KOMBIT

Guideline for .Net Sample Service

Version 1.0

**Introduction**

The following document describes how to configure the .Net-based sample service.

This is a soap service that authenticates the caller with a token issued by an STS compliant with the KOMBIT Støttesystemer specification for STS. The service has a simple ping method, that requires no input and which returns a statically configured text message.

In the following, this service is also referred to as “user context”, because it is a service that is invoked with a user context.

In the following, a “user system” refers to the caller of the service. This is because, in the KOMBIT Støttesystemer information model, a caller of a service is referred to as an “Anvendersystem”, and “user system” or “using system” are the best translations for this term.

After completing this guide, the .Net-based sample service will be configured.

Setting up the .Net-based service in IIS is outside the scope of this document. This is described in the document “All\_guideline\_setup sites IIS.docx”.

It is assumed that the reader is a .Net-developer knowledgeable in the following technologies used to develop this .Net-based sample. This includes:

* C#
* Microsoft.Net framework v4.5
* Microsoft Windows Server Operating System
* Microsoft Internet Information Systems (IIS)
* X509v3 Certificates
* Windows Communications Foundation
* SOAP protocol
* WS-Trust XML protocol

# Table of Contents

[Table of Contents 3](#_Toc418869032)

[Prerequisites 4](#_Toc418869033)

[Configuring The Service 4](#_Toc418869034)

[IIS website 4](#_Toc418869035)

[Configuration 4](#_Toc418869036)

[Installing The Anvendersystem (User System) 5](#_Toc418869037)

[Configuration Of The Anvendersystem (User System) 5](#_Toc418869038)

[Calling The Service Using the Anvendersystem (User Context) 6](#_Toc418869039)

# Prerequisites

This document requires that the following prerequisites are satisfied:

* Setting up the .Net-based samples according to the guide “All\_guideline\_setup sites IIS.docx”
* Logging is done to the folder c:\temp. This folder must exist for logging to work.

# Configuring The Service

### IIS website

This guideline assumes that the URL of the service is:

[https://adgangsstyringeksempler.test-stoettesystemerne.dk/Service](https://adgangsstyringeksempler.test-stoettesystemerne.dk/STSUserContext)/

### Configuration

Some changes to the properties in the configuration file Service\web.config may be required:

* **ServiceAddress**: The address where this service is deployed**. ServiceServiceCertificateThumbprint**: The thumprint of a certificate which is used as service certificate for the service endpoint.
  + In this sample, the supplied certificate “KombitTestSigningCertificate.p12” is used.
  + The certificate must exist in LocalMachine\My
* **StsSigningCertificateThumbprint**: The thumprint of a certificate which is issuer of the received security token.
  + In this sample, the supplied certificate “KombitTestSigningCertificate.p12” is used.
  + The certificate must exist in LocalMachine\TrustedPeople
* **ResponseMessage**: The response message from the Ping-method on the service.
* **SoapMessageLogLocation**: a folder to store all the received request to this service and its response to client.
* **serilog:minimum-level**: specify the level of logging. Log files are stored in the Logs\ folder.

## Installing The Anvendersystem (Consumer)

The Anvendersystem is implemented as a set of unit tests that can be found in the project:

Kombit.Samples.Consumer

These are also located in the folder:

Kombit.Samples\Tests\Kombit.Samples.Consumer

The purpose of the test cases is to simulate how to send an RST issue request and process the response from a WS-Trust service, this includes:

* How to generate security token request
* How to sign the security token request.
* How to send the request to WS-Trust service.
* In the sample, we test it against our STS test stub.
* How to process the response from WS-Trust service.
* How to handle error if there is from WS-Trust service.

It also simulates how to use the issued token to send a request to the service and process the response from the service. This includes:

* How to send the request to the service with an issued token.
* How to process the response from this web service.

# Configuration Of The Anvendersystem (User System)

Some changes to the properties in the configuration file

Tests\Kombit.Samples.Consumer\Kombit.Samples.Consumer.dll.config

May be required, depending on the specific environment where the tests are executed.

* **BaseAddress**: the address where STS and Anvendersystem (user context) is deployed.
* **AValidClientCertificateThumbprint**: the thumprint of a certificate which is used to imitate a valid client credential which will be accepted to receive a security token.
  + Any valid X509 certificate may be used, e.g. the following supplied certificate “CertificateAnvendersystem.p12”
  + The certificate must be located in LocalMachine\My.
* **StsServiceCertificateThumbprint**: thumprint of a certificate which is used as service certificate for certificate endpoint.
  + In this sample, the supplied certificate “KombitTestSigningCertificate.p12” is used.
  + The certificate must exist in LocalMachine\My.
* **StsServiceCertificateDNSIdentity**: the dns identity of sts service certificate.
* **StsCertificateEndpoint**: the certificate endpoint address.
* **StsMexEndpoint**: The mex endpoint address of the STS
* **HeaderSigningAlgorithm**: a signing algorithm used to sign header of soap message
* **AValidOnBehalfOfCertificateThumbprint**: thumbprint of a certificate which is used on proxy onbehalfof element or used as client certificate to request onbehalfof token.
* **ServiceAddress**: the service which will accept request authenticated by issued token. In this sample, it is the address of deployed service "service".
* **ServiceServiceCertificateThumbprint**: the service certificate of the above service. The certificate must exist in LocalMachine\My.
* **ServiceServiceCertificateDNSIdentity**: the dns identity of service endpoint.
* **ExpectedResponseMessage**: The expected response message from service service.
* **BppValue**: Expected Bpp value in base64 encoded format.
* **SoapMessageLogLocation**: a folder to store all the soap message sent and received to sts and service.
* **serilog:minimum-level**: specify the level of logging. Log files are stored in the Logs\ folder.

# Calling The Service Using the Anvendersystem (User Context)

Open a browser and browse to:

[https://adgangsstyringeksempler.test-stoettesystemerne.dk/Service/](https://adgangsstyringeksempler.test-stoettesystemerne.dk/STSUserContext/)

to be greeted with a welcome page.

Sample code which demonstrates how to call the service can be found in the class:

Kombit.Samples.Consumer.Consumer

The following test case demonstrates how to call the STS and then use the issued token to call a service.

SendRstAndThenExecuteServiceServiceSuccessfully