

Grundkommandos

forge	Starte Forge
command-list	Zeigt die verfügbaren Kommandos an
man <command>	Erläutert das angefragte Kommando
project-new	Erstellt interaktiv ein neues Projekt und wechselt in dessen Kontext
cd ..	Geht einen Kontext zurück (auf dem Stack)
cd <suite-m>	Betrifft einen neuen Kontext z.B. ein Unterverzeichnis, ein Projekt oder eine Datei
exit	Beendet das Programm

Forge Download: <http://forge.jboss.org/download>
 Systemvoraussetzung: Java7 und gesetzte JAVA_HOME Variable
 Installation unter Linux: `curl http://forge.jboss.org/sh | sh`

Container und Instanzverwaltung

Mangels Doku habe ich mit ein paar kleinen Analysen versucht herauszufinden wie die Container und Instanzen verwaltet werden. Dabei habe ich einfach in verschiedenen Terminalfenstern forge gestartet und festgestellt:

- * Jedes Terminal startet einen eigenen Forge Container
- * Der erste gestartete Forge Container scheint als Template bei der Initialisierung der anderen Forge Container zu dienen - er wird nach Beenden von Forge scheinbar mit dem aktuellen Stand gespeichert und dient anschließend wieder als Template (evtl. durch Systemsperrern auf Dateien in denen der Zustand gesichert wird - unklar wo der Zustand gesichert wird)
- * Kommandos scheinen als Singleton im Container gestartet zu werden und nach Beendigung des Kommandos weiter zu existieren. Ein erneuter Aufruf eines Kommandos kennt daher schon Belegungen interner Felder z.B. per Prompt injizierter Werte

Dynamische Vorbelegung der Auswahl

```
projectDirName.setDefaultVa
new Callab le< -
String >() {
@Override
public
String call() {
if
(artifactId.getValue() == null) {
return " myp roj ect ";
}
return artifa ctI d.g etV alue();
}
});
```

Anzuwenden im eigenen UICommand in der Methode:

```
@Override
public void initializeUI(UIBuilder builder) throws Exception {
```

Eigenes Kommando definieren

```
public class RadesNewLibraryProject extends
AbstractUICommand implements UICommand {
@Inject
protected ResourceFactory resourceFactory;
@Inject
@WithAttributes(label = " Group ID:"
, required = true, defaultV alue = " com.gi thu -
b.m yUs ern ame ")
protected UIInput<String> groupId;
@Override
public UICommand Metadata getMet ada -
ta( UIContext context) {
return Metada ta.f or Com man -
d(R ade sNe wLi bra ryP roj ect.class)
.name( " my- pro -
jec t-g ene rat or")
.description -
("Cr eates a RADES lib projec t.")
.category (Ca -
teg ori es.c re ate ("Pr oje ct/ Gen era tio n"));
}
@Override
```



By **Huluvu424242**
(FunThomas424242)

Not published yet.
 Last updated 20th November, 2017.
 Page 1 of 2.

Sponsored by **ApolloPad.com**
 Everyone has a novel in them. Finish
 Yours!
<https://apollopad.com>

Eigenes Kommando definieren (cont)

```
public boolean isEnabled (UI Context
context) {
    return true; // logik ob commando
verfügbar oder nicht
}
@Override
public void initializeUI (UI Builder
builder) throws Exception {
    buildera.add(groupId);
}
@Override
public Result execute (UIExecutionCo -
ntext context) throws Exception {
    final UIContext uiContext =
context.getUIContext();
    final UIOutput log = uiCont -
ext.getProvider().getOutput();
    final UIPrompt prompt = contex -
t.getPrompt();
    return Results
        .success("Kommando
'rades -new-libproject' wurde erfolg reich
ausgeföhrt.");
}
}
```

Klasse oben zeigt die wichtigsten Definitionen.



By [Huluvu424242](#)
([FunThomas424242](#))

cheatography.com/funthomas424242/
github.com/Huluvu424242

Not published yet.
Last updated 20th November, 2017.
Page 2 of 2.

Sponsored by [ApolloPad.com](#)
Everyone has a novel in them. Finish
Yours!
<https://apollopad.com>