
Protokoll zur SEW-Übung

„Login mit JSF“

SEW
4AHITM 2015/16

Yunus Sari

Note:

Betreuer: Michael Borko

Version 1.0

Begonnen am 12. Juni 2016

Beendet am 14. Juni 2016

INHALTSVERZEICHNIS

1	Aufgabenstellung	1
2	Aufwandschätzung	1
3	Vorbereitung.....	2
4	Umsetzung	6
5	Demonstration	8
6	Versionisierung	10
7	Zeitaufzeichnung	10

1 AUFGABENSTELLUNG

Folge der Anleitung und erstelle das erweiterte Login-Beispiel!

Abgabe: Code auf GitHub pushen

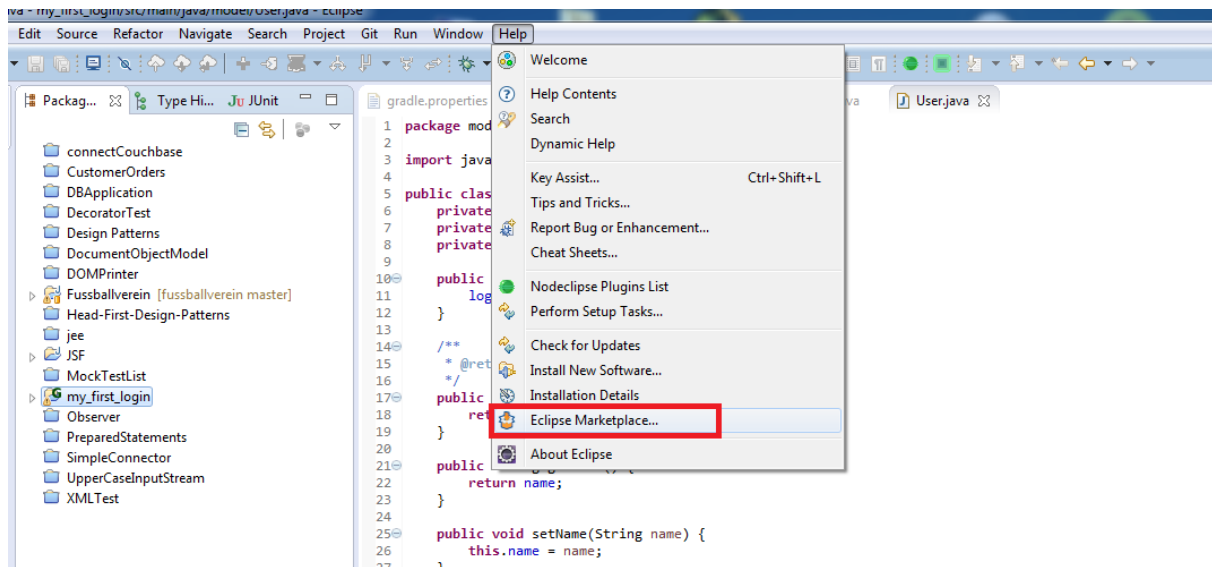
2 AUFWANDSCHÄTZUNG

Arbeitspaket	Geschätzter Aufwand
Installation und Konfiguration der Umgebung	0 min – schon bereit
Programmierung	1 h 30 min
Protokollierung	1 h 30 min
Summe	3 h 00 min

