Integrationsvejledning for Virksomhedsservice Light

Version 1.0

NEM REFUSION

KOMBIT
Kommunernes it-fællesskab

Edora A/S

Indhold

Integrationsvejledning for Virksomhedsservice Light Version 1.0

Formål

Virksomhedsservice Light er en SOAP-baseret webservice, der muliggør adgang til og behandling af indberetninger direkte fra eksterne systemer.

- Servicen eksponerer metoder på **engelsk**, mens request- og response-elementer benytter **danske navne**.
- Integration sker via **SOAP-protokollen**, hvilket sikrer standardiseret kommunikation mellem systemer.

Involverede systemer

Tabellen nedenfor giver et overblik over de involverede systemer og deres rolle i integrationen.

ROLE	SYSTEM	BESKRIVELSE
Provider	NemRefusion	System der udstiller SOAP-servicen og håndterer ind- og udgående anmodninger.
Consumer(s)	Eksterne systemer	Systemer der integrerer med NemRefusion for at indsende og hente data via servicen.

Information og Dokumentation

Før integration med **Virksomhedsservice Light** anbefales det at gennemgå **snitfladebeskrivelsen** for at få en klar forståelse af serviceopsætningen og XML-strukturen, der anvendes ved kald til endpoints.

Detaljerede specifikationer og XML-eksempler findes i **snitfladebeskrivelsen**:

KOMBIT - NemRefusion

Adgang til servicen

Virksomhedsservice Light er ikke offentligt tilgængelig. Adgang kan opnås ved at kontakte **NemRefusion** efter anskaffelse af de nødvendige certifikater.

NemRefusion - Supportportal

Anskaffelse af certifikater

For at tilgå services, er der behov for to *OCES3 certifikater*. Et til at kalde servicen og et til at signere.

Der findes to typer **OCES3-certifikater**:

- **TEST-certifikat** – anvendes til testmiljøet og anbefales ved udvikling og validering.
- **PROD-certifikat** – anvendes i produktionsmiljøet.

Begge certifikater er af typen **Systemcertifikat**. For at få oprettet de nødvendige certifikater skal I kontakte jeres **MitID-ansvarlige**.

Yderligere information og vejledning til anskaffelse af **TEST** og **PROD** certifikater findes her: https://viden.stil.dk/display/OFFintegrationsplatformen/OCES3+certifikater

Registrering af Klientcertifikat

For at kunne tilgå Virksomhedsservice Light, skal klienten autentificere sig med et OCES3-systemcertifikat. Ud over at certifikatet skal være gyldigt og installeres korrekt, kræves det, at thumbprintet registreres hos NemRefusion.

Sådan registreres certifikatet:

1. Åbn Windows Certificate Store (certmgr.msc).
2. Find certifikatet under "Current User" → "Personal" → "Certificates".
3. Højreklik på certifikatet og vælg "Properties".
4. Kopiér værdien under "Thumbprint".
5. Send thumbprintet til NemRefusion support, så det kan registreres i systemet.

Bemærk: Servicen afviser alle anmodninger fra et certifikat, der ikke er registreret.

Opsætning af klienten

Bemærk, at adgang til følgende kræver et klientcertifikat. Se det relevante afsnit ovenfor for yderligere information.

For at teste denne service skal du oprette en klient og bruge den til at kalde servicen. Dette kan gøres ved at bruge værktøjet **svcutil.exe** fra kommandolinjen med følgende syntaks:

```
svcutil.exe https://vslight.nemrefusion.dk/VSLight.svc?wsdl
```

Du kan også tilgå servicebeskrivelsen som en enkelt fil via:

```
https://vslight.nemrefusion.dk/VSLight.svc?singleWsdl
```

Dette vil generere en **konfigurationsfil** og en **kodefil**, som indeholder klientklassen. Tilføj disse filer til din klientapplikation, og brug den genererede klientklasse til at kalde servicen.

Eksempel:

C#

```
class Test
{
    static void Main()
    {
        VSLightClient client = new VSLightClient();

        // Brug variablen 'client' til at kalde operationer på servicen.

        // Luk altid client.
        client.Close();
    }
}
```

Hvis der opstår problemer med adgang til **WSDL** og **XSD-filer** pga. certifikatkrav, kan filerne hentes og bruges lokalt.

