

SOSI-Gateway

Guide til udviklere

Indhold

Indledning.....	3
Forudsætninger	3
Afhængigheder	3
SOSI-GW	3
Projektstruktur	3
Releases	4
Tests.....	4

Version	Dato	Beskrivelse	Forfatter
1	21-06-2016	Første version, udarbejdet med udgangspunkt i Udviklingsvejledning for SOSIGW 1.0	OBJ

Indledning

Dette dokument beskriver hvordan man kommer i gang med at tilpasse eller videreudvikle SOSI-GW ud fra kildeteksten.

Forudsætninger

Der forudsættes erfaring med java-udvikling og brug af open source projekter. Læs også installationsvejledningen, der godt nok er rettet mod drift og test, men hvis beskrivelse af teknisk setup ikke gentages her.

Afhængigheder

SOSI-GW afhænger af en lang række af andre open source projekter. Til at styre disse afhængigheder benyttes "Ivy", som er et Ant-plugin (se <http://ant.apache.org/ivy> for mere information). Konfiguration af Ivy og basal build-struktur hentes fra projektet "trifork-common", som er inkluderet i SOSI-GW-projektet. Det forventes at der findes Java ($\geq 1.5.0$) og Ant ($\geq 1.7.0$) i omgivelsen.

SOSI-GW

SOSI-GW projektet ligger på nspop svn-repository på følgende adresse:

<https://svn.nspop.dk/svn/trifork/sosi-gw/trunk/>

Miljøvariablen "JAVA_HOME" skal være sat og pege på det JDK, der svarer til den Java, der er den aktive. Det er ikke tilstrækkeligt med et JRE. (Eksempel: export JAVA_HOME=/usr/lib/jvm/java-6-sun)

Udfør "ant" i projektet. Dette henter en hel række open source projekter fra nettet og bygger projektet. Herefter kan Eclipse åbne projektet, med alle dependencies på deres rette plads. Bemærk, at hvis projektet åbnes i Eclipse før Ant er kørt, vil der være fejl i projektet. Disse fejl forsvinder efter Ant er kørt.

Projektstruktur

SOSIGW-projektet er et Eclipse-projekt, og har følgende struktur:

src/	Indeholder kildekode samt statisk konfiguration og andre ressource-filer
test/	Indeholder unit- og integrationstests
performance/	Indeholder JMeter performancetests
doc/	Indeholder al dokumentation på nær Javadoc, som kan genereres via ant javadoc.
etc/	Indeholder filer, der anvendes i forbindelse med byggeprocessen, f.eks. XML-skemaer-WSDL-filer mv.
launch/	Indeholder launchere, der kan anvendes i forbindelse med udvikling af GWT i Hosted Mode
lib/	Indeholder de jarfiler, projektet benytter sig af. Bemærk at dette bibliotek laves automatisk af Ant, og det skal ikke ligge i Subversion

target/	Indeholder alt der genereres i forbindelse med byggeprocessen, dvs kompilerede filer, genererede source-filer og andre artefakter. Dette dir skal heller ikke ligge i Subversion.
trifork-common/	Indeholder et basalt byggemiljø baseret på Ant og Ivy.
vendor/	Indeholder dokumentation på nogle af de 3.parts-jarfiler der anvendes.
web/	Indeholder de filer, der skal pakkes med i den færdige war-fil på nær class- og jar-filer
ixy.xml	Beskriver projektets afhængigheder.
ChangeLog	Beskriver hvilke ændringer, der er sket i de forskellige versioner.

Releases

Et nyt release laves ved først at opdatere ChangeLog-filen med ændringerne fra sidste release. Herefter checkes det at projektet kan bygges og at alle tests kører uden fejl. Herefter køres release.sh med det nye versionsnummer som argument, f.eks. ./release.sh 1.0.5. Hvis der releases til nsp køres scriptet nsp-release.sh, ligeledes med versionsnummer.

Udvikles der under Windows kan release foretages manuelt med følgende trin:

- Lav en svn update for at sikre, at den seneste version er hentet.
- Lav et nyt tag i SVN baseret på trunk, der hedder vx.y.z hvor x.y.z er det nye versionsnummer.
- Kør ant clean
- Pak projektet til en tar.gz-fil, der hedder sosigw-x.y.x.tar.gz
- Placer denne fil i svn under releases
- Kør ant for at bygge en war-fil
- Omdøb war-filen (som ligger under target) til sosigw-x.y.z.war
- Placer war-filen sammen med tar.gz-filen i svn

Tests

Unit- og integrationstests er placeret under test/, og følger den samme pakkestruktur som de klasser der testes. Der anvendes normal Junit 3 til at implementere tests.

Tests kan køres på to måder: Enten via Ant (ant test) eller via Eclipse. I Eclipse højreklikkes på test og under Run As vælges Unit Test.

Så vidt muligt navngives testmetoder efter deres hensigt – dvs. f.eks. testConditionShouldBeTrue() i stedet for test1().

Når tests køres via Ant laves der desuden coverage måling via Clover. Coverage-rapporter kan genereres med Ant coverage-report, som derefter kan findes under target/coverage.