3 VORBEREITUNG

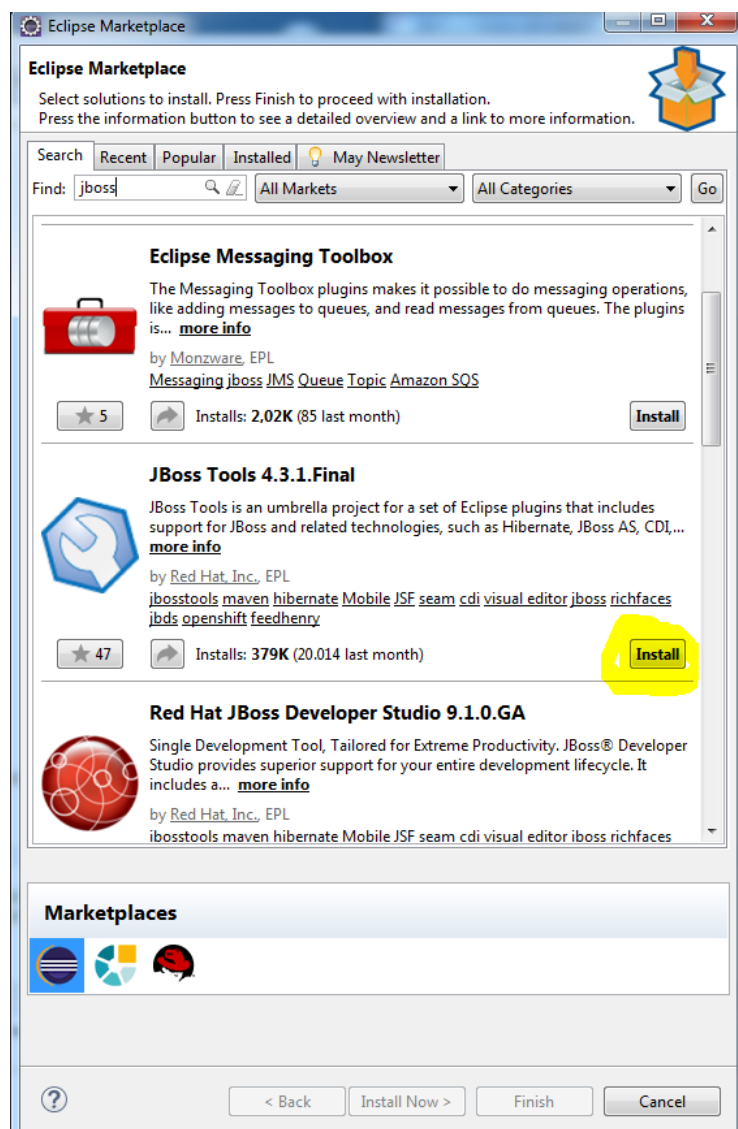
Installation der JBoss Tools:

Man geht in den Eclipse Marketplace



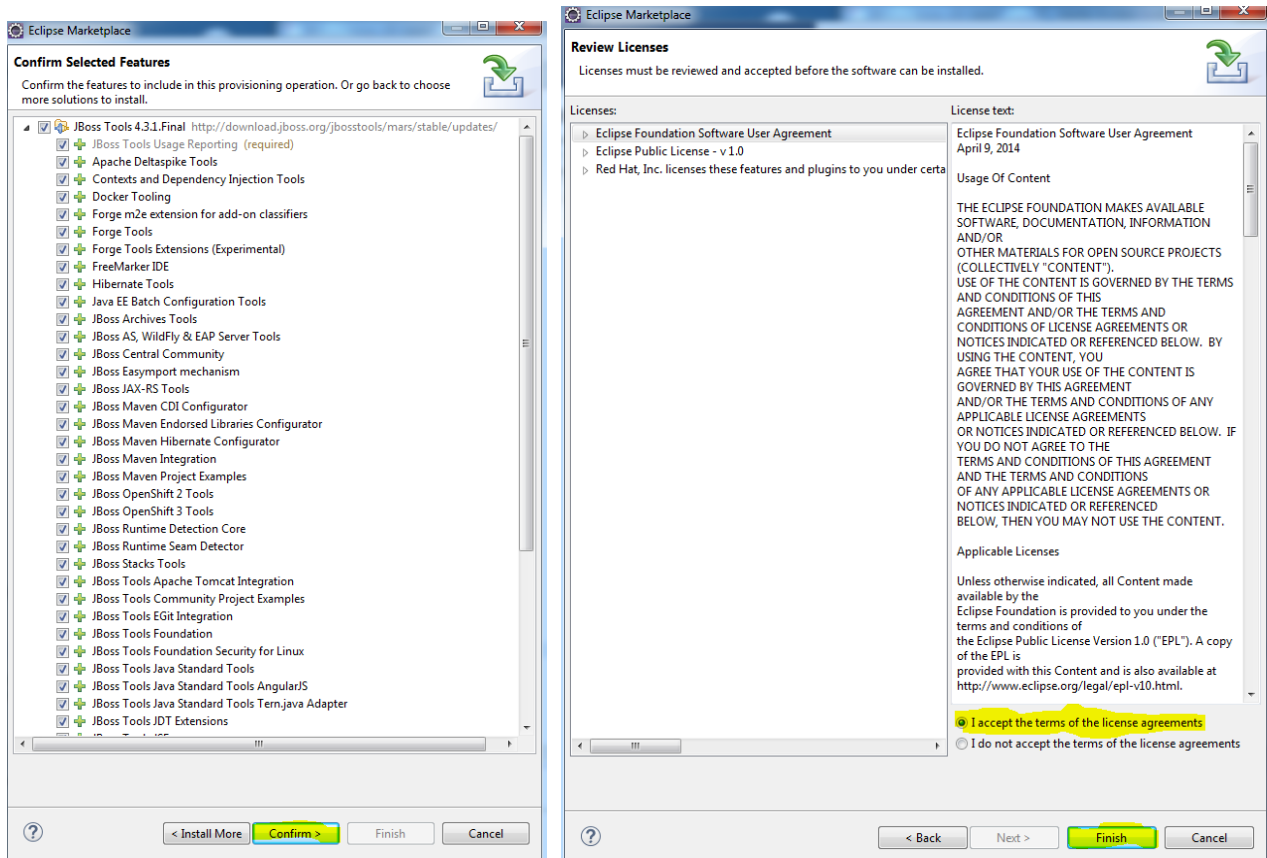
Und sucht nach JBoss.

