REVATURE

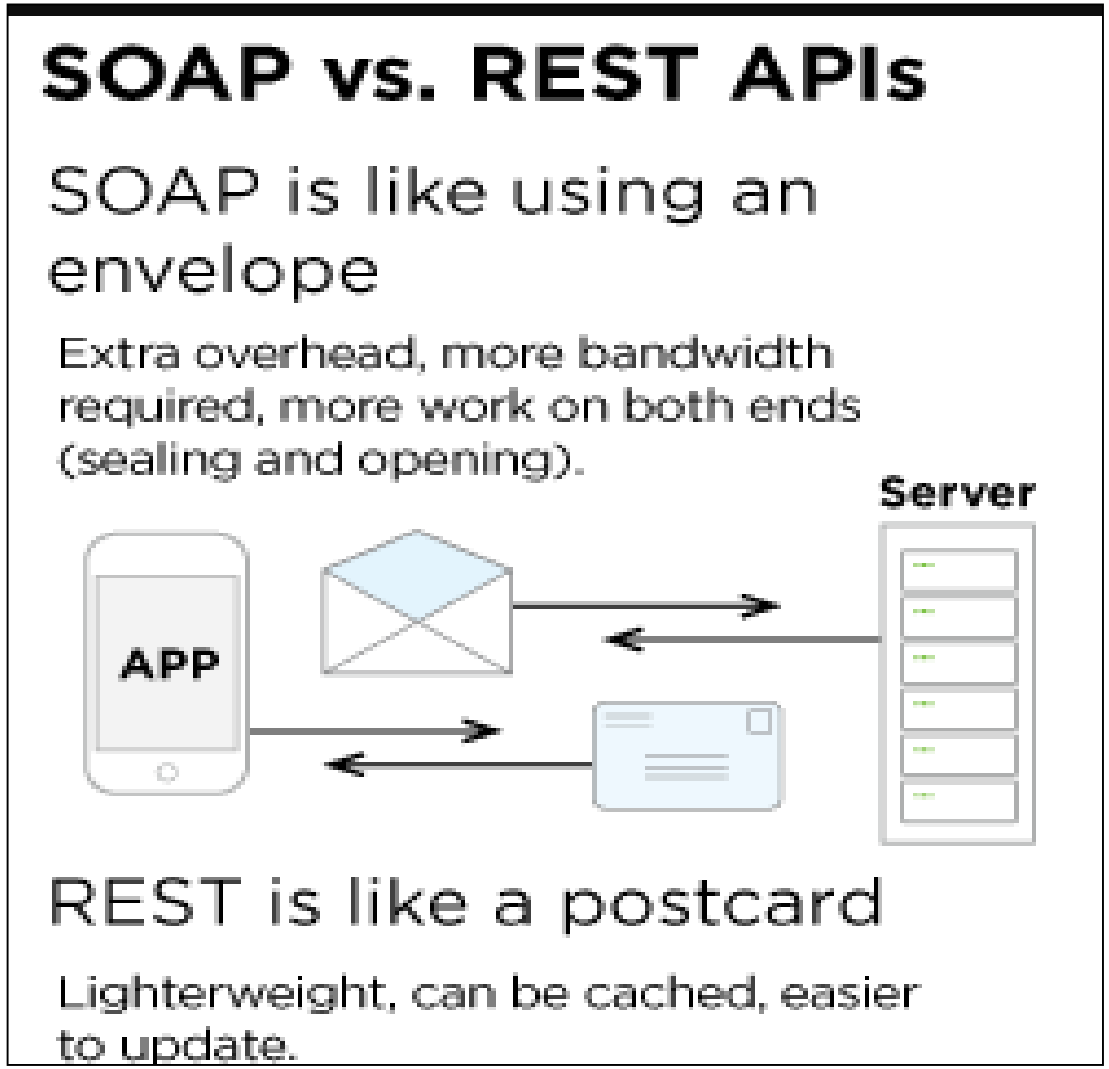
SERVICENOW OVERVIEW

REVATURE

ServiceNow – SOAP Webservices

Webservices -SOAP

- Demo for send and receive data using SOAP web Services
- Main difference between SOAP and REST is where SOAP uses extra overhead and provides envelope for sealing at one end and opening at another end.
- SOAP uses Web Services Description Language which is standard way of describing web service



Webservices -SOAP

- A WSDL file typically consists of the following sections:
- The types element - which defines the data types (XML elements) that are used by the web service.
- The message elements - each of which defines a message exchanged with the web service.
- portType elements - which combine multiple messages into a single operation - for synchronous operations, this is usually one input and one output.
- The binding element - which defines exactly how each operation will take place over the network (SOAP, in this example).
- The service element - which says where the service can be accessed from - in other words, its endpoint.

WSDL Elements

- **Definition** – It is the root element of all WSDL documents. It defines the name of the web service, declares multiple namespaces used throughout the remainder of the document, and contains all the service elements described here.
- **Data types** – The data types to be used in the messages are in the form of XML schemas.
- **Message** – It is an abstract definition of the data, in the form of a message presented either as an entire document or as arguments to be mapped to a method invocation.
- **Operation** – It is the abstract definition of the operation for a message, such as naming a method, message queue, or business process, that will accept and process the message.
- **Port type** – It is an abstract set of operations mapped to one or more end-points, defining the collection of operations for a binding; the collection of operations, as it is abstract, can be mapped to multiple transports through various bindings.
- **Binding** – It is the concrete protocol and data formats for the operations and messages defined for a particular port type.
- **Port** – It is a combination of a binding and a network address, providing the target address of the service communication.
- **Service** – It is a collection of related end-points encompassing the service definitions in the file; the services map the binding to the port and include any extensibility definitions.

