

Software-Architektur

Blatt 2

Prof. Dr. Oliver Braun

Fakultät für Informatik und Mathematik
Hochschule München

Letzte Änderung: 27.01.2020 21:31

Erste Schritte mit Play

Laden Sie sich den [Typesafe Activator](#) herunter. Auf den Laborrechnern unter Linux sollte er bereits installiert sein.

Um den Activator auf der Kommandozeile zu nutzen, fügen Sie den Pfad der PATH-Variable hinzu.

Um ein initiales Play/Scala-Projekt zu erzeugen geben Sie in einem Verzeichnis Ihrer Wahl folgendes ein:

```
$ activator new MoviestorePlay play-scala
```

Dieses Kommando erzeugt ein Verzeichnis `MoviestorePlay`. Wechseln Sie in das Verzeichnis `MoviestorePlay`.

Der Typesafe Activator dient nicht nur dazu Projekte aus Templates zu erstellen, sondern er kapselt auch das Scala Build Tool, kurz SBT.

Mit dem Kommando

```
$ activator
```

kommen Sie in die “Activator-Shell” und können direkt Kommandos wie `compile`, `test` oder `run` eingeben. Sie können die Kommandos aber auch direkt als Argumente übergeben, also z.B.

```
$ activator run
```

Interessant ist beispielsweise das Präfix `~`. Damit können Sie ein Kommando bei jeder Änderung eines Sourcefiles neu ausführen lassen, z.B. `~ run`.

IntelliJ bringt selbst Unterstützung für SBT mit, wenn Sie die Scala-IDE for Eclipse nutzen, müssen Sie zusätzlich ein Terminal mit dem Activator/SBT betreiben.

In IntelliJ können Sie ein SBT-Projekt direkt importieren. In Eclipse geben Sie

```
$ activator eclipse
```

ein um ein Eclipse-Projekt zu erzeugen. Danach können Sie es in Eclipse importieren.

Ein Beispiel-Projekt

Clonen Sie das Beispielprojekt <https://github.com/ob-cs-hm-edu/PizzaInitial.git> und wenn Sie sich die Git-Commits der Reihe nach ansehen, können Sie nachvollziehen wie aus dem initialen Projekt das vorliegende geworden ist.

Für Scalastyle und Scoverage müssen Sie folgenden Zeilen an die Datei `project/plugins.sbt` anfügen (inkl. Leerzeilen)

```
addSbtPlugin("org.scalastyle" %% "scalastyle-sbt-plugin" % "0.6.0")
```

```
addSbtPlugin("org.scoverage" % "sbt-scoverage" % "1.0.4")
```

Über den Activator (bzw. in IntelliJ) können Sie es laufen lassen.

Wenn Sie sich die [scalastyle-config.xml](#) herunterladen und top-level speichern und dann

```
$ activator scalastyle
```

bekommen Sie die Überprüfung, die auch mein [Jenkins](#) macht. Das Ergebnis stellt der Jenkins übrigens als “Checkstyle Trend” dar.

Mit

```
$ activator doc
```

erzeugen Sie die ScalaDoc im Verzeichnis `target/scala-2.11/api`.

Das Projekt enthält noch im Verzeichnis `test` noch zwei einfache [Specs2](#)-Tests. Diese können Sie mit

```
$ activator test
```

ausführen.

Mit

```
$ activator coverage test
```

gefolgt von

```
$ activator coverageReport
```

können Sie einen Report unter `target/scala-2.11/scoverage-report` erzeugen, der Ihnen die Test-Abdeckung zeigt. Im Jenkins-Projekt ist dies als “Scala Code Coverage” zu sehen.

Außerdem baut Jenkins noch aus der Datei `README.txt` mit [Pandoc](#) eine HTML-Seite.

Übrigens ist das einzige echte `activator`-Kommando `new`. Der Rest wird an `sbt` weitergereicht. Der Jenkins hat z.B. gar keinen `activator` und macht dann

```
$ sbt coverage test
```

```
...
```