Wenn man JBoss Tools 4.3.1.Final gefunden hat klickt man auf Install.

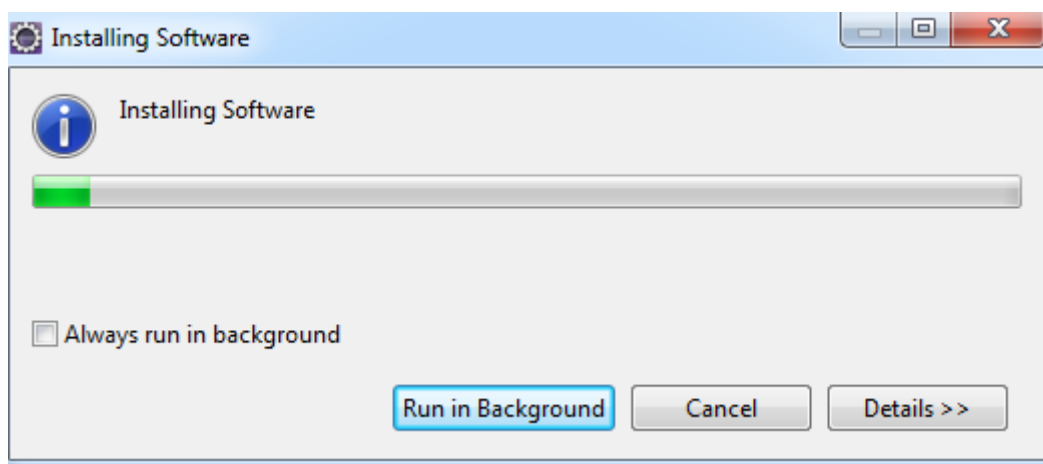


Nach dem klicken auf Install erscheint zuerst ein Menü zum Auswählen der Features. Man klickt auf confirm.

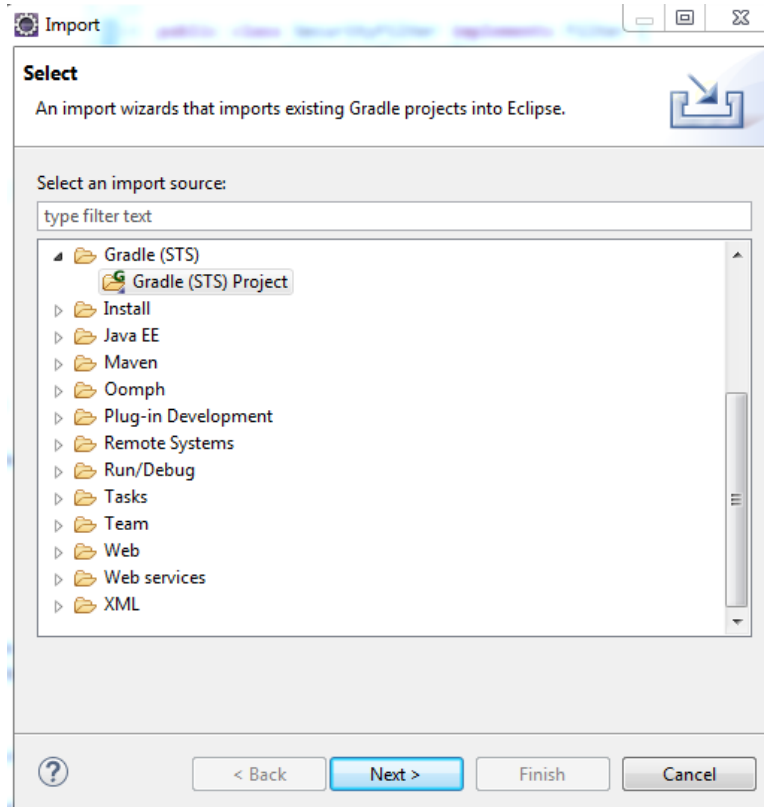
Beim nächsten Fenster erscheinen die license agreements. Nach dem sorgfältigen Lesen klickt man auf „I accept the license terms of the agreements“ und dann auf Finish.



