

Master of Advanced Studies
Software Engineering 2018

Modul Software Testing



Übungsumgebung

Juni 2019

Thomas Briner
thomas.briner@gmail.com

Inhaltsverzeichnis

Übungsumgebung	3
Schritt 1: Vorbereitung der benötigten Tools.....	3
Schritt 2: Klonen des Repository in lokales Dateisystem.....	5
Schritt 3: Projekte ab command line bauen	6
Schritt 4: Projekte in IDE öffnen.....	7
Schritt 5: System under Test starten.....	10

Übungsumgebung

Die praktischen Übungen werden basierend auf dem Code durchgeführt, wie er im Repository unter <https://github.com/thomasbriner/HSR-SW-Testing-Code> verfügbar ist.

Die Übungsumgebung besteht aus Java Code, der für die Umsetzung der Unit Tests verwendet wird, einem Webshop als System under Test und einem Rahmen für die Durchführung von Selenium Tests.

Schritt 1: Vorbereitung der benötigten Tools

Um die Übungsumgebung nutzen zu können, sind verschiedene Tools notwendig.

Tool	Beschreibung	Quelle für Installation	Verifikation für korrekte Installation
Java JDK Version 8	<p>Als Programmiersprache wird Java verwendet.</p> <p>Damit die Umgebung reibungslos funktioniert, muss die Java Version 8 verwendet werden.</p> <p>Falls bereits eine höhere Version installiert ist, so muss zusätzlich die JDK Version 8 installiert und die Systempfade entsprechend angepasst werden.</p>	http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html	<p>Auf Command Line:</p> <pre>\$ java -version java version "1.8.0_131" Java(TM) SE Runtime Environment (build 1.8.0_131-b11) Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)</pre>

Maven	Als Build Tool für den Webshop wird Maven verwendet	Download: https://maven.apache.org/download.cgi Installation: https://maven.apache.org/install.html	Auf Command Line, ev. im entsprechenden Installationsverzeichnis <pre>\$ mvn -version Apache Maven 3.5.3 Maven home: /usr/local/Cellar/maven/3.5.3/libexec ... Default locale: de_CH, platform encoding: UTF-8 OS name: "mac os x", version: "10.13.3", arch: "x86_64", family: "mac"</pre>
Gradle	Ein Teil der Module arbeitet mit Gradle als Build Tool	Installation: https://gradle.org/install/	Auf Command Line, ev. im entsprechenden Installationsverzeichnis: <pre>\$ /opt/gradle/gradle-4.2.1/bin/gradle -version ----- Gradle 4.2.1 ----- Build time: 2017-10-02 15:36:21 UTC Revision: a88ebd6be7840c2e59ae4782eb0f27fbe3405ddf ...</pre>
Git	Das Repository verwendet Git als Versionsverwaltungstool	Download: https://git-scm.com/downloads Installation: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git	Auf Command Line: <pre>\$ git -version git version 2.14.3 (Apple Git-98)</pre>
IDE	Als Entwicklungsumgebungen wird IntelliJ oder Eclipse empfohlen.	IntelliJ: https://www.jetbrains.com/help/idea/install-and-set-up-product.html Eclipse: http://www.eclipse.org/downloads/eclipse-packages/ z.B. Eclipse IDE for Java Developers	IDE startet fehlerfrei

Schritt 2: Klonen des Repository in lokales Dateisystem

Um mit dem Code arbeiten zu können, erstellen Sie eine lokale Kopie des ganzen Repositorys.

1. Command-Line öffnen
2. Lokal an gewünschtem Ort in Verzeichnis wechseln

```
$ mkdir MAS2018-Testing  
$ cd MAS2018-Testing/
```

3. Repository klonen

```
$ git clone https://github.com/thomasbriner/HSR-SW-Testing-Code.git  
Cloning into 'HSR-SW-Testing-Code'..  
remote: Counting objects: 959, done.  
remote: Compressing objects: 100% (550/550), done.  
remote: Total 959 (delta 340), reused 953 (delta 337), pack-reused 3  
Receiving objects: 100% (959/959), 27.23 MiB | 5.19 MiB/s, done.  
Resolving deltas: 100% (340/340), done.
```