Webservices -SOAP

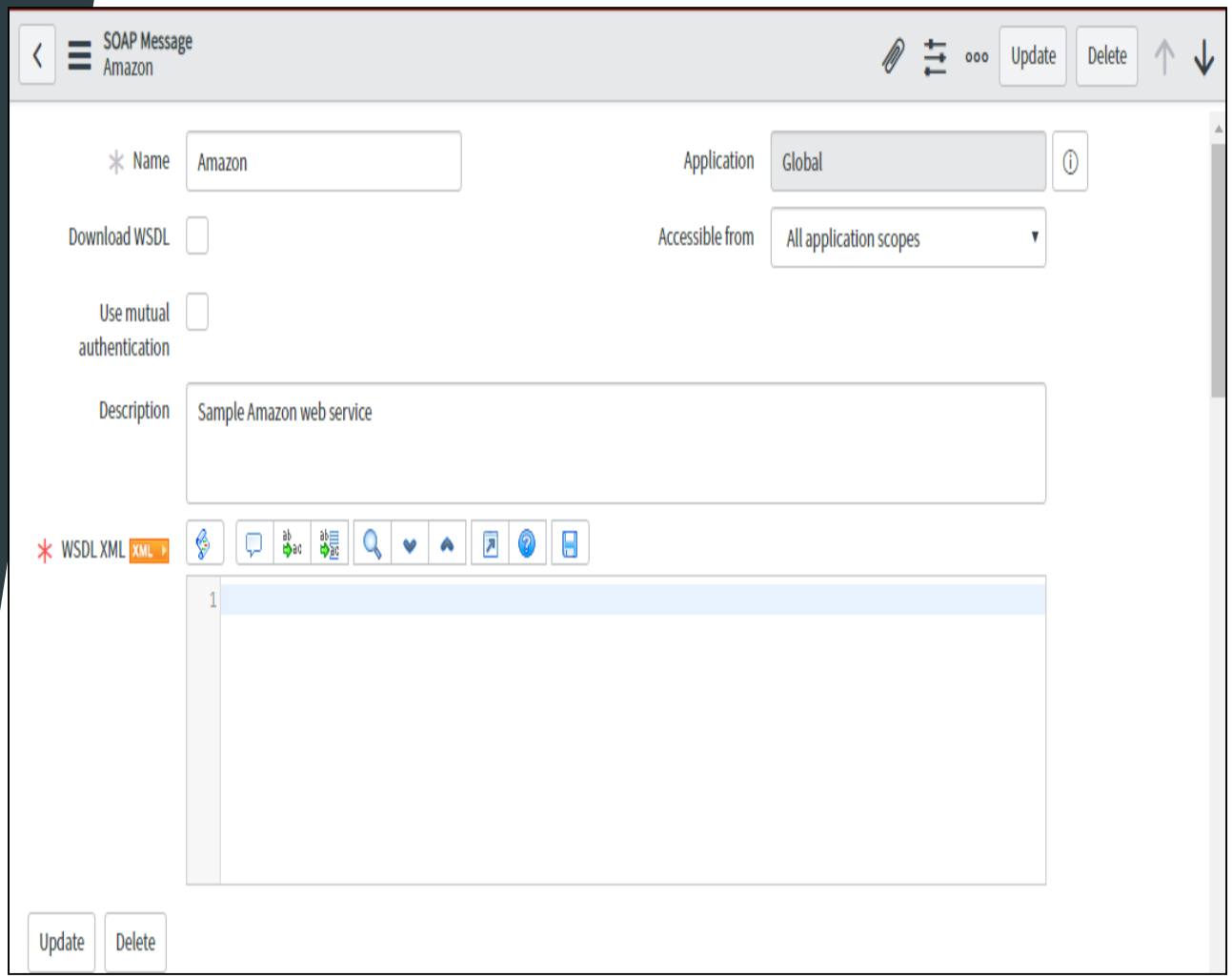
- We will look in the demo how to generate SOAP message records,create outbound SOAP messages,Populate SOAP message functions,variable substituiton and creating SOAPMessageV2scripting API to call SOAP web services.

Webservices -SOAP

- Information needed to send SOAP requests are stored in SOAP message records.
- Each record stores in the format of WSDL
- 2 ways to provide WSDL to SOAP message