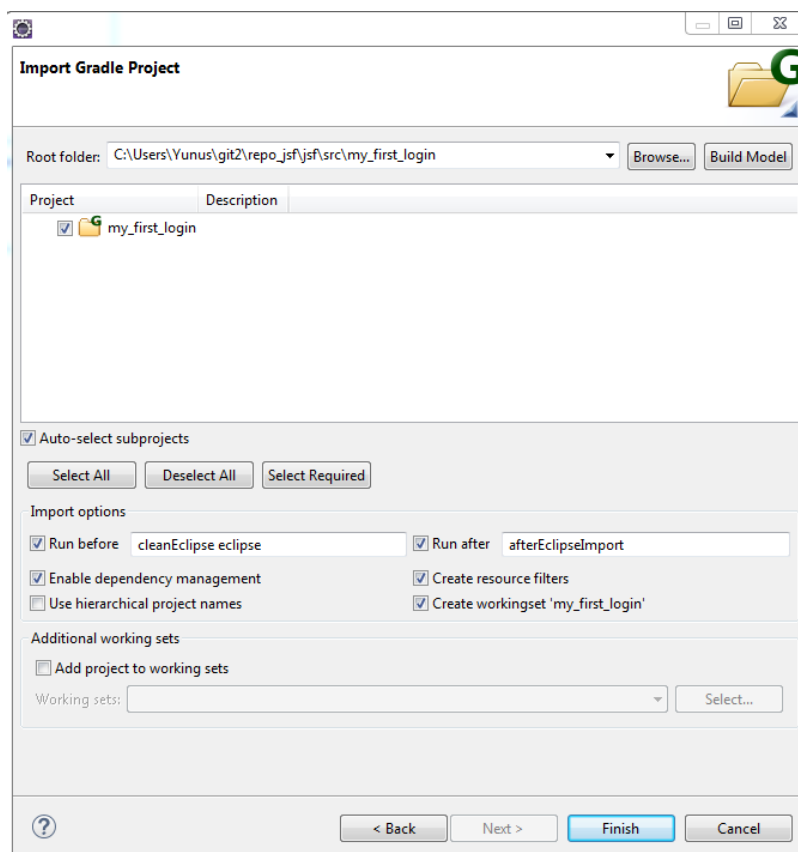
Nun erscheint das Installationsfenster. JBoss wird installiert.



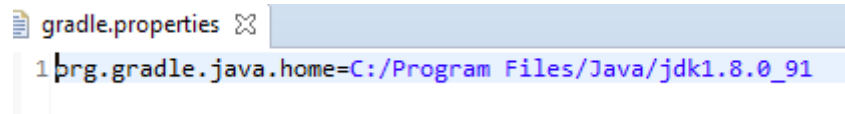
Importieren eines vorhandenen Gradle Projekts:



Das jeweilige erstellte Projekt wird ausgewählt und mit finish importiert.



Unter gradle.properties wurde der Path zu Java angepasst.



```
gradle.properties
1 org.gradle.java.home=C:/Program Files/Java/jdk1.8.0_91
```

Mit „gradle jettyrun“ im Ordner wird es gestartet:

```
3.06.2016 00:19 <DIR>      .
3.06.2016 00:19      478 .classpath
8.05.2016 07:14      275 .gitignore
2.06.2016 23:52 <DIR>      .gradle
3.06.2016 00:19      458 .project
3.06.2016 00:19 <DIR>      .settings
3.06.2016 00:19 <DIR>      bin
2.06.2016 23:52 <DIR>      build
8.05.2016 07:14      645 build.gradle
2.06.2016 23:52 <DIR>      gradle
3.06.2016 00:16          54 gradle.properties
8.05.2016 07:14      5.046 gradlew
8.05.2016 07:14      2.314 gradlew.bat
16.06.2016 09:33      648 settings.gradle
2.06.2016 23:52 <DIR>      src
                        8 Datei(en),      9.918 Bytes
                        8 Verzeichnis(se), 55.652.065.280 Bytes frei

::\Users\Yunus\git2\repo_jsf\jsf\src\my_first_login>gradle jettyrun
compileJava
processResources
classes
· Building 75% > :jettyRun > Running at http://localhost:8080/login
```

