

Wissensnetzwerke im Grid

Single Sign-On Server

Arbeitspaket 1

Februar 2012

Milad Jason Daivandy



Inhaltsverzeichnis

1	Einleitung	2
2	Installation	3
3	Konfiguration	4
3.1	Initialisierungskonfiguration der Java Webapplikation - web.xml	4
3.2	Applikationsspezifische Einstellungen - config.xml	4
3.3	Logging-Einstellungen - log4j.properties	7
4	Betrieb	8
5	Schnittstellen	8
5.1	Web browser	8
5.2	Web service	9
6	Schnellstart	9

1 Einleitung

Der Single Sign-On Server fungiert als zentrale Sicherheitskomponente in WisNetGrid und zeichnet verantwortlich für

- WisNetGrid-Nutzerkonten
 - Registrierung
 - Verwaltung durch den jeweiligen Nutzer
- Zentrale Authentisierung von WisNetGrid-Nutzern
 - nach dem Single Sign-On - Konzept
 - ein Login- bzw. Logoutvorgang bei einem WisNetGrid-Dienst meldet einen WisNetGrid-Nutzer automatisch an allen weiteren WisNetGrid-Diensten an bzw. ab
- Zentrale Authorisation von WisNetGrid-Nutzern

-
- Trust Delegation
 - Ermöglicht einem WisNetGrid-Nutzer, WisNetGrid-Dienste in seinem Auftrag handeln zu lassen
 - Kernaspekt des Ressourcenföderationskonzepts von WisNetGrid (siehe [4])
 - Administration von WisNetGrid-Nutzerkonten
 - Auflistung aller WisNetGrid-Nutzer
 - Rollenverwaltung aller WisNetGrid-Nutzer
 - Schnittstellen (siehe Unterabschnitt 5)
 - Web browser
 - RESTful Web service API

Jeder WisNetGrid-Dienst [1] delegiert dabei mittels entsprechender Client-Softwarebibliotheken (siehe Abschnitt 5) sicherheits-bezogene Aufgaben an den Single Sign-On Server und muss somit keine eigene IT-Sicherheitsschicht bereitstellen.

2 Installation

Der Single Sign-On Server wird in einer ZIP-Datei geliefert, die abhängig von der Softwareversion *sso-server-\$version.zip* benannt ist.

Diese Datei ist in ein Verzeichnis zu entpacken, das im weiteren Verlauf dieser Dokumentation als *\$Installationsverzeichnis* referenziert wird und folgende Unterverzeichnisse enthält:

- bin
 - Executables für Unix/Linux
 - Windows
- conf
 - Initialisierungskonfiguration
 - Applikationsspezifische Konfiguration
- doc
 - Dieses Dokument
- lib
 - Java-Bibliotheken
- logs
 - Logs

3 Konfiguration

Nach der Installation liegen zwei Konfigurationsdateien vor:

- *\$Installationsverzeichnis/conf/web.xml*
 - dient der Initialisierung der Java Webapplikation
 - referenziert config.xml
- *\$Installationsverzeichnis/conf/config.xml*
 - regelt applikationsspezifische Einstellungen

3.1 Initialisierungskonfiguration der Java Webapplikation - web.xml

Diese Datei erfordert nach der Installation des Single Sign-On Server (siehe Abschnitt 2) keinerlei Änderungen. Dennoch zeigt die nachfolgende Tabelle etwaige Anpassungsmöglichkeiten auf.

Name	Standardwert	Beschreibung
display-name	WisNetGrid Single Sign-On Server	Name der Webapplikation (bestimmt den Titel im Web browser)
context-param/param-value-class	file:conf/config.xml	Pfad zur applikationsspezifischen Konfiguration
filter/filter-name	sso-server	Pfad der Webapplikation, z.B. https://meine-domain.com:9999/sso-server, muss mit dem nachfolgenden Parameter übereinstimmen!
filter-mapping/filter-name	sso-server	Filter - Zugriff, muss mit dem vorhergehenden Parameter übereinstimmen!