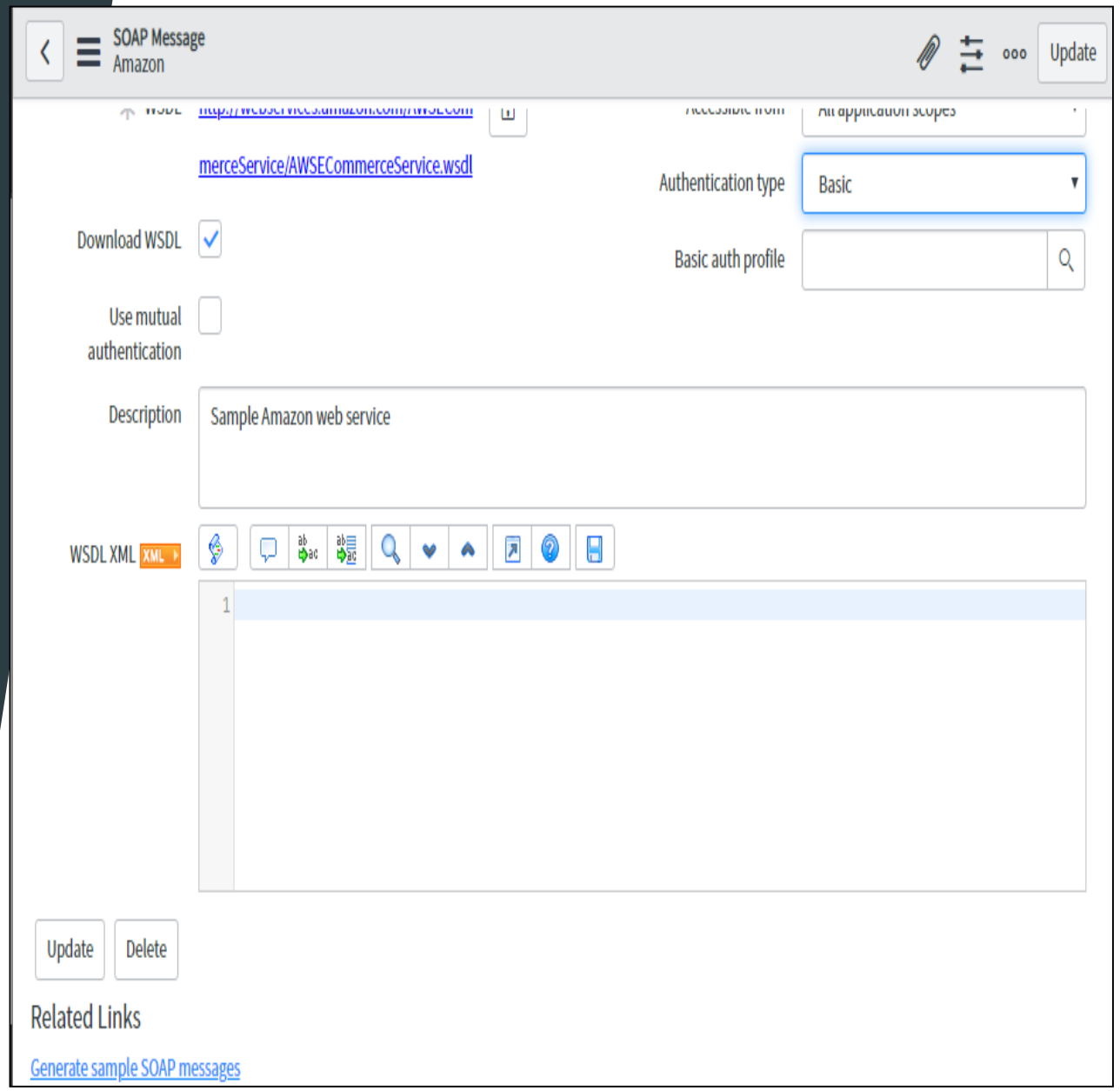
  - Download WSDL

  -Copy Paste the entire WSDL XML text



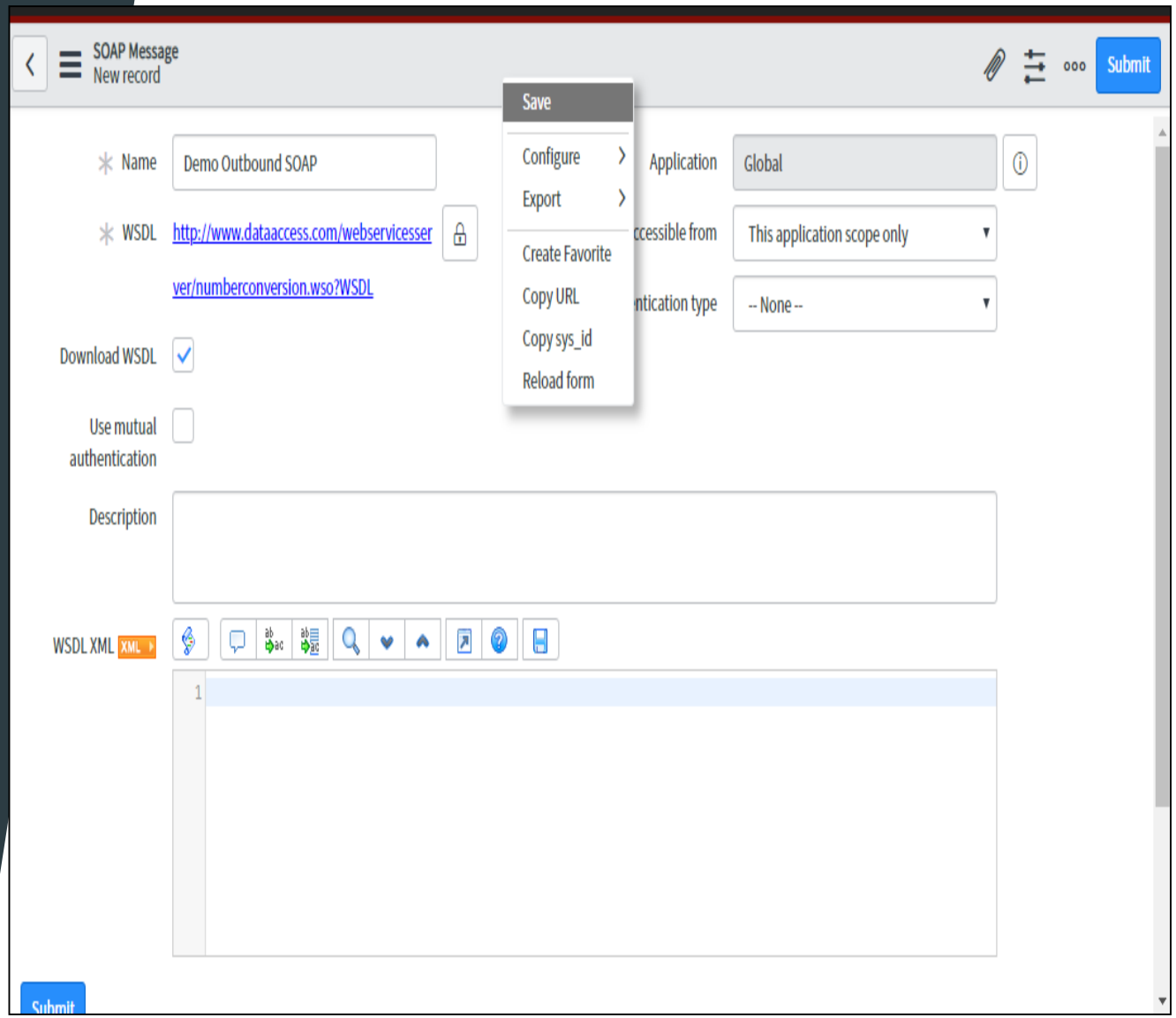
Webservices -SOAP

- If we are downloading WSDL via URL,the URL must be accessible from SNOW instance and cannot be behind a firewall
- It provides basic authentication for WSDL retrieval
- After we have WSDL file location or have the WSDL document, we go ahead and generate SOAP message functions for each operations defined in the WSDL