4 UMSETZUNG

ToDo: (trainer.xhtml)

```
<!-- TODO
outputLabel fuer msg.left
outputLabel fuer card.left mit id "left"
selbiges fuer msg.right bzw. card.right -->
```

Durchführung:

```
<h:outputLabel value="#{msg.left}" />
<h:outputLabel id="left" value="#{card.left}" />
<h:outputLabel value="#{msg.right}" />
<h:outputLabel id="right" value="#{card.right}" />
```

ToDo: (trainer.xhtml)

```
<!-- TODO outputLabel und outputText fuer result -->
```

Durchführung:

```
<h:outputLabel value="Output of Results: " />
<h:inputText id="result" value="#{card.result}" />
```

ToDo: (loginView.xhtml)

```
<!-- TODO
outputLabel fuer msg.user
inputText fuer user.name inkl. LoginValidator
outputLabel fuer msg.password
inputSecret fuer user.password -->
```

Durchführung:

```
<h:outputText value="#{msg.user}" />
<p:inputText value="#{user.name}" />
<h:outputLabel value="#{msg.password}" />
<p:password value="#{user.password}" label="Password" required="true" />
```

ToDo: (loginView.xhtml)

```
<!-- TODO commandButton mit Text msg.login, user.login wird aufgerufen -->
```

Durchführung:

```
<p:commandButton value="#{msg.Login}" action="#{user.Login}"/>
```

ToDo: (trainer.xhtml)

```
<!-- TODO drei buttons: card.show / msg.show, card.next / msg.next, user.logout /
msg.logout -->
```

Durchführung:

```
<p:commandButton action="#{card.show}" ajax="false" value="#{msg.show}" />
<p:commandButton action="#{card.next}" ajax="false" value="#{msg.next}"
type="submit" />
<p:commandButton action="#{user.logout}" ajax="false" value="#{msg.logout}" />
```

ToDo: (Card.java)

```
// TODO left und right zufaellig befuellen
```

Durchführung:

```
left = (int)(Math.random() * 100);
right = (int)(Math.random() * 100);
```


ToDo: (User.java)

```
// TODO name und password ueberpruefen - im Fehlerfall "failed" returnen,  
// ansonsten "loggedIn" auf true setzen und "success" returnen
```

Durchführung:

```
if(name.equals("ysari") && password.equals("pass")){  
    loggedIn = true;  
    return "success";  
} else {  
    return "failed";  
}
```

5 DEMONSTRATION

Nach dem eingeben der URL:

Train your Brain

User	<input type="text"/>
Password	<input type="password"/>

Login

Bei falscher Eingabe der Logindaten :

Train your Brain

Login failed!

[Please log in again!](#)

Bei richtiger Eingabe der Logindaten (User: ysari; Passwort: pass)

Train your Brain

Result = left * right	
Left Side	30
Right Side	52

Output of Results:

Show Result

New Test

Logout

Nach dem Klick auf „Show Result“:

Train your Brain

Result = left * right	
Left Side	30
Right Side	52
Output of Results:	1560

Show Result **New Test** **Logout**

Nach dem Klick auf „New Test“:

Train your Brain

Result = left * right	
Left Side	85
Right Side	58
Output of Results:	0

Show Result **New Test** **Logout**

Nach dem Klick auf „Logout“:

Train your Brain

You logged out successfully.

[Please log in again!](#)

6 VERSIONISIERUNG

Link zur Respository:

<https://github.com/ysari-tgm/jsf>

Link zum Source Code:

https://github.com/ysari-tgm/jsf/tree/master/src/my_first_login

Link zum Protokoll:

<https://github.com/ysari-tgm/jsf/tree/master/protocol/abgabe2>

7 ZEITAUFZEICHNUNG

Arbeitspaket	Tatsächlicher Aufwand
Installation und Konfiguration der Umgebung	0 min – schon bereit
Programmierung	1 h 00 min
Protokollierung	1 h 15 min
Summe	2 h 15 min