Sådan downloader du WSDL-filerne lokalt:

1. Åbn en browser og gå til: https://vslight.nemrefusion.dk/VSLight.svc?wsdl
2. Gem filen som **VSLight.wsdl**.
3. Åbn den gemte WSDL-fil i en teksteditor og find referencer til `?xsd=xsd0` osv.
4. Download de tilhørende XSD-filer manuelt:
   https://vslight.nemrefusion.dk/VSLight.svc?xsd=xsd0
   https://vslight.nemrefusion.dk/VSLight.svc?xsd=xsd1
5. Gem dem som **xsd0.xsd** og **xsd1.xsd** i samme mappe som VSLight.wsdl.
6. Opdater VSLight.wsdl til at pege på de lokale XSD-filer:

```
<import namespace="..." schemaLocation="xsd0.xsd"/>
<import namespace="..." schemaLocation="xsd1.xsd"/>
```

7. Tilføj den lokale VSLight.wsdl til dit projekt i **Visual Studio** og opret en **service reference**.

Fordel ved at bruge en lokal WSDL:

- Omgår adgangsproblemer ved certifikatgodkendelse.
- Hurtigere udvikling, da servicen ikke behøver at være tilgængelig.

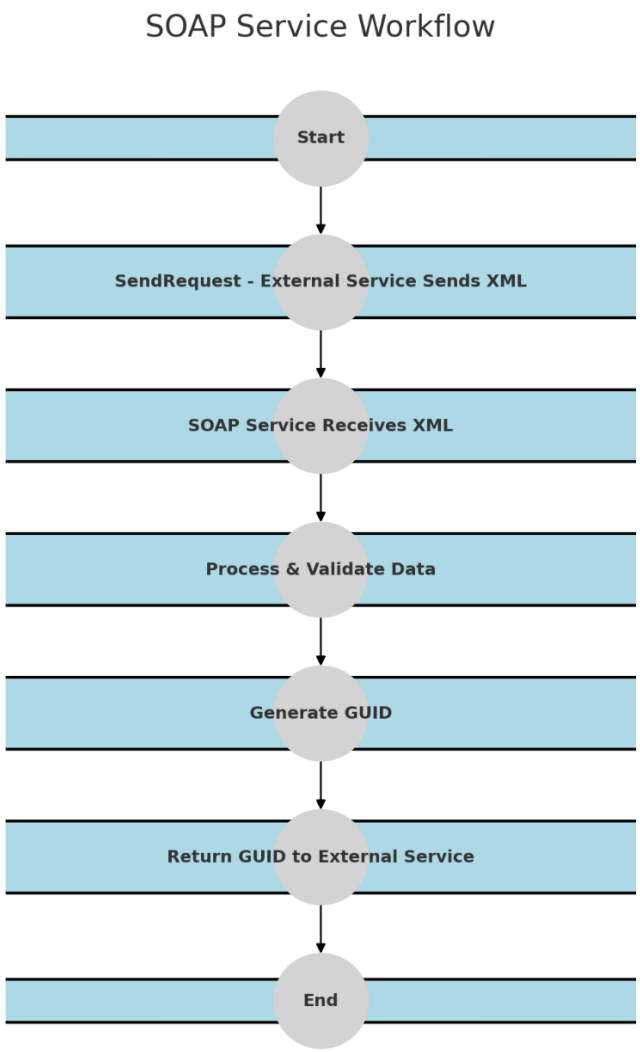
Bemærk: Når integrationen er testet, skal klienten skifte tilbage til den officielle service-URL.

Business Flow

SendRequest

Nedenstående diagram illustrerer det overordnede **workflow** for SOAP-servicen, der håndterer XML-baserede anmodninger fra eksterne systemer. Workflowet beskriver, hvordan en anmodning sendes, behandles og returneres med et unikt identifikationsnummer (**GUID**).