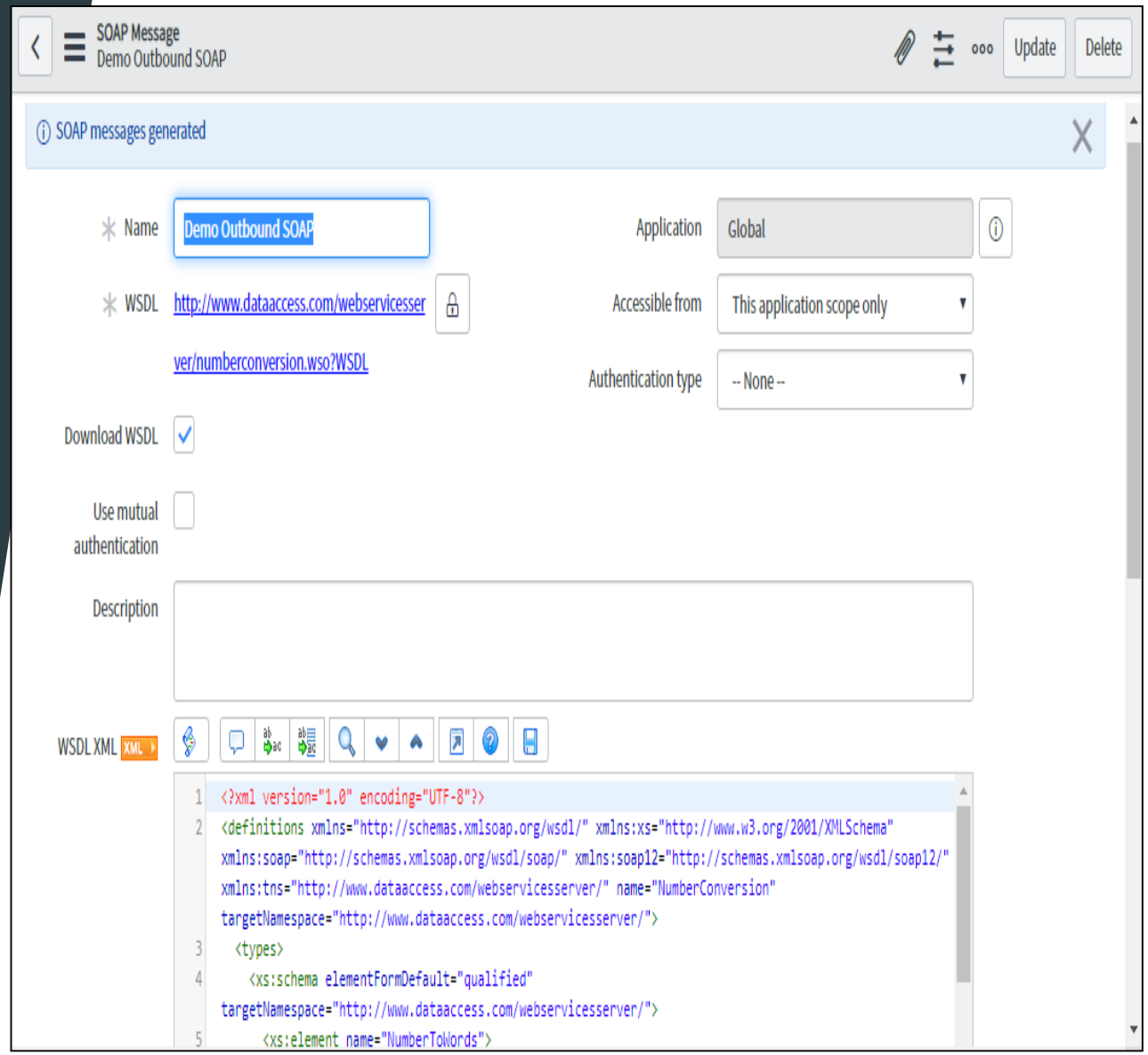
Webservices -SOAP

- Lets create a new SOAP message Demo Outbound SOAP.
- We will use WSDL URL http://www.dataaccess.com/webservicesserver/numberconversion.wso?WSDL
- Unlock the WSDL field and paste the WSDL URL.Locking it back prevents accidental changes to the URL
- Save it and generate sample SOAP messages



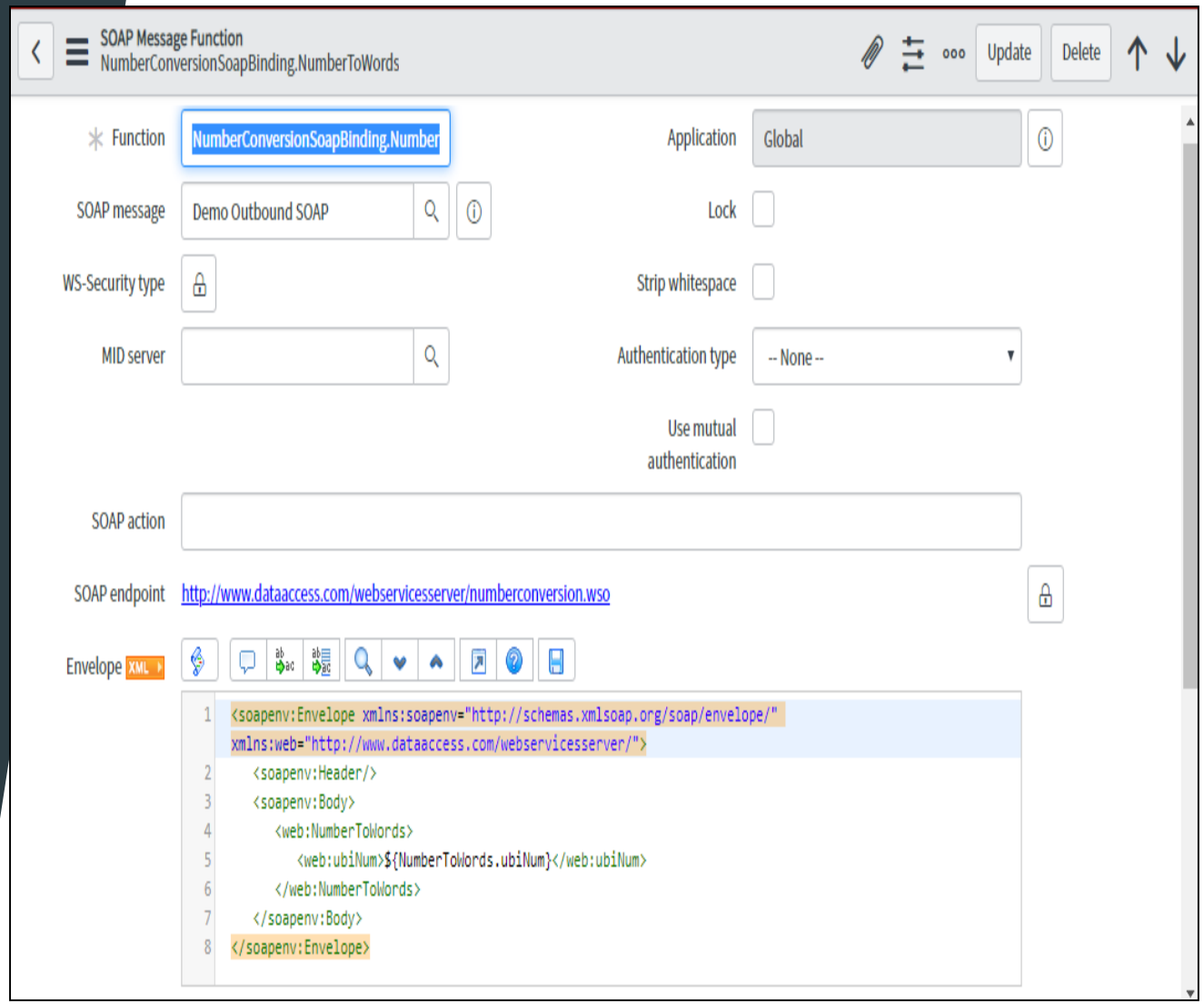
Webservices -SOAP

- This populates WSDL XML field and produces SOAP message funtions based on the WSDL.



