

Navision Stat 9.4

11. december 2019
ØSY/NSIR/
MILE, RASEG

Bruger dokumentation Webservice til generisk integration

Overblik

Introduktion

Dokumentet giver en beskrivelse af anvendelsen af den webservice der er tilknyttet GIS (Generisk Integration Snitflade) til Navision Stat.

Indhold

Bruger dokumentation.....	1
Webservice til generisk integration	1
Overblik.....	1
Introduktion.....	1
Målgruppe	1
Seneste ændring.....	1
Beskrivelse	2
Systemoverblik	2
WSDL dokumentation	3
Webservice beskrivelse.....	3
Webservice sikkerhed.....	4
Test af afsendelse til GIS Webservice.....	4
Klient stub'er	5
Microsoft .Net 3.5.....	5
Java.....	8

Målgruppe

Beskrivelsen er specielt rettet mod udviklere af klient programmer til webservices.

Seneste ændring

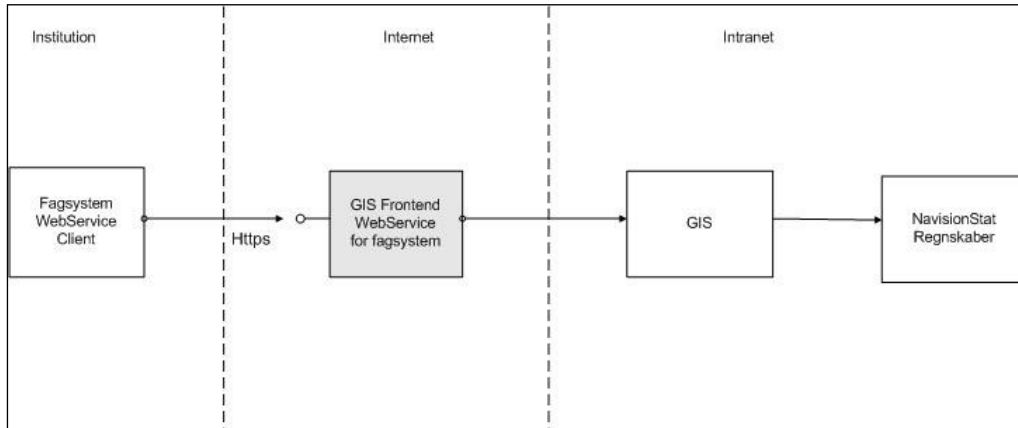
Publiceret første gang: 07.08.2009

Seneste ændring: 11.12.2019

Beskrivelse

Systemoverblik

Den generelle integration til Navision Stat kan tilgås vha. en offentlig webservice.



Et eksternt fagsystem kan anvende GIS ved at sende et forespørgende xml-dokument til GIS's offentligt tilgængelige webservice. Webservicen sender forespørgslen videre til GIS funktionaliteten i Navision Stat, der behandler forespørgslen og generer et svar som et nyt xml-dokument, der returneres tilbage via GIS webservicen til fagsystemet.

WSDL dokumentation

Webservicens interface er dokumenteret af WSDL standarden. I WSDL dokumentation indgår følgende filer¹:

- www.oes.dk.NavisionStatIntegrationService.2009.05.25.wsdl
- tempuri.org.wsdl
- www.oes.dk.NavisionStatIntegrationService.2009.05.25.xsd
- schemas.microsoft.com.2003.10.Serialization.xsd

Bemærk at endpoint-adressen i klient programmet skal tilrettes til den webservice som klienten skal tilgå. Ud over endpoint-adressen skal klient programmet bruge oplysninger om: Servicebrugernavn til servicen, Password til servicen og EAN nummeret til regnskabet.

Webservice beskrivelse

GIS webservicen er opbygget med én operation.

Service navn	NavisionStatIntegrationService
Operationer	InvokeNavisionStatIntegration
Input parametre	<ul style="list-style-type: none"> - companyIdentifier (string) -- EAN nr. til regnskab - immediateExecuting (string) -- J/N for straks udførelse - NSRequest (string) -- Input xml til GIS/NavisionStat
Output parameter	<ul style="list-style-type: none"> - NSResponse (string) -- Output xml fra GIS/NavisionStat

¹Ved opbygning af klienten kan man fx via Visual Studio hjælpeværktøjet svcutil.exe, anvende de fire WSDL filer som input. Programmet genererer 2 output filer:

- en C# klassefil (ofte kaldet en proxy fil), som skal indgå i klient programmet.
- fil der indeholder configurations definitioner, der skal anvendes af klient programmet, herunder et (forslag) til en endpoint adresse (URL til den webservice som klienten skal kalde).



Regnskabet EAN nummer (parameter "companyIdentifier") skal være registreret på GIS webservicen og anvendes til at dirigere forespørgslen videre til det korrekte Navision Stat regnskab. Hvis EAN nummer ikke er registreret, returneres en fejl. Registreringen foretages

Hvis parameteren "immediateExecuting" = J, betyder det at forespørgslen behandles inden svaret sendes retur til klienten.

Hvis parameteren "immediateExecuting" = N, betyder det at der umiddelbart returneres et "minimalt" svar til klienten med Status="Afventer". Forespørgslen bliver efterfølgende behandlet og svaret til klienten lægges "på lager". For at få fat i svaret skal klienten sende forespørgslen ind igen.

Hvis der opstår en fejl ifm. afviklingen af webservicen, returneres en SOAP fault.

Webservice sikkerhed

GIS webservicen er udført som en .NET WCF service med binding security mode = "TransportWithMessageCredential". Dvs. at servicen anvender HTTPS protokol på transport niveau til kryptering af indholdet af meddelelsen og server certifikat til identifikation overfor klient. Klientens identifikation (credentials) skal overføres på message niveau til servicen.

