

Socom Quickstart Guide

24. November 2014

Inhaltsverzeichnis

1	Git	1
2	Socom starten	1
2.1	Starten mit Eclipse	1
2.2	Starten als Runnable-Jar	2
3	Am SocomWeb GWT-Projekt mit Eclipse arbeiten	3
4	Socom testen	4
5	Socom säubern (Ursprungszustand)	5

1 Git

https://dev.kom.e-technik.tu-darmstadt.de/gitlab/users/sign_in

2 Socom starten

2.1 Starten mit Eclipse

Bevor die Projekte importiert werden muss das Google Plugin für GWT installiert werden. Dazu die entsprechende Eclipse-Update-Site aus https://developers.google.com/eclipse/docs/getting_started suchen und folgende Features installieren:

Google Plugin for Eclipse x.x aus der Kategorie Google Plugin for Eclipse

Google Web Toolkit SDK 2.4.0 aus der Kategorie SDKs

(Optional) GWT-Designer-Features aus der Kategorie GWT Designer for GPE

Um Socom mit Eclipse zu starten müssen folgend die im git vorliegenden Projekte **SocomAPI**, **SocomWeb** & **SocomWebGraph** importiert werden.

Das SocomAPI Projekt enthält einen Ordner *database*. In diesem ist eine JAR enthalten mit der die Datenbank gestartet werden kann. Die benötigten Optionen sind in der Datei *server.properties* gespeichert (Port, User, Passwort, ...). Um die JAR zu starten steht für Windows die vorgefertigte *database-server.bat* und für Unix die *database-server.sh* zur Verfügung¹.

Wird die Datenbank gestartet sollte die Ausgabe folgendermaßen aussehen:

```
{rhaban@rhaban-ubuntu:~/git/neu_rhaban/SocomAPI/database$ ./database-server.sh
[Server@296672d6]: [Thread[main,5,main]]: checkRunning(false) entered
[Server@296672d6]: [Thread[main,5,main]]: checkRunning(false) exited
[Server@296672d6]: Startup sequence initiated from main() method
[Server@296672d6]: Loaded properties from [/home/rhaban/git/neu_rhaban/SocomAPI/database/server
.properties]
[Server@296672d6]: Initiating startup sequence...
[Server@296672d6]: Server socket opened successfully in 26 ms.
[Server@296672d6]: Database [index=0, id=0, db=file:data/Socomdb, alias=Socom] opened
successfully in 1075 ms.
[Server@296672d6]: Startup sequence completed in 1103 ms.
[Server@296672d6]: 2012-10-02 13:25:33.440 HSQLDB server 2.2.8 is online on port 9001
[Server@296672d6]: To close normally, connect and execute SHUTDOWN SQL
[Server@296672d6]: From command line, use [Ctrl]+[C] to abort abruptly
```

Nachdem die Datenbank² gestartet wurde, kann die SocomAPI gestartet werden. Die Datenbank wird automatisch beim Start von Socom ausgeführt. Dazu muss sich im Projektarchiv der Ordner 'database' befinden. Dazu in Eclipse im importierten SocomAPI-Projekt das Package *de.tud.kom.Socom* aufsuchen und die Java-Klasse *DeploySocom* mit der enthaltenen main-Methode starten. Standardmäßig startet der Server auf Port 7999 8080, was aber in 'socomconfig.ini' (./war/WEB-INF/) angepasst werden kann.

Gleichzeitig wird mit der API der Web-Container für GWT gestartet. Das fertig kompilierte GWT-Projekt befindet sich dazu auch in diesem Projekt im Ordner *war*.

Es bietet sich an, nach dem Importieren der Projekte in Eclipse einen Symlink auf Betriebssystemebene von *SocomWeb|war* auf *SocomAPI|war* anzulegen. So ist das Web-Projekt beim Starten mit Socom immer auf dem Stand der letzten GWT-Kompilierung. Unter *Linux* im Order *SocomAPI* den Ordner *war* löschen und folgendes Kommando ausführen: `ln -s ../SocomWeb/war war`

Unter *Windows (Vista oder neuer)* im Order *SocomAPI* den Ordner *war* löschen und folgendes Kommando ausführen: `mklink /J war ../SocomWeb\war`

Weiterarbeiten in Kapitel 4.

2.2 Starten als Runnable-Jar

Manchmal bietet es sich an die API lieber als kleine Runnable-JAR zu starten. Beispielsweise wenn kleine Änderungen schnell auf einem externen Server getestet werden sollen ohne dass immer das komplette Projekt kopiert werden muss.