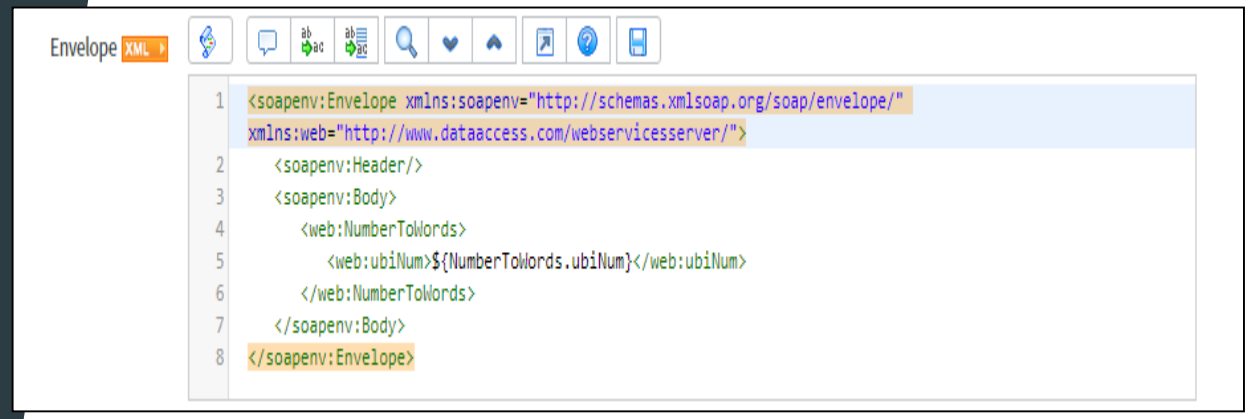
Webservices -SOAP

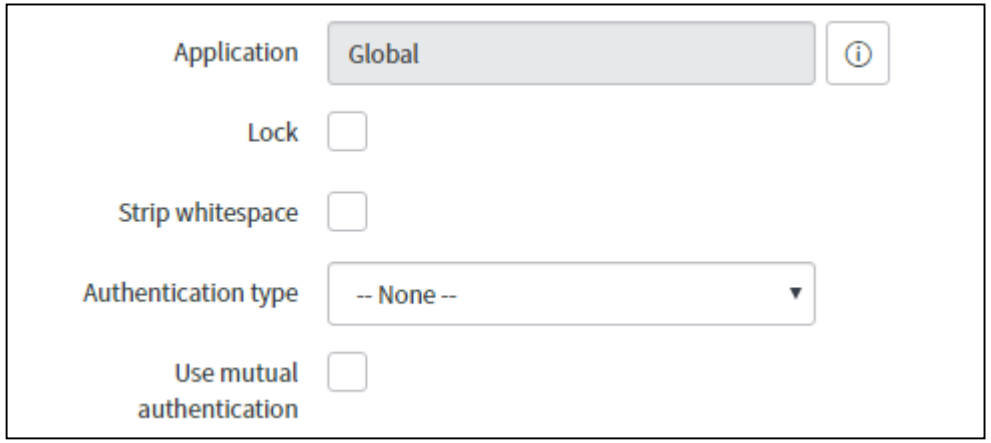
- In these records some fields are auto-populated from the WSDL.
- NumberConversionSoapBinding.NumberToWords is the SOAP function to invoke on the SOAP Webservice.
- SOAP action header is available to be set if the SOAP web service requires it.
- SOAP end point specifies the URL of the SOAP Webservice



Webservices -SOAP

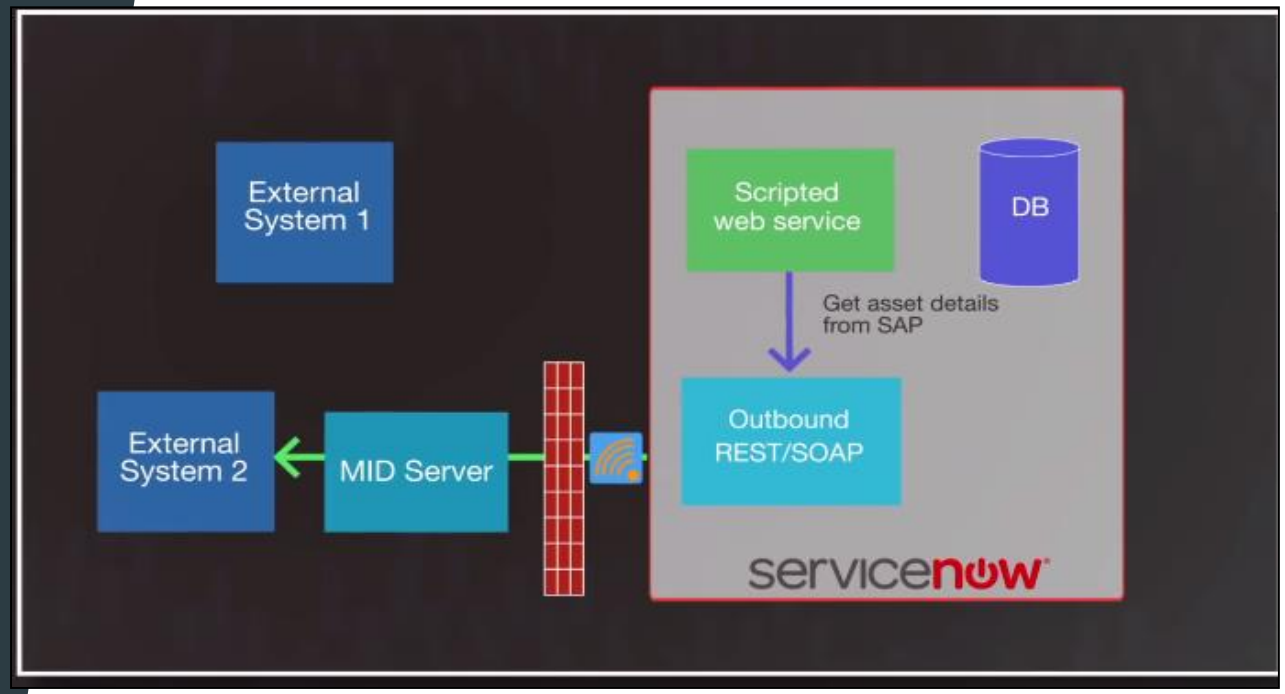
- Envelope contains the payload to be sent to the webservice.
- If the third party requires WS security and use mutual Authentication type information can be configured here.
- https://docs.servicenow.com/bundle/london-application-development/page/integrate/inbound-soap/concept/c_WS-Security.html