4. Verifikation: Überprüfen Sie, dass die Inhalte vorhanden sind

```
$ ls -l HSR-SW-Testing-Code/  
total 112  
-rw-r--r-- 1 briner wheel 34978 7 Jun 09:53 LICENSE  
-rw-r--r-- 1 briner wheel 4802 7 Jun 09:53 README.md  
drwxr-xr-x 9 briner wheel 288 7 Jun 09:53 admin  
drwxr-xr-x 9 briner wheel 288 7 Jun 09:53 api  
-rw-r--r-- 1 briner wheel 574 7 Jun 09:53 boot-community-demo.iml  
drwxr-xr-x 9 briner wheel 288 7 Jun 09:53 core  
-rw-r--r-- 1 briner wheel 6457 7 Jun 09:53 pom.xml  
drwxr-xr-x 14 briner wheel 448 7 Jun 09:53 selenium-tests  
drwxr-xr-x 9 briner wheel 288 7 Jun 09:53 site
```

Schritt 3: Projekte ab command line bauen

Für diesen Schritt ist es notwendig, eine Verbindung ins Internet zu haben.

1. Navigieren Sie auf der Commandline in den entsprechenden Ordner HSR-SW-Testing-Code

```
$ cd HSR-SW-Testing-Code/
```

2. Bauen Sie die Maven Projekte mit folgendem Befehle:

```
$ mvn clean install
```

Die Ausführung startet mit diesem Output:

```
[INFO] Scanning for projects...
[INFO] -----
[INFO] Reactor Build Order:
[INFO]
[INFO] Broadleaf Spring Boot Community Demo [pom]
[INFO] Community Demo Core [jar]
[INFO] Community Demo Site [jar]
[INFO] Community Demo Admin [jar]
[INFO] Community Demo API [jar]
[INFO]
[INFO] -----< com.mycompany-community:boot-community-demo >-----
[INFO] Building Broadleaf Spring Boot Community Demo 1.0.0-SNAPSHOT [1/5]
[INFO] -----[ pom ]-----
[INFO]
```

und endet mit:

```
[INFO] -----
[INFO] Reactor Summary:
[INFO]
[INFO] Broadleaf Spring Boot Community Demo 1.0.0-SNAPSHOT SUCCESS [ 0.822 s]
[INFO] Community Demo Core ..... SUCCESS [ 4.436 s]
[INFO] Community Demo Site ..... SUCCESS [ 6.991 s]
[INFO] Community Demo Admin ..... SUCCESS [ 1.844 s]
[INFO] Community Demo API 1.0.0-SNAPSHOT ..... SUCCESS [ 2.274 s]
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 17.025 s
[INFO] Finished at: 2018-06-10T20:52:41+02:00
[INFO] -----
```

3. Wechseln sie nun ins Unterverzeichnis selenium-tests

```
$ cd selenium-tests/
```

- 4.
5. Starten sie den Gradle Build für dieses Projekt

```
$ gradle build
```

6. Der Output sollte so aussehen:

```
Starting a Gradle Daemon (subsequent builds will be faster)

> Task :compileTestJava
Note: /Users/briner/tmp/HSR-SW-Testing-Code/selenium-
tests/src/test/java/ch/testing/selenium/weekenddiscount/util/FileUtil.
java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

BUILD SUCCESSFUL in 9s
4 actionable tasks: 4 executed
```

Damit ist das Projekt erfolgreich gebaut.

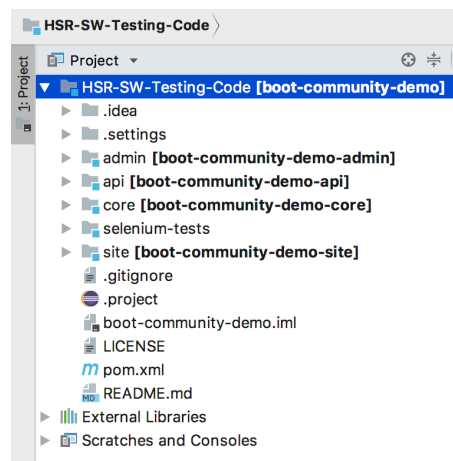
Schritt 4: Projekte in IDE öffnen

IntelliJ

- Mit „File → Open“ das Verzeichnis HSR-SW-Testing-Code öffnen
- Die 5 Module werden angezeigt:
 - admin
 - api
 - core
 - selenium-tests
 - site