Um Socom als Runnable-Jar zu erhalten muss im Kontextmenü des SocomAPI-Projekts (per Rechtsklick) die Option *Export* ausgewählt werden. Dann *Runnable JAR* und ent-

¹Wird die Konsole zum Starten unter Unix verwendet muss, um Probleme zu vermeiden, zuerst in den Ordner gewechselt werden (cd ../database) bevor die .sh gestartet wird. Für Windows einfach draufklicken, das Terminal startet.

²Port: 9001, User: Socom, Passwort: Socom4kom

sprechend dem Dialog folgen. Es bietet sich an die Option *Copy required libraries into a sub-folder next to the generated JAR* zu aktivieren. So bleibt das letztendliche, auszutauschende Socom-Projekt wirklich sehr klein, da keine Bibliotheken etc. neu kopiert werden müssen.

Neben der entstandenen JAR (im Folgenden *server.jar* genannt) und dem Ordner mit den Bibliotheken (hier *server_lib*) werden noch weitere Ordner benötigt. Sie können allesamt aus dem SocomAPI Projekt kopiert werden. Diese sind:

database Die Datenbank.

ffmpeg Binärdateien zur Audiokonvertierung durch die API.

war Das Web-Projekt.

config Die Konfiguration für Log4j.

Nachdem alle Dateien vorhanden sind ~~muss zunächst wieder die Datenbank gestartet werden (siehe 2.1, wobei der kopierte enthaltene *database* Ordner genutzt werden kann).~~ Daraufhin kann die JAR gestartet werden³. Socom kann nun genutzt werden. Der Ordner kann natürlich beliebig rumgeschoben werden und die *server.jar* ausgetauscht (aktualisiert). Das Web-Projekt ist davon allerdings unabhängig und kann nur durch Ersetzen des *war*-Ordners aktualisiert werden.

3 Am SocomWeb GWT-Projekt mit Eclipse arbeiten

Das Web-Projekt kann in Eclipse beliebig bearbeitet werden. Änderungen können mithilfe einer Neu-Compilierung übernommen werden. Durch die Installation des Google-Plugins ist in der Werkzeugleiste ein neues Symbol mit einem G entstanden. Dessen Menü bietet die Option *GWT-Compile-Project* an. Dort das Projekt SocomWeb auswählen, wenn es nicht ausgewählt ist und mit *Compile* bestätigen. Der Compiliervorgang kann bis zu 2 Minuten dauern und sollte bei Erfolg etwa folgende Konsolenausgaben liefern (Momentan vorhandene gelbe Warnungen können ignoriert werden):

```
Compiling module de.tud.kom.Socom.web.SocomWeb
  Compiling 6 permutations
    Compiling permutation 0...
    Compiling permutation 1...
    Compiling permutation 2...
    Compiling permutation 3...
    Compiling permutation 4...
    Compiling permutation 5...
  Compile of permutations succeeded
Linking into /home/rhaban/git/neu_rhaban/SocomWeb/war/Socomweb
Link succeeded
Compilation succeeded — 52,692s
```

Das neu kompilierte Projekt befindet sich im Ordner *war*. Dieser kann in das SocomAPI-Projekt geschoben werden um dort eine aktuelle Version zu erhalten oder in den Ordner neben die Runnable-JAR kopieren um dort eine aktuelle Version zu haben. Logisches verlinken der Ordner vermeidet lange Kopiervorgänge.

³java -jar server.jar

Die Haupt HTML-Seite des Projekts befindet sich im war Ordner unter dem Namen *Socom-Web.html*. Entsprechend existiert eine *.css*-Datei. Im Ordner *data* werden Daten für verschiedenste Dinge wie Influence-Uploads, Spiele-Bilder, Achievement-Bilder, u.v.m. gespeichert. Der *temp*-Ordner enthält Dateien die aus der Datenbank als File abgelegt wurden um sie in der Webseite präsentieren zu können.

