



SOSIGW

- Udviklingsvejledning for SOSIGW 1.0

Indeks

Indeks	1
Revisionshistorik	2
Indledning	2
Forudsætninger	2
Afhængigheder	2
SOSI-GW	2
Projektstruktur	3
Releases	3
Tests	4

Version	Dato	Ansvarlig	Kommentarer
1	09-10-2009	jre	Initiel version
2	14-09-2011	tgk	Tilføjet script til nsp-release

Revisionshistorik

Version	Dato	Ændring	Ansvarlig
1	09/10/2009	Initiel version af dokument	jre
2	14/09/2011	Tilføjet script til nsp-release	tgk

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 softwarebørsens svn-repository på følgende adresse: <https://svn.softwareborsen.dk/sosi-gw/>

Environmentvariablen "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.

På T4 er det under udvikling behageligt at benytte "Inplace-deployment", som betyder at T4 udfører det kode, ens editor kompilerer for en, uden at skulle redeploye eller genstarte serveren. Dette gøres ved at aktivere "autorestart" på den systemcontainer, der deployes til, og deploye "sosig.map" med "-inplace" fremfor at deploye "target/sosigw.war".

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. De primære filer er odt-filerne, pdf-filerne kan potentielt være uddaterede.
- etc/ indeholder filer, der anvendes i forbindelse med byggeprocessen, fx XML-skemaer- WSDL-filer mv.
- launch/ indeholder launchere, der kan anvendes i forbindelse med udvikling af GWT i Hosted Mode
- lib/ indeholder de jar-filer, 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-jar-filer 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, fx ./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 fx `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`.