Verifikation:

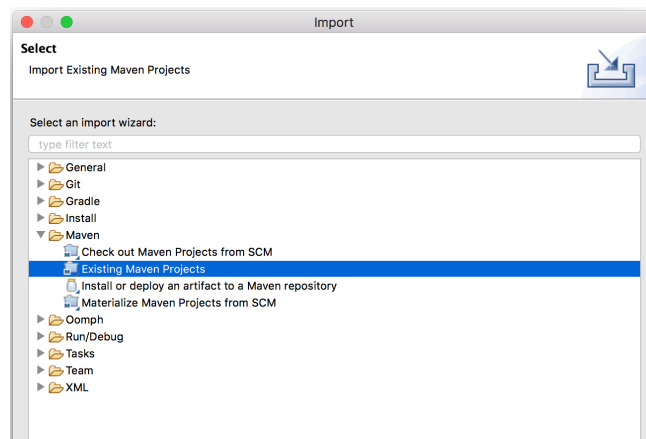
Das Projekt mit den 5 Modulen wird angezeigt:



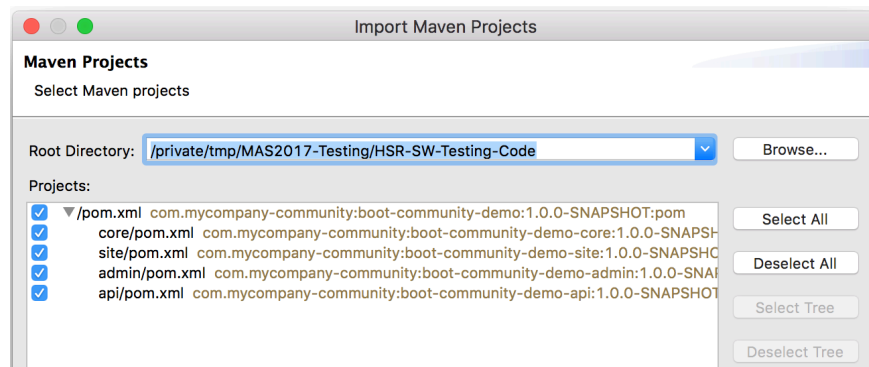
Eclipse

- Maven Projekte importieren:

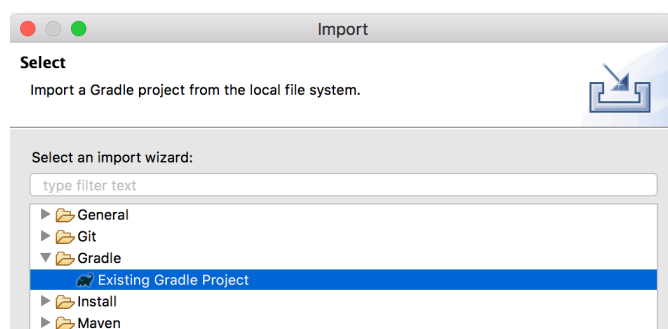
- Import Projects → Maven → Existing Maven Projects



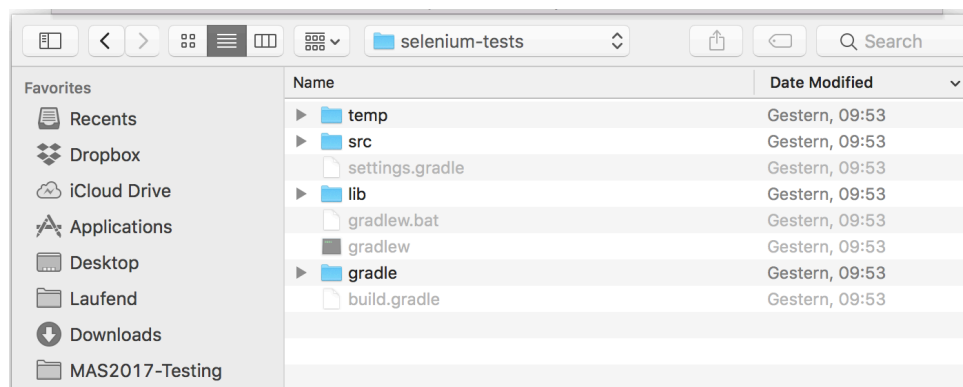
- Verzeichnis HSR-SW-Testing-Code wählen → Parent pom.xml mit 4 Child-POM werden angezeigt



