Composer Help

Signed SOAP Requests

4/4/2025

Signed SOAP Requests

Contents

The Web Service block enables Composer applications to invoke Web Services, which require message-level authentication. The message level security support provided by the Web Service block is limited to one-way signed SOAP requests from the Composer application to the Web Service. Web Services can then verify that the request received from a Composer application includes a valid certificate.

Prerequisites

The prerequisites are:

- Web Service is able to verify only the signature (Timestamp, UsernameToken and Encryption are not supported).
- Web Service sends an unsigned response, i.e., Web Service is not configured to process outgoing response (only InflowSecurity is configured).
- X.509 certificate for the client is available and is trusted by the Web Service. Certificates can be purchased from a certificate authority or can be generated (for testing) using tools, such as OpenSSL.
- Certificates should be based on one of the supported encryption algorithms, RSA or DSA.
- Certificate Stores:
- For Java projects, certificates and keys should be available in a Java Keystore (*.jks file). OpenSSL and Keytool (available in JDK 1.6) can be used to create and import certificates.
- For .NET projects, certificates and keys should be available in the Windows Certificate Store. OpenSSL can be used to create certificates and Certificates (snap-in in Microsoft Management Console) can be used to import certificates.
- For .NET projects, WSE 3.0 (runtime) should be installed on the machine running Composer.

Enabling Signing of SOAP Messages

To enable signing of SOAP messages, set the Authentication Type property in the Security section to one of the following values

- SOAPDigitalSignatureAuthentication -- for signing messages when not using HTTP Basic authentication.
- SOAPSignatureWithHTTPBasicAuthentication -- for signing messages when used along with HTTP Basic Authentication (Security Basic Authentication Credentials section is specified)

Once enabled to sign the request, the application will need information regarding the public key (certificate) and private key (key) as below:

- Certificate Store Name (.NET only) -- Windows Store Name containing the client certificate and private key. Value should be one of the following predefined Windows Certificate Stores or the name of a custom Store in which the certificate and key are stored. Note that this Store should contain the client certificate (should include the private key as well).
- AddressBook -- The X.509 Certificate Store for other users.

- My -- The X.509 Certificate Store for personal certificates.
- TrustedPeople -- The X.509 Certificate Store for directly trusted people and resources.
- Certificate Alias -- Alias that identifies the certificate and key in the Store. For .NET projects, this refers to the subject of the certificate, e.g., CN=ComposerCertificate.
- Certificate or Key Store Location -- Path to the Certificate Store location containing the certificate and private key. In .NET, the value should be set to one of the following:
- StoreLocation.LocalMachine (default when value is not one of these)
- StoreLocation.CurrentUser
- Key Algorithm -- Algorithm to be used for encryption. This is the same as the algorithm that was specified when the key was generated; usually received from the certificate authority issuing the certificate.
- Key Store Password (Java only) -- Java Key Store password for the key store specified as the key store location.
- Private Key Alias (Java only) -- Alias by which the private key is identified in the key store.
- Private Key Password (Java only) - Password to access the private key to be used when signing a message. For .NET projects, it is expected that the password be stored as part of the settings for the certificate.

At run time, the Composer application will create a SOAP message and then sign it using its private key. The signed message will include an encrypted signature in the SOAP header and the SOAP request as the body. This signed message is sent to the Web Service for processing. Web Service will decrypt the signature using the client certificate (public key previously imported into the Web Service certificate store) and hence authenticating that the source of the request is valid.

Signature Validation Failure Causes

Signature Validation by the Web Service may fail for the following reasons:

- Syntax of request (signature) doesn't conform to the policy enforced by the Web Service. Example: Timestamp is required by the Web Service but was not included in the request because Composer doesn't support Timestamp policy.
- Validation of signature failed. Example: Web Service uses RSA key, but the request was signed using DSA key.
- Application validation policy rejects the request. Example: Signature created by an untrusted key.

Once signature validation is successful, the Web Service will process the request and then send the unsigned response back to the Composer application. Composer processes the response without signature validation. The above will ensure that Web Services will process requests only from legitimate clients, the Composer application being one of them.