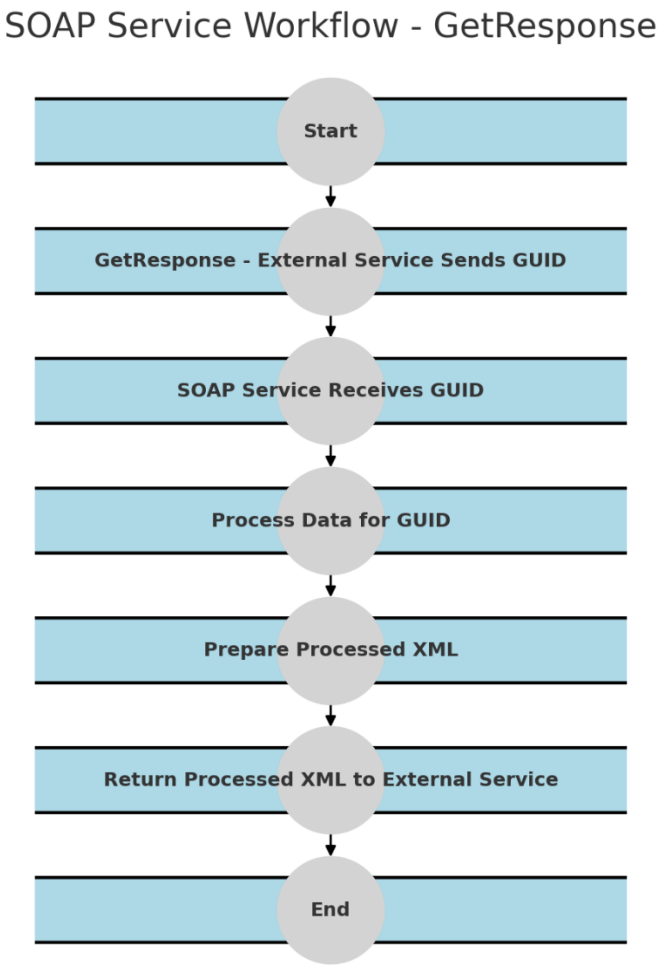
Diagrammet giver et visuelt overblik over de vigtigste trin i processen og understøtter forståelsen af servicen's funktionsmåde.



GetResponse

Dette diagram illustrerer workflowet for **GetResponse**, hvor en ekstern service sender et **GUID** for at anmode om resultater, der er genereret på baggrund af tidligere indsendt data. Servicen modtager GUID'et, behandler de tilhørende data og returnerer et XML-svar med de relevante resultater.

Diagrammet giver et visuelt overblik over, hvordan servicen håndterer forespørgsler og sikrer en struktureret udveksling af data mellem systemerne.



Support

Nødvendig information og vejledning om **Virksomhedsservice Light** findes på **Virk NemRefusion**: https://virk.nemrefusion.dk/information-and-help/20250647 . Her kan du finde dokumentation, retningslinjer og ofte stillede spørgsmål.

For teknisk support, fejlmelding eller yderligere assistance kan **NemRefusion Support** kontaktes via: https://support.edora.dk/servicedesk/customer/portal/10 . Dette supportcenter håndterer henvendelser vedrørende integration, adgang og tekniske problemer.

Brug af service og dataspecifikation

Data Specifikation

SOAP anvendes som kommunikationsprotokol. Detaljeret dokumentation og snitfladebeskrivelse kan findes hos KOMBIT - NemRefusion under dokumentationssektionen.

KALD OG ENDPOINTS

Endpoints i VSLight følger et 'anmod og hent'-princip. Først sendes en signeret anmodning til behandling via et POST-kald. Derefter kan resultatet hentes med et GET-kald ved at angive den tilhørende identifikator.

METODE	BESKRIVELSE	INPUT	OUTPUT
SendRequest	Modtager et XML-baseret request for at tilføje en ny anmodning til CompanyServiceLight-systemet.	XML-dokument med indberetning	UUID (string) som reference til anmodningen
GetRequest	GUID (string)	Henter resultatet af en tidligere anmodning baseret på dets GUID.	XML-dokument med svardata

Sammenhængen mellem metoderne er sekventiel:

1. SendRequest **opretter en ny anmodning** og returnerer et unikt UUID.
2. GetResponse **henter resultatet** af denne anmodning ved hjælp af UUID'et.