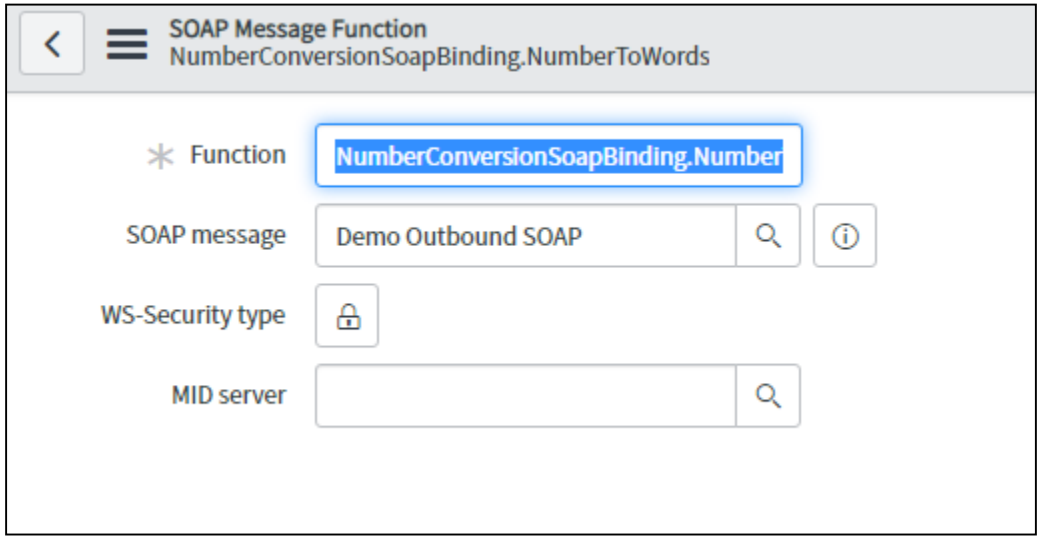




Webservices -SOAP

- If we have outbound message to be sent via midserver then we need to configure MID server and specify it in Mid server in the SOAP message function





Webservices -SOAP

- Mutual Authentication allows client and server to establish trust.

  Note : Mutual authentication is not available when making outbound web service calls through a midsever.

- Selecting mutual authentication provides protocol profile where we can select previously defined protocol to be used.

Webservices -SOAP

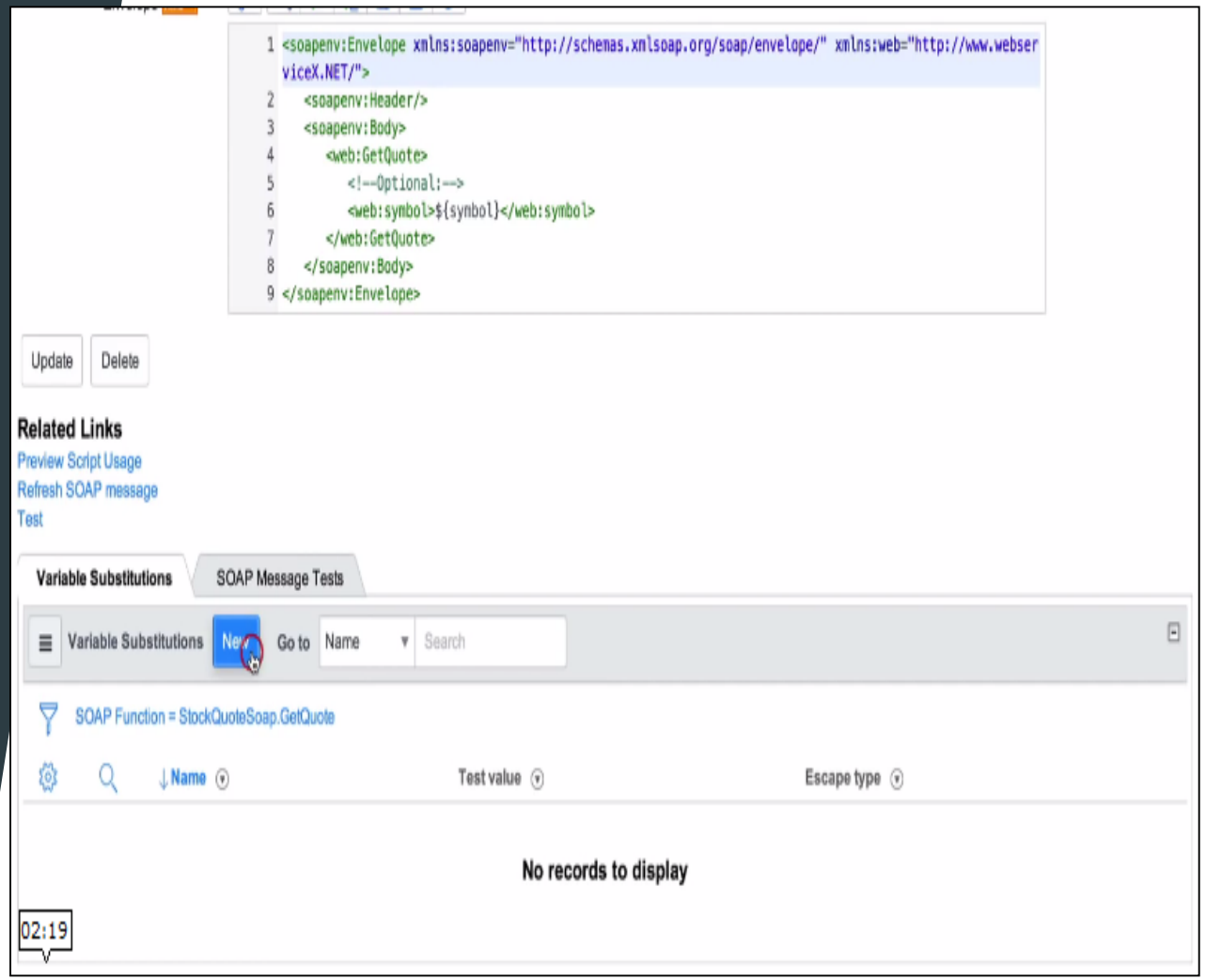
- Envelope field defines the message to be sent to the third party webservice end point
- Envelope data is setup with template parameters for variable substitution.
- Variable substitution allows to accept static content and converting to dynamic by mapping them.

