Accessing IPNew Web Services with certificates and claims-based authentication

v1.0

Contents

Introduction

With the introduction of the new version of RBP, Trading Platform, Information Platform, IPData and TSOData, the access of web services will slightly differ from the previous method. Web services can be accessed by using SOAP 1.1. and WS-Trust 1.3. protocols, with certificate and claims-based authentication.

Purpose of the document

The present document provides information on the renewed RBP's web services (server-server connection) for those using these services for preparation and adaptation purposes. The description is valid for the Regional Booking Platform, the Trading Platform and the Information Platform, but it is not valid to the IPData and TSOData platforms.

User Authentication

The services are exclusively available for authenticated users. The authentication of the users is done by an external authentication service (Active Directory Federation Services – AD FS), authentication must be performed prior to invoking the web service. The data package containing the data of the authenticated user (SAML token) must be attached to the web service request message to RBP, TP or IP in order that the server serves these requests.

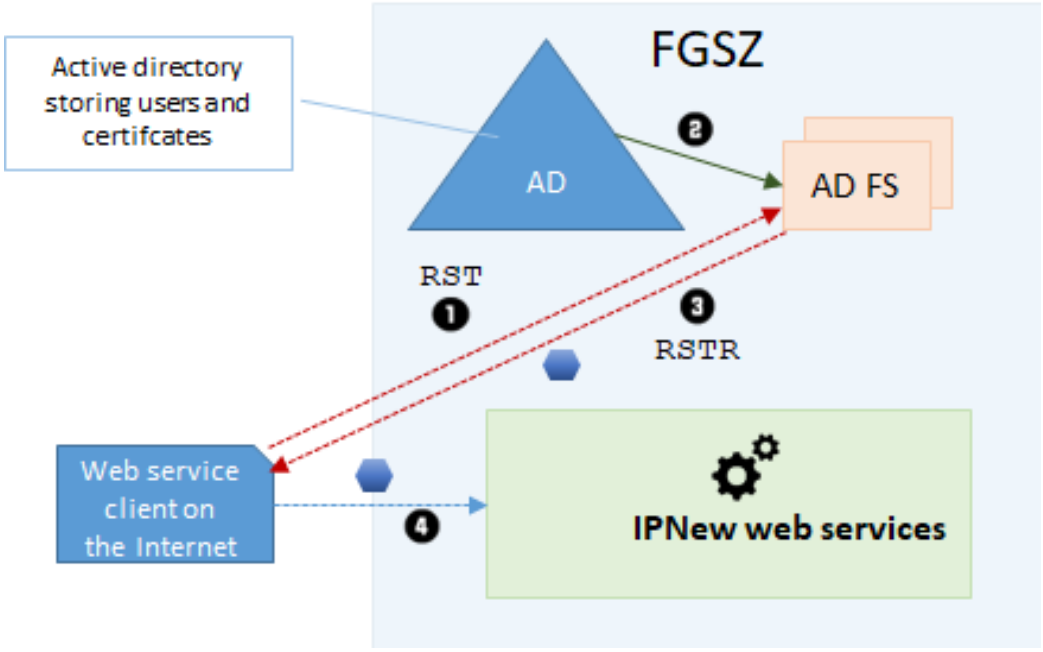
Process of Authentication in Case of SOAP Requests

In case of requesting the RBP, TP or IP web services, the requesting application (client) must use the WS-Trust 1.3. protocol. Although the WS-Trust protocol is fairly complicated and complex, in this case one should get acquainted only with the relevant parts thereof.

On the server side, two components have important role in authentication besides the directory:

- Relying Party (RP): The application or service that performs the business process, in this case RBP, TP or IP.
- Security Token Service (STS or Identity Provider or Issuer): the service performing the authentication and token issuance, in this case the AD FS. In case of the WS-Trust, the STS itself is also a SOAP webservice.

The authentication and requesting of webservice happens in multiple steps as described below:



1.) As a first step, the client requests a special end point of the AD FS, the RST (Request Security Token) method, in which request it describes what kind of token it requests (e.g. SAML 1.1) and which RP it wants to communicate with. The request must contain the authentication data. In the production environments, only certificate-based authentication is foreseen to be permitted.
2.) The AD FS checks the authentication data.

3.) If the authentication data provided are correct, the AD FS provides the required token in a RSTR (Request Security Token Response) response.

4.) The client stores the token, and then attaches it to the RBP, TP or IP webservice requests in the SOAP message (in the security header) within the validity period of the token.

Web Service Addresses in the test environment

ADFS WSDL: https://sts.test.rbp.eu/adfs/services/trust/mex

ADFS endpoint for username/password authentication:
https://sts.test.rbp.eu/adfs/services/trust/13/usernamemixed

ADFS endpoint for X.509 certificate based authentication:
https://sts.test.rbp.eu/adfs/services/trust/13/certificatemixed

Web Service Addresses in the production environment

ADFS WSDL: https://sts.rbp.eu/adfs/services/trust/mex

ADFS endpoint for username/password authentication:
https://sts.rbp.eu/adfs/services/trust/13/usernamemixed

ADFS endpoint for X.509 certificate based authentication:
https://sts.rbp.eu/adfs/services/trust/13/certificatemixed

In case of business web services the address is available in the corresponding interface documentation. The <servername> placeholder should be replaced with the actual value for the respective environment. In the test environment the value of **ipnew.test.fgsz.hu** should be used. In production environment the value of **ipnew.fgsz.hu** should be used

Creating a Sample Web service client with Visual Studio

When creating a web service client, it is recommended to use a modern development tool (e.g. Visual Studio), which significantly simplifies the task by hiding complicated technical details. For testing purposes, a simple Console Application is enough. For leveraging certificate-based authentication, you need an X. 509 client certificate that was previously assigned by FGSZ to your user account. You should also import the certificate with its private key on the client machine into a certificate store (e.g. Current User/Personal) to make it easier to refer to from code.

In this sample application the FGSZNNOService will be used.

app.config settings

Start by creating a Visual Studio Project and adding a Service Reference to it. Provide the WSDL for the Web service when prompted. When adding a Service Reference, Visual Studio sets up the app.config file and also generates code in the background which refers the sts.test.fgsz.hu endpoint of ADFS. Our customers have to replace in the app.config file in the `issuer address`, `issuerMetadata address` and the binding name nodes the `sts.test.fgsz.hu` to `sts.test.rbp.eu`. In the production environment the value of `sts.rbp.eu` should be used.

Here is a sample app.config:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
  </startup>
  <system.serviceModel>
    <bindings>
      <ws2007FederationHttpBinding>
        <binding name="WS2007FederationHttpBinding_IFGSZNNOServiceSoapWrapper">
          <security mode="TransportWithMessageCredential">
            <message establishSecurityContext="false" issuedKeyType="BearerKey">
              <!--<issuer
address="https://sts.test.rbp.eu/adfs/services/trust/13/usernamemixed"
                  binding="ws2007HttpBinding" bindingConfiguration="adfs" />-->
              <issuer
address="https://sts.test.rbp.eu/adfs/services/trust/13/certificatemixed"
                  binding="ws2007HttpBinding" bindingConfiguration="adfs" />
              <issuerMetadata address="https://sts.test.rbp.eu/adfs/services/trust/mex" />
              <tokenRequestParameters>
                <trust:SecondaryParameters xmlns:trust="http://docs.oasis-open.org/ws-
sx/ws-trust/200512">
                  <trust:KeyType xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-
trust/200512">http://docs.oasis-open.org/ws-sx/ws-trust/200512/Bearer</trust:KeyType>
                  <trust:CanonicalizationAlgorithm xmlns:trust="http://docs.oasis-
open.org/ws-sx/ws-trust/200512">http://www.w3.org/2001/10/xml-exc-
c14n#</trust:CanonicalizationAlgorithm>
                  <trust:EncryptionAlgorithm xmlns:trust="http://docs.oasis-open.org/ws-
sx/ws-trust/200512">http://www.w3.org/2001/04/xmlenc#aes256-
cbc</trust:EncryptionAlgorithm>
                </trust:SecondaryParameters>
              </tokenRequestParameters>
            </message>
          </security>
        </binding>
      </ws2007FederationHttpBinding>
      <ws2007HttpBinding>
        <binding name="adfs">
          <security mode="TransportWithMessageCredential">
            <transport clientCredentialType="None" />
            <!--<message clientCredentialType="UserName"
negotiateServiceCredential="false" establishSecurityContext="false" />-->
            <message clientCredentialType="Certificate" negotiateServiceCredential="false"
establishSecurityContext="false" />
          </security>
        </binding>
      </ws2007HttpBinding>
    </bindings>
    <behaviors>
      <endpointBehaviors>
        <behavior name="certificateEndpointBehavior">
          <clientCredentials>
```

