Example: Add WS-Addressing elements to SOAP requests

In this example scenario we will add WS-Addressing related elements to the SOAP header of requests.

Do the following:

1. In Integration Studio create an extension and define an action that will handle the WS-Addressing change.
   In example below we defined an action in Integration Studio called "AddWSAddressing" with no input parameters.

![AddWSAddressing](image)

2. Click 'Edit Source Code .NET'. Set the project target framework and add a reference to the System.ServiceModel assembly.

   Enter the code below, replacing the MssAddWSAddressing function placeholder that Integration Studio created for you:

```csharp
// required 'using' statements at the beginning of the file
using System.ServiceModel;
using System.ServiceModel.Channels;
using OutSystems.SOAP.API;

/* ... */

// replace the 'MssAddWSAddressing' function placeholder with the following code
public void MssAddWSAddressing() {
    SOAPClient client = SoapRequest.GetCurrentClient();
    SecurityMode securityMode = SecurityMode.Transport;
    Binding binding = new WSHttpBinding(securityMode);
```
client.Endpoint.Binding = binding;

3. Quit Visual Studio .NET and, back in Integration Studio, publish the extension by clicking the "1-Click Publish" toolbar icon or by pressing F5.

4. In Service Studio, add a reference to the "AddWSAddressing" action of your extension in your application module.

5. In the flow of the SOAP callback of your SOAP Web Service, i.e. the flow of "OnBeforeRequestAdvanced", drag the "AddWSAddressing" action to the flow.

6. Publish the application module and test the application, checking that the WS-Addressing elements are present in the web service requests made by the OutSystems platform for the web service you just configured.