Det er vigtigt, at klienten venter på, at servicen har færdigbehandlet en anmodning, før GetResponse kaldes, bemærk at indsendelsen kun lægger anmodningen i kø til at blive behandlet og der kan gå lidt tid før servicen kommet til at behandle den. Hvis resultatet ikke er klar, kan klienten lave en ny forespørgsel senere.

DATA KONVERTERING

Ved afsendelse af en anmodning via SendRequest(XElement XML), skal data konverteres til en XML-struktur, der overholder VSLight's specifikationer. Dette indebærer:

- Strukturering af XML: Alle nødvendige elementer og attributter skal være korrekt defineret.
- Namespace og signatur: XML-dokumentet skal inkludere påkrævede namespaces og eventuelle digitale signaturer.
- Feltnavnsmapping: Interne datafelter skal mappes til de tilsvarende XML-felter.

Læs mere omkring strukturen i snitfladebeskrivelsen.

TILFØJELSE AF DIGITAL SIGNATUR TIL XML-ANMODNINGER I EN C# APPLIKATION

Denne guide beskriver, hvordan du **signerer en XML-anmodning** digitalt, inden den sendes til **Virksomhedsservice Light**. Signaturen sikrer, at data ikke er blevet ændret undervejs og er et krav for at kommunikere med servicen.

For at integrere med servicen korrekt skal du:

1. **Indlæse et gyldigt certifikat** fra Windows Certificate Store.
2. **Tilføje en digital signatur** til XML-dokumentet.
3. **Sende den signerede XML** via SOAP-webservicen.

1. Indlæsning af OCES3-certifikat

Virksomhedsservice Light kræver, at XML-anmodninger signeres med et **OCES3-systemcertifikat**. For at hente dette certifikat fra Windows Certificate Store kan du bruge følgende metode:

```
using System.Security.Cryptography.X509Certificates;

private static X509Certificate2 LoadCertificate(string thumbprint)
{
    var store = new X509Store(StoreName.My, StoreLocation.CurrentUser);
    store.Open(OpenFlags.ReadOnly);
    var result = store.Certificates.Find(X509FindType.FindByThumbprint, thumbprint, false);

    if (result.Count == 0)
    {
        throw new Exception("Certifikat ikke fundet. Kontrollér thumbprint og
installation.");
    }

    return result[0];
}
```

Forklaring

- Certifikatet hentes i dette eksempel fra **CurrentUser's "My" store** baseret på dets **thumbprint**.
- Hvis certifikatet ikke findes, kastes en fejl.
- Certifikatet bruges senere til at signere XML-data.

En alternativ metode er at **konfigurere det direkte i app.config eller web.config**.

Eksempel på app.config med klientcertifikat

```
<system.serviceModel>
  <behaviors>
    <endpointBehaviors>
      <behavior name="ClientCertificateBehavior">
        <clientCredentials>
          <clientCertificate storeLocation="LocalMachine"
                             storeName="My"
                             x509FindType="FindByThumbprint"
                             findValue="THUMBPRINT_HER" />
        </clientCredentials>
      </behavior>
    </endpointBehaviors>
  </behaviors>

  <client>
    <endpoint address="https://vslight.nemrefusion.dk/VSLight.svc"
              binding="basicHttpBinding"
              bindingConfiguration="BasicHttpBinding_IVSLight"
              contract="VSLightService.IVSLight"
              name="BasicHttpBinding_IVSLight"
              behaviorConfiguration="ClientCertificateBehavior" />
```

```
  </client>
</system.serviceModel>
```

2. Signering af XML-anmodningen

Når du har hentet certifikatet, kan du tilføje en **digital signatur** til XML-anmodningen. Det gøres ved at:

