KOMBIT

Guideline for .Net Sample Service

Version 1.0

# kombit-service-net

Sample .Net WCF service and consumer using STS.

## Introduction

The following document describes how to configure the .Net-based sample service. After completing this guide, the .Net-based sample service will be configured.

This soap service 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, an Anvendersystem is also referred to as a service consumer because an Anvendersystem consumes services.

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 (WCF)
* SOAP protocol
* WS-Trust XML protocol

## The Service

Before the sample service can be configured, the necessary prerequisites to run the sample must be in place. This includes:

* Windows Server 2012R2
* Web Role
* IIS installed
* ASP.Net v5
* .Net v4.5

In addition, Visual Studio 2015 is required to build the sample.

## Setup

To build and configure the sample service, do the following:

1. Either clone the repository <https://github.com/Safewhere/kombit-service-net.git> to C:\kombit-service-net, or unpack the provided zip-file kombit-service-net.zip to C:\kombit-service-net.
2. Open C:\kombit-service-net\Kombit.Samples.Service.sln in Visual Studio, and build the solution.
3. Make sure an SSL certificate that covers the DNS name service.projekt-stoettesystemerne.dk is present inLocalMachine\My certificate store.
4. Open the Hosts-file, and map the DNS name service.projekt-stoettesystemerne.dk to localhost (127.0.0.1).
5. Create the folder c:\temp that will be used for logging, and grant Network Service full control to the folder.
6. Create a new IIS web application:
   1. The Site name should be service.projekt-stoettesystemerne.dk
   2. The Physical pathshould be C:\kombit-service-net\Kombit.Samples.Service
   3. The Binding type should be HTTPS
   4. The Host name should be service.projekt-stoettesystemerne.dk
   5. Select an appropriate SSL certificate, that matches the host name chosen in the previous step
7. Grant the application pool identity for the web application read and execute permissions to C:\kombit-service-net
8. Import all p12 files located in C:\kombit-service-net\Certificates to LocalMachine\My:
9. Grant the application pool identity for the web application read permission to the private key for all certificates imported in the previsoun step.
10. Import C:\kombit-service-net\Certificates\StsServiceCertificate.cer to LocalMachine\TrustedPeople
11. Open a browser and point it to <https://service.projekt-stoettesystemerne.dk/>

The sample service is now build and configured, and ready to be tested.

### Configuration Parameters

Some changes to the properties in the configuration file Kombit.Samples.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. The certificate must exist in LocalMachine\My
* StsSigningCertificateThumbprint The thumprint of the certificate that is used by the STS to sign tokens. The certificate must exist in LocalMachine\TrustedPeople
* ResponseMessage The response message from the Ping-method on the service.

## The Anvendersystem (Service Consumer)

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

Some changes to the properties in the configuration file:  
\Kombit.Samples.Consumer\Kombit.Samples.Consumer.dll.config

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

* StsBaseAddress the address where STS and Anvendersystem (user context) is deployed.
* AValidClientCertificateThumbprint the thumprint of a certificate that is assigned to an Anvendersystem on the STS. This certificate must be located in LocalMachine\My
* StsServiceCertificateThumbprint thumprint of a certificate which is used as service certificate for certificate endpoint. This certificate must be located in LocalMachine\My
* StsServiceCertificateDNSIdentity The DNS identity of the STS service certificate. This is the DNS identity of the certificate that is referred to by StsServiceCertificateThumbprint.
* StsCertificateEndpoint the STS certificate endpoint address.
* StsMexEndpoint The MEX endpoint address of the STS
* AnvenderContext The Anvenderkontekst to use for the RequestSecurityToken request sent to the STS.
* AValidOnBehalfOfCertificateThumbprint The thumbprint of a certificate which is used for proxy-OnBehalfOf element or used as client certificate to request OnBehalfOf token.
* ServiceBaseAddress The address of a service which will accept requests authenticated by a token issued by the STS. In this sample this is the address of deployed service.
* ServiceAddress The relative path of the service.
* ServiceServiceCertificateThumbprint the service certificate of the above service. The certificate must be located in LocalMachine\My.
* ServiceServiceCertificateDNSIdentity the DNS identity of service endpoint. This is the DNS identity of the certificate that is referred to by ServiceServiceCertificateThumbprint.
* ExpectedResponseMessage The expected response message from the service.
* BppValue The expected OIO BPP value in base64 encoded format.
* SoapMessageLogLocation a folder to store all SOAP messages sent and received to and from the STS and the service.
* serilog:minimum-level specifies the logging level. Log files are stored in the Logs\ folder.

## Calling The Service Using the Anvendersystem (Service Consumer)

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