```
            <serviceCertificate>
              <authentication certificateValidationMode="None" revocationMode="NoCheck"/>
            </serviceCertificate>
          </clientCredentials>
        </behavior>
      </endpointBehaviors>
    </behaviors>
    <client>
      <endpoint
address="https://ipnew.test.fgsz.hu/Fgsz.Ip.Web.UAT/Services/SoapFGSZNNOService.svc"
          binding="ws2007FederationHttpBinding"
bindingConfiguration="WS2007FederationHttpBinding_IFGSZNNOServiceSoapWrapper"
          contract="FgszService.IFGSZNNOServiceSoapWrapper"
name="WS2007FederationHttpBinding_IFGSZNNOServiceSoapWrapper"
          behaviorConfiguration="certificateEndpointBehavior" />
    </client>
  </system.serviceModel>
  <system.net>
    <defaultProxy enabled="true" useDefaultCredentials="true">
      <!--<bypasslist> … </bypasslist>
      <proxy> … </proxy>
      <module> … </module>-->
    </defaultProxy>
  </system.net>
</configuration>
```

Sample .net/C# code

The following C# code, or similar, can be used to call the service. The sample code (and the app.config above) is used for certificate-based authentication. If you want to use username/password authentication, you must comment out or delete the line below "// Certificate Authentication", then remove comments ("//") from the two lines under "//Windows Authentication" and enter the correct username and password instead of the asterisk characters. In the app.config, replace the "Issuer Address" and "Message clientCredentialType" elements with the username version (lines currently commented out).

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Security;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;

namespace WsClientTest
{
    class Program
    {
        static void Main(string[] args)
        {
            System.Net.ServicePointManager.ServerCertificateValidationCallback =
                delegate (object sender, X509Certificate certificate, X509Chain chain,
SslPolicyErrors sslPolicyErrors)
                { return true; };

            var proxy = new FgszService.FGSZNNOServiceSoapWrapperClient();

            // Certificate authentication
            // Replace "tempClientcert" with the subject of your certificate
```

```
proxy.ClientCredentials.ClientCertificate.SetCertificate(StoreLocation.CurrentUser,
StoreName.My, X509FindType.FindBySubjectName, "tempClientcert");

//proxy.ClientCredentials.ServiceCertificate.Authentication.CertificateValidationMode =
System.ServiceModel.Security.X509CertificateValidationMode.None;
            //proxy.ClientCredentials.ServiceCertificate.Authentication.RevocationMode =
X509RevocationMode.NoCheck;

            // Windows Authentication
            //proxy.ClientCredentials.UserName.UserName = "testuser@ipnewtest.local";
            //proxy.ClientCredentials.UserName.Password = "********";
            string result = proxy.CheckAlive();
            Console.WriteLine(result);

            Console.ReadLine();

        }
    }
}
```

Example of message exchange

By running the sample program, you can monitor traffic similar to the following HTTP message exchange.

Requesting a Token

Request Sample

```
POST https://sts.test.rbp.eu/adfs/services/trust/13/certificatemixed HTTP/1.1
Content-Type: application/soap+xml; charset=utf-8
Host: sts.test.rbp.eu
Content-Length: 4638
Expect: 100-continue
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"><s:Header><a:Action
s:mustUnderstand="1">http://docs.oasis-open.org/ws-sx/ws-
trust/200512/RST/Issue</a:Action><a:MessageID>urn:uuid:f9aa5885-174e-4fa8-8e38-
4b35520078a5</a:MessageID><a:ReplyTo><a:Address>http://www.w3.org/2005/08/addressing/anony
mous</a:Address></a:ReplyTo><a:To s:mustUnderstand="1"
u:Id="_1">https://sts.test.rbp.eu/adfs/services/trust/13/certificatemixed</a:To><o:Securit
y s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"><u:Timestamp u:Id="_0"><u:Created>2018-03-
09T12:35:18.852Z</u:Created><u:Expires>2018-03-
09T12:40:18.852Z</u:Expires></u:Timestamp><o:BinarySecurityToken u:Id="uuid-d4b24929-31e2-
49ad-95a3-b8f4e8275032-1" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-x509-token-profile-1.0#X509v3" EncodingType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
1.0#Base64Binary">MIIDBjCCAfKgAwIBAgIQY2AkvW4NyY9ARCBjGn1CKjAJBgUrDgMCHQUAMBsxGTAXBgNVBAMT
EFJvb3RDYUNsaWVudFRlc3QwHhcNMTgwMzAxMDkxNDIzWhcNMzkxMjMxMjM1OTU5WjAZMRcwFQYDVQQDEw50ZW1wQ2
xpZW50Y2VydDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANVOrgbeL/kgdRrD3DnnQZ6GO7+LaLDAPum/
bdgFeA4LkCiYUP6QHGCUw57kEZ1ZaKU+jJHRfxZzbussuuILGNJaw3gEy533R7O2MNeDyglvSSYHU9mnfKkirO/Jj/
asn6Wkk+W6zwXkbH3uTRV3Z6OYv5r55QK/4cryCsvz1qYASLYBe0gW9nhoW8V79a6OU+Ydw8pXZsTaz7xbzfX4am6q
+TpoGYCEBpknI0Golt/H7MaOud8XybUv/RnqjIUetbzzOf36Ynom1sviiKXv29H3QT91E0BaLKXj4AhBsvQ245nf1k
KVT/gd0oQ4gWqP30ZMWJub9fK0nF24UiZUDpECAwEAAaNQME4wTAYDVR0BBEUwQ4AQ3osAzHoU12fYNO9rdfjTvqEd
MBsxGTAXBgNVBAMTEFJvb3RDYUNsaWVudFRlc3SCEJ563a9OH6OmRjCpyUVHj1kwCQYFKw4DAh0FAAOCAQEAVKv8ox
/Wi+dmncXccJZDQjuVhYEKRvgXCNWlJcfhf57SsIq4CV4P9NSKaKAsez1sDWFSJ3Luava9nZT/hx6mcC/enCulo8gW
4P1gsaGz2Z4pgtI8V7cF/vkYA4k4od9wTnN82zzH1szdHQuN2CWTo+QmvK9QhNffuwehBL5dJGPcxKWq1ldZzQ76Kb
```

```
KdzR2+NHN24m7qYZ+pcNEge5yENT3XFqHMVXFRaBnjqkBpzCLDvYhvt9BQC2e7DAPK5fngdMvfut1Qgh43V4LAnW4K
Ts5ZldMHriWidAs73VcysxidYKFgAOD6IrrQW71wdTUGkWUqdvF0/UjgRCMm9yUQGQ==</o:BinarySecurityToke
n><Signature
xmlns="http://www.w3.org/2000/09/xmldsig#"><SignedInfo><CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/><SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/><Reference
URI="#_0"><Transforms><Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/></Transforms><DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/><DigestValue>FlASDEG7MEZw8w0FX2G4VD3yX
y0=</DigestValue></Reference><Reference URI="#_1"><Transforms><Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/></Transforms><DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/><DigestValue>GFhi1zZiaiGWSD/4Ovt2XyoaQ
vw=</DigestValue></Reference></SignedInfo><SignatureValue>CQoemS3joXqnHzS54lE8u34BC74jzHDQ
u7vkQJF/8T46OllXxhmhVuSuxb7ubCzX2M3AnKynBtBVq08ZnWyFBhVABOqWfhICOCqAUl9hvEkPNloW97ZWExjxmI
saX0WHpNpdWcojG/szZOGsTF6yivCaXoIYz4hBUxYyCbX6iYYbK96vuLHZJhWTrkwagyxnu/3oCVJHIielHmS6nl8z
sJehMBOo2OKFr1s3zKOiVIdfG4Xjt5In1JobxT6cSnQKDBKJduJQG6Nn/2WYEwZxIpqexlc+SMeQEMq3xYeK5dk3uC
WMr+4a0GbHh9d1lRnpXE04h9XA3O6zl2p2kWLyrg==</SignatureValue><KeyInfo><o:SecurityTokenRefere
nce><o:Reference ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-
token-profile-1.0#X509v3" URI="#uuid-d4b24929-31e2-49ad-95a3-b8f4e8275032-
1"/></o:SecurityTokenReference></KeyInfo></Signature></o:Security></s:Header><s:Body><trus
t:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-
trust/200512"><trust:RequestType>http://docs.oasis-open.org/ws-sx/ws-
trust/200512/Issue</trust:RequestType><wsp:AppliesTo
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"><EndpointReference
xmlns="http://www.w3.org/2005/08/addressing"><Address>https://ipnew.test.fgsz.hu/Fgsz.Ip.W
eb.UAT/Services/SoapIpDataSecurityService.svc</Address></EndpointReference></wsp:AppliesTo
><trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-
trust/200512/Bearer</trust:KeyType><trust:SecondaryParameters>&#xD;
                  <trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-
trust/200512/Bearer</trust:KeyType>&#xD;
                  <trust:CanonicalizationAlgorithm>http://www.w3.org/2001/10/xml-exc-
c14n#</trust:CanonicalizationAlgorithm>&#xD;
                  <trust:EncryptionAlgorithm>http://www.w3.org/2001/04/xmlenc#aes256-
cbc</trust:EncryptionAlgorithm>&#xD;

</trust:SecondaryParameters></trust:RequestSecurityToken></s:Body></s:Envelope>
```