1. Oprette en **SignedXml** instans.
2. Tilføje en **reference** til hele dokumentet.
3. Generere signaturen og indsætte den i XML-dokumentet.

```
 1. using System.Security.Cryptography.Xml;
 2. using System.Xml;
 3.
 4. private static XmlDocument SignXmlDocument(XmlDocument xmlDocument, X509Certificate2
certificate)
 5. {
 6.     // Opretter SignedXml objekt og angiver signeringsnøgle fra certifikatet
 7.     SignedXml signedXml = new SignedXml(xmlDocument)
 8.     {
 9.         SigningKey = certificate.GetRSAPrivateKey()
10.     };
11.
12.     // Tilføjer en reference til hele XML-dokumentet
13.     Reference reference = new Reference { Uri = "" };
14.     XmlDsigEnvelopedSignatureTransform env = new XmlDsigEnvelopedSignatureTransform();
15.     reference.AddTransform(env);
16.     signedXml.AddReference(reference);
17.
18.     // Tilføjer certifikatoplysninger til signaturen
19.     KeyInfo keyInfo = new KeyInfo();
20.     keyInfo.AddClause(new KeyInfoX509Data(certificate, X509IncludeOption.ExcludeRoot));
21.     signedXml.KeyInfo = keyInfo;
22.
23.     // Genererer signaturen og tilføjer den til XML-dokumentet
24.     signedXml.ComputeSignature();
25.     XmlElement signatureElement = signedXml.GetXml();
26.     xmlDocument.DocumentElement.PrependChild(xmlDocument.ImportNode(signatureElement,
true));
27.
28.     return xmlDocument;
29. }
```

Forklaring

- **SigningKey** sættes til certifikatets **private key**.
- **Reference** sikrer, at hele XML-dokumentet signeres.
- **KeyInfo** tilføjes for at inkludere nødvendige certifikatoplysninger.
- **ComputeSignature()** beregner signaturen og tilføjer den til XML.

3. Serialisering og afsendelse af den signerede XML

Når XML-dokumentet er signeret, skal det sendes til servicen. Først serialiseres objektet til et XmlDocument, hvorefter det signeres og sendes via SOAP-webservicen.

```
1. private static async Task<string> SendRequest<E>(E request)
```

```
2. {
3.     XmlDocument xmlDoc = SerializeToXmlDocument(request);
4.     XmlDocument signedXml = SignXmlDocument(xmlDoc,
LoadCertificate("DINE_CERTIFIKAT_THUMBPRINT"));
5.
6.     return await _vsLightClient.SendRequestAsync(signedXml.DocumentElement);
7. }
```

Forklaring

- **SerializeToXmlDocument(request)** konverterer objektet til XML.
- **SignXmlDocument()** signerer XML-dokumentet med certifikatet.
- **SendRequestAsync()** sender den signerede XML til SOAP-servicen.

TILFØJELSE AF OCES3-CERTIFIKAT TIL HTTPCLIENT I C#

Når certifikatet er hentet (her genbruger vi metoden `LoadCertificate` fra det foregående afsnit, husk det ikke må være samme certifikat der bruges), skal det tilføjes til en **HttpClient**, så alle anmodninger automatisk bruger det til autentificering.

Dette gøres ved at oprette en **HttpClientHandler**, der indeholder certifikatet:

```
 1. using System.Net.Http;
 2.
 3. private static HttpClient CreateHttpClientWithCertificate(string certThumbprint)
 4. {
 5.     var cert = LoadCertificate(certThumbprint);
 6.
 7.     var handler = new HttpClientHandler();
 8.     handler.ClientCertificates.Add(cert); // Tilføjer certifikatet til HTTP-anmodninger
 9.
10.     return new HttpClient(handler);
11. }
```

Forklaring

- **HttpClientHandler** bruges til at konfigurere klienten.
- **Certifikatet tilføjes** til klienten for at sikre, at alle anmodninger autentificeres.
- En **HttpClient** oprettes med denne konfiguration.

Når HttpClient er konfigureret med certifikatet, kan du sende en anmodning til servicen:

```
 1. private static async Task SendRequest()
 2. {
 3.     var httpClient = CreateHttpClientWithCertificate("DINE_CERTIFIKAT_THUMBPRINT");
 4.
 5.     var requestContent = new StringContent("<xml>Din anmodning her</xml>", Encoding.UTF8,
"application/xml");
 6.
 7.     HttpResponseMessage response = await httpClient.PostAsync(
 8.         "https://vslight.nemrefusion.dk/vslight.svc", requestContent
 9.     );
10.
11.     if (response.IsSuccessStatusCode)
12.     {
```

```
        Console.WriteLine("Anmodning sendt succesfuldt.");
    }
    else
    {
        Console.WriteLine($"Fejl: {response.StatusCode}");
    }
}
```

Forklaring

- **Opretter en HttpClient** med OCES3-certifikatet.
- **Sender en POST-anmodning** med XML-data til servicen.
- **Logger succes eller fejl** afhængigt af responsen.

