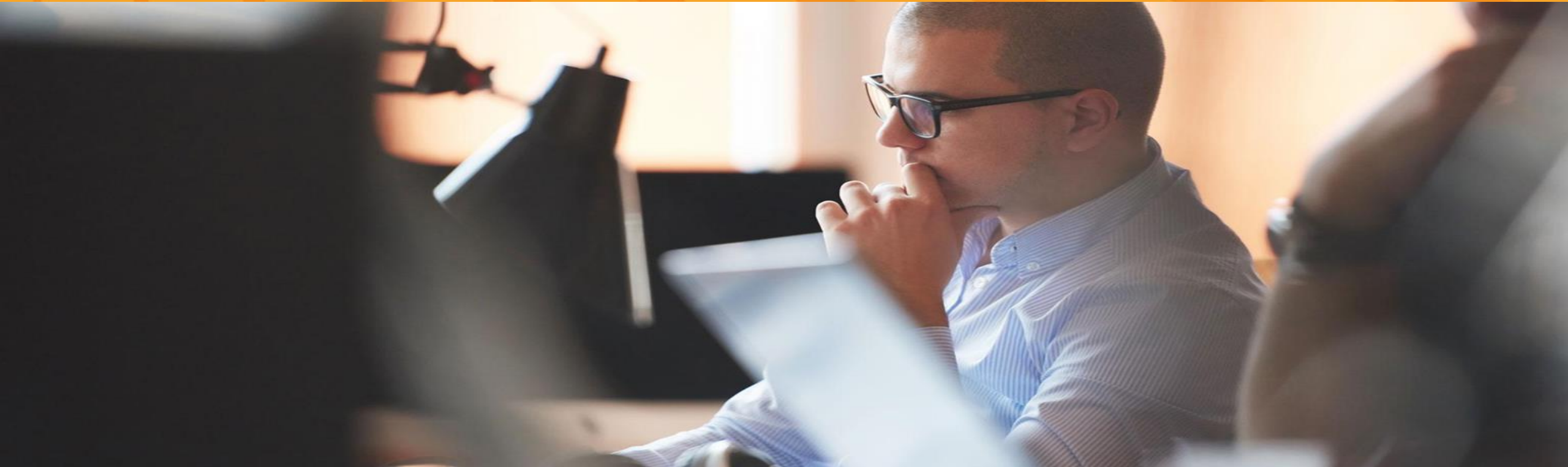


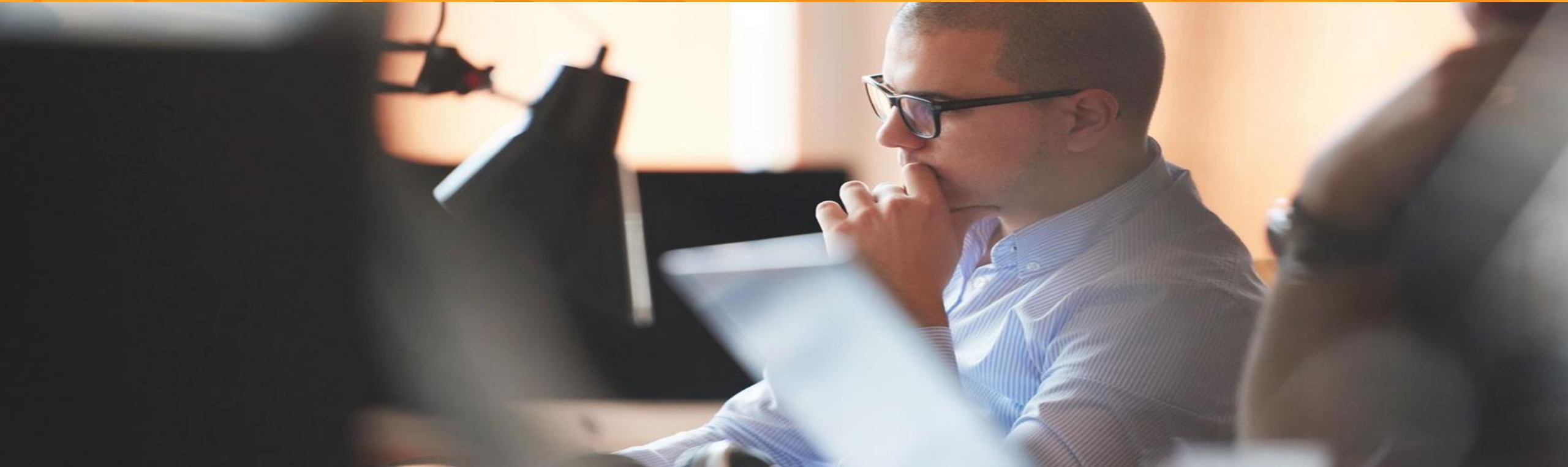


SERVICENOW OVERVIEW





ServiceNow – SOAP Webservices



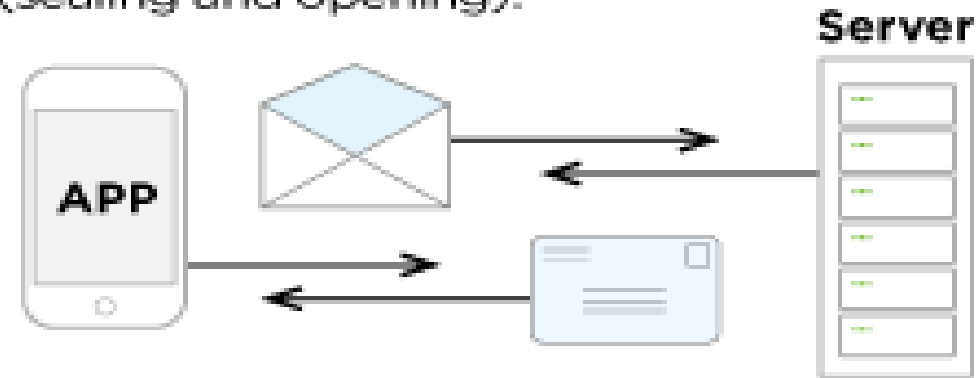
Webservices -SOAP

- ▶ 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

SOAP vs. REST APIs

SOAP is like using an envelope

Extra overhead, more bandwidth required, more work on both ends (sealing and opening).



REST is like a postcard

Lighterweight, can be cached, easier to update.

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.

- SOAP message records
- Creating outbound SOAP messages
- Populating the SOAP Message Functions related list
- Using variable substitution
- Testing the SOAP message
- Previewing the script to use across the platform
- Using the SOAPMessageV2 scripting 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

The screenshot shows a web application interface for managing SOAP messages. The title bar indicates 'SOAP Message Amazon'. The main form contains the following fields and controls:

- Name:** A text input field containing 'Amazon'.
- Application:** A dropdown menu set to 'Global'.
- Download WSDL:** A checkbox that is currently unchecked.
- Use mutual authentication:** A checkbox that is currently unchecked.
- Accessible from:** A dropdown menu set to 'All application scopes'.
- Description:** A text area containing 'Sample Amazon web service'.
- WSDL XML:** A section with a red asterisk icon and a tab labeled 'WSDL XML'. Below this is a large text area for pasting XML content, with a line number '1' visible at the top.

At the bottom of the form, there are 'Update' and 'Delete' buttons. The top right of the interface includes a toolbar with icons for editing, deleting, and navigating, along with 'Update' and 'Delete' buttons.

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

The screenshot shows the 'SOAP Message' configuration page in the Amazon Snow Console. The page title is 'SOAP Message Amazon'. At the top right, there are icons for a document, a list, and a menu, along with an 'Update' button. The main configuration area includes a 'WSDL' field with a URL: <http://mws.amazonaws.com/mwsdl.com/mercerService/AWSECommerceService.wsdl>. Below this, there are checkboxes for 'Download WSDL' (checked) and 'Use mutual authentication' (unchecked). To the right, there is a dropdown for 'Authentication type' set to 'Basic' and a search box for 'Basic auth profile'. A 'Description' field contains the text 'Sample Amazon web service'. Below the description, there is a 'WSDL XML' section with a list of XML files, currently showing one item. At the bottom, there are 'Update' and 'Delete' buttons, and a 'Related Links' section with a link to 'Generate sample SOAP messages'.

