

Udviklingsvejledning til web service (INSPIRE-søgetjeneste)

Indholdsfortegnelse

Indholdsfortegnelse	1
Indledning	1
Generelt	1
.NET og C#	2
Transaktioner	3
Java	5
Transaktioner	6
Testmiljø	7
Referencer	8

Indledning

Geodata-info.dk har et web service interface, der udstiller operationer specificeret i de tekniske retningslinjer for INSPIRE Discovery Services version 2.12 samt standarden OGC Catalogue Services Specification 2.0.2 - ISO Metadata Application Profile for CSW 2.0, version 1.0.0. En del af disse operationer udstilles som operationer i en SOAP-tjeneste, som er beskrevet ved følgende WSDL:

<http://www.geodata-info.dk/csw?wsdl>

Det har imidlertid vist sig, at gængse programmeringsværktøjer har problemer med at autogenerere kode ud fra den givne WSDL. Det er fx ikke muligt at benytte den til at tilføje en service reference i MS Visual Studio 2010. Man må derfor benytte en mere manuel fremgangsmåde ved udvikling af programmel, der tilgår tjenesten.

Denne vejledning beskriver hvordan man programmatisk kan tilgå Geodata-info.dks web service interface med SOAP (HTTP POST). Der gives eksempler på tilgang fra .NET/C# og Java.

Der henvises til brugervejledningen [1] for en nærmere beskrivelse af operationerne i søgetjenesten og til CSW-standardens [2] for en fuldstændig reference til CSW.

Generelt

SOAP-tjenesten tilgås ved HTTP POST, dvs. en SOAP-besked i XML sendes til tjenesten med en POST-operation i HTTP-protokollen.

Når det ikke er muligt at autogenerere kode fra den givne WSDL, har man ikke automatisk operationer og datastrukturer for beskeder til rådighed. Man skal derfor selv opbygge de SOAP-beskeder, der skal sendes. Dette gøres typisk ved simpel sammensætning af tekststrengene, men man kan naturligvis også implementere specialiserede datastrukturer til formålet.

Eksemplerne i de herpå følgende afsnit har til formål at illustrere afsendelse af SOAP-beskeder til tjenesten ved HTTP POST. Opbygning af SOAP-beskederne samt fortolkning og præsentation af svarene er ikke behandlet videre. Eksemplerne er vejledende og er ment som hjælp til at komme i gang – de er ikke fyldestgørende mht. parametre og fejlhåndtering.

.NET og C#

Hvis man benytter MS Visual Studio 2010 og .NET-frameworket, kan man benytte sig af indbyggede faciliteter til at sende HTTP-requests, som illustreret i det følgende eksempel.

Klassen ExampleSoapClient indeholder metoden httpSoapTest, der sender en tekst til SOAP-tjenesten på Geodata-info.dk, og metoden simpleSoapRequest, der opbygger et GetCapabilities-request som en tekststreng.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Net;
using System.IO;

namespace WindowsFormsApplication1
{
    class ExampleSoapClient
    {
        string endPoint = "http://www.geodata-info.dk/csw-test";

        public string httpSoapTest(string soapXml)
        {
            //Dan xml-dokument fra tekststreng
            System.Xml.XmlDocument doc = new System.Xml.XmlDocument();
            doc.InnerXml = soapXml.ToString();

            try
            {
                //Opret request til tjenesten og indstil parametre
                HttpWebRequest req = (HttpWebRequest)WebRequest.Create(endPoint);
                req.Timeout = 180000;
                req.Method = "POST";
                req.Headers.Add("SOAPAction:");
                req.ContentType = "application/soap+xml;charset=utf-8";

                //Send xml-dokument
                Stream reqStrm = req.GetRequestStream();
                doc.Save(reqStrm);
                reqStrm.Close();

                //Hent svar
                WebResponse resp = req.GetResponse();
                Stream respStrm = resp.GetResponseStream();
                StreamReader strmRdr = new StreamReader(respStrm);
                string str = strmRdr.ReadToEnd();
                strmRdr.Close();

                return str;
            }
            catch (Exception e)
            {
                System.Console.WriteLine();
                System.Console.Write(e.ToString());
            }
        }
    }
}
```

```

        System.Console.WriteLine();
    }

    return null;
}

public string simpleSoapRequest()
{
    //Dan tekststreng med GetCapabilities-request i SOAP-wrapping
    string str = "<soap:Envelope xmlns:soap="
        + "\"http://www.w3.org/2003/05/soap-envelope\" "
        + "xmlns:csw=\"http://www.opengis.net/cat/csw/2.0.2\" "
        + "xmlns:ows=\"http://www.opengis.net/ows\">"
        + "<soap:Header/>"
        + "<soap:Body>"
        + " <csw:GetCapabilities service="
        + " \"http://www.opengis.net/cat/csw\">"
        + " <ows:AcceptVersions>"
        + " <ows:Version>2.0.2</ows:Version>"
        + " </ows:AcceptVersions>"
        + " </csw:GetCapabilities>"
        + "</soap:Body>"
        + "</soap:Envelope>";

    return str;
}
}
}

```

For at benytte eksemplet i MS Visual Studio 2010, kan man oprette en Windows Forms Application og oprette en ny klasse, ExampleSoapClient, med kodeeksemplet. Man kan derefter benytte følgende kodestump til at invokere eksemplet fra sit program:

```

ExampleSoapClient sc = new ExampleSoapClient();
string soapResp = sc.httpSoapTest(sc.simpleSoapRequest());
System.Console.WriteLine();
System.Console.Write(soapResp);
System.Console.WriteLine();

```

I eksemplet opbygges SOAP-beskeden direkte som en tekststreng, og svaret bliver blot skrevet til konsollen. I praksis vil der naturligvis ligge et større arbejde i at opbygge korrekte forespørgsler og efterfølgende præsentere svarene. Fejlhåndtering i eksemplet er ligeledes rudimentær.

Transaktioner

Man har mulighed for at indsætte, opdatere og slette sine metadata ved hjælp af CSW Transaction-operationen. Det kræver brugernavn og password at administrere sine metadata records, og derfor skal SOAP-requests til Transaction-operationen indeholde en header med brugernavn og password på formen vist i følgende eksempel:

```

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:oas="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:csw="http://www.opengis.net/cat/csw/2.0.2"
xmlns:ogc="http://www.opengis.net/ogc">

```

```

<soap:Header>
  <oas:Security>
    <oas:UsernameToken>
      <oas:Username>Brugernavn</oas:Username>
      <oas:Password>Password</oas:Password>
    </oas:UsernameToken>
  </oas:Security>
</soap:Header>
<soap:Body>
  <csw:Transaction ... >
    ...
  </csw:Transaction>
</soap:Body>
</soap:Envelope>

```

Brugerstyring er ikke en del af CSW-standarden. Dvs. denne angivelse af brugernavn og password til en CSW-operation er speciel for geodata-info.dk, men bygger på OASIS Standard 200401, Web Services Security UsernameToken Profile 1.0.

Eksempel på en metode, der danner en forespørgsel, som sletter metadata records med titlen "geodatainfo test":

```

public string simpleTransactionRequest(string login, string passwd)
{
  //Dan tekststreng med Delete-request i SOAP-wrapping
  string str = "<soap:Envelope xmlns:soap=\"http://www.w3.org/2003/05/soap-envelope\" "
    + "xmlns:csw=\"http://www.opengis.net/cat/csw/2.0.2\" "
    + "xmlns:ows=\"http://www.opengis.net/ows\" "
    + "xmlns:oas=\"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd\">"
    + "<soap:Header>"
    + " <oas:Security>"
    + " <oas:UsernameToken>"
    + " <oas:Username>" + login + "</oas:Username>"
    + " <oas:Password>" + passwd + "</oas:Password>"
    + " </oas:UsernameToken>"
    + " </oas:Security>"
    + " </soap:Header>"
    + "<soap:Body>"
    + " <csw:Transaction "
    + " xmlns:csw=\"http://www.opengis.net/cat/csw/2.0.2\" "
    + " xmlns:ogc=\"http://www.opengis.net/ogc\" "
    + " service=\"CSW\" version=\"2.0.2\" >"
    + " <csw:Delete>"
    + " <csw:Constraint version=\"1.1.0\">"
    + " <ogc:Filter>"
    + " <ogc:PropertyIsEqualTo>"
    + " <ogc:PropertyName>title</ogc:PropertyName>"
    + " <ogc:Literal>geodatainfo test</ogc:Literal>"
    + " </ogc:PropertyIsEqualTo>"
    + " </ogc:Filter>"
    + " </csw:Constraint>"
    + " </csw:Delete>"
    + " </csw:Transaction>"
    + "</soap:Body>"
    + "</soap:Envelope>";

  return str;
}

```

For flere eksempler og en nærmere gennemgang af CSW-operationer henvises til brugervejledningen [1].

Java

Der findes forskellige Java-biblioteker, der kan benyttes til at sende HTTP POST. I nærværende eksempel benyttes biblioteket `java.net`, da det er et standardbibliotek. Man kunne fx også have benyttet Apache HttpClient, der er et populært java-bibliotek til kommunikation over HTTP-protokollen.

Som i det foregående eksempel indeholder klassen `ExampleSoapClient` metoden `httpSoapTest`, der sender en tekst til SOAP-tjenesten på `Geodata-info.dk`, og metoden `simpleSoapRequest`, der opbygger et `GetCapabilities-request` som en tekststreng.

```
package exampleapplication;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

public class ExampleSoapClient {

    String endPoint = "http://www.geodata-info.dk/csw-test";

    public String httpSoapTest(String soapXml) {
        try {
            //Dan forbindelse til tjenesten med HTTP POST
            URL url = new URL(endPoint);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setDoOutput(true);
            conn.setConnectTimeout(180000);
            conn.setRequestMethod("POST");
            conn.addRequestProperty("Content-Type",
                "application/soap+xml;charset=utf-8");

            //Send xml
            OutputStreamWriter wr = new OutputStreamWriter(conn.getOutputStream());
            wr.write(soapXml);
            wr.close();

            //Hent svar
            int httpStatus = conn.getResponseCode();
            if (httpStatus == HttpURLConnection.HTTP_OK) {
                BufferedReader rd =
                    new BufferedReader(new InputStreamReader(conn.getInputStream()));
                StringBuilder str = new StringBuilder();
                String line = rd.readLine();
                while (line != null) {
                    str.append(line);
                    str.append("\n");
                    line = rd.readLine();
                }
                return str.toString();
            } else {
```

```

        throw new Exception("HTTP error: " + httpStatus);
    }
}
catch (Exception e){
    System.out.println();
    System.out.println(e.toString());
}

return null;
}

public String simpleSoapRequest(){
    //Dan tekststreng med GetCapabilities-request i SOAP-wrapping
    String str = "<soap:Envelope xmlns:soap="
        + "\"http://www.w3.org/2003/05/soap-envelope\" "
        + "xmlns:csw=\"http://www.opengis.net/cat/csw/2.0.2\" "
        + "xmlns:ows=\"http://www.opengis.net/ows\">"
        + "<soap:Header/>"
        + "<soap:Body>"
        + " <csw:GetCapabilities service="
        + " \"http://www.opengis.net/cat/csw\">"
        + " <ows:AcceptVersions>"
        + " <ows:Version>2.0.2</ows:Version>"
        + " </ows:AcceptVersions>"
        + " </csw:GetCapabilities>"
        + "</soap:Body>"
        + "</soap:Envelope>";
    return str;
}
}

```

For at benytte eksemplet i fx Eclipse kan man oprette et java-projekt og en ny klasse, ExampleSoapClient, med kodeeksemplet. Man kan derefter benytte følgende program til at invocere eksemplet:

```

public class ExampleProgram {
    public static void main(String[] args) {
        ExampleSoapClient sc = new ExampleSoapClient();
        String soapResp = sc.httpSoapTest(sc.simpleSoapRequest());
        System.out.println();
        System.out.println(soapResp);
    }
}

```

Som i det foregående eksempel opbygges SOAP-beskeden direkte som en tekststreng, og svaret bliver blot skrevet til konsollen.

Transaktioner

Som beskrevet tidligere, kan man oprette, opdatere og slette sine metadata records med CSW Transaction-operationen, forudsat at man inkluderer brugernavn og password i SOAP-headeren.

Eksempel på en metode, der danner en forespørgsel, som sletter metadata records med titlen "geodatainfo test":

```

public String simpleTransactionRequest(String login, String passwd)

```

```

{
    //Dan tekststreng med Delete-request i SOAP-wrapping
    String str = "<soap:Envelope xmlns:soap="
        + "\"http://www.w3.org/2003/05/soap-envelope\" "
        + "xmlns:csw=\"http://www.opengis.net/cat/csw/2.0.2\" "
        + "xmlns:ows=\"http://www.opengis.net/ows\" "
        + "xmlns:oas=\"http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd\">"
        + "<soap:Header>"
        + " <oas:Security>"
        + "   <oas:UsernameToken>"
        + "     <oas:Username>" + login + "</oas:Username>"
        + "     <oas:Password>" + passwd + "</oas:Password>"
        + "   </oas:UsernameToken>"
        + " </oas:Security>"
        + " </soap:Header>"
        + "<soap:Body>"
        + " <csw:Transaction "
        + "   xmlns:csw=\"http://www.opengis.net/cat/csw/2.0.2\" "
        + "   xmlns:ogc=\"http://www.opengis.net/ogc\" "
        + "   service=\"CSW\" version=\"2.0.2\" >"
        + "     <csw:Delete>"
        + "       <csw:Constraint version=\"1.1.0\">"
        + "         <ogc:Filter>"
        + "           <ogc:PropertyIsEqualTo>"
        + "             <ogc:PropertyName>title</ogc:PropertyName>"
        + "             <ogc:Literal>geodatainfo test</ogc:Literal>"
        + "           </ogc:PropertyIsEqualTo>"
        + "         </ogc:Filter>"
        + "       </csw:Constraint>"
        + "     </csw:Delete>"
        + "   </csw:Transaction>"
        + "</soap:Body>"
        + "</soap:Envelope>";

    return str;
}

```

Som svar på en Transaction-operation får man en statusrapport:

```

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Body>
    <csw:TransactionResponse xmlns:csw="http://www.opengis.net/cat/csw/2.0.2">
      <csw:TransactionSummary>
        <csw:totalInserted>0</csw:totalInserted>
        <csw:totalUpdated>0</csw:totalUpdated>
        <csw:totalDeleted>0</csw:totalDeleted>
      </csw:TransactionSummary>
    </csw:TransactionResponse>
  </env:Body>
</env:Envelope>

```

Der henvises i øvrigt til brugervejledningen [1].

Testmiljø

Geodata-info.dk stiller et testsite tilgængeligt. Dette site omfatter både et servicesnit og en portaldel svarende til live-systemet. Man kan således redigere data ved at tilgå

tjenesterne og efterfølgende kontrollere data i portaldelen, eller man kan oprette data i portaldelen og efterfølgende fremsøge data med tjenesterne.

Testsitets web service interface findes på følgende adresse:

<http://www.geodata-info.dk/csw-test>

Testsitets portaldel findes på følgende adresse:

<http://test.geodata-info.dk>

For at benytte Transaction-operationer i testsitets servicesnit og få adgang til testsitets registrantdel, hvor man kan administrere metadata, skal man oprette sig som testbruger på følgende adresse:

<http://test.geodata-info.dk/nybruger>

Oprettelse af testbrugere foregår på samme måde som oprettelse af brugere i live-systemet, idet man sender en anmodning om oprettelse som testbruger og efterfølgende modtager mails med brugernavn og password, når anmodningen er godkendt.

Referencer

[1] Brugervejledning til web service (INSPIRE søgetjeneste. <http://www.geodata-info.dk/Portal/BrugervejledningWebservice.pdf>)

[2] OGC Catalogue Services Specification 2.0.2 - ISO Metadata Application Profile for CSW 2.0, version 1.0.0.

http://portal.opengeospatial.org/files/?artifact_id=21460