3.2 Applikationsspezifische Einstellungen - config.xml

3.2.1 Beispielkonfiguration: Server

Die folgende Tabelle zeigt die Servereinstellungen auf, zusammengefasst innerhalb des folgenden XML-Elements:

```
<bean id="configuration" class="wisnetgrid.security.sso.Configuration"
    factory-method="get">

    <!-- Server - Einstellungen -->
    <property name="host" value="127.0.0.1" />
```

```
<property name="port" value="9999" />
<property name="restPort" value="8183" />

<!-- SSL - Einstellungen -->
<property name="ssl" value="true" />
<property name="keyStore" value="src/main/resources/conf/
    keystore_sso-server.jks" />
<property name="keyStorePass" value="sso-server" />
<property name="trustStore" value="src/main/resources/conf/
    keystore_sso-server.jks" />
<property name="trustStorePass" value="sso-server" />

<!-- Temporaeres Verzeichnis -->
<property name="tmpDir" value="./tmp" />

<!-- Session - Einstellungen -->
<property name="sessionLifetime" value="1800000" />
<property name="sessionValidationInterval" value="30000" />

<!-- E-Mail - Unterstuetzung -->
<property name="systemMail" value="wisnetgrid2009@gmail.com" />
<property name="systemMailUser" value="wisnetgrid2009@gmail.com" /
    >
<property name="systemMailPassword" value="wisnetgrid2009" />
<property name="systemMailSmtpHost" value="smtp.gmail.com" />
<property name="systemMailSmtpPort" value="587" />
<property name="systemMailSmtpAuthenticate" value="true" />
<property name="systemMailTLS" value="true" />
<property name="contactEMail" value="j.daivandy@fz-juelich.de" />
</bean>
```

Auflistung 1: SSO-Server-Konfiguration

Name	Standardwert	Beschreibung	Optional
host	-	Öffentliche IP bzw. Domainname	nein
ssl	true	(De)aktiviert den SSL-Modus. Ohne SSL können Nachrichten abgehört werden. Daher wird ein aktivierter SSL-Modus dringend empfohlen.	nein
port	9999	TCP-Port für Web browser - Zugriff	nein
restPort	8183	TCP-Port für REST Web service - Zugriff	nein
keyStore	conf/keystore.jks	Pfad zum Java Key-store (Teil der SSL-Konfiguration)	bei inaktivem SSL
keyStorePass	sso-server	Passwort zum Zugriff auf o. g. Java Keystore	bei inaktivem SSL
trustStore	conf/keystore.jks	Pfad zum Java Truststore (Teil der SSL-Konfiguration, Üblicherweise im applikationsspezifischen Java Keystore enthalten)	bei inaktivem SSL
trustStorePass	sso-server	Passwort zum Zugriff auf o. g. Java Truststore	bei inaktivem SSL
tmpDir	./tmp	Temporäres Verzeichnis	ja
sessionLifetime	1800000	Gültigkeitsdauer für eine ruhende Nutzersitzung (in Millisekunden)	ja
sessionValidationInterval	30000	Invalidierungsintervall für abgelaufene Nutzersitzungen (in Millisekunden)	ja
systemMail	-	E-Mail-Adresse des SSO-Server	ja
systemMailUser	-	Benutzername zu 'system-Mail'	ja
systemMailPasswort	-	Passwort zu 'systemMail'	ja
systemMailSmtpHost	-	SMTP-Server zu zu 'systemMail'	ja
systemMailSmtpPort	-	SMTP-Port zu zu 'system-Mail'	ja
systemMailSmtpAuthenticate	-	SMTP-Authentisierung zu 'systemMail' (true — false)	ja
contactMail	-	Alternative E-Mail-Adresse zu 'systemMail' für Supportanfragen	ja
	6		

Die Datenbankeinstellung erfolgt im selben Dokument, jedoch innerhalb eines anderen XML-Elements.

3.2.2 Beispielkonfiguration: eingebettete Datenbankverbindung

```
<bean id="dataSource" class="org.h2.jdbcx.JdbcDataSource">
  <property name="URL" value="jdbc:h2:$DB_DIR/$DB_NAME" />
</bean>
```

Auflistung 2: Eingebettete Datenbankverbindung

Die eingebettete Datenbank wird mit folgenden WisNetGrid-Nutzern populiert:

Name	Kennwort	Rolle	Beschreibung
sso-admin	sso-admin	sso_admin	Administratorzugriff auf Single Sign-On Server
sso-user	sso-user	sso_user	Nutzerzugriff auf Single Sign-On Server
webapp-admin	webapp-admin	webapp_admin	Administrator eines Diensteanbieters, der IT-Sicherheit an den Single Sign-On Server delegiert

3.2.3 Beispielkonfiguration: TCP-basierte Datenbankverbindung

```
<bean id="dataSource" class="org.h2.jdbcx.JdbcDataSource">
  <property name="URL" value="jdbc:h2:tcp://$DB_HOST:$DB_PORT/$DB_NAME" />
  <property name="user" value="$DB_USER" />
  <property name="password" value="$DB_PASS" />
</bean>
```

Auflistung 3: TCP-basierte Datenbankverbindung

Dieser Modus erfordert eine installierte und für den Single Sign-On Server erreichbare Instanz der Single Sign-On Database [2].

3.3 Logging-Einstellungen - log4j.properties

Der Single Sign-On Server verwendet das Logging-Framework Apache Log4J, an dessen Dokumentation [3] für tiefgreifende Einstellungsoptionen verwiesen wird. Diese Datei erfordert nach der Installation des Single Sign-On Server (siehe Abschnitt 2) keinerlei Änderungen. Dennoch zeigt die nachfolgende Tabelle etwaige Anpassungsmöglichkeiten auf.

Name	Standardwert	Beschreibung
log4j.rootLogger	INFO, Appender	Anstelle <i>INFO</i> sind noch folgende Log-Level möglich: <ul style="list-style-type: none"> • OFF • WARN • TRACE • FATAL • ERROR • DEBUG • ALL
log4j.appender.Appender.File	logs/sso-server.log	Pfad zur zu füllenden Log-Datei

4 Betrieb

Nach Installation und korrekter Konfiguration lässt sich der Single Sign-On Server mittels der Skripten unter *\$Installationsverzeichnis/bin* in Betrieb nehmen.

Betriebssystem	Start	Neustart	Stop
Unix/Linux	sso-server.sh start	sso-server.sh restart	sso-server.sh stop
Windows	sso-server.bat	-	CTRL + C bzw. entsprechenden Java-Prozess über den Taskmanager stoppen

Etwaige Probleme beim Start sind standardmäßig unter *\$Installationsverzeichnis/logs/sso-server.log* einsehbar (siehe Abschnitt 3.3).

5 Schnittstellen

5.1 Web browser

- `http(s)://$host:$webPort/sso-server` aufrufen
- WisNetGrid-Konto registrieren

5.2 Web service

Client-Bibliotheken und REST Interfaces sind nach Installation und Inbetriebnahme des Single Sign-On Servers abrufbar unter **http(s)://\$host:\$restPort/sso-service**.

6 Schnellstart

- Installationsanweisungen ausführen: Abschnitt 2
- Einstellungen für Host, TCP-Ports und SSL vornehmen: Unterabschnitt 3.2
- Anweisungen für Inbetriebnahme ausführen: Abschnitt 4
 - beim ersten Start wird die Datenbank automatisch im eingebetteten Modus installiert (siehe Abschnitt 3.2.2)

Der Single Sign-On Server ist nun über folgende Schnittstellen erreichbar:

- Web browser: `http(s)://$host:$webPort/sso-server`
- REST Web service: `http(s)://$host:$restPort/sso-service`

Literatur

- [1] Jason Milad Daivandy. *Gesamtsicht auf Komponenten der Ressourcenföderationsschicht*, 2012. Dokumentation.
- [2] Jason Milad Daivandy. *Single Sign-On Database*, 2012. Dokumentation.
- [3] Ceki Gülcü. *Logging-log4j Wiki*, 2001. Dokumentation.
- [4] Denis Hünich. *Ressourcenföderator*, 2012. Dokumentation.