

# **Programmers Guide - Serviceplatform**

Udarbejdet for:

KOMBIT A/S

Halfdangsgade 8

2300 København S

# Indhold

1	Introduction.....	3
2	Calling Serviceplatformen Web Services .....	3
3	Calling Redirect Services.....	4
4	Sample Code.....	4
4.1	Java .....	5
4.1.1	Demo-java.zip .....	5
4.1.2	Kombit-client.zip.....	5
4.2	C#/.Net .....	6
4.2.1	Establish trust .....	6

## 1 Introduction

This document contains a short introduction to calling web services exposed by KOMBITs Serviceplatform<sup>1</sup>.

For information about specific services see the online service catalogue or the document "SSE/12746/IFS/0001 Services on the Serviceplatform".

The reader is assumed to be familiar with SSL, the Serviceplatform self-services solution, and has established a Connection Agreement (CA) and Service Agreement (SA) to use when calling services. The reader must further have acquired a suitable SSL certificate, see below.

## 2 Calling Serviceplatformen Web Services

All services on the Serviceplatform are SOAP1.1 web services, which must be called using SSL and client certificate authentication. The client certificate used when calling must be a VOCES- or FOCES certificate<sup>2</sup>.

In addition to the certificate, any service call to the Serviceplatform must provide an Invocation Context. This is comprised by a number of parameters identifying the caller and the agreement allowing the caller to use the called service on the Serviceplatform. The required parameters are:

1. ServiceAgreementUUID – a unique identifier of the service agreement that the caller has made with a municipality.
2. ServiceUUID – a unique identifier of the service that the caller wants to use.
3. UserUUID – a unique identifier of the municipality on behalf of which the caller accesses the service.
4. UserSystemUUID – a unique identifier of the system that calls the service.

All of the values required for calling a service can be found by logging into the self-services solution, and inspecting the service agreement.

Further, the caller can as part of the invocation context provide the following optional parameters:

5. OnBehalfOfUser – identification of a user in the calling system.
6. CallersServiceCallIdentifier – an identification of the service call provided by the caller.
7. AccountingInfo – a message from the caller to be used for accounting of the service call.

These last three parameters are logged by the Serviceplatform to an audit log, when the service is called by the caller.

An example of a call to "CVR service":

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:x="http://rep.oio.dk/eogs/xml.wsdl/"
xmlns:ns="http://serviceplatformen.dk/xml/schemas/InvocationContext/1/"
xmlns:x.1="http://rep.oio.dk/eogs/xml.schema/">
  <soapenv:Header/>
  <soapenv:Body>
```

---

<sup>1</sup> See: <http://www.serviceplatformen.dk>

<sup>2</sup> FOCES and VOCES certificates are presently issued by NETS, see: <https://www.nets-danid.dk/>

```

<x.:GetLegalUnitRequest level="2">
  <ns:InvocationContext>
    <ns:ServiceAgreementUUID>43fb7e80-3f80-11e2-a32b-
d4bed98c63db</ns:ServiceAgreementUUID>
    <ns:UserSystemUUID>17b22dc2-3f80-11e2-a32b-d4bed98c63db</ns:UserSystemUUID>
    <ns:UserUUID>fb21b665-3f7f-11e2-a32b-d4bed98c63db</ns:UserUUID>
    <ns:ServiceUUID>93a48b42-3945-11e2-9724-d4bed98c63db</ns:ServiceUUID>
  <ns:CallersServiceCallIdentifier>DEMO</ns:CallersServiceCallIdentifier>
</ns:InvocationContext>
  <x.1:LegalUnitIdentifier>78834412</x.1:LegalUnitIdentifier>
</x.:GetLegalUnitRequest>
</soapenv:Body>
</soapenv:Envelope>

```

### 3 Calling Redirect Services

Redirect services are called in the same way as other services, except that the request including invocation context parameters must be supplied as URL parameters.

An example of a redirect call to the “Post Danmark FindOs Service”:

```

https://prod.serviceplatformen.dk/service/Redirect/Redirect/1?serviceAgreementUUID=uuid&userSystemUUID=uuid&userUUID=uuid&onBehalfOfUser=kbh&serviceUUID=uuid&callersServiceCallIdentifier=kombit&accountingInfo=account&postnr=4000&husnr=1&vejnavn=Algade

```

where the *uuid* parts are meant to be proper UUIDs<sup>3</sup>.

The client sending the request must be able to follow redirects (HTTP Code 307), and must use the same certificate in the call to the Serviceplatform as it would use in a call to the service directly.

### 4 Sample Code

As part of this programmer’s guide, small samples illustrating how to call a service on the Serviceplatform are available in both Java and C#/.Net.

The samples cannot run out of the box, as a connection agreement, service agreement, WSDL location, and OCES certificate must be setup in the environment (External test or Production) prior to calling.

By default the demo clients are set up to run against the external test environment. To make them run against production environment, change the URL’s defining the endpoints to production endpoints.

---

<sup>3</sup> UUIDs are in the canonical form.

All places to change in the sample code are marked with "TO\_BE\_MODIFIED".

## 4.1 Java

Two Java downloads are available:

- Demo-java.zip is simple example with all configuration in one single Java class calling the CVR service
- Kombit-client.zip is a mini framework that might serve as a starting point. Based on configuration in a property file, it makes it easy to obtain a Java client for all services.

### 4.1.1 Demo-java.zip

The Java CVRClient can be loaded into the IDE Eclipse, by creating a new project in the folder CVRClient, and run from the IDE after the configuration parameters are changed to match the calling system.

Most of the sample code is auto generated using the tool java2wsdl. See <http://cxf.apache.org/docs/java-to-wsdl.html> for information about this tool.

The file to modify and run is CvrPortType\_CvrService\_Client, in the package:  
dk.oio.rep.eogs.xml\_wsdl

The client is correctly configured if it runs and prints out the following:

Invoking searchLegalUnit...

Result: LegalUnitIdentifier=19435075

*Note that Demo-java has been tested with Java v 1.6.0\_38*

### ***Establish trust***

In order to call services it is necessary to establish trust to the Serviceplatform. This is done in 2 steps.

- 1) Download and save the server certificate from environment you wish to run on (<https://www.exttest.serviceplatformen.dk> or <https://www.prod.serviceplatformen.dk>). This may be done via a web browser, for instance Mozilla Firefox.
- 2) Add the downloaded certificate into your trust store (can be done using keytool – see example below):

```
keytool -import -alias serviceplatform.server -file serviceformen.dk.cer -  
keystore cacerts
```

### 4.1.2 Kombit-client.zip

The zip contains a compiled version ready to run from the command line (using "Person stamdata (lokal)").

See further instructions in the included readme.txt.

Moreover the code might serve as a starting point, when a new client is developed. See the class `ConnectionTest.java` for inspiration – it uses “Person stamdata” but could easily be rewritten to call another service.

Trust is automatically established using the included trustStore containing the CA used on external test and production.

*Note that Kombit-client has been tested with Java v 1.7.0\_51-b13 and 1.8.0\_25*

## 4.2 C#/.Net

The solution can be opened in Visual Studio 2010 and run from the IDE, after the configuration parameters are changed to match the calling system. The sample code is developed using WCF.

Most of the sample code is auto generated using VS2010.

Files to modify and run are `Program.cs` and `app.config`.

The client is correctly configured if it runs and prints out the following:

```
Invoking searchLegalUnit...
```

```
Result: LegalUnitIdentifier=19435075
```

```
Press Enter to complete
```

### 4.2.1 Establish trust

In order to call services it is necessary to establish trust to the Serviceplatform. This is done in 2 steps.

- 1) Download and save the server certificate from environment you wish to run on (<https://exttestwww.serviceplatformen.dk> or <https://www.serviceplatformen.dk>). This may be done via a web browser, for instance Mozilla Firefox.
- 2) Add the downloaded certificate into your trust store (can be done using `mmc.exe`, see [http://msdn.microsoft.com/en-us/library/bb950259\(v=bts.10\).aspx](http://msdn.microsoft.com/en-us/library/bb950259(v=bts.10).aspx) for usage of this tool)