Response Sample

```
HTTP/1.1 200 OK
Content-Length: 6803
Content-Type: application/soap+xml; charset=utf-8
Server: Microsoft-HTTPAPI/2.0
Date: Fri, 09 Mar 2018 12:35:21 GMT

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"><s:Header><a:Action
s:mustUnderstand="1">http://docs.oasis-open.org/ws-sx/ws-
trust/200512/RSTRC/IssueFinal</a:Action><a:RelatesTo>urn:uuid:f9aa5885-174e-4fa8-8e38-
4b35520078a5</a:RelatesTo><o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"><u:Timestamp
u:Id="_0"><u:Created>2018-03-09T12:35:21.894Z</u:Created><u:Expires>2018-03-
09T12:40:21.894Z</u:Expires></u:Timestamp></o:Security></s:Header><s:Body><trust:RequestSe
curityTokenResponseCollection xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-
trust/200512"><trust:RequestSecurityTokenResponse><trust:Lifetime><wsu:Created
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">2018-03-09T12:35:21.894Z</wsu:Created><wsu:Expires xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">2018-03-
09T13:35:21.894Z</wsu:Expires></trust:Lifetime><wsp:AppliesTo
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"><wsa:EndpointReference
xmlns:wsa="http://www.w3.org/2005/08/addressing"><wsa:Address>https://ipnew.test.fgsz.hu/F
gsz.Ip.Web.UAT/Services/SoapIpDataSecurityService.svc</wsa:Address></wsa:EndpointReference
```

```
></wsp:AppliesTo><trust:RequestedSecurityToken><saml:Assertion MajorVersion="1"
MinorVersion="1" AssertionID="_4e0cb07c-4a87-44bf-92ce-0b3bd02582bb"
Issuer="http://sts.test.rbp.eu/adfs/services/trust" IssueInstant="2018-03-
09T12:35:21.894Z" xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"><saml:Conditions
NotBefore="2018-03-09T12:35:21.894Z" NotOnOrAfter="2018-03-
09T13:35:21.894Z"><saml:AudienceRestrictionCondition><saml:Audience>https://ipnew.test.fgs
z.hu/Fgsz.Ip.Web.UAT/Services/SoapIpDataSecurityService.svc</saml:Audience></saml:Audience
RestrictionCondition></saml:Conditions><saml:AttributeStatement><saml:Subject><saml:Subjec
tConfirmation><saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:Confirm
ationMethod></saml:SubjectConfirmation></saml:Subject><saml:Attribute AttributeName="name"
AttributeNamespace="http://schemas.xmlsoap.org/ws/2005/05/identity/claims"><saml:Attribute
Value>IPNEWTEST\eburany</saml:AttributeValue><saml:AttributeValue>tempClientcert</saml:Att
ributeValue></saml:Attribute><saml:Attribute AttributeName="upn"
AttributeNamespace="http://schemas.xmlsoap.org/ws/2005/05/identity/claims"><saml:Attribute
Value>eburany@ipnewtest.local</saml:AttributeValue></saml:Attribute></saml:AttributeStatem
ent><saml:AuthenticationStatement
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:X509-PKI"
AuthenticationInstant="2018-03-
09T12:35:21.878Z"><saml:Subject><saml:SubjectConfirmation><saml:ConfirmationMethod>urn:oas
is:names:tc:SAML:1.0:cm:bearer</saml:ConfirmationMethod></saml:SubjectConfirmation></saml:
Subject></saml:AuthenticationStatement><ds:Signature
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"><ds:SignedInfo><ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/><ds:SignatureMethod
Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/><ds:Reference
URI="#_4e0cb07c-4a87-44bf-92ce-0b3bd02582bb"><ds:Transforms><ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/><ds:Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/></ds:Transforms><ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/><ds:DigestValue>IbJHvrT2lrACeoovH/76H
rez/8yXgQ1UM3y63O0fdO8=</ds:DigestValue></ds:Reference></ds:SignedInfo><ds:SignatureValue>
KKu6oopRUSUvDuIdDuo7y/x+0HsRLs125yeHzCRMEcgtHldYXyBQIvuSjqoDkO8scg9Gyib9U8AUQJAVlXxsnZt2Wn
x19webYrk0nH7VEZffoJpycQHjWN6+R3lHsXc8OwBqtF+m1VlPEImNhSv+cmCnHiMkNfGCxEZMt3vIZR01oDCSFm3u
05enOnILq4hkgb7kd/jhSfa7kZHgzjyY6QmfcYIRBPu27p9zyT495DVSR+L9etkZHlBrvzCPTTh3TbX9CTWh1uZUsE
++mu/nmY84VUV2nwdqd1fwKQM2nAeh02HzmGy/a2kHZdF6GIAlg2NNmtcrQX49NUv7a3CZFg==</ds:SignatureVa
lue><KeyInfo
xmlns="http://www.w3.org/2000/09/xmldsig#"><X509Data><X509Certificate>MIIDEDCCAfigAwIBAgIQ
HIgjaxlQUI9Fpn/u6ofQnjANBgkqhkiG9w0BAQsFADBEMUIwQAYDVQQDEzlBREZTIFNpZ25pbmcgLSBpcG5ld3Rlc3
RhZC5ub3J0aGV1cm9wZS5jbG91ZGFwcC5henVyZS5jb20wHhcNMTgwMjE3MTYxMjUzWhcNMTkwMjE3MTYxMjUzWjBE
MUIwQAYDVQQDEzlBREZTIFNpZ25pbmcgLSBpcG5ld3Rlc3RhZC5ub3J0aGV1cm9wZS5jbG91ZGFwcC5henVyZS5jb2
0wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC138SfR2bsFXZobfoWIDZrqRy8//lyF06iVYf20Qa/rJyB
lZkKZFvwXt5L0+Nzt0FqbRHv37qAbEwuu0NyNzVzFa99hd3AncKuYHXgJr5PFEXNUu7eI2nxtpxQNsjgedxqtlR2dR
s/k3wvWy+ltEQKYyZ3maZmXEz+vU2kQhbzajTne0C+XEjNGic5+dhBbCKcRU7aYUljZJy/Nv+U5hMGCCwAomziZ5rA
PtpO9vU781zUhczpDa6tPqQiY4N/a8qong+kYV1lLyJvluYkaizHzMyJU7b0aKTZfV8vMuEERc6WDseI4HfdhtTY0O
M55TpQgBaphW80x/8/OgG85w3rAgMBAAEwDQYJKoZIhvcNAQELBQADggEBAAucCoXRfg4mnwfzBVah36NAQn2kCfda
U3sqet8nQXPXWStU28lP2tiF8pspz5Bb1Uw1bEO3MwGm6+o2YPd6gnYpwYtWdmCYYFIZgQRU8nCzk4zw98I2O6JXs7
kDMAocaIInIjs//4SnVHH3PSe29lw26ga2yAk6WmMSdPWze6zCspNskBjww8wvhhK9GknAPXWu0gCyHfG+ehEzfxXI
x4ieJEodFNgmGuk97SWnMAkZdBlrZ1mVZWAMQI7AtaOu6JjvgDOGgmwyHMvEV83GdKOvKzR+vCb3X41L4Me/Fym7nx
WZBEiRDqZEzidA6EQK2ceuiobFzhKx/xwJIoVknUI=</X509Certificate></X509Data></KeyInfo></ds:Sign
ature></saml:Assertion></trust:RequestedSecurityToken><trust:RequestedAttachedReference><o
:SecurityTokenReference k:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
profile-1.1#SAMLV1.1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:k="http://docs.oasis-open.org/wss/oasis-wss-wssecurity-
secext-1.1.xsd"><o:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-
token-profile-1.0#SAMLAssertionID">_4e0cb07c-4a87-44bf-92ce-
0b3bd02582bb</o:KeyIdentifier></o:SecurityTokenReference></trust:RequestedAttachedReferenc
e><trust:RequestedUnattachedReference><o:SecurityTokenReference
k:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1"
xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd" xmlns:k="http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-
1.1.xsd"><o:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
profile-1.0#SAMLAssertionID">_4e0cb07c-4a87-44bf-92ce-
0b3bd02582bb</o:KeyIdentifier></o:SecurityTokenReference></trust:RequestedUnattachedRefere
nce><trust:TokenType>urn:oasis:names:tc:SAML:1.0:assertion</trust:TokenType><trust:Request
Type>http://docs.oasis-open.org/ws-sx/ws-
```

trust/200512/Issue</trust:RequestType><trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Bearer</trust:KeyType></trust:RequestSecurityTokenResponse></trust:RequestSecurityTokenResponseCollection></s:Body></s:Envelope>

