

# Gemeinsame Entwicklungsplattformen

2016.06 2017.06  
2017-08-21

by Jens Kupferschmidt, Kathleen Neumann

*Der Abschnitt beschreibt Werkzeuge und Zugänge  
zu den Entwicklerbereichen von MyCoRe.*

## Table of contents

|   |   |
|---|---|
| 1 Git-Zugang.....                                     | 2 |
| 1.1 Integration von MyCoRe-Git in Eclipse.....        | 2 |
| 1.2 Weiteres Repositories von MyCoRe-Entwicklern..... | 3 |
| 2 Subversion-Zugang.....                              | 3 |
| 3 Nutzung von Eclipse.....                            | 3 |



## 1 Git-Zugang

Das Entwicklerteam hat sich entschlossen, die MyCoRe-Quellcodes vom eigenen Subversion-Server auf GitHub zu verschieben. Neben einer besseren Wahrnehmung von MyCoRe in der Open Source Gemeinde sollen die besseren Team-Arbeitsmöglichkeiten von Git ausgenutzt werden. Die Umstellung fand im Oktober 2016 für den aktuellen Master-Code (trunk) und das release 2016.06 statt. Die Code-Entwicklung ist dabei weiter an JIRA und BAMBOO gekoppelt.

Alle Projekte der MyCoRe Community sind unter <https://github.com/MyCoRe-Org> zu finden.

Hinweise zum Umgang mit Git von Kathleen Neumann sind hier zu finden.

### 1.1 Integration von MyCoRe-Git in Eclipse

Um das MyCoRe-Git-Repository auszuchecken und in Eclipse zu integrieren sollte zuerst *Git* auf dem lokalen Rechner installiert werden. Anschließend ist die Konfiguration entsprechend der obigen Doku zu setzen. Nun können in Eclipse folgende Aktivitäten ausgeführt werden:

- in die Git-Ansicht wechseln
- `clone a git repository` auswählen
- als Source `https://github.com/MyCoRe-Org/mycore.git` eingeben
- eigenes Git Login und Pw eintragen
- `deselect all` und dann `select master` auswählen
- Directory festlegen (z. B. `~/workspace`); origin als Bezeichner lassen
- ... und laden

Nun ist der Codestand im Git-View sichtbar. Jetzt soll der Code auch in der Java-Ansicht verfügbar sein. Dazu sind folgende Schritte notwendig.

- in die Git-Ansicht wechseln
- im *Working Tree* der ausgecheckten Daten `Import projects`
- dort `Import General Project` auswählen
- anschließend in die Java-Ansicht wechseln und mit
- `Import existing Maven projects` die gewünschten MyCoRe-Komponenten auswählen

Um vom Master auf ein bestimmtes Release zu wechseln sind folgende Schritte zu tun.

- in die Git-Ansicht wechseln
- auf dem Master rechte Maustaste `Fetch from upstream`
- da unter *Remote tracking* `checkout as new branch` auswählen

#### Anschalten der Warnung für die pom.xml

Hierfür ist die jeweilige `pom.xml` zu öffnen. Ober erscheint eine Warnung. Auf der Warnung rechte Maustaste und `Make goal update-project-license as ignored in eclipse build` markieren.

## 1.2 Weiteres Repositories von MyCoRe-Entwicklern

---

Neben dem Repository auf GitHub betreiben einige Entwickler noch eigene Repositories auf GitLab-Grundlage.

- Scientific Git des URZ der Uni Leipzig - Ansprechpartner ist Jens Kupferschmidt

## 2 Subversion-Zugang

---

Der **alte** Quellcode der MyCoRe-Kerns und einiger Anwendungen, wird auf einem Subversion-Server an der Universität Duisburg-Essen verwaltet. **Achtung, die Codestände von bereits auf Git migrierten Projekten sind veraltet und sollten nur noch zur Information genutzt werden!!!** Die Nutzung des SVN wird schrittweise nach GitHub oder anderen Git-Systemen migriert.

Aktuell gib es zwei Zugangsmöglichkeiten: über HTTP im lesenden Zugriff und über SSH mit Schreibrechten (sofern der Entwickler Zugang zum System hat [der Server wird von Frank Lützenkirchen verwaltet]).

<http://www.mycore.de/svn/>

<svn+ssh://server.mycore.de/svn/>

### MIME-Typ für alle neuen Dateien

Alle neuen Dateien, die bei MyCoRe mit Subversion verwaltet werden müssen das SVN Property `svn:mime-type` ausgefüllt haben. Bei Mime-Typ `text/*` muss zusätzlich noch da Property `svn:eol-style` beschrieben sein. Für die häufigsten Dateiarten gibt es eine Konfiguration zum herunterladen, die ans Ende der Datei `config` im Verzeichnis `.subversion` des Nutzerverzeichnis angefügt wird. Der entsprechende Bereich trägt den Namen `[auto-props]`.

## 3 Nutzung von Eclipse

---

Die Entwicklungsumgebung Eclipse leistet nicht nur hilfreiche Dienste bei der Formatierung des Java-Codes. Mit ihr kann auch die Syntaxprüfung der Java-Klasse wie auch ihre Einbettung in das Gesamtprojekt leicht überwacht werden. Dazu sind einige Installationen und Einstellungen erforderlich. Diese werden dann sowohl für die Kernanwendung wie auch für die Applikationen genutzt.

Zuerst ist die Eclipse-Entwicklungsumgebung zu installieren. Von den Entwicklern wird als Basis die **Eclipse IDE for Java EE Developers** genutzt und auch empfohlen.

Zur Bearbeitung von I18N-Sprachdateien hat sich der ResourceBundle-Editor (besonders für Sprachen außerhalb des Lateinischen Alphabetes) bewährt. Weiterhin sollte nach der Installation die Code-Style-Definition von MyCoRe importiert werden, mehr dazu im Abschnitt Java-Code Formatierung.

Um eine direkte Jira-Anbindung im Eclipse zu realisieren, muss der Jira-Connector über den Eclipse-Marketplace installiert werden. Danach muss das Jira über MyLyn eingebunden werden.

Für weitere Einstellungen im Editor kann AnyEdit empfohlen werden. Damit kann beispielsweise konfiguriert werden, dass überzählige Leerzeichen z.B. am Zeilenende aus dem Code entfernt werden.