FEJLHÅNDTERING

Ved fejl kan servicen returnere SOAP Faults. Almindelige fejl inkluderer:

FEJLKODE	FEJLMEDDELELSE	BESKRIVELSE
101	"RequestUUID does not contain a valid UUID. (Example: 98f44b5b-5a45-4ad9-b13e-4b9366e5c01f)"	UUID'et i GetResponse-kaldet er ugyldigt eller mangler.
103	"Request XML could not be delivered to an underlying system. Please try again later."	SendRequest kunne ikke levere XML-data til det bagvedliggende system.
104	"Response XML could not be acquired from an underlying system. Please try again later."	GetResponse kunne ikke hente resultatet fra det bagvedliggende system.
InternalServiceFault	"The server was unable to process the request due to an internal error..."	Intern fejl i servicen, som kræver fejlsøgning.
Certifikatfejl	"Client certificate is not a registered subscriber."	Klientens certifikat er ikke registreret i systemet.
Manglende certifikat	"No client certificate thumbprint."	Servicen kunne ikke finde et thumbprint i klientens certifikat.

For at håndtere disse fejl korrekt i en klientapplikation kan du bruge en try-catch-blok, der fanger FaultException og logger fejlen:

```
try
{
    XmlElement response = await client.GetResponseAsync(uuid);
}
catch (FaultException ex)
{
    Console.WriteLine($"SOAP Fault modtaget: {ex.Message}");
```

```
 8. }
 9. catch (Exception ex)
10. {
11.     Console.WriteLine($"Uventet fejl: {ex.Message}");
12. }
```

Hvis du får fejlkode 103 eller 104, kan du forsøge at sende anmodningen igen senere. Hvis fejlkode 101 returneres, skal du sikre, at det UUID, du bruger, er korrekt.

Maksimal Datastørrelse og Timeout

Ved brug af GetResponse kan der returneres store mængder data. For at undgå fejl ved store svar bør klienten justere **WCF-konfigurationen** for at tillade større meddelelser.

Sådan justeres konfigurationen i WCF-klientens app.config eller web.config:

Tilføj eller rediger følgende i <bindings>-sektionen

```
<bindings>
  <basicHttpBinding>
    <binding name="BasicHttpBinding_IVSLight"
             maxBufferSize="65536000"
             maxReceivedMessageSize="65536000"
             closeTimeout="00:01:00"
             openTimeout="00:01:00"
             receiveTimeout="00:10:00"
             sendTimeout="00:01:00">
      <security mode="Transport">
        <transport clientCredentialType="Certificate"/>
      </security>
    </binding>
  </basicHttpBinding>
</bindings>
```

Forklaring af justeringerne:

- **maxReceivedMessageSize**: Øger den maksimale størrelse for en modtaget besked til **65 MB**.
- **maxBufferSize**: Øger bufferen for at matche max message size.
- **receiveTimeout**: Forlænger tidsgrænsen til **10 minutter** for langsomme svar.

Bemærk: Hvis svaret er større end maxReceivedMessageSize, vil anmodningen fejle med en **"Message size exceeded"** fejl.

Datasikkerhed og Fortrolighed

VSLight-service håndterer **personfølsomme og forretningskritiske data**, herunder **CVR-numre, virksomhedsoplysninger og signerede anmodninger**.

For at sikre fortrolighed og integritet implementeres følgende foranstaltninger:

- **Transportkryptering (TLS 1.2+)** sikrer sikker dataoverførsel.
- **Adgangskontrol** begrænser adgang til autoriserede systemer og brugere.

- **Logging og overvågning** registrerer adgang til følsomme oplysninger.
- **Dataopbevaring** sker kun i nødvendigt omfang, og følsomme data slettes eller anonymiseres efter gældende regler.

Følsomme data håndteres i overensstemmelse med **GDPR** og interne sikkerhedspolitikker.