Calling a Web Service with the acquired token

Request Sample

```
POST https://ipnew.test.fgsz.hu/Fgsz.Ip.Web.UAT/Services/SoapFGSZNNOService.svc HTTP/1.1
Content-Type: application/soap+xml; charset=utf-8
Host: app.example.com
Content-Length: 5029
Expect: 100-continue
Accept-Encoding: gzip, deflate

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"><s:Header><a:Action
s:mustUnderstand="1">http://service.fgsz.hu/IFGSZNNOServiceSoapWrapper/CheckAlive</a:Actio
n><a:MessageID>urn:uuid:55832272-64d9-420a-a946-
69de57a66df9</a:MessageID><a:ReplyTo><a:Address>http://www.w3.org/2005/08/addressing/anony
mous</a:Address></a:ReplyTo><a:To
s:mustUnderstand="1">https://ipnew.test.fgsz.hu/Fgsz.Ip.Web.UAT/Services/SoapFGSZNNOServic
e.svc</a:To><o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"><u:Timestamp
u:Id="_0"><u:Created>2018-03-09T12:35:21.904Z</u:Created><u:Expires>2018-03-
09T12:40:21.904Z</u:Expires></u:Timestamp><saml:Assertion MajorVersion="1"
MinorVersion="1" AssertionID="_4e0cb07c-4a87-44bf-92ce-0b3bd02582bb"
Issuer="http://sts.test.rbp.eu/adfs/services/trust" IssueInstant="2018-03-
09T12:35:21.894Z" xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"><saml:Conditions
NotBefore="2018-03-09T12:35:21.894Z" NotOnOrAfter="2018-03-
09T13:35:21.894Z"><saml:AudienceRestrictionCondition><saml:Audience>https://ipnew.test.fgs
z.hu/Fgsz.Ip.Web.UAT/Services/SoapFGSZNNOService.svc</saml:Audience></saml:AudienceRestric
tionCondition></saml:Conditions><saml:AttributeStatement><saml:Subject><saml:SubjectConfir
mation><saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:ConfirmationMe
thod></saml:SubjectConfirmation></saml:Subject><saml:Attribute AttributeName="name"
AttributeNamespace="http://schemas.xmlsoap.org/ws/2005/05/identity/claims"><saml:Attribute
Value>IPNEWTEST\eburany</saml:AttributeValue><saml:AttributeValue>tempClientcert</saml:Att
ributeValue></saml:Attribute><saml:Attribute AttributeName="upn"
AttributeNamespace="http://schemas.xmlsoap.org/ws/2005/05/identity/claims"><saml:Attribute
Value>eburany@ipnewtest.local</saml:AttributeValue></saml:Attribute></saml:AttributeStatem
ent><saml:AuthenticationStatement
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:X509-PKI"
AuthenticationInstant="2018-03-
09T12:35:21.878Z"><saml:Subject><saml:SubjectConfirmation><saml:ConfirmationMethod>urn:oas
is:names:tc:SAML:1.0:cm:bearer</saml:ConfirmationMethod></saml:SubjectConfirmation></saml:
Subject></saml:AuthenticationStatement><ds:Signature
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"><ds:SignedInfo><ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"></ds:CanonicalizationMethod><ds:SignatureMethod
Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"></ds:SignatureMethod><ds:Reference URI="#_4e0cb07c-4a87-44bf-92ce-
0b3bd02582bb"><ds:Transforms><ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"></ds:Transform><ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"></ds:Transform></ds:Transforms><ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"></ds:DigestMethod><ds:DigestValue>IbJH
vrT2lrACeoovH/76Hrez/8yXgQ1UM3y63O0fdO8=</ds:DigestValue></ds:Reference></ds:SignedInfo><d
s:SignatureValue>KKu6oopRUSUvDuIdDuo7y/x+0HsRLs125yeHzCRMEcgtHldYXyBQIvuSjqoDkO8scg9Gyib9U
8AUQJAVlXxsnZt2Wnx19webYrk0nH7VEZffoJpycQHjWN6+R3lHsXc8OwBqtF+m1VlPEImNhSv+cmCnHiMkNfGCxEZ
Mt3vIZR01oDCSFm3u05enOnILq4hkgb7kd/jhSfa7kZHgzjyY6QmfcYIRBPu27p9zyT495DVSR+L9etkZHlBrvzCPT
Th3TbX9CTWh1uZUsE++mu/nmY84VUV2nwdqd1fwKQM2nAeh02HzmGy/a2kHZdF6GIAlg2NNmtcrQX49NUv7a3CZFg=
=</ds:SignatureValue><KeyInfo
xmlns="http://www.w3.org/2000/09/xmldsig#"><X509Data><X509Certificate>MIIDEDCCAfigAwIBAgIQ
HIgjaxlQUI9Fpn/u6ofQnjANBgkqhkiG9w0BAQsFADBEMUIwQAYDVQQDEzlBREZTIFNpZ25pbmcgLSBpcG5ld3Rlc3
```

```
RhZC5ub3J0aGV1cm9wZS5jbG91ZGFwcC5henVyZS5jb20wHhcNMTgwMjE3MTYxMjUzWhcNMTkwMjE3MTYxMjUzWjBE
MUIwQAYDVQQDEzlBREZTIFNpZ25pbmcgLSBpcG5ld3Rlc3RhZC5ub3J0aGV1cm9wZS5jbG91ZGFwcC5henVyZS5jb2
0wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC138SfR2bsFXZobfoWIDZrqRy8//lyF06iVYf20Qa/rJyB
lZkKZFvwXt5L0+Nzt0FqbRHv37qAbEwuu0NyNzVzFa99hd3AncKuYHXgJr5PFEXNUu7eI2nxtpxQNsjgedxqtlR2dR
s/k3wvWy+ltEQKYyZ3maZmXEz+vU2kQhbzajTne0C+XEjNGic5+dhBbCKcRU7aYUljZJy/Nv+U5hMGCCwAomziZ5rA
PtpO9vU781zUhczpDa6tPqQiY4N/a8qong+kYV1lLyJvluYkaizHzMyJU7b0aKTZfV8vMuEERc6WDseI4HfdhtTY0O
M55TpQgBaphW80x/8/OgG85w3rAgMBAAEwDQYJKoZIhvcNAQELBQADggEBAAucCoXRfg4mnwfzBVah36NAQn2kCfda
U3sqet8nQXPXWStU28lP2tiF8pspz5Bb1Uw1bEO3MwGm6+o2YPd6gnYpwYtWdmCYYFIZgQRU8nCzk4zw98I2O6JXs7
kDMAocaIInIjs//4SnVHH3PSe29lw26ga2yAk6WmMSdPWze6zCspNskBjww8wvhhK9GknAPXWu0gCyHfG+ehEzfxXI
x4ieJEodFNgmGuk97SWnMAkZdBlrZ1mVZWAMQI7AtaOu6JjvgDOGgmwyHMvEV83GdKOvKzR+vCb3X41L4Me/Fym7nx
WZBEiRDqZEzidA6EQK2ceuiobFzhKx/xwJIoVknUI=</X509Certificate></X509Data></KeyInfo></ds:Sign
ature></saml:Assertion></o:Security></s:Header><s:Body><CheckAlive
xmlns="http://service.fgsz.hu"
xmlns:b="http://schemas.datacontract.org/2004/07/WsClientTest.FgszService"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance"/></s:Body></s:Envelope>
```


