

Open Source-basierte Softwareentwicklung

Praktikum 5 - Build Management

Szenario

In dieser Woche lernen Sie die ersten Schritte mit dem Build-Werkzeug »Maven«.

Im ersten Teil des Praktikums arbeiten Sie mit der Kommandozeile und erstellen ein neues Maven-Projekt. Im zweiten Teil des Praktikums verwalten Sie Ihr Projekt mit GitHub und verwenden Sie Eclipse um Maven-Operationen aus der GUI auszuführen.

Arbeitsschritte

Schritt 1

Installieren Sie – falls noch nicht vorhanden – mindestens das Java 8 JDK und Maven (von <https://maven.apache.org>) lokal auf Ihren Rechner. Passen Sie eventuell Ihren Pfad an, sodass der Befehl »mvn --help« in einem beliebigen Verzeichnis gefunden und ausgeführt wird.

Schritt 2

Erzeugen Sie mit folgendem Kommando (Achtung, alles in einer Zeile!) in einem lokalen Arbeitsverzeichnis ein neues Maven Projekt mit einem Namen (artifactId) Ihrer Wahl:

```
mvn archetype:generate -B -DarchetypeArtifactId=maven-archetype-quickstart  
-DarchetypeVersion=1.3 -DgroupId=de.hfu -DartifactId=praktikum
```

Überschreiben Sie dann die Datei `pom.xml` mit der Version aus: <http://webtech.informatik.hs-furtwangen.de/oss/pom.xml>

Schritt 3

Erstellen Sie ein Java Programm, bestehend aus einer Quelltextdatei, mit einer `main`-Methode, welches eine Zeichenkette von der Tastatur einliest, in Großbuchstaben umwandelt und wieder in die Konsole ausgibt.

Schritt 4

Mit »mvn package« im Hauptverzeichnis Ihres Projekts können Sie den Maven-Build ausführen. Damit Ihre Jar-Datei auch ausführbar wird, muss eine sogenannte Manifest-Datei erzeugt werden. Maven kann diese generieren, ändern Sie in der Datei `pom.xml` die Zeichenkette »de.hfu.IHREHAUPTKLASSE« in Ihre Klasse mit der `main`-Methode, also z.B.

»de.hfu.HauptKlasse«, ohne Dateiendung. Weitere Informationen auf der Webseite: *Make The Jar Executable* <http://maven.apache.org/shared/maven-archiver/examples/classpath.html#Make>

Schritt 5

Führen Sie Ihre Jar-Datei aus: »`java -jar target/[Projektname].jar`«

Die eingegebene Zeichenkette wird in Großbuchstaben ausgegeben.

Schritt 6

Erstellen Sie ein neues Projekt in Ihrem GitHub-Account und fügen Sie dort alle Dateien (bis auf das target-Verzeichnis, welches Sie mit der `.gitignore` Datei ausschließen können) aus Ihrem Arbeitsverzeichnis hinzu.

Schritt 7

Starten Sie Eclipse und überprüfen Sie, ob die Erweiterung »Maven Integration for Eclipse« bereits installiert ist (wenn es unter Help/Installation Details.../Features auftaucht, ist es installiert). Wenn nicht, gehen Sie auf den Karteireiter »Search« im Dialog »Eclipse Marketplace...«, tippen Sie »maven« ein und klicken Sie »install« im Eintrag »Maven Integration for Eclipse«.

Schritt 8

Importieren Sie Ihr Maven-Projekt in Eclipse (File / Import... / Existing Maven Projects). Das Projekt sollte automatisch als Maven-Projekt (kleines blaues M links oben im Projekt-Icon) und als Git-Projekt (kleine Gelbe Tonne rechts unten im Projekt-Icon) erkannt werden.

Schritt 9

Ergänzen Sie Ihre Java-Klasse um einige JavaDoc-Kommentare. Um die Reports zu erstellen, müssen Sie das Build-Target »`site`« ausführen. In Eclipse richten Sie dazu ein Run-Configuration ein (Toolbar: schwarzer kleiner Pfeil nach unten neben dem grünen Ausführen-Pfeil / Run Configurations...). Maven-Build doppelklicken, rechts »Browse Workspace« und Projekt auswählen, sowie »`site`« bei »`goals`« eintragen. »Run« anklicken. Die Reports werden erstellt, sie befinden sich in `target/site/` und können mit einem Web-Browser angeschaut werden.