

Question

[Tom Philippi](#) · Aug 15, 2017

Adding soap header to imported WSDL

I have an imported WSDL from Salesforce and I am trying to send a login-request to the Salesforce-servers via InterSystems. The login requires a username and a password in the body, and an organizationId in the header. I am having trouble filling the organizationId in the header.

Via SoapUI I can send a successful message to the Salesforce-servers. The message should look like this:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:enterprise.soap.sforce.com">
  <soapenv:Header>
    <urn:LoginScopeHeader>
      <urn:organizationId>MyOrganizationId</urn:organizationId>
    </urn:LoginScopeHeader>
  </soapenv:Header>
  <soapenv:Body>
    <urn:login>
      <urn:username>MyUsername</urn:username>
      <urn:password>MyPassword</urn:password>
    </urn:login>
  </soapenv:Body>
</soapenv:Envelope>
```

In InterSystems Studio, I firstly imported the WSDL using the Studio's Soap Add-in. Among other things, this gave the following results:

1. A business operation class containing the login method:

```
Method login(pRequest As Request.loginRequest, Output pResponse As
Response.loginResponse) As %Library.Status
{
  Set ..Adapter.WebServiceClientClass = "SforceService.Soap"
  Set tSC =
  ..Adapter.InvokeMethod("login",..result,pRequest.username,pRequest.password) Quit:$$$ISERR(tSC) tSC
  Set tSC = pRequest.NewResponse(.pResponse) Quit:$$$ISERR(tSC) tSC
  Set pResponse.result=$get(result)
  Quit $$$OK
}
```

2. A webclient class containing the login-method (proxy?_

```
/// Login to the Salesforce.com SOAP Api
Method login(username As %String(REQUIRED=1), password As %String(REQUIRED=1)) As
SforceService.tns.LoginResult(XMLNAME="result") [ Final, ProcedureBlock = 1, SoapBindingStyle =
document, SoapBodyUse = literal, WebMethod ]
{
```

```
Quit ..WebMethod("login").Invoke($this, "", .username, .password)
}
```

3. The webclient class also contains an XdataMap noting that the LoginScopeHeader may be included with the login-request. The LoginScopeHeader contains the organizationId.

```
...
<method name="login">
  <request>
    <header name="LoginScopeHeader" class="SforceService.tns.LoginScopeHeader"/>
  </request>
</method>
...
```

As you can see the Header is referred to, but not included in as a parameter in the method.

Of course, I need to expand the method so that I can pass organizationId as a parameter. A first step in to this solution would be to hard code the organizationId in login-request. In the InterSystems documentation I found an example how to add the header using the HeadersOut-function (<https://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY...>). I implemented this in the web client class as follows:

```
/// Login to the Salesforce.com SOAP Api
Method login(username As %String(REQUIRED=1), password As %String(REQUIRED=1)) As
SforceService.tns.LoginResult(XMLNAME="result") [ Final, ProcedureBlock = 1, SoapBindingStyle =
document, SoapBodyUse = literal, WebMethod ]
{
  Set header=##class(SforceService.tns.LoginScopeHeader).%New()
  Set header.organizationId = "MyOrganizationId"
  Do ..HeadersOut.SetAt(header, " SforceService.tns.LoginScopeHeader")

  Quit ..WebMethod("login").Invoke($this, "", .username, .password)
}
```

However, when I compile and run this code and view the soap log the header is still missing:

```
08/16/2017 10:55:13 *****
Output from Web client with SOAP action =
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xmlns:s='http://www.w3.org/2001/XMLSchema'>
  <SOAP-ENV:Body><login xmlns="urn:enterprise.soap.sforce.com"><username
xsi:type="s:string">test</username><password xsi:type="s:string">test</password></login></SOAP-
ENV:Body>
</SOAP-ENV:Envelope>
```

What did I do wrong?

[#SOAP #Ensemble](#)

Source URL: <https://community.intersystems.com/post/adding-soap-header-imported-wsdl>