Response Sample

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Fri, 09 Mar 2018 12:35:22 GMT
Content-Length: 1567

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <s:Header>
      <a:Action
s:mustUnderstand="1">http://domain.service.fgsz.hu/IFGSZNNOServiceSoapWrapper/CheckAliveRe
sponse</a:Action>
      <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <u:Timestamp u:Id="_0">
            <u:Created>2018-09-02T09:11:03.596Z</u:Created>
            <u:Expires>2018-09-02T09:16:03.596Z</u:Expires>
         </u:Timestamp>
      </o:Security>
   </s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <CheckAliveResponse xmlns="http://domain.service.fgsz.hu">
         <CheckAliveResult>The service is alive.</CheckAliveResult>
      </CheckAliveResponse>
   </s:Body>
</s:Envelope>
```

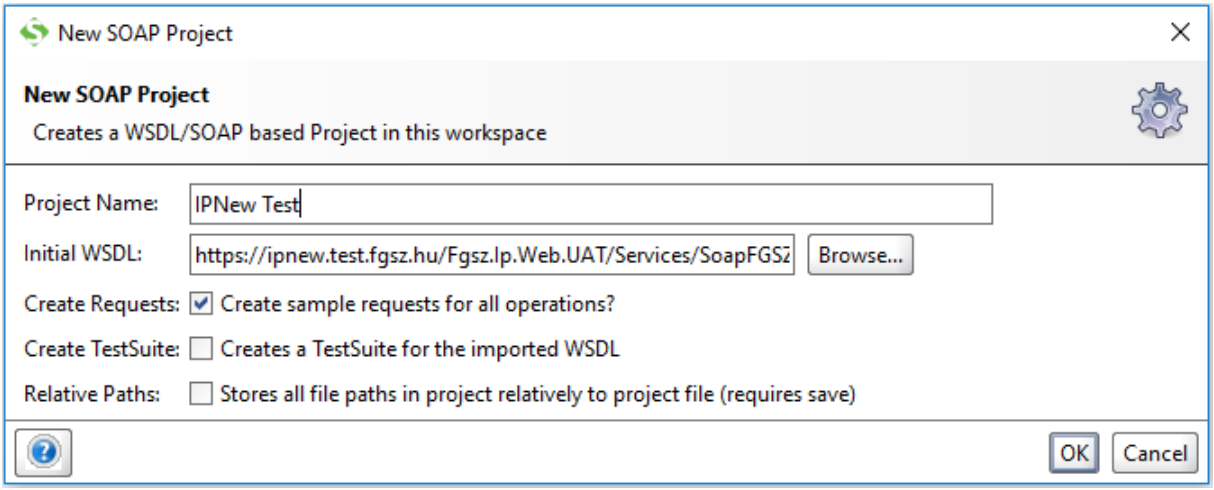
Testing with SoapUI

You can test the Web services without the creation of a program, by using a free tool such as SoapUI (https://www.soapui.org/) as described below.

Creating a SoapUI Project

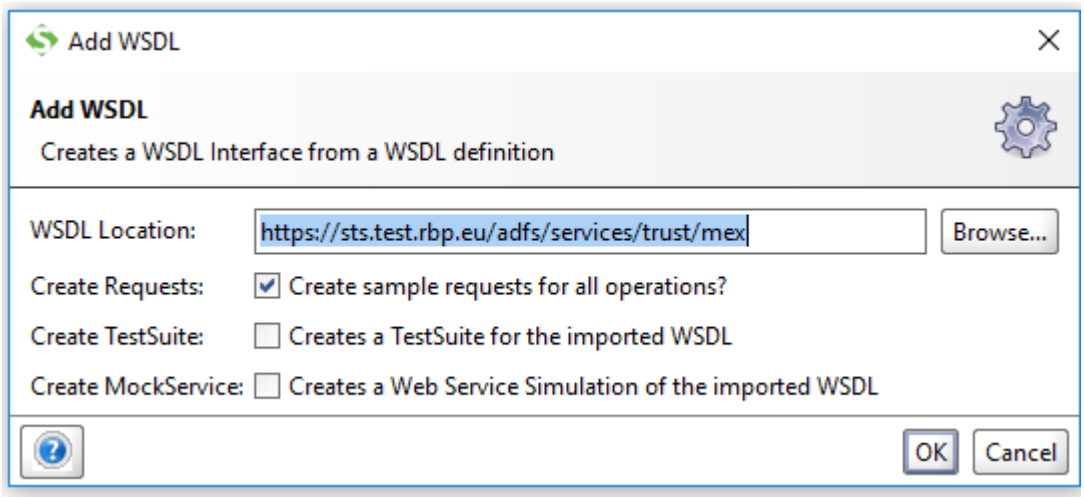
Create a new SOAP project, e.g. IPNew Test (File menu-> New SOAP Project menu item). In the Initial WSDL field enter the WSDL address of the IPNew Web service you want to use (e.g. https://ipnew.test.fgsz.hu/Fgsz.Ip.Web.UAT/Services/SoapFGSZNNOService.svc?singleWsdl

)



A new project is created, with an interface corresponding to the WSDL and with a sample request message for each Web method. We will use them during testing.

You should also add the ADFS Web services actions to your project. Right click on the name of the project and then choose "Add WSDL". In the WSDL Location box, type the address for ADFS WSDL, such as: https://sts.test.rbp.eu/adfs/services/trust/mex. Sample messages for ADFS service methods are then created.

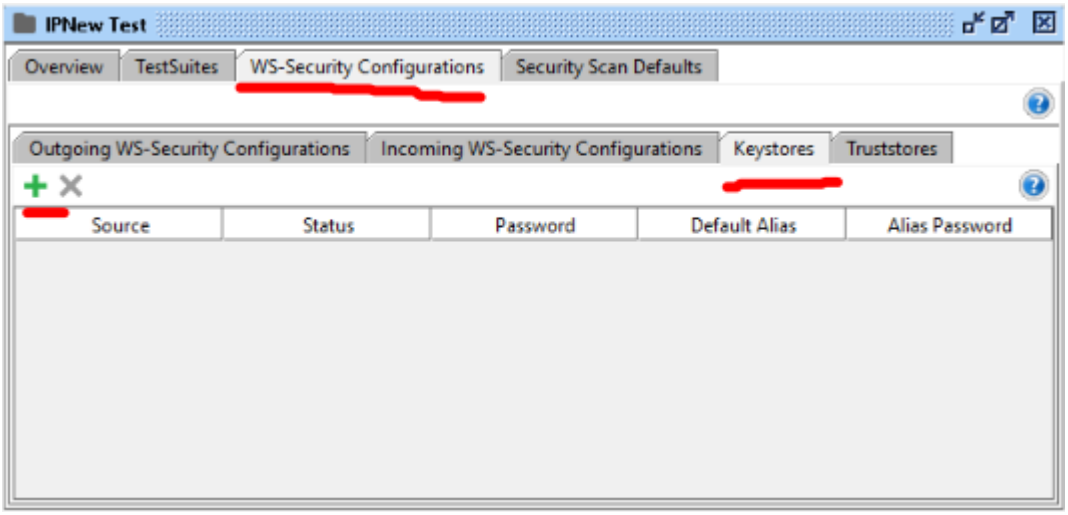


Set up Project parameters

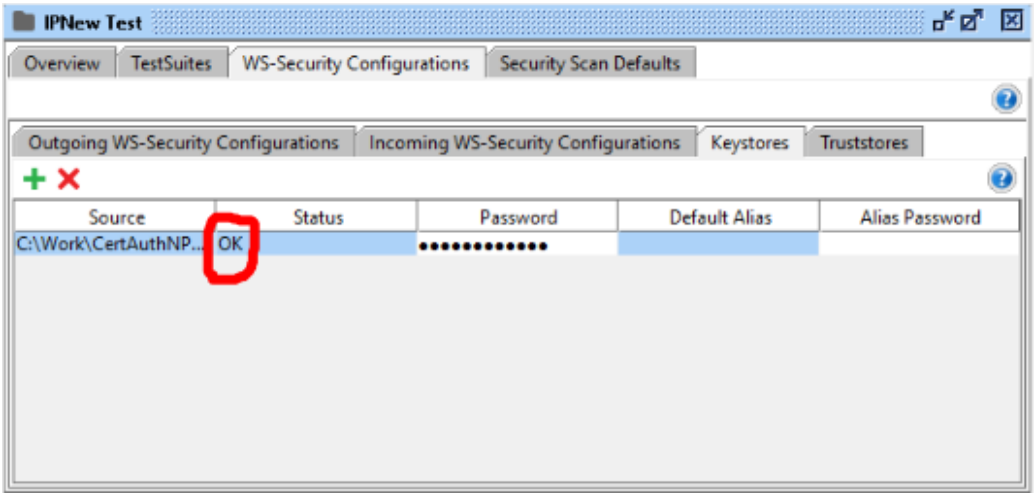
Set parameters required for certificate-based authentication

For certificate-based authentication we need a client certificate, which is previously assigned by FGSZ to an AD user. To invoke the Web service, you will need a file containing the private and public key of the certificate (e.g. TestClient.pfx).

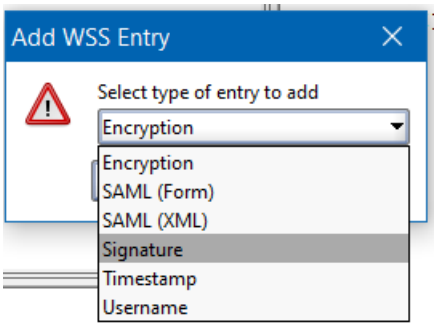
Double-click the SoapUI Project name, in the window that appears, click WS-Security Configuration tab, and the Keystores Tab inside it.



