How to deploy and start the chatbot

Requirements

- Software
  - Java 8, set JAVA_HOME and add Java to the PATH
  - Maven
  - Gradle
  - Git

Install the Software on Ubuntu

- `sudo apt-get install maven`
- `sudo apt-get install gradle`
- `sudp apt-get install git`
- `sudo apt-get install openjdk-8-jdk-headless`
- `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

Checkout the Code from the Repository

- `git clone git@gitlab.dai-labor.de:IRML/Chatbot.git`

or

- `export GIT_SSL_NO_VERIFY=true`
- `git clone https://gitlab.dai-labor.de/IRML/Chatbot.git`
- Example outout
  - `Cloning into 'Chatbot'...`
  - `Username for 'https://gitlab.dai-labor.de': andreas`
  - `Password for 'https://andreas@gitlab.dai-labor.de':`
  - `remote: Counting objects: 29561, done.`
  - `remote: Compressing objects: 100% (450/450), done.`
  - `remote: Total 29561 (delta 319), reused 785 (delta 237)`
  - `Receiving objects: 100% (29561/29561), 410.76 MiB | 39.53 MiB/s, done.`
  - `Resolving deltas: 100% (15248/15248), done.`
  - `Checking connectivity... done.`

Build the Project

- Maven build in ~/Chatbot
  - `cd Chatbot`
  - `mvn clean install`
- Maven assembly build in ssds-index
  - `cd ssds-index`
  - `mvn assembly:assembly`
  - Expected Output `[INFO] Reading assembly descriptor: src/main/assembly/src.xml [INFO] Building tar: /home/tester/Chatbot/ssds-index/target/ssds-admin.tar.gz [INFO] -- --------- [INFO] BUILD SUCCESS [INFO] ------------------------ --`

- Build the grails webapp
  - Go to the wepp-app directory and run gradle build `cd ../webapp/gradle war`
  - Expected output `BUILD SUCCESSFUL Total time: 2 mins 0.221 secs This build could be faster, please consider using the Gradle Daemon: https://docs.gradle.org/2.10/userguide/gradle_daemon.html`

Download and Unpack the Server Software

- Download the Jetty server and the Solr-Server from the DAI webserver
  - `[ -d "Downloads" ] || mkdir Downloads`
  - `cd Downloads`
  - `wget http://dainas.aot.tu-berlin.de/~andreas@DAI/SSDS-RESOURCE/ssds-app-server.tgz`
  - `wget http://dainas.aot.tu-berlin.de/~andreas@DAI/SSDS-RESOURCE/ssds-solr-server.tgz`
  - `tar xvzf ssds-app-server.tgz --directory ..` start app appears - mod date old
  - `tar xvzf ssds-solr-server.tgz --directory ..` start solr appears - mod date old
  - `cd $HOME`
  - `ln -s Chatbot/ssds-data-berlin` creates “deprecated alias” on mac on home dir
  - `ln -s Chatbot/ssds-data-hamburg`
  - `ln -s Chatbot/ssds.properties`

Configure the Environment and the Path

- create a file `$HOME/ssds.environment`
- adapt the `root.path` in the `ssds.properties` for the desired environment

"root.dir"
: "/Users/mega/home/ssds.environment",

Create a backup

- backup old stuff
  - `[ -d "ssds-backup" ] || mkdir ssds-backup`

Install the new system

```
cd $HOME
MYDATE=`date +'%Y%m%d-%H%M%S'`
HOSTNAME=`hostname`
BACKUPDIR=$HOME/ssds-backup/$MYDATE
INDEXBASEDIR=$HOME/ssds-solr/data/cores
# cp $HOME/Chatbot/ssds.properties .

# stop solr
cd $HOME
./stop-solr.sh
sleep 10

mkdir -p $BACKUPDIR/ssds-solr/data/cores

mkdir -p $HOME/ssds-solr/data/cores
cp -r $HOME/Chatbot/ssds-index/solr/* $INDEXBASEDIR/..
./start-solr.sh
echo "sleep 120 seconds"
sleep 120

[ -d "HOME/ssds-admin" ] || mv $HOME/ssds-admin $BACKUPDIR
mkdir -p $HOME/ssds-admin
```

dunno
dunno
dunno
ok

```
tar xvzf $HOME/Chatbot/ssds-index/target/ssds-admin.tar.gz --directory
$HOME/ssds-admin
cd $HOME/ssds-admin
./start-data-update.sh
sleep 10

cd $HOME
./stop-app.sh
mkdir -p $BACKUPDIR/ssds-app/webapps
mv $HOME/ssds-app/webapps/va $BACKUPDIR/ssds-app/webapps
mv $HOME/ssds-app/webapps/va.war $BACKUPDIR/ssds-app/webapps
cp $HOME/Chatbot/webapp/build/libs/va.war $HOME/ssds-app/webapps/

cd $HOME
echo "sleep for 60 seconds"
sleep 60
./start-app.sh
```