- Alles selektieren und importieren
- selenium-tests projekt mit gradle importieren:
 - Import Projects → Gradle → Existing Gradle Projects

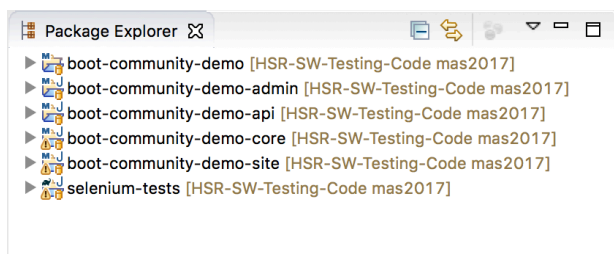


- HSR-SW-Testing-Code/selenium-tests wählen



Verifikation:

Die 5 Projekte sollten im Workspace zur Verfügung stehen:



Schritt 5: System unter Test starten

1. Navigieren Sie auf der Commandline in den entsprechenden Ordner HSR-SW-Testing-Code

```
$ cd HSR-SW-Testing-Code/
```

2. Wechseln Sie in den Ordner site

```
$ cd site/
```

3. Starten Sie die Webapplikation mit mvn spring-boot:run

```
$ mvn spring-boot:run
```

Die Ausführung startet mit diesem Output:

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.mycompany-community:boot-community-demo-site >-----
[INFO] Building Community Demo Site 1.0.0-SNAPSHOT
[INFO] -----[ jar ]-----
```

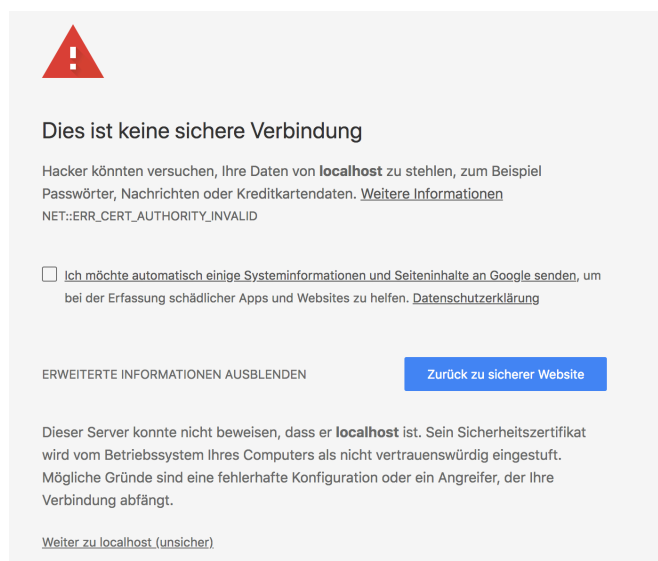
und kann länger dauern (mehrere Minuten), bis das System gestartet ist und der Output so aussieht:

```
2018-06-10 21:02:43.308 INFO o.b.c.s.s.s.index.SolrIndexServiceImpl : Indexing operation completed in 0:00:01.031
2018-06-10 21:02:43.319 INFO o.b.c.s.s.s.index.SolrIndexServiceImpl : Finished building entire Solr index in 0:00:01.346
2018-06-10 21:02:43.320 INFO c.c.c.s.s.SolrIndexCleanupServiceImpl : All indexes rebuilt at startup because value was
create
2018-06-10 21:02:43.630 INFO s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8443 (https) 8080 (http)
2018-06-10 21:02:43.642 INFO com.community.SiteApplication : Started SiteApplication in 41.362 seconds (JVM
running for 42.282)
```

4. Öffnen Sie im Browser die folgende URL:

<http://localhost:8080/>

Sie werden auf die Seite weitergeleitet und je nach Browsereinstellungen erscheint eine Fehlerseite bezüglich der Authentizität des Serverzertifikates.



Fügen Sie hier eine Ausnahme hinzu, so dass die Seite angezeigt wird. Dieser Fehler tritt auf, weil für die lokale Seite kein gültiges Zertifikat besteht.

Nun sollten Sie die Startseite des Webshops „Heat Clinic“ sehen:

