

# .NET Framework klienteksempel



Klienteksemplet virker ikke i ASP.NET Core



Anvendelsen af klienteksemplet forudsætter, at kaldet kommer fra en IP-adresse som er godkendt hos STIL. Har I spørgsmål er der hjælp at finde på [denne side](#).

Klienteksemplet er et Visual Studio projekt, der har til formål at vise, hvordan man integrerer til STILs integrationsplatform, IPL. Eksemplet inkluderer også konfiguration af IPL sikkerhedsmodel i Visual Studio.

1

## Download klienten

Du kan downloade klienteksemplet nedenfor.



helloworldclient.zip



Certifikatet i klienteksemplet er udløbet. Det nye certifikat fra Nets kan findes her:

<https://www.nets.eu/dk-da/kundeservice/nemid-tjenesteudbyder/NemID-tjenesteudbyderpakken/Pages/OCES-II-certifikat-eksempler.aspx>

2

## Importere Integrationsplatformens servercertifikat



Certifikatet kan hentes på [denne side](#)

1. Flyt certifikatet ind i helloworldclient-mappen, så certifikat-filen ligger sammen med README.md-filen.
2. I Visual Studio åbnes HelloWorldClientTests.cs filen og på linje 26 ændres string value for variabelen serviceCertificateFileName til det fulde filnavn for Integrationsplatformens certifikat inkl. filtype, så hvis certifikatet blev navngivet "Servercertifikat\_03-02-2021.cer" skal serviceCertificateFileName have denne værdi.

## 3

### Import af WSDL i Visual Studio

1. I Visual Studio højreklikkes på "Connected Services" og der vælges "Add Service Reference...".
2. I vinduet, der åbner, indtastes i "Address" feltet url for WSDL og der klikkes herefter på "Go" knappen. Visual Studio søger på url'en og finder servicen, hvorefter den fremkommer i feltet "Services" og webservicens operationer kan ses i feltet "Operations", når man har klikket på servicen i feltet "Services".
3. Klik herefter på "OK" knappen, hvorved Visual Studio importerer WSDL'en og genererer den nødvendige kode til at kunne kalde webservicen gennem en klient.

### Eksempel

[blocked URL](#)

## 4

### Afvikling af test i klienteksemplet

Klienteksemplet indeholder et testprojekt. Testprojektet benytter XUnit testframeworket. Projektets tests kan afvikles ved at bygge projektet, hvorefter projektets tests bliver synlige og kan afvikles. Skulle de ikke blive synlige, kan de findes ved at vælge vinduet Test Explorer.

Herfra kan testene afvikles ved at klikke på "Run All" eller ved at højreklikke på hver enkelt test, hvorefter der kan vælges "Run" eller "Debug".

### Eksempel

[blocked URL](#)

## 5

### Klar til at kalde HelloWorld på Integrationsplatformen.

Alle andre operationer end ping kræver signering af requestet. Dette er vist i KlientEksempel-projektet med testmetoden `GetAliveWithParameterHelloWorldReturnsData`. Testmetoden `PingHelloWorldReturnsUp` benytter også IPLs sikkerhedsmodel og signerer requestet. Dette er for at vise, at det på ping-operationen ikke gør nogen forskel, om der benyttes signering eller ej.



#### Problemer med http statuskode 400?

##### Symptom

Det er ikke muligt at se indholdet af beskeden fra Integrationsplatformen når Integrationsplatformen sender 'Bad request' pga. at request ikke overholder skemaet for servicen.

##### Årsag

Fejlbeskeder angående skemavalideringsfejl giver http 400 fra IPL, og .NET kan ikke som standard udlæse indholdet af disse.

## Løsning

Erhvervsstyrelsen har lavet et eksempel på kode som gør, at selve beskeden fra IPL kan udlæses, selvom den kommer med http 400 (Bad request). Se kodeeksempel nedenfor.

### Eksempelkode

```
        /// SOAP faults might be
returned with a http code 400 (Bad request),
        /// if that is the case we need to manually
get the SOAP fault from the WebException

        private Exception
GetSoapFaultFromHttpException(System.Net.
WebException e)
        {
            StringBuilder sb = new StringBuilder();
            Stream s = e.Response.
GetResponseStream();
            // Try to read the fault
            try
            {
                byte[] readBuffer = new
byte[1000];

                int count = 0;
                bool finish = false;
                while (!finish)
                {
                    count = s.Read(readBuffer, 0,
readBuffer.Length);
                    if (count ==
0)
                    {
                        // EOF
                        finish = true;
                    }
                    else
                    {
                        sb.Append(System.Text.
Encoding.UTF8.GetString(readBuffer, 0, count));
                    }
                }
            }
            catch
            {
                return e;
            }
            finally
            {
                s.Close();
            }
            // Try to make it a SOAP faultobject
            // move declaration here, for easy
debugging
            MemoryStream memStream = null;
            Message message;
            MessageFault msgFault;
            Exception exception;
            try
            {
                ;
                using (memStream = new MemoryStream
(Encoding.Default.GetBytes(sb.ToString())))
                {
                    XmlTextReader xmlReader = new
XmlTextReader(memStream);
                    message = Message.CreateMessage
(xmlReader, int.MaxValue, MessageVersion.
Soap12WSAddressing10);
                }
                //OiosiMessage oio = new OiosiMessage
```

```

(message);
        msgFault = MessageFault.CreateFault
(message, int.MaxValue);
        exception =
CreateFaultWasReceivedException(new FaultException
(msgFault));
    }
    catch (Exception)
    {
        string errorMessage = sb.ToString();
        if (string.IsNullOrEmpty
(errorMessage))
        {
            if (e.GetBaseException() != null)
            {
                // The e.Message is just:
                'Protocol Error'.
                // Better error description
                here:
                errorMessage = e.
GetBaseException().Message;
            }
            else
            {
                errorMessage = e.Message;
            }
        }
        return new Exception
(errorMessage);
    }
    return
exception;
}

```