

LU16a - Die Zwölf-Faktoren-App

In der Praxis haben sich verschiedene Vorgehen bewährt. Im Folgenden möchte ich die Grundsätze der Zwölf-Faktoren-App (<https://12factor.net/de/>) vorstellen.

1. Codebase

Eine im Versionsmanagementsystem verwaltete Codebase, viele Deployments

□ Empfehlung: Git △ Auf Regelmässigkeit achten

2. Abhängigkeiten

Abhängigkeiten explizit deklarieren und isolieren

□ Empfehlung: Maven(pom.xml) oder Gradle(build.gradle) bei Java, Pip(requirement.txt) bei Python, Npm/Yarn(package.json) bei JavaScript/TypeScript

3. Konfiguration

Die Konfiguration in Umgebungsvariablen ablegen

□ Empfehlung: application.properties und application.yml bei Spring

4. Unterstützende Dienste

Unterstützende Dienste als angehängte Ressourcen behandeln

□ Datenbank-Adresse nicht hartcoden. Empfehlung: .properties-Datei

5. Build, release, run

Build- und Run-Phase strikt trennen

! Das Thema wird im nächsten Semester behandelt.

6. Prozesse

Die App als einen oder mehrere Prozesse ausführen

□ Zustandslosigkeit durch Persistierung sicherstellen

7. Bindung an Ports

 Dienste durch das Binden von Ports exportieren

□ Ports in den Konfigurationen festlegen

8. Nebenläufigkeit

Mit dem Prozess-Modell skalieren

□ Aufgrund Zustandslosigkeit kann Skaliert werden, ohne manuelle Thread-Manipulation

9. Einweggebrauch

Robuster mit schnellem Start und problemlosen Stopp

□ Stopp durch Spring-Framework sichergestellt

10