Stamdataservice på NSP

Guide til udviklere

Indholdsfortegnelse

1 Formål 3

1.1 Metode og rapportens opbygning 3

1.2 Målgruppe og læsevejledning 3

2 System design 4

2.1 Seal 4

2.2 Properties 4

3 Opsætning af udviklingsmiljø 5

3.1 Kildekode 5

3.2 Byggemiljø 5

3.3 Autogenereret kode 6

3.4 Database setup 7

3.5 Test 7

3.6 IDE 7

3.6.1 Eclipse 7

3.7 IntelliJ Idea IDE 8

3.8 Distribution 9

4 Sådan køres performancetest for CPR-services 10

4.1 Query-parametre 13

4.2 Forudsætninger 14

4.3 Udførsel 16

5 Ændringslog 17

6 Referencer og kilder 19

# Formål

Nærværende dokument er en guide til nye udviklere af stamdataservicen på NSP. Guiden gennemgår på overordnet plan de aktiviteter, der er nødvendige for at kunne videreudvikle stamdataservicen.

## Metode og rapportens opbygning

Efter nærværende introduktion vil dokumentet gennemgå de væsentligste dele af opsætningen af et lokalt udviklingsmiljø og afvikling af test/performancetest.

Dokumentet forudsætter, at læseren har grundig kendskab til Java udvikling, webservices og Maven. Kendskab til JBoss applikationsserver vil yderligere hjælpe læseren, men er ikke en forudsætning.

I dokumentet benyttes følgende notationer:

Markering af **scripts** og **kommandoer**.

Markering af **advarsler**

Markering af **referencer til filer**.

## Målgruppe og læsevejledning

Den primære målgruppe for dokumentet er systemudviklere.

# System design

Stamdataservicen består af 2 hovedmoduler (DoDi og NSP), der hver er opdelt i en række komponenter. Der henvises til [DESIGN] for yderligere information vedrørende DoDi og NSP.

**Komponenter på DoDi:**

DoDi applikationerne er en række selvstændigt kørende applikationer, til overvågning af indkomne XML filer. Indkomne XML filer indlæses, valideres og gemmes herefter i DoDi’ens database.

**NSP:**

Hver NSP komponent er designet som en Servlet 2.4 web-applikation og benytter Guice til dependency injection.

Servlet paths, filters, osv. konfigureres direkte i koden og ikke i web.xml.

Hvert modul indeholder en *ApplicationContextListener.java* fil der fungerer som entry-point til applikationen. Det anbefales, at man som ny udvikler på projektet kigger koden igennem med denne fil som udgangspunkt.

## Seal

Stamdataservicen benytter Seal.java til håndtering af forespørgsler og svar i webservice snitfladen.

Seal.java bygger i nuværende version (2.1.x) på commons-logging, hvilket konflikter med JBoss AS6. I pom.xml filerne er commons-logging derfor fjernet, og erstattet med en slf4j commons-logging proxy.

Derudover er XercesImpl også fjernet, da den på tilsvarende måde konflikter med JBoss AS6[[1]](#footnote-1).

## Properties

NSP komponenterne benytter Guice til konfigurationsstyring. Konfiguration styres via filen *config.properties*, der pakkes sammen med WAR-filen.

*Stamdata* importerne styres af java properties filer. Hver stamdata importer har en default konfigurations fil (default-config.properties) som er deployet sammen med war filen, denne kan overstyres med en properties fil lagt i jboss uden for war filen (config.properties)

# Opsætning af udviklingsmiljø

Opsætningen af udviklingsmiljøet for stamdataservicen forudsætter, at følgende elementer allerede er installeret på udviklerens maskine:

* Java Developer Kit 6.0\_x
* Et passende udviklingsmiljø
* Maven 3.x
* Virtualbox version 4.1.18 (til DoDi-komponenterne)
* Vagrant version 1.0.3 (til DoDi-komponenterne)
* MySQL database 5.1.x (til NSP-komponenterne)
* JBoss AS6 (til NSP-komponenterne)

## Kildekode

Kildekoden er placeret i to forskellige github-repositories:

### NSP

Kildekoden er placeret her:

<https://github.com/trifork/sdm/>

Koden checkes ud på følgende måde:

% git clone https://github.com/trifork/sdm/

### DoDi

Kildekoden er placeret her:

<https://github.com/trifork/sdm2/>

Koden checkes ud på følgende måde:

% git clone https://github.com/trifork/sdm2/

## Byggemiljø

Stamdataprojektet anvender Maven som byggesystem [MAVEN]. Strukturen følger de generelle anbefalinger for Maven projekter, og er struktureret med en parent pom.xml og en projekt pom.xml fil for hvert underprojekt.

Subprojekterne er opbygget efter Maven layout konventionen.

## Database setup

### NSP

Tester man med en ”NSP in a box” (NIAB) skal mysql databasen som udgangspunkt konfigureres i på NIAB hosten. (Se [NIAB] for konfiguration af NIAB)

Opret en bruger i mysql databasen med bruger navn ”sdm” og password ”papkasse”:

% mysql -uroot -p

Enter password: <root password til mysql>[[2]](#footnote-2)

mysql> CREATE USER 'sdm'@'localhost' IDENTIFIED BY 'papkasse';

Stamdataservicens database og tabeller oprettes via:

db/schema.sql

*Schema.sql* scriptet eksekveres på følgende måde:

% mysqladmin -u root create sdm\_warehouse -p

% mysql -u root -p

Enter password: <root password til mysql>

mysql> grant all privileges on sdm\_warehouse.\* to sdm@localhost identified by 'papkasse' with grant option;

mysql> exit

% mysql -u sdm sdm\_warehouse < db/schema.sql -p

Enter password: papkasse

Ovenstående forudsætter, at man eksekverer dette, mens man står et sted hvor schema.sql ligger i /db/schema.sql.

Det anbefales, at den nuværende navngivning af databasen bibeholdes. Ønsker man at etablere en database med et alternativt navn, skal dette tilrettes i modulernes konfigurationsfiler.

Bemærk, at ændringer i konfigurationsfilerne har systemmæssige konsekvenser, og derfor bør kun velovervejede ændringer committes.

### DoDi

DoDi-projekterne anvender Vagrant+Puppet til at opsætte udviklingsmiljøet.

For at få bootet og opsat en udviklingsmaskine, køres følgende

% vagrant up; vagrant provision

Vagrant provision burde ikke være nødvendigt, men er det pt. på grund af en timing-fejl med jboss.

Når vagrant har bygget den virtuelle maskine, er den sat op med den korrekte version af JBoss 7 samt en MySQL der er konfigureret til at have en root-bruger med password papkasse og en sdm4-bruger med password sdm4.

Der er også automatisk oprettet en tom database med standard-navnet for databasen, (sdm\_warehouse) og en datasource i JBoss med det JNDI-navn, som applikationen som default forventer at kunne slå datasource op.

Applikationen kører ved opstart automatisk databaseskema på databasen ved hjælp af frameworket Flyway.

Der er forwardet følgende porte ind i den virtuelle maskine, så man kan tilgå jboss og mysql fra sin lokale maskine

8080 -> 8080 (dvs. at man kan tilgå <http://localhost:8080/>)

3307 -> 3306 (dvs. at man kan have en urelateret mysql kørende lokalt og tilgå SDM-mysql’en på den virtuelle maskine på url localhost:3307)

## Autogenereret kode

Start med at køre følgende:

% mvn clean install

Stamdataprojektet benytter JAX-WS i webservice snitfladerne i CPR-WS projektet.

Snitfladekoden er autogeneret og skal ved opdatering af de associererede WSDL filer opdateres med kommandoen:

% mvn generate-sources

Som bagvedliggende implementering af JAX-WS benyttes Oracle’s reference implementering. Denne kan konfigureres ved at ændre i filen

<CPR-modul>/…/WEB-INF/sun-jaxws.properties

## Test

Installationen kan verificeres ved at eksekvere stamdataservicens test suite.

Stamdataservicen benytter JUnit og Mockito til test.

Testkoden er for hvert modul lokaliseret i:

src/test/java

Test suiten afvikles ved at udføre følgende kommando i projektroden:

% mvn test

Kommandoen kan også udføres under de individuelle moduler, hvorved kun undermodulets test udføres.

Installationen kan yderligere verificeres ved at udføre kommandoen:

% mvn verify

Denne kommando validerer code coverage og kode konventionerne for projektet.

Kode konventionerne følger reglerne defineret i filen:

config/checkstyle.xml

## IDE

Stamdataservicen kan principielt udvikles i enhver Java IDE, der forstår Maven projekters opbygning.

I dette dokument beskrives kort opsætning for to af de pt. mest udbredte Java IDE’er: Eclipse og IntelliJ.

### Eclipse

Eclipse er ikke født med Maven support, og det anbefales derfor, at man installerer m2eclipse inden stamdataservicen hentes ind i Eclipse:

http://www.eclipse.org/m2e/

Herefter importeres projekterne i Eclipse via ”import”:

Alternativt kan man importere projektet ved at udføre følgende kommando:

% mvn eclipse:eclipse

Og herefter importere projektet på normal vis i Eclipse.

Kommandoen genererer Eclipse projektfilerne (.project og .classpath) for roden og hvert undermodul. Denne metode kræver dog, at kommandoen udføres hver gang man ændrer i pom filerne.

## IntelliJ Idea IDE

IntelliJ Idea er født med Maven support, og stamdataservicen kan derfor direkte importeres. Projektet importeres i IntelliJ ved under ”Create new project” at vælge ”Import project from external model”. Herefter udvælges roden af stamdataservicen, hvorefter projektet importeres.

Det anbefales i den sammenhæng, at man krydser af i ”Import Maven projects automatically”, hvorefter IntelliJ selv detekterer nye moduler i projektet.

Alternativt kan man importere projektet ved at udføre følgende kommando:

% mvn idea:idea

Herefter kan projektet importeres på normal vis i IntelliJ.

*Obs! Denne metode kræver dog, at kommandoen udføres hver gang man ændrer i pom filerne.*

## Distribution

Stamdataservicen kan bygges til distribution eller lokal test ved at udføre:

% mvn package

Dette generer en række WAR filer, der efterfølgende kan deployeres lokalt eller i produktion.

Kodeord til serverne og databaserne skal indhentes hos NSP-operatøren.

# Performance test af autorisationsservicen og kopi-register-servicen

Dette afsnit beskriver hvordan man kører performance af autorisationsservicen og kopi-register-servicen.

## Krav

For at køre performance test på Stamdata Projektet er det nødvendigt at have:

|  |  |
| --- | --- |
| **Krav** | **Installation og test** |
| Stamdatas kildekode er checket ud. | %git clone https://github.com/trifork/sdm/ git clone https://github.com/trifork/sdm/ |
| Maven3 installereret | Se maven.org for installation  % mvn –v (i prompt) |
| Adgang til internettet | - |
| Adgang til Test STS’en (dvs. være på et netværk hvor den er tilgængelig.) | Check at  <http://niab01.nsp-test.netic.dk:8080/sts/services/SecurityTokenService?wsdl> viser wsdl i browser. Udskift evt. niab01.nsp-test.netic.dk:8080 men den Test STS host og port du anvender. |
| NSP in a box (NIAB) kørende via vmware’s vmplayer | Følg vejledningen i [niab] og gennemfør smoketesten for at checke at den fungerer. |

### Deployment

Fra Netic kan man downloade et vmware image der er opsat til at kunne afvikle NSP. Dette kaldes ”NSP in a box” (NIAB). Dette image kan rekvireres ved henvendelse hos operatøren.

### Konfiguration

Inden man kører test skal man konfigurere stamdata komponenterne til at køre i development mode. Fra kildekoden skal

/nsp/authorization-lookup-ws/src/test/resources/test-config.properties

omdøbes til

stamdata-authorization-lookup-ws.properties

og lægges i

$JBOSS\_HOME/server/default/conf/

På samme måde skal

/nsp/batch-copy-ws/src/test/resources/test-config.properties

omdøbes til

stamdata-batch-copy-ws.properties

og lægges i

$JBOSS\_HOME/server/default/conf/

Check at følgende property:

security=dgwsTest

er sat i

stamdata-batch-copy-ws.properties

Dette gør, at komponenterne godkender requests med ID Kort underskrevet af Test STS’en.

Når filerne er ændret skal JBoss genstartes:

% sudo /etc/init.d/jboss restart

### Konfiguration af database

Det CVR-nummer der bliver brugt til tests skal være oprettet i stamdatatabellerne ”Client” og ”Client\_permissions”. Dette gøres på følgende måde:

1. I tabellen ”Client” oprettes et element med:
   1. ”name” sat til ”Region Syd” og
   2. ”subjectSerialNumber ” sat til ”CVR:19343634-UID:1234”
2. Aflæs det generede id og opret i tabellen ”Client\_permission” en indgang med
   1. ”client\_id” sat til det genererede i tabellen ”Client”
   2. ”permission” sat til ”cpr/person/v1”.

### Kørsel

Performance tests køres ved at køre følgende kommandoer fra folderen ”common/performance” i stamdata’s kildekode:

% mvn -Pperformancetest integration-test site -Dhostname=<host> -Dport=<port>

Hvor <host> er adressen på nsp-serveren, hvor Stamdataservice er deployet. Hvis du kører mod NIAB så anvend dens IP-adresse som <host> og anvend 8080 som <port>.

## Testrapporter

HTML-rapporter bliver lagt i

common/performance/target/site/

når alle tests er færdige.

Her ligger performancerapporter fra kørsel mod autorisationsservicen (authorization-ws-test-report.html) og kopi-register-servicen (replication-ws-report.html).

# Performancetest af CPR-services

Test køres med current dir i

nsp/cpr-ws

For StamdataPersonLookup testes den case, hvor request indeholder fornavn+efternavn, og der søges efter personer der matcher dette. Denne case er udvalgt, fordi den vurderes at belaste serveren mest (i forhold til fx casen med opslag ud fra cpr-numre).

## Krav

For at køre performance test på Stamdata Projektet er det nødvendigt at have:

|  |  |
| --- | --- |
| **Krav** | **Installation og test** |
| Stamdatas kildekode er checket ud. | %git clone https://github.com/trifork/sdm/ git clone https://github.com/trifork/sdm/ |
| Maven3 installereret | Se maven.org for installation  % mvn –v (i prompt) |
| Adgang til internettet | - |
| Adgang til Test STS’en (dvs. være på et netværk hvor den er tilgængelig.) | Check at  <http://niab01.nsp-test.netic.dk:8080/sts/services/SecurityTokenService?wsdl> viser wsdl i browser. Udskift evt. niab01.nsp-test.netic.dk:8080 men den Test STS host og port du anvender. |
| NSP in a box (NIAB) kørende via vmware’s vmplayer | Følg vejledningen i [niab] og gennemfør smoketesten for at checke at den fungerer. |
| CPRABBS servicen deployeret (på NIAB) | I dette dokument er der en miniguide hertil men det anbefales at konsultere dokumenterne  [BRS-guide til anvendere] og [BRS-driftvejledning] |

### Konfiguration

Inden man kører test, skal man konfigurere stamdata komponenterne til at køre i development mode. Fra kildekoden skal

/nsp/cpr-ws/src/test/resources/test-config.properties

omdøbes til

stamdata-cpr-ws.properties

og lægges i (anvend sudo)

$JBOSS\_HOME/server/default/conf/

## Performance testdata

Filen

nsp/cpr-ws/src/test/resources/getPersonDetailsQueryParameters.csv

indeholder komma-separerede par af fornavn, efternavn der benyttes til kaldene til StamdataPersonLookup-servicen.

Der skal rettes i denne fil, så der angives mere end to fornavn, efternavn-par.

Alle fornavn, efternavn-par skal vælges så de hver giver mindst ét hit i den database der køres med. Fornavn, efternavn par som ikke giver et hit markeres som fejlede af JMeter-sampleren.

På samme måde indeholder filen

nsp/cpr-ws/src/test/resources/ getSubscribedPersonDetailsQueryParameters.csv

et cvr-nummer pr. linie. Disse cvr-numre benyttes til kaldene til StamdataPersonLookupWithSubscription-servicen.

Der skal rettes i denne fil, så der angives det ønskede antal cvr-numre.

De cvr der angives, behøver ikke nødvendigvis alle give hits (men nogle af dem skal). Sampleren markerer kun requests hvor samtlige person-opslag lykkedes.

Cvr-numre giver hits, første gang der kaldes for dem, givet at det angivne cpr-nummer i Cpr subscription-servicen er angivet til at have abonnement på mindst ét cvr-nummer, der findes i databasen.

## Forudsætninger

### CPR-WS Konfiguration

Inden man kører test skal man konfigurere stamdata komponenterne til at køre i development mode. Fra kildekoden skal

/nsp/cpr-ws/src/test/resources/test-config.properties

omdøbes til

stamdata-cpr-ws.properties

og lægges i (anvend sudo)

$JBOSS\_HOME/server/default/conf/

### CPR-WS database opsætning

For at kunne køre testen igennem med de leverede input-parametre, skal dette script indlæses i Stamdatas database:

src/test/resources/data\_needed\_for\_performancetests.sql

f.eks. ved at køre (på NIAB hvis det er det man har valgt)

% mysql -usdm -p < nsp/cpr-ws/src/test/resources/data\_needed\_for\_performancetests.sql

Password: papkasse

indeholder eksempeldata for henholdsvis cpr-abonnementer og ændrede cpr-numre til cprabbs-servicen.

For at give et realistisk dataload, skal der indsættes en væsentligt større mængde data i begge tabeller. Disse data skal matche data i Stamdatas person-tabel.

Både Stamdata og cprabbs-servicen skal være konfigureret til at acceptere Id-kort udstedt af SOSI test-STS’en. Se vejledning til anvendere [BRS-guide til anvendere] og [BRS-driftvejledning] for detaljer og afsnit 5.4 for en miniguide.

## CPRABBS-Service miniguide

Der skal være adgang til en kørende cprabbs-service. Dette kan f.eks. opnås ved at deploye cpr-abbs war filen i på NIAB fra seneste BRS release. Denne cpr-abbs war fil ligger i et BRS release her:

<versions-nr>/cprabbs-war-<versions-nr>.war

### CPRABBS-service konfiguration

Fra kildekoden skal

/brs/common/src/main/resources/brs.local-deploy.properties

omdøbes til :

brs.properties

og lægges i (anvend sudo):

$JBOSS\_HOME/server/default/conf/

I Stamdatas konfiguration skal property

cprabbs.service.endpoint.host

cprabbs.service.endpoint.port

cprabbs.service.endpoint.path

sættes til cprabbs-servicens endpoint, fx værdierne

cprabbs.service.endpoint.host=localhost

cprabbs.service.endpoint.port=8080

cprabbs.service.endpoint.path=/cprabbs/service/cprabbs

### CPRABBS-service databaseopsætning

Opret og tilpas CPRABBS notifikationsdatabasen:

$ mysql -f -uroot -p < init-mysql-nsp\_reg\_notification.sql

(ignorér fejl)

$ mysql --database=nsp\_reg\_noti -uroot -p < create-cprsubscription-tables.sql

$ mysql --database=nsp\_reg\_noti -uroot -p < mysql-cprsubscription-alter-tables.sql

Opret og tilpas testdatabase over personer med CPR-nummer opdateringer i stamdata databasen:

$ mysql -f -uroot -p < init-mysql-nsp\_stamdata.sql

$ mysql --database=nsp\_stamdata -uroot -p < create-test-stamdata-tables.sql

$ mysql --database=nsp\_stamdata -uroot -p < mysql-test-stamdata-alter-tables.sql

Fyld data i CPRABBS notifikationsdatabasen:

$ mysql --database=nsp\_stamdata -uroot -p < src/test/resources/cprabbs\_data\_needed\_for\_performancetests\_nsp\_stamdata.sql

Opret og tilpas testdatabase over personer med CPR-nummer opdateringer i stamdata databasen:

$ mysql --database=nsp\_reg\_noti -uroot -p < src/test/resources/cprabbs\_data\_needed\_for\_performancetests\_nsp\_reg\_noti.sql

## Udførsel

Man udfører testen ved at køre følgende (som én linie, uden liniebrud):

rm -rf target/chronos/&& mvn integration-test site -Pperformancetest -DskipTests -DnumberOfIterations=5 -DnumberOfThreads=10 -Dhostname=localhost -Dport=8080

(rm-delen sørger for, at Chronos-pluginn’et kører testen igen, selvom der ligger et resultat fra tidligere).

Værdierne af de 4 properties til sidst kan rettes. De betyder følgende:

|  |  |
| --- | --- |
| **Property** | **Beskrivelse** |
| numberOfIterations | hvor mange loops skal hver tråd køre |
| numberOfThreads | hvor mange tråde skal der startes |
| Hostname | på hvilken host kører StamdataPersonLookup-servicen |
| Port | på hvilken port kører StamdataPersonLookup-servicen |

## Testrapporter

Når kommandoen har kørt, er testrapporter tilgængelige i

target/site

Denne html fil:

StamdataPersonLookupGetPersonDetails-report.html

Viser resultatet af den test, der forespørger på personer via fornavn, efternavn par. Resultater af den performancetest, der er abonnements (altså afhænger cpprabs-servicen) forefindes i

StamdataPersonLookupWithSubscriptionGetSubscribedPersonDetails-report.html

# Tips og tricks

I de følgende beskrives problemer og deres løsninger:

## JBoss out of memory

### Beskrivelse

I JBoss’s boot.log:

$JBOSS\_HOME/server/default/log/boot.log

Logger JBoss noget i stil med ”out of memory” og nævner “permgenspace”

### Løsning

Forøg JBoss permgen space ved at ændre linien indeholdende:

JAVA\_OPTS="-Xms2048m -Xmx2048m -XX:MaxPermSize=**256**m -Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"

i filen

$JBOSS\_HOME/bin/run.conf

til

JAVA\_OPTS="-Xms2048m -Xmx2048m -XX:MaxPermSize=**512**m -Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"

## JBoss kan ikke skrive til “trancsaction.log”

### Beskrivelse

I JBoss’s boot.log:

$JBOSS\_HOME/server/default/log/boot.log

Logger JBoss noget i stil med ”transaction.log” og ”cannot write”

### Løsning

Ignorér denne fejl.

# Ændringslog

Nyeste udgave af dette dokument kan erhverves ved henvendelse til NSP-operatøren.

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Dato** | **Ændring** | **Ansvarlig** |
| 1.0 | 28/4-2011 | Initielt Dokument | Trifork |
| 1.1 | 6/10-2011 | Opdateret med CPR tjenester | Trifork |
| 1.2 | 8/12-2011 | Kvalitetssikret af Lakeside | Lakeside |
| 1.3 | 22/12-2011 | Opdateret bla. med performance test af autorisationsservicen og kopi-register-servicen | Trifork |
| 1.4 | 20/8-2012 | Tilføjet DoDi-specifikke afsnit der beskriver SDM4 | Trifork  jrf@trifork.com |

# Referencer og kilder

|  |  |  |
| --- | --- | --- |
| Reference-id | Indhold / Overskrift | Henvisning |
| [MAVEN] | Welcome to Apache Maven | <http://maven.apache.org/> |
| [NIAB] | NSP in a box | Kan rekvireres ved henvendelse til operatøren inklusiv vejledning i anvendelse og konfiguration. |
| [BRS-guide til anvendere] | Guide til anvendere | Ligger i doc bibliotek i en BRS release |
| [BRS-driftvejledning] | Driftvejledning | Ligger i doc bibliotek i en BRS release |

1. Fra Seal.java 2.1.2 er XercesImpl ikke længere inkluderet i Seal.java. [↑](#footnote-ref-1)
2. Root password til mysql sættes med følgende kommando:

   % mysqladmin -u root password NEWPASSWORD [↑](#footnote-ref-2)