SOAP Message
Amazon

WSDL <http://mws.amazonaws.com/mwsdl.com/mercerService/AWSECommerceService.wsdl>

Download WSDL ☒

Use mutual authentication ☐

Authentication type Basic

Basic auth profile

Description Sample Amazon web service

WSDL XML XML

1

Update Delete

Related Links

[Generate sample SOAP messages](#)

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

The screenshot shows a web application interface for configuring a SOAP message. The title bar indicates "SOAP Message New record". A "Save" button is in the top right corner. The form contains the following fields and controls:

- Name:** "Demo Outbound SOAP"
- WSDL:** <http://www.dataaccess.com/webservicesserver/numberconversion.wso?WSDL>. A lock icon is present to the right of the URL.
- Download WSDL:** ☒
- Use mutual authentication:** ☐
- Description:** An empty text area.
- Application:** "Global"
- Accessible from:** "This application scope only"
- Authentication type:** "-- None --"

A context menu is open over the WSDL field, showing the following options:




- Save
- Configure >
- Export >
- Create Favorite
- Copy URL
- Copy sys_id
- Reload form

At the bottom, there is a "WSDL XML" section with a toolbar containing icons for XML, JSON, and other data formats. Below the toolbar is a large text area for the WSDL content, with a tab labeled "1". A "Submit" button is located at the bottom left of the form.

Webservices -SOAP

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

SOAP Message
Demo Outbound SOAP



UpdateDelete

SOAP messages generated

X

* Name

Demo Outbound SOAP

Application

Global

Info

* WSDL

<http://www.dataaccess.com/webservicesserver/ver/numberconversion.wso?WSDL>

Lock

Accessible from

This application scope only

Authentication type

-- None --

Download WSDL

☒











Use mutual authentication

☐

Description

WSDL XML

XML



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:tns="http://www.dataaccess.com/webservicesserver/" name="NumberConversion"
  targetNamespace="http://www.dataaccess.com/webservicesserver/">
3   <types>
4     <xs:schema elementFormDefault="qualified"
  targetNamespace="http://www.dataaccess.com/webservicesserver/">
5       <xs:element name="NumberToWords">
```

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

The screenshot shows the configuration interface for a SOAP Message Function. The title bar indicates the function is 'NumberConversionSoapBinding.NumberToWords'. The interface includes several fields and controls:

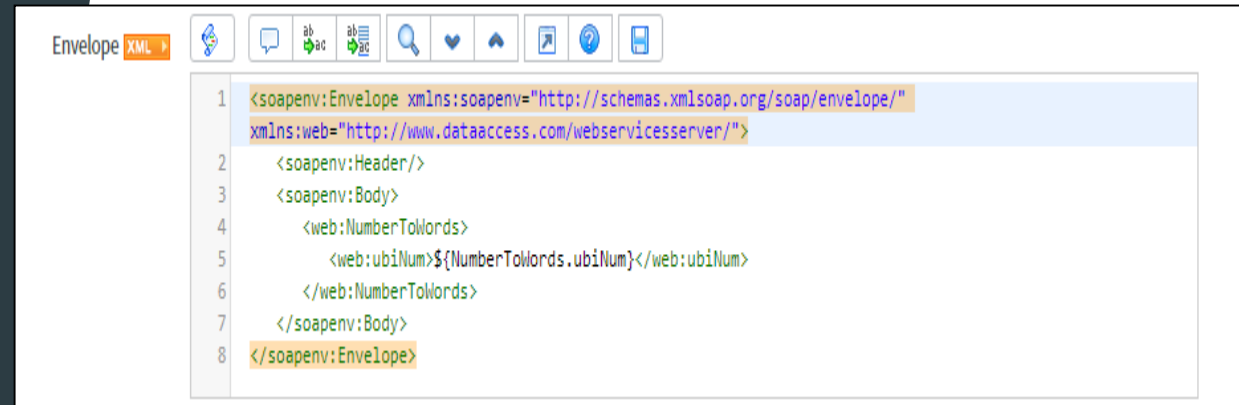
- Function:** A dropdown menu showing 'NumberConversionSoapBinding.Number'.
- Application:** A dropdown menu showing 'Global'.
- SOAP message:** A text field containing 'Demo Outbound SOAP'.
- Lock:** A checkbox that is currently unchecked.
- WS-Security type:** A dropdown menu showing a lock icon.
- Strip whitespace:** A checkbox that is currently unchecked.
- MID server:** A text field with a search icon.
- Authentication type:** A dropdown menu showing '-- None --'.
- Use mutual authentication:** A checkbox that is currently unchecked.
- SOAP action:** An empty text field.
- SOAP endpoint:** A text field containing the URL 'http://www.dataaccess.com/webservicesserver/numberconversion.wso'.
- Envelope:** A section with a toolbar and a text area containing XML code.

The XML code in the Envelope section is as follows:

```
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
2   xmlns:web="http://www.dataaccess.com/webservicesserver/">
3   <soapenv:Header/>
4   <soapenv:Body>
5     <web:NumberToWords>
6       <web:ubiNum>${NumberToWords.ubiNum}</web:ubiNum>
7     </web:NumberToWords>
8   </soapenv:Body>
9 </soapenv:Envelope>
```

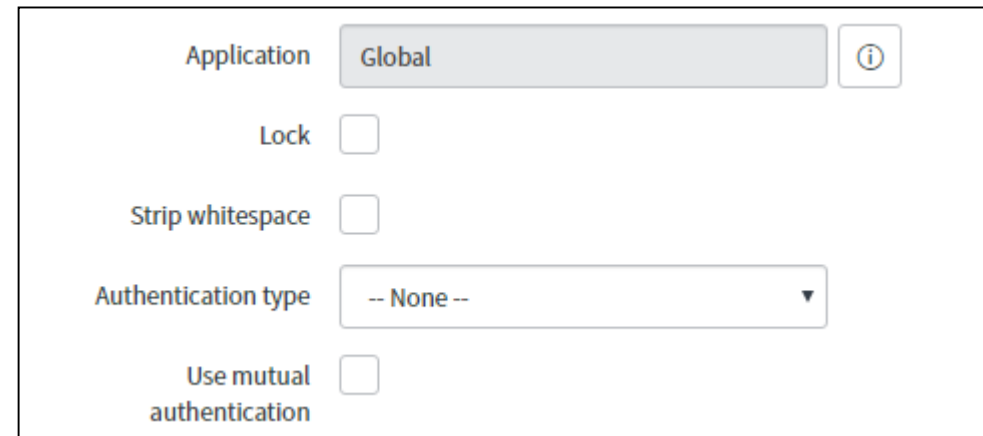
Webservices -SOAP

- ▶ 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



The screenshot shows a code editor with a toolbar at the top. The XML content is as follows:

```
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://www.dataaccess.com/webservicesserver/"
2   <soapenv:Header/>
3   <soapenv:Body>
4     <web:NumberToWords>
5       <web:ubiNum>${NumberToWords.ubiNum}</web:ubiNum>
6     </web:NumberToWords>
7   </soapenv:Body>
8 </soapenv:Envelope>
```

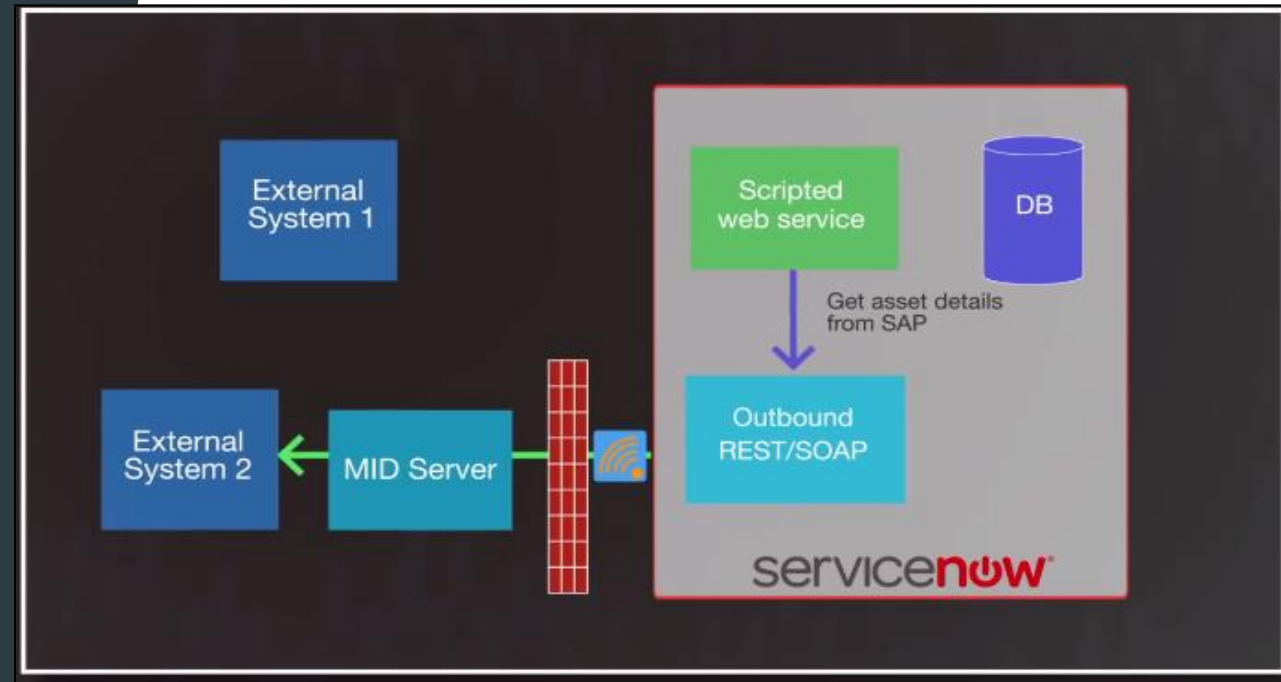


The screenshot shows a configuration form with the following fields and options:

- Application: Global (with an information icon)
- Lock: ☐
- Strip whitespace: ☐
- Authentication type: -- None -- (dropdown menu)
- Use mutual authentication: ☐

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



< ≡ SOAP Message Function
NumberConversionSoapBinding.NumberToWords

* Function

SOAP message

WS-Security type


MID server

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.

Use mutual authentication	<input checked="" type="checkbox"/>
Protocol profile	<input type="text"/> 

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}