Transport Specification

Tekniske Kendetegn

Virksomhedsservice Light er en **SOAP-baseret** webservice, der følger et **request/response**-mønster. Kommunikationen sker over **HTTPS**, hvilket sikrer sikker dataoverførsel mellem systemer.

- **Exchange Pattern**: Request/Response
- **Transport Protocol**: HTTPS (SOAP over HTTP)
- **Coupling Characteristics**:
    - **SendRequest(XElement XML)** er **synkron**, hvor klienten sender en XML-anmodning og modtager et unikt reference-ID.
    - **GetResponse(string RequestUUID)** muliggør **asynkron datahentning**, da en ekstern service kan anmode om det færdigbehandlede svar baseret på en tidligere anmodning.

Servicen understøtter både **indsendelse af data (SendRequest)** og **hentning af resultatdata (GetResponse)**, hvilket muliggør en fleksibel integration mellem eksterne systemer og **NemRefusion**.

Dataudvekslingsformat

Virksomhedsservice Light udveksler data i **XML-format** via en **SOAP-baseret** webservice over **HTTPS**.

- **Request:** XML via SendRequest(XElement XML)
- **Response:** XML via GetResponse(string RequestUUID)

Alle udvekslinger følger et defineret **XML-skema (XSD)** for validering og struktur.

Eksterne Krav og Begrænsninger

Brugen af **Virksomhedsservice Light** er underlagt følgende eksterne begrænsninger:

- **Adgangskontrol:**
    - Kræver **to OCES3-systemcertifikater** (ét til autentificering og ét til signering).
    - Adgang gives kun efter aftale med **NemRefusion**.
- **Afhængigheder:**

- Kald til **GetResponse** kræver et gyldigt **RequestUUID**, som først opnås via **SendRequest**.
- XML-strukturen skal følge et **foruddefineret XSD-skema** for at blive accepteret af servicen.

- **Kommunikationskrav:**
  - Servicen er kun tilgængelig via **HTTPS (SOAP over HTTP)** og kræver korrekt konfigureret **TLS 1.2+**.

Disse begrænsninger skal overholdes for en vellykket integration med servicen.

Kald af Servicen

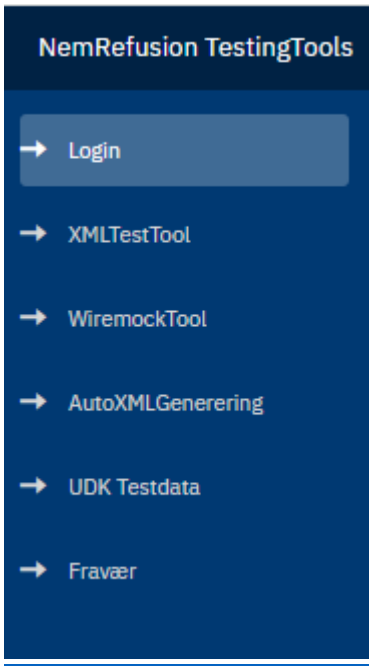
Virksomhedsservice Light kaldes **on demand**, hvor eksterne systemer sender en anmodning via **SendRequest**, og resultatet hentes senere via **GetResponse**. Servicen er ikke batch-baseret og aktiveres kun ved specifikke forespørgsler.

Service Endpoints

Environment	Endpoint URL
External Test	https://nemrefusion-apps-exttest.edora.dk/CompanyServiceLight/VSLight.svc
Produktion	https://vslight.nemrefusion.dk/vslight.svc

TEST

Løsningen kan testes mod ekstern Test, det er muligt at tilgå miljøet og skabe brugere og se sager der ligger på miljøet via. Test værktøjet: https://nemrefusion-apps-test.edora.dk/testingtools/login



Login indeholder brugere man kan bruge eller man kan skabe sine egne.

XMLTestTool giver mulighed for at indsende XML som virksomhed eller Kommune.

Wiremock viser og tillader oprettelse af test opsætning på miljøet.

Fravær giver mulighed for at søge på alle sager på miljøet, og se XML (så man kan oprette en sag via UI og hente XML så man har noget at udvikle efter).

Der ydes **ikke** support på test værktøjet så det er fri leg, dog bedes man ikke ændre på testdata / test opsætning man ikke selv har oprettet.