  Format ${variable_name}

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://www.dataaccess.com/webservicesserver/">
   <soapenv:Header/>
   <soapenv:Body>
      <web:NumberToWords>
         <web:ubiNum>${NumberToWords.ubiNum}</web:ubiNum>
      </web:NumberToWords>
   </soapenv:Body>
</soapenv:Envelope>
```

Webservices -SOAP

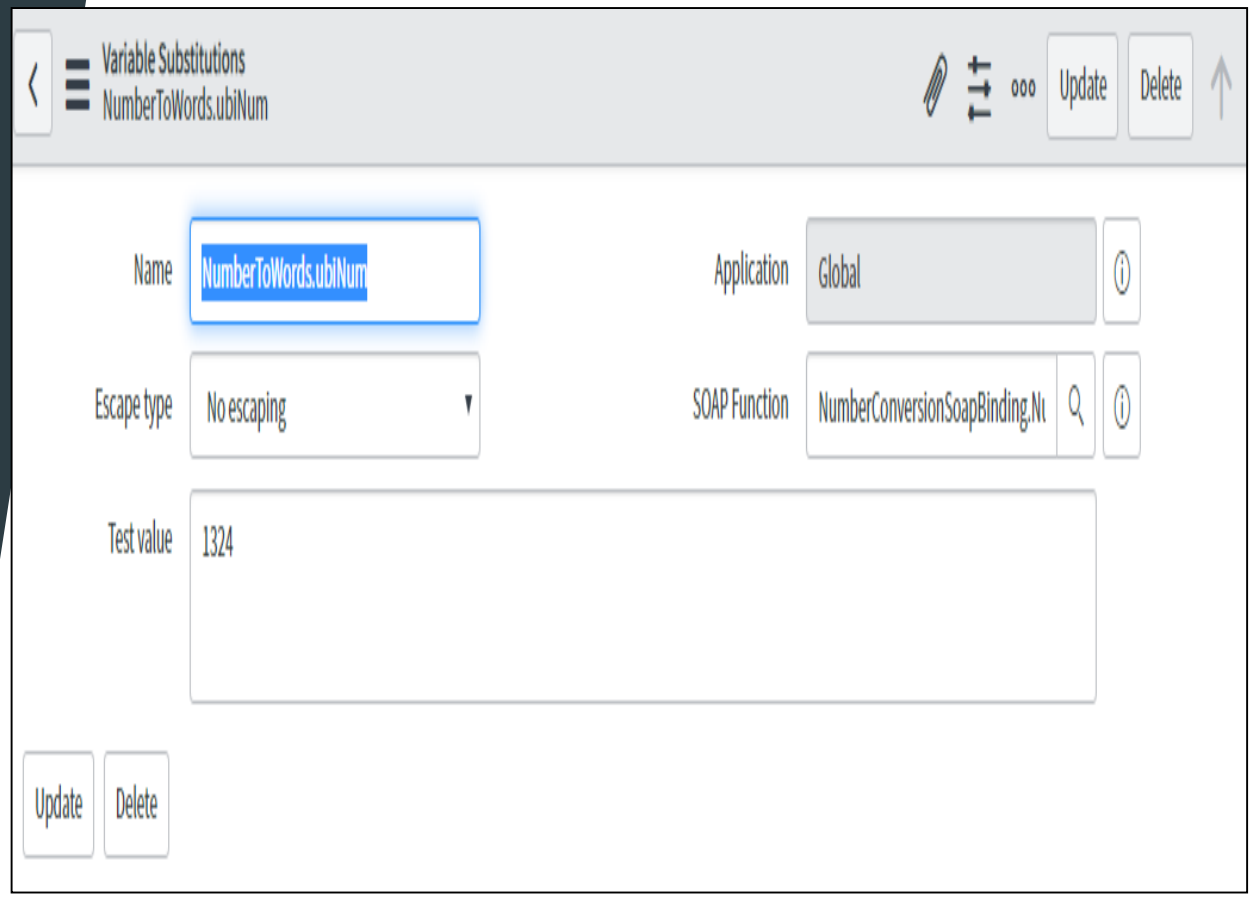
- After the webservice is configured we can use the test UI action to test and resolve any issues using variable substitution.
- Variable substitution is defined in the related list for testing purposes only but can be overridden by scripting API



Webservices -SOAP

- **Lets create new variable for test value for Name :NumberToWords.ubiNum**

  and Test value 1234
- Update the variable

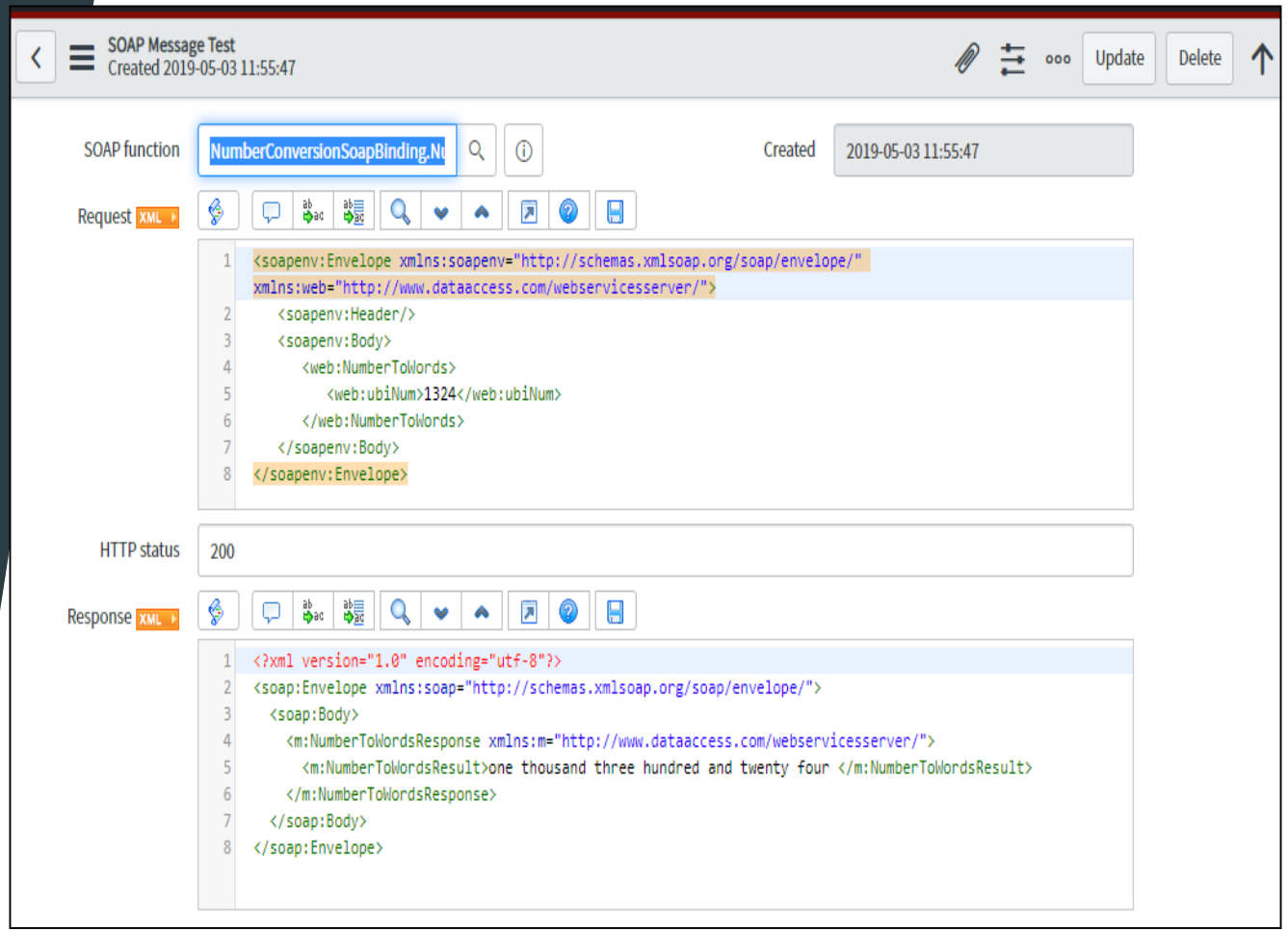


Webservices -SOAP

- Lets now Test the SOAP message function
- It includes the Request sent and the Response that is received
- Here the variable is substituted by number we are specifying
- HTTP status code is reported
- In the response we observe the data which was required from the third party web service.
- Hence this outbound webservice is working fine

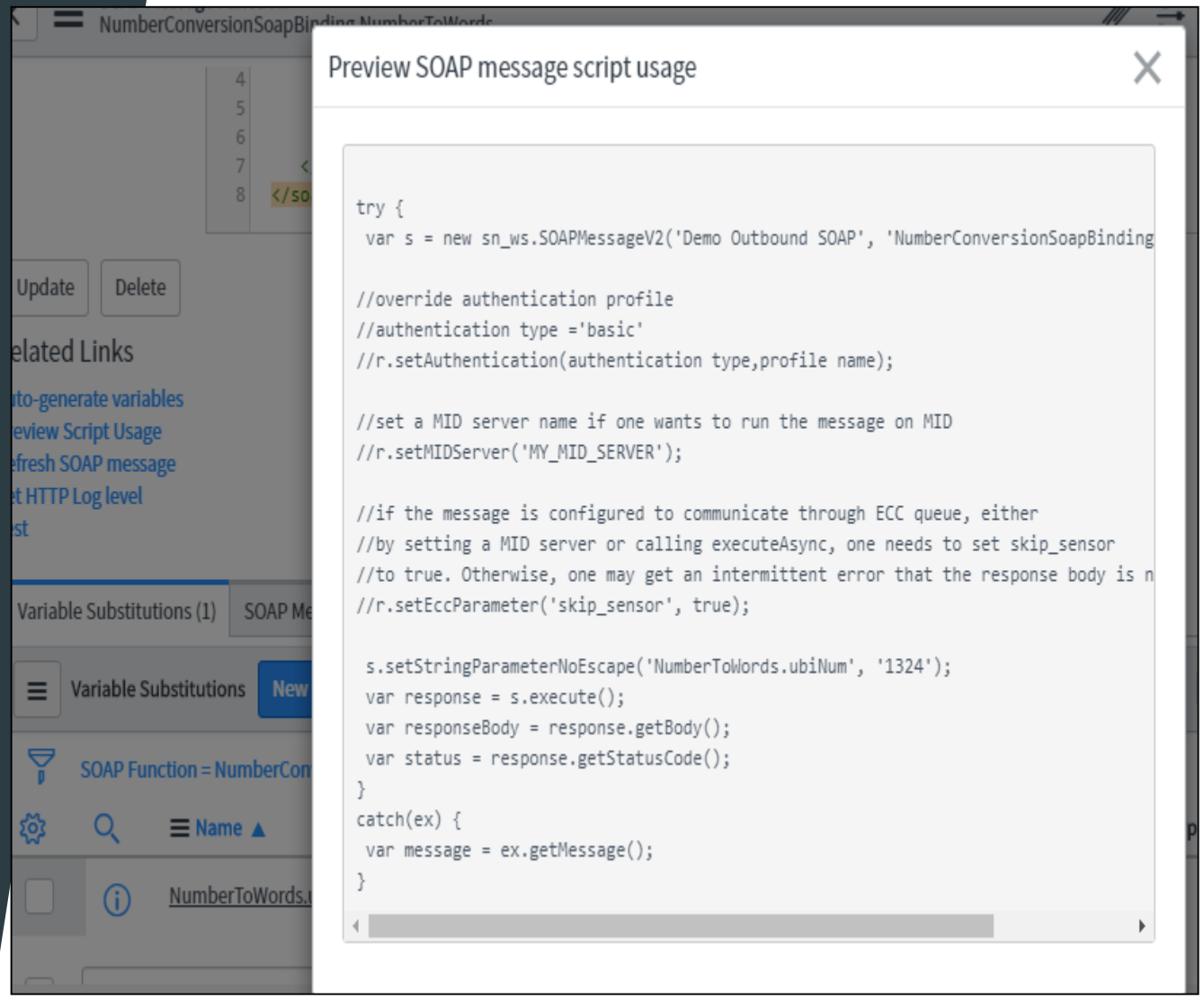
Related Links

Auto-generate variables
Preview Script Usage
Refresh SOAP message
Set HTTP Log level
Test



Webservices -SOAP

- In the SOAP message function Related links clicking on Preview script usage displays the script which is used to call the SOAP message.