Click the green + icon, and in the File Selection window that appears, select the certificate file. If the file is password-protected, enter the password in the next window and press OK. Verify that "OK" is displayed in the Status field.



Switch to "Outgoing WS-Security Configurations" tab, and then click the green + icon. In the window that appears, add any name for the configuration, such as "clientCert". In the bottom section of the window, click the green + icon (Add a new WSS Entry). From the dropdown list select the Signature option.



Fill in the fields of the Signature window as follows:

Keystore: the keystore we have previously added (certificate file).

Alias: the name of the key in the keystore.

Password: the password used to protect the certificate file.

Key Identifier Type: Binary Security Token

Signature Algorithm: Http://www.w3.org/2000/09/xmldsig#rsa-sha1

Signature Canonicalization: http://www.w3.org/2001/10/xml-exc-c14n#

Digest Algorithm: Http://www.w3.org/2000/09/xmldsig#sha1

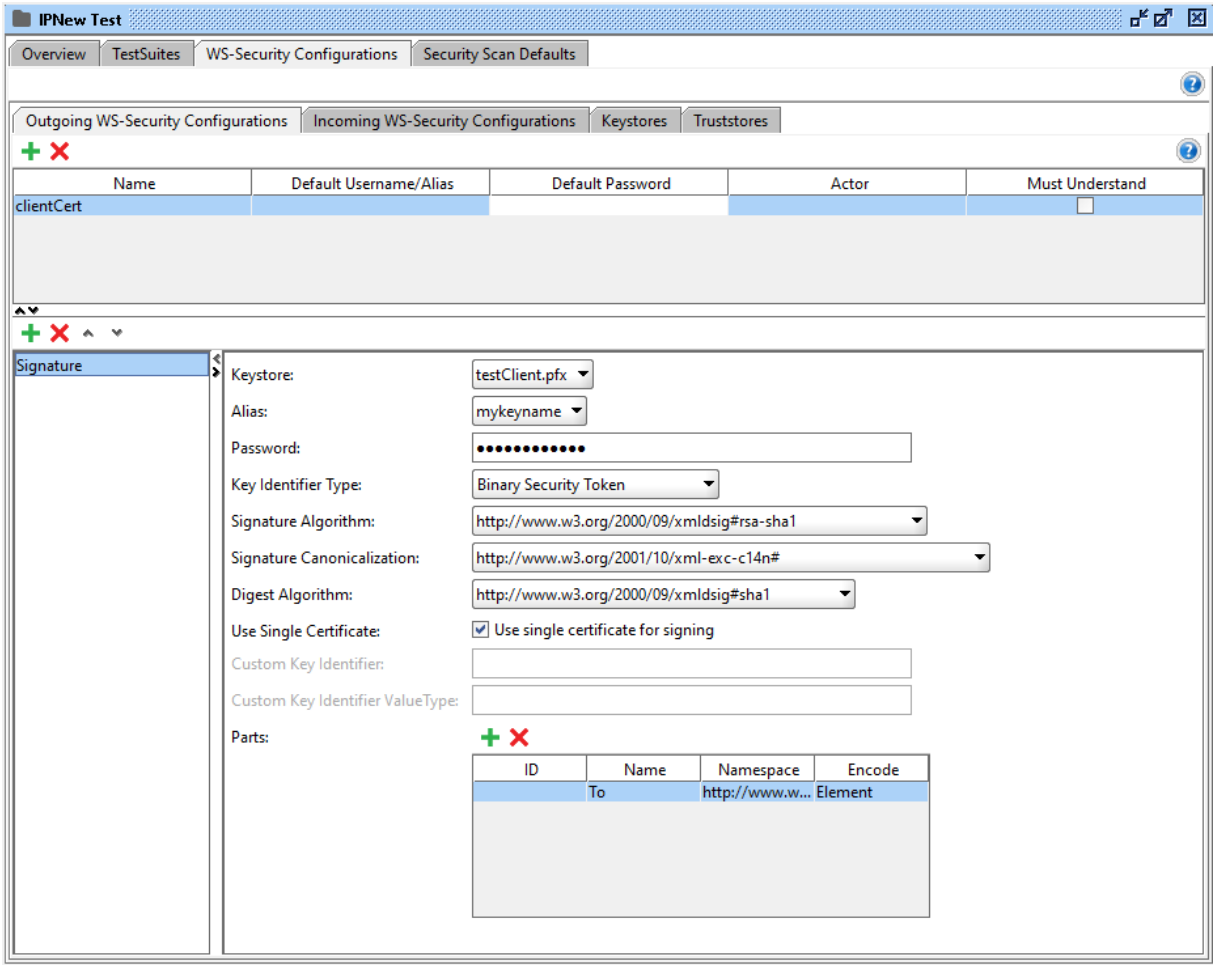
Check the “Use Single Certificate” checkbox.

In the lower “Parts” section, enter a line with the following information:

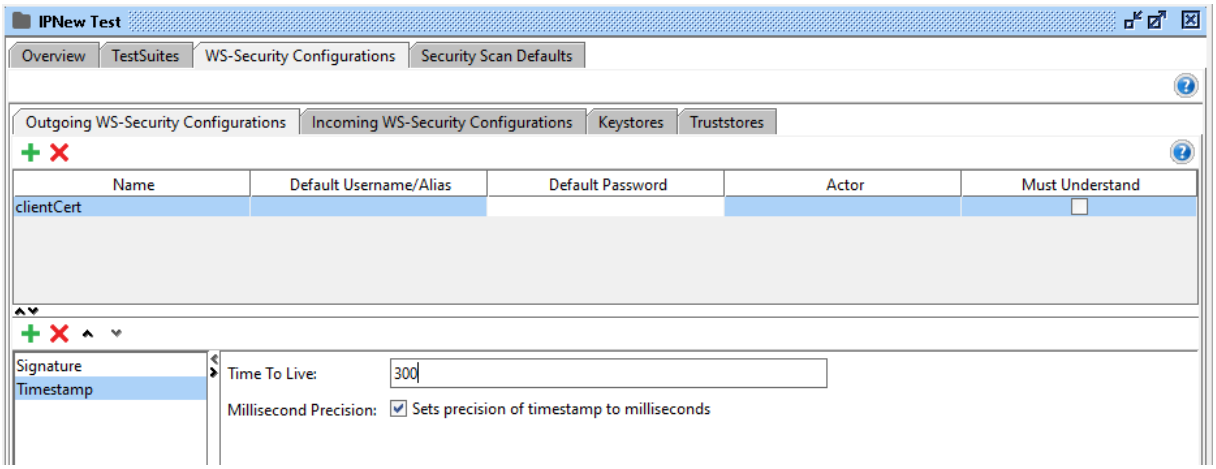
Name Column: To

Namespace Column: http://www.w3.org/2005/08/addressing

Encode Column: Element

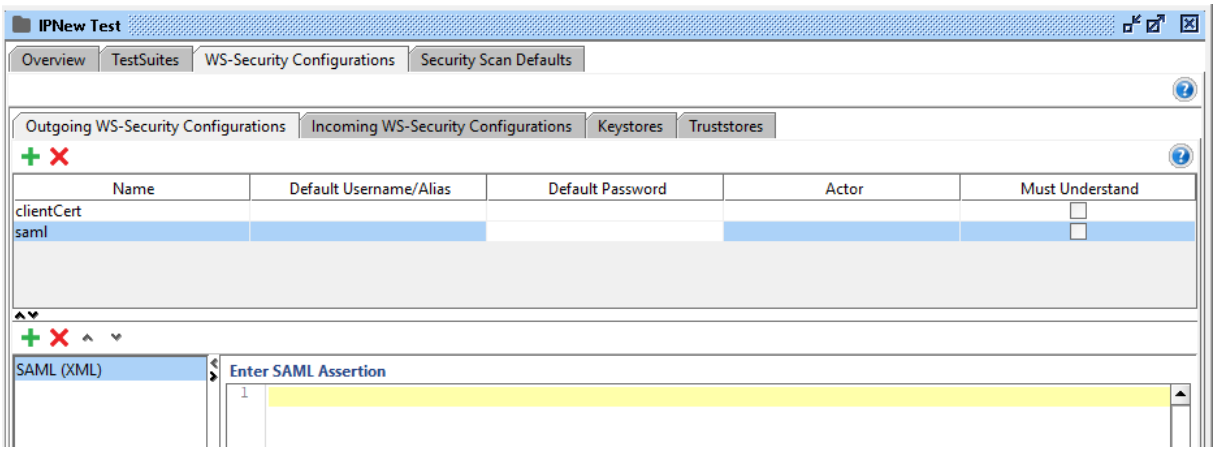


Create another line under the Signature entry using the green “+” icon and select “Timestamp”. Enter 300 into the Time To Live box.



Create another configuration line under clientCert using the top green “+” icon. The name should be saml.

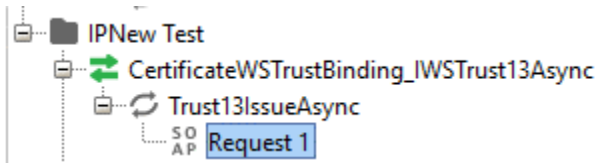
In the lower section, add a new Entry, the type must be SAML (XML). Do not add a value for it yet.



Creating Requests

Creating a Token Request for certificate-based identification

In the Navigator tree on the left, under ADFS methods, expand the “CertificateWSTRustBinding_IWSTrust13Async” branch down to “Request 1” level. Double-click “Request 1”.



In the XML view in the left pane, replace the green XML comments with the following XML:

<ns:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</ns:RequestType>

<wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">

<EndpointReference xmlns="http://www.w3.org/2005/08/addressing">

<Address>https://ipnew.test.fgsz.hu/Fgsz.Ip.Web.UAT/Services/SoapFGSZNNOService.svc</Address>

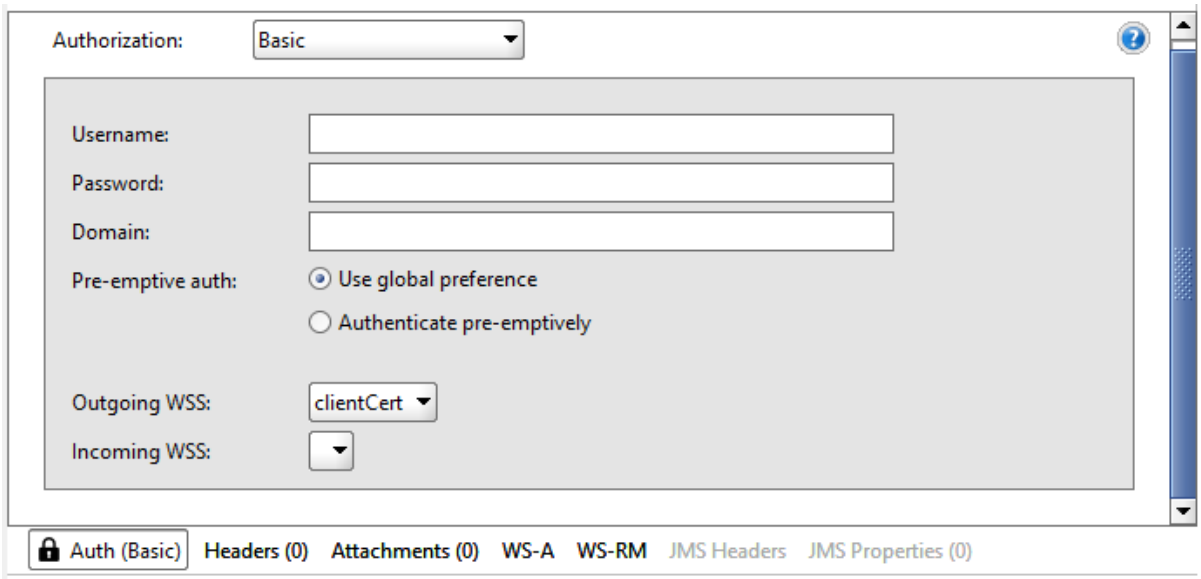
</EndpointReference>

</wsp:AppliesTo>

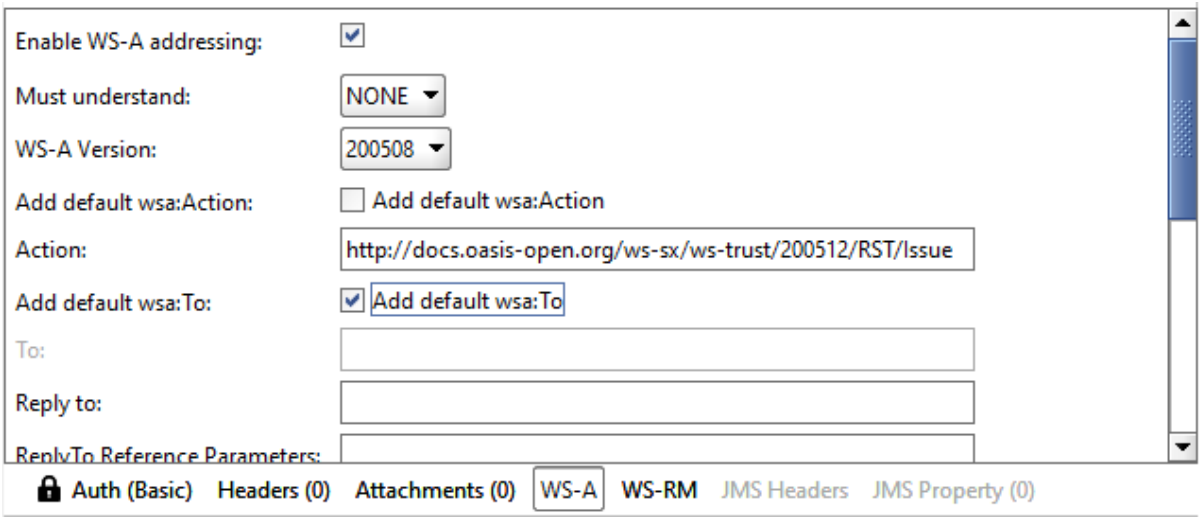
<ns:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Bearer</ns:KeyType>