4 Socom testen

Ist Socom erfolgreich gestartet sollte die Ausgabe folgende Form haben:

```
2013-45-22 15:45:59 [main] INFO de.tud.kom.socom.DeploySocom - ***Server is running on rhaban
-mint/127.0.1.1@8080
2013-45-22 15:45:59 [main] INFO de.tud.kom.socom.database.HSQLAccess - Start Database Process
2013-45-22 15:45:59 [main] INFO de.tud.kom.socom.database.HSQLAccess - [Server@40c19080]: [
Thread[main,5,main]]: checkRunning(false) entered
2013-45-22 15:45:59 [main] INFO de.tud.kom.socom.database.HSQLAccess - [Server@40c19080]: [
Thread[main,5,main]]: checkRunning(false) exited
2013-45-22 15:45:59 [main] INFO de.tud.kom.socom.database.HSQLAccess - [Server@40c19080]:
Startup sequence initiated from main() method
2013-45-22 15:45:59 [main] INFO de.tud.kom.socom.database.HSQLAccess - [Server@40c19080]:
Loaded properties from [/home/rhaban/Workspace/Socom/SocomAPI/database/server.properties]
2013-45-22 15:45:59 [main] INFO de.tud.kom.socom.database.HSQLAccess - [Server@40c19080]:
Initiating startup sequence...
2013-45-22 15:45:59 [main] INFO de.tud.kom.socom.database.HSQLAccess - [Server@40c19080]:
Server socket opened successfully in 10 ms.
2013-46-22 15:46:00 [main] INFO de.tud.kom.socom.database.HSQLAccess - [Server@40c19080]:
Database [index=0, id=0, db=file:data/Socomdb, alias=Socom] opened successfully in 871 ms.
2013-46-22 15:46:00 [main] INFO de.tud.kom.socom.database.HSQLAccess - Tables already exists
- skip.
2013-46-22 15:46:00 [main] INFO de.tud.kom.socom.database.HSQLAccess - ***Database ready
```

Ist Socom im Ursprungszustand (siehe 5) so muss zunächst die Datenbank mit einigen Beispieldaten gefüllt werden bevor sie genutzt werden kann. Beispielsweise muss ein Spiel inkl. Spielinstanz erstellt werden bevor überhaupt ein Nutzer erstellt werden kann. Die Folgenden URLs sind zu nutzen um Socom testen zu können.

Vor jede URL muss die Adresse + Port des Server sowie der Zusatz *servlet* gegeben werden. Also muss vor den beschriebene URL-Fragmente folgendes stehen:

`http://localhost:8080/servlet/`

`game/addgame?game=Portal&genre=Adventure&password=portalpw&mastersecret=123a45b6`

Hiermit wird das Spiel Portal der Genre Adventure mit dem Passwort portalpw gespeichert.

`game/addgameinstance?game=Portal&password=portalpw&version=1.0&description=Portal123`

Erstellt eine neue Spielinstanz mit Versionsnummer (String) und einer Beschreibung.

`game/addgamescene?game=Portal&password=portalpw&version=1.0&sceneid=a&name=Szene A`
`game/addgamescene?game=Portal&password=portalpw&version=1.0&sceneid=b&name=Szene B`
`game/addgamescenerelation?game=Portal&version=1.0&password=portalpw&parent=a&child=b`

Erstellt zwei Szenen A und B und verbindet Sie mit einer Relation zueinander.

```
user/createuser?username=rhaban&password=meinpw&game=Portal&version=1.0&
gamepassword=portalpw&visibility=2
```

Erstellt einen neuen Nutzer namens *Rhaban* mit Passwort *meinpw* und initialisiert seine Spielzugehörigkeit zu Portal, Version 1.0. Die Sichtbarkeit des Profils ist mit 2 auf 'public' gesetzt.

```
user/validateuser?username=rhaban&password=meinpw&game=Portal&version=1.0&
gamepassword=portalpw
```

Loggt den Nutzer ein. Wird ein Browser genutzt speichert diese automatisch das gesendete Cookie und sendet es entsprechend zurück, sodass der Nutzer bei folgenden Requests eingeloggt ist. Für weitere Requests am Besten im SocomAPI Projekt den Ordner *doc_public* beachten oder die Manager der einzelnen Komponenten im Quelltext durchsuchen. Demnächst folgt ein weiteres Test-Projekt bzw. ein Mini-GUI-Client.

5 Socom säubern (Ursprungszustand)

Um Socom in den Ursprungszustand zu bringen muss zunächst Socom gestoppt werden. Desweiteren muss der Ordner *data* aus dem Datenbank-Ordner gelöscht werden. Die Datenbank ist nun leer. Desweiteren kann der Log-Ordner natürlich gelöscht werden. Im *war* Ordner sollte darüber hinaus aus dem Ordner *data* der komplette Ordner *influence_data* gelöscht werden. Zuletzt sollte der Ordner *temp* aus dem Ordner *war* entfernt werden. Beim erneuten starten sind nun keine Spuren von alten Laufzeiten enthalten.