```
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:web="http://www.dataaccess.com/webservicesserver/">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <web:NumberToWords>
5       <web:ubiNum>${NumberToWords.ubiNum}</web:ubiNum>
6     </web:NumberToWords>
7   </soapenv:Body>
8 </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

The screenshot displays a web application interface for managing SOAP messages. At the top, a text area contains an XML SOAP envelope with the following structure:

```
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://www.webserviceX.NET/">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <web:GetQuote>
5       <!--Optional:-->
6       <web:symbol>${symbol}</web:symbol>
7     </web:GetQuote>
8   </soapenv:Body>
9 </soapenv:Envelope>
```

Below the XML editor are 'Update' and 'Delete' buttons. A 'Related Links' section provides links for 'Preview Script Usage', 'Refresh SOAP message', and 'Test'. The main interface is divided into two tabs: 'Variable Substitutions' and 'SOAP Message Tests'. The 'Variable Substitutions' tab is active, showing a table with columns for 'Name' and 'Search'. A 'New' button is highlighted with a red circle. Below the table, a filter 'SOAP Function = StockQuoteSoap.GetQuote' is applied. The table is currently empty, displaying 'No records to display'. A timer in the bottom left corner shows '02:19'.

Webservices -SOAP

- ▶ Lets create new variable for test value for Name :NumberToWords.ubiNum and Test value 1234
- ▶ Update the variable

The screenshot shows the 'Variable Substitutions' dialog box for the variable 'NumberToWords.ubiNum'. The dialog has a title bar with a back arrow, a menu icon, the text 'Variable Substitutions' and 'NumberToWords.ubiNum', and icons for edit, expand/collapse, and a list. There are 'Update' and 'Delete' buttons in the top right. The main area contains four fields: 'Name' (text box with 'NumberToWords.ubiNum'), 'Application' (dropdown menu with 'Global'), 'Escape type' (dropdown menu with 'No escaping'), and 'SOAP Function' (text box with 'NumberConversionSoapBinding.N'). Each of these four fields has an information icon to its right. Below these fields is a large 'Test value' text area containing '1234'. At the bottom left are 'Update' and 'Delete' buttons.

Field	Value
Name	NumberToWords.ubiNum
Application	Global
Escape type	No escaping
SOAP Function	NumberConversionSoapBinding.N
Test value	1234

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](#)

The screenshot displays a web application interface for testing SOAP messages. The title bar indicates 'SOAP Message Test' and 'Created 2019-05-03 11:55:47'. The main area shows the 'SOAP function' as 'NumberConversionSoapBinding.N' and the 'Created' timestamp as '2019-05-03 11:55:47'. Below this, the 'Request' is shown in XML format, and the 'Response' is also shown in XML format. The 'HTTP status' is reported as '200'.

SOAP function: NumberConversionSoapBinding.N

Created: 2019-05-03 11:55:47

Request XML:

```
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
2   xmlns:web="http://www.dataaccess.com/webservicesserver/"
3   >
4   <soapenv:Header/>
5   <soapenv:Body>
6     <web:NumberToWords>
7       <web:ubiNum>1324</web:ubiNum>
8     </web:NumberToWords>
9   </soapenv:Body>
10 </soapenv:Envelope>
```

