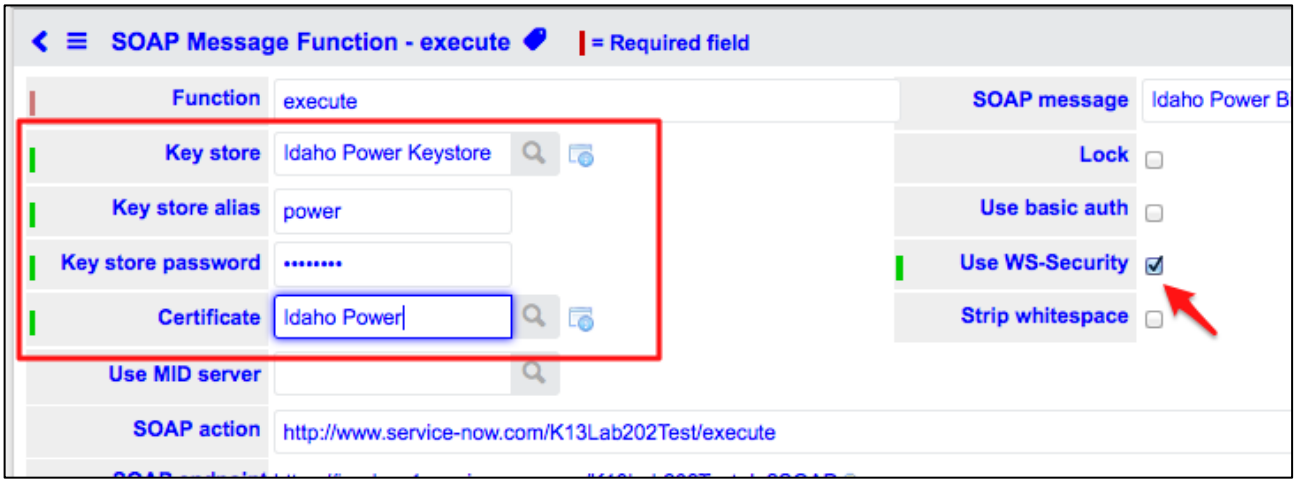


2. Open the **execute** SOAP Message Function.



©2014 ServiceNow, Inc. All rights reserved.

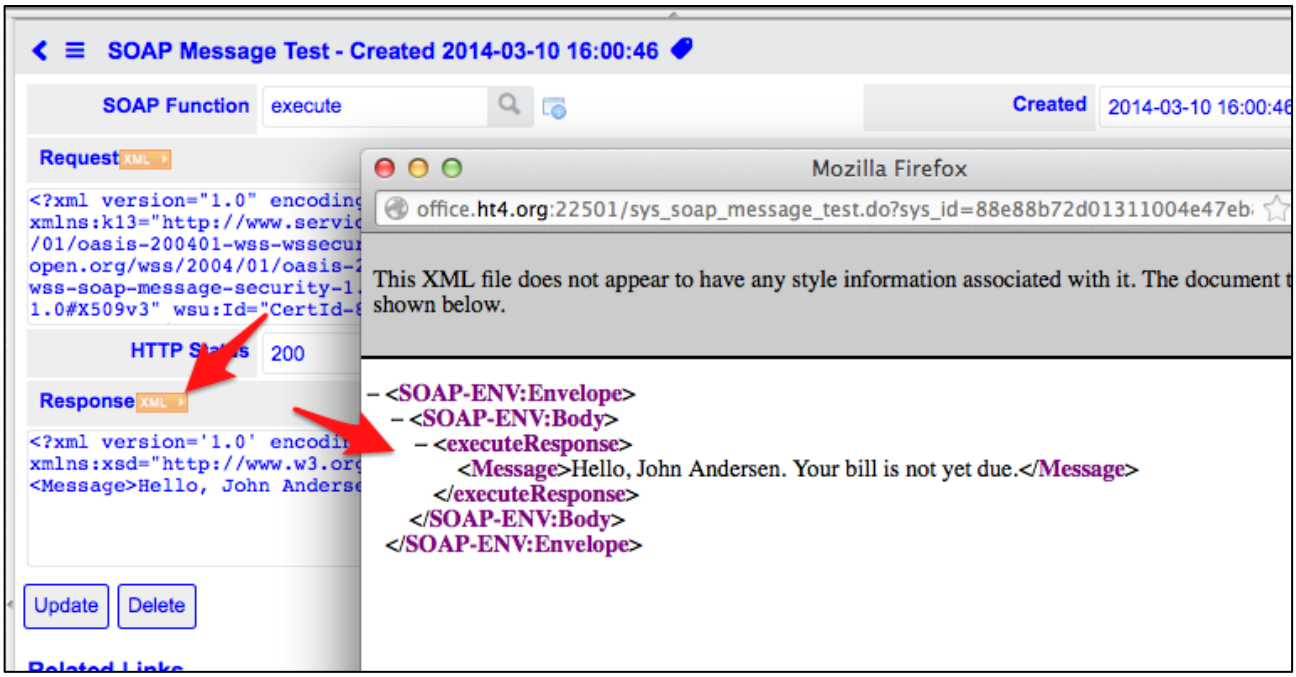
3. Click the Use WS-Security check box and fill in the following information:
    a) Key Store: **Idaho Power Keystore**
    b) Key store alias: **power**
    c) Key store password: **password**
    d) Certificate: **Idaho Power**



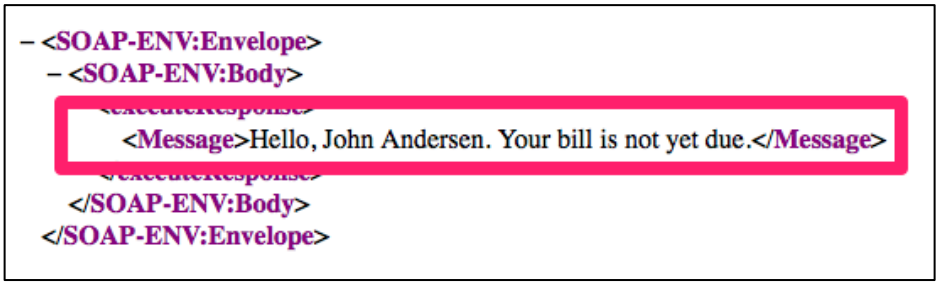
4. Click the **Test** link to submit the request.

©2014 ServiceNow, Inc. All rights reserved.

5. Once the response comes back, click the Response **XML** button to view the formatted response.



6. The response should have a message that contains your name in the text.



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.