- We observe here that the setStringParameter is used to override the values for variable substitution at run time.
- We can copy and use this script anywhere that run server side script.Ex:Business rules,Script includes,script actions and Background scripts



Webservices -SOAP

- The response body provides the response we saw when we tested this function.
- Status code is retrieved by he method var status = response.getStatusCode();
- The script uses SOAP messageV2 API which has different available functionality
- https://docs.servicenow.com/app_store/dev_portal/API_reference/SOAPMessageV2/concept/c_SOAPMessageV2API.html

```
try {
 var s = new sn_ws.SOAPMessageV2('Demo Outbound SOAP', 'NumberConversionSoapBinding

//override authentication profile
//authentication type ='basic'
//r.setAuthentication(authentication type,profile name);

//set a MID server name if one wants to run the message on MID
//r.setMIDServer('MY_MID_SERVER');

//if the message is configured to communicate through ECC queue, either
//by setting a MID server or calling executeAsync, one needs to set skip_sensor
//to true. Otherwise, one may get an intermittent error that the response body is n
//r.setEccParameter('skip_sensor', true);

 s.setStringParameterNoEscape('NumberToWords.ubiNum', '1324');
 var response = s.execute();
 var responseBody = response.getBody();
 var status = response.getStatusCode();
}
catch(ex) {
 var message = ex.getMessage();
}
```

Thankyou