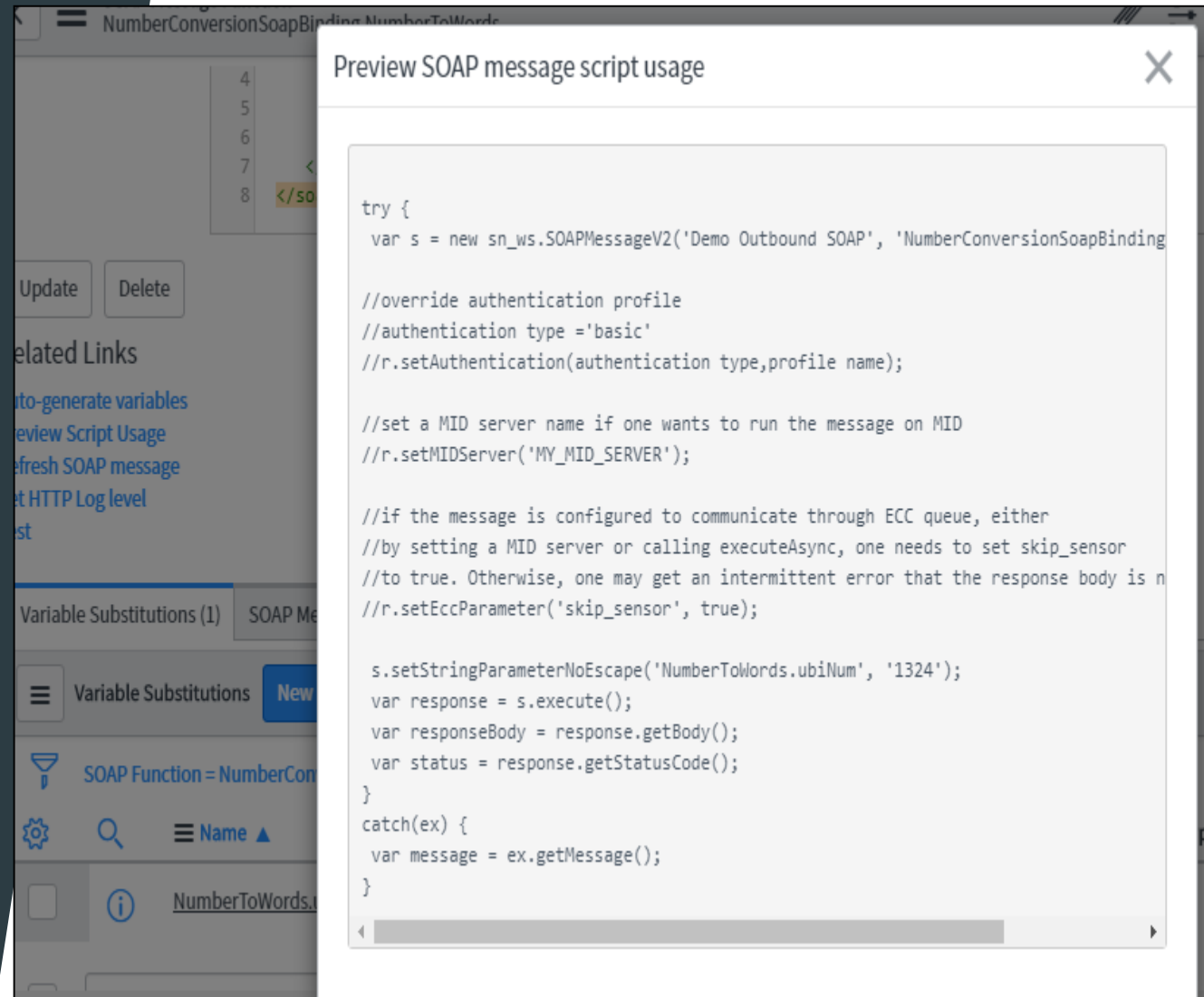
HTTP status: 200

Response XML:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
3   >
4   <soap:Body>
5     <m:NumberToWordsResponse xmlns:m="http://www.dataaccess.com/webservicesserver/"
6       <m:NumberToWordsResult>one thousand three hundred and twenty four</m:NumberToWordsResult>
7     </m:NumberToWordsResponse>
8   </soap:Body>
9 </soap:Envelope>
```

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



The screenshot displays a web application interface for configuring a SOAP message function. The main window is titled "NumberConversionSoapBinding-NumberToWords". It features a "Preview SOAP message script usage" dialog box in the foreground, which contains a JavaScript script. The script is used to call the SOAP message function and handle the response. The background window shows the "NumberToWords" function configuration, including a "Variable Substitutions" section and a "SOAP Function" dropdown menu.

```
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();  
}
```

Webservices -SOAP

- ▶ The response body provides the response we saw when we tested this function.
- ▶ Status code is retrieved by the 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