Make sure that the Address XML element contains the correct Web service address (modify as necessary).

Click “Auth” at the bottom of the window, in the window that appears, select “Add New Authorization Item” from the “Authorization” box, and then in the popup window the “Basic” option. From the Outgoing WSS list select the “clientCert” configuration.



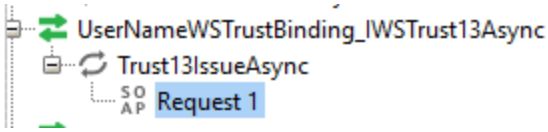
Click on the “WS-A” tab. Select the “Add default wsa:To” option.



Click the green Play button to make the Web service call. The Web service reply that includes a SAML token is displayed in the right XML view.

Creating a Token Request for username/password authentication

In the Navigator tree on the left, under ADFS methods, expand the “UserNameWSTrustBinding_IWSTrust13Async” branch down to “Request 1” level. Double-click “Request 1”.



In the left XML view, replace the <Soap:Header/> element with the following XML fragment:

<soap:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">

<wsse:Security soap:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<wsse:UsernameToken wsu:Id="UsernameToken-348E3F3C81E7E21F39152094195633514">

<wsse:Username>testusername@ipnewtest.local</wsse:Username>

<wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">password</wsse:Password>

</wsse:UsernameToken>

</wsse:Security> <wsa:To>https://sts.test.rbp.eu/adfs/services/trust/13/usernamemixed</wsa:To>

</soap:Header>

In the XML view in the left pane, replace the green XML comments with the following XML:

<ns:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</ns:RequestType>

<wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">

<EndpointReference xmlns="http://www.w3.org/2005/08/addressing">

<Address>https://ipnew.test.fgsz.hu/Fgsz.Ip.Web.UAT/Services/SoapFGSZNNOService.svc</Address>

</EndpointReference>

</wsp:AppliesTo>

<ns:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Bearer</ns:KeyType>

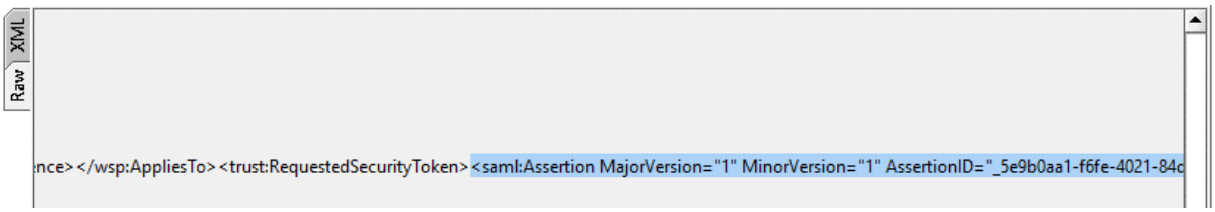
Make sure that the Address XML element contains the correct Web service address (modify as necessary).

Click the green “Play” button to make the web service call. The web service reply that includes a SAML token is displayed in the right XML view.

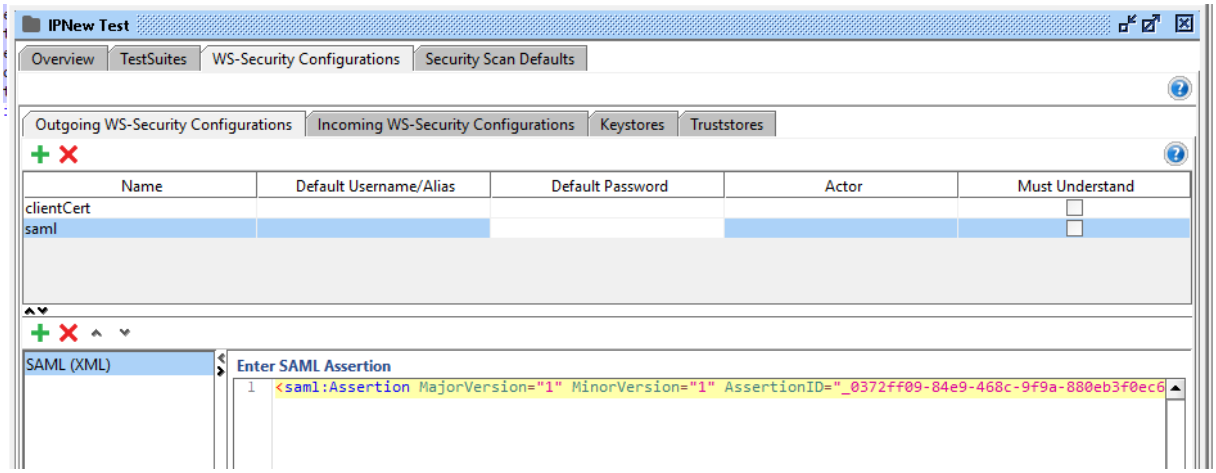
Creating an IPNew Web Service Call Request

1. From the XML View switch to the Raw View:

2. Copy a fragment of the XML that begins with <saml:Assertion> and ends with </saml:Assertion>. Pay close attention to make sure that there are no changes made to any character, no whitespaces or new lines inserted and that formatting is not changed because it will invalidate the digital signature.



3. Copy the extracted part into the “saml” configuration item we created in an earlier step (“Enter SAML Assertion” field):



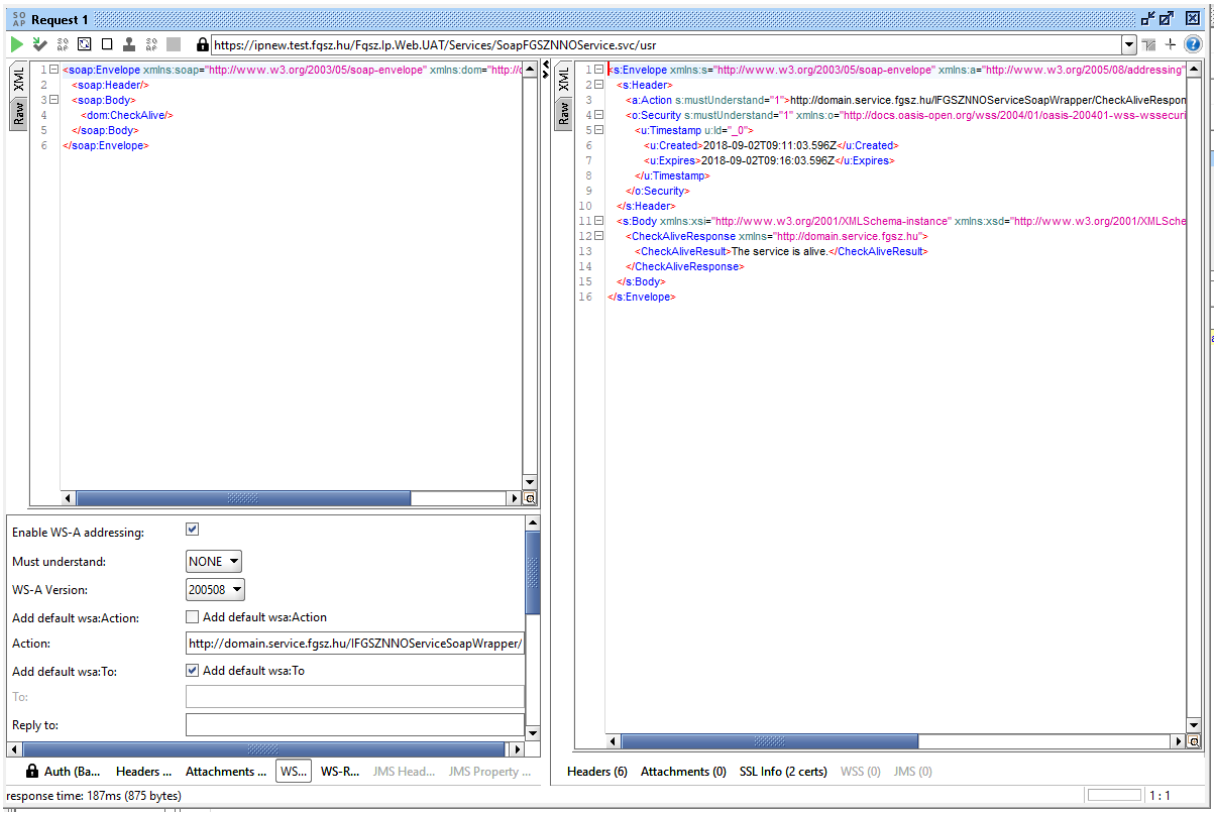
4. In the Navigator tree on the left, under IPNew web service methods, expand the desired branch down to “Request 1” level. Double-click “Request 1”.

5. Click “Auth” at the bottom of the window, in the window that appears, select “Add New Authorization Item” from the “Authorization” box, and then in the popup window the “Basic” option. From the Outgoing WSS list select the “saml” configuration.

6. Click on the “WS-A” tab. Select the “Add default wsa:To” option.

7. If necessary, edit the sample XML created by SoapUI.

8. Click the green “Play” button to make the web service call. The web service response is displayed in the right XML view.



Known issues and error messages

- **"At least one security token in the message could not be validated."** error: it's likely that some time has elapsed since invoking the ADFS web service and the validity of the token has expired. The ADFS web service must be called again to obtain a fresh token, and the saml configuration element must be updated with the new <SAML:Assertion> element.
- **"An error occurred when verifying security for the message."** error, possible causes are:
    - If the <saml:Assertion> fragment was not copied from the Raw View, or if any character has changed compared to the original, then the system will assume that the message was compromised since the signature does not match the message. In this case carefully repeat the first three steps in section "Creating an IPNew Web Service Call Request" then execute the call again.
    - In case the certificate that is used for authentication contains accented characters (e.g. in the name field of the certificate owner) it may happen, that the token response is not encoded correctly in SOAP UI and the saml:Assertion section contains unreadable garbage characters. This causes an error with message "An error occurred when verifying security for the message." in subsequent web service calls. To work around this, correct the wrong characters and use the modified saml:Assertion XML in subsequent steps.

Links

- WS-Trust 1.3 – OASIS: http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html
- SOAP 1.1: http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
- SoapUI: https://www.soapui.org/