Test af afsendelse til GIS Webservice

Integrationsudvikleren må selv stå for at få opsat et testmiljø eller –regnskab til test af integrationen. Hvis databasen hostes hos SIT eller KMD kan man sende en forespørgsel ind via SAM service portal. I øvrige tilfælde må man selv tage kontakt til hoster.

Klient stub'er

Moderniseringsstyrelsen har udarbejdet to enkle klient stub'er (i java og .Net), som man kan benytte ved udviklingen af GIS webservice klienter. Stub'erne håndterer al kommunikation med GIS webservicen inkl. sikkerhed.

Microsoft .Net 3.5

Moderniseringsstyrelsen har udarbejdet en klientstub, som du kan benytte til webservice klienter udarbejdet i .Net. Stub'en håndterer kommunikationen med webservice inklusiv håndtering af sikkerhed. Stub'en er udarbejdet i .Net 3.5 SP1 og Windows Communication Foundation (WCF).

Stub'en kan downloades fra Moderniseringsstyrelsens hjemmeside og kan umiddelbart anvendes i et .Net program.

Her gengives indholdet af README.txt filen:

>>>>

.Net klientstub til ØS GIS-service

Vejledning i brug af denne stub:

1. Opret et nyt projekt i Visual Studio
2. I dit projekt skal du lave en reference til den udpakkede dll-fil (GISClientProxy.dll)
3. Tilføj den udpakkede applikationskonfigurationsfil (app.config) til dit projekt
4. Skriv dit klientprogram. Det kunne se således ud:

```
using System;

using System.Collections.Generic;

using System.Text;

namespace TestAfGISClientStub
{
    class Program
    {
        static void Main(string[] args)
        {
            string endpoint = "https://test3.oes.dk/NavisionStatIntegrationServiceTest/NavisionStatIntegrationService.svc";

            string username = "GISuser";

            string password = "Gis010203";

            string companyIdentifier = "5798009811400";

            string immediateExecution = "j";

            string nsRequest = "TestTestTest";

            string nsResponse = "";

            GISClientStub.IGISClient stub = GISClientStub.GISClientFactory.GetInstance(endpoint, username, password);

            nsResponse = stub.invokeNavisionStatIntegration(companyIdentifier, immediateExecution, nsRequest);

            System.Console.Out.WriteLine(nsResponse);

            System.Console.In.Read();
        }
    }
}
```

5. Kør klientprogrammet.

6. Du bør få nedenstående output, som viser at der er forbindelse til en GIS service.

```
<?xml version="1.0"?><NavisionErrorMsg xmlns="urn:www-Navision-com:NavisionError-  
rorMsg.xml">
```

```
<ErrorNo/><ErrorMessage>This message is for C/AL programmers:
```

```
This Automation variable has not been instantiated.
```

```
You can instantiate it by either creating or assigning it.
```

```
</ErrorMessage><ErrorSource></ErrorSource><BusId/></NavisionErrorMsg>
```

6. For yderligere oplysninger vedrørende valide forespørgsler til GIS henvises til dokumentet vedrørende dette.

Copyright Økonomistyrelsen 2009

<<<<

Hvis du selv vil udvikle din egen klientstub i .Net kan du søge informationen her:

<http://msdn.microsoft.com/en-us/library/ms733133.aspx>

Java

Moderniseringsstyrelsen har udarbejdet en klientstub, som du kan benytte til udarbejdelse af webservice klienter i java. Stub'en håndterer kommunikationen med webservicen inklusiv håndtering af sikkerhed. Stub'en er udarbejdet i Metro 1.5, som er Sun's referenceimplementation af jax-ws specifikationen. Stub'en benytter også XWSS 3.0, som er en open source implementering af WS-Security.

Stub'en kan downloades fra Moderniseringsstyrelsen hjemmeside og kan umiddelbart anvendes i et java program.

Stub'en er senest testet OK på jre1.8.0_231.

Nedenfor gengives README.txt filen:

```
>>>>
```

```
Java klientstub til ØS GIS-service
```

```
Vejledning i brug af denne stub:
```

1. Sørg for, at du benytter en JRE 1.6 eller JDK 1.6 - kan fx downloades fra <http://java.sun.com>.
2. Inkluder jar-filerne gis.jar og xws-security.jar i classpath.
3. Skriv et klientprogram, som fx kan se således ud:


```
import dk.oes.navisionsstatintegration.client.GisClient;

import dk.oes.navisionsstatintegration.client.GisClientFactory;

public class TestClient {

    public static void main(String[] args) {

        GisClient gisClient =
            GisClientFactory.getInstance("<brugernavn>", "<password>",
                "https://test3.oes.dk/NavisionStatIntegrationServiceTest/NavisionStatIntegrationService.svc");

        String response =
            gisClient.invokeNavisionStatIntegration("5798009811400", "j", "test");

        System.out.println(response);

    }

}
```

4. Kør klientprogrammet. Output burde være:

```
<?xml version="1.0"?>

<NavisionErrorMsg xmlns="urn:www-Navision-com:NavisionErrorMsg.xml">

<ErrorNo/><ErrorMessage>This message is for C/AL programmers:
```

This Automation variable has not been instantiated.

You can instantiate it by either creating or assigning it.

```
</ErrorMessage><ErrorSource></ErrorSource><BusId/></NavisionErrorMsg>
```

Se evt. API-dokumentationen i api-doc-biblioteket.

Copyright Økonomistyrelsen 2009

<<<<

Hvis du selv vil udvikle din egen klientstub i Java kan du finde softwaren her:

<https://metro.dev.java.net/1.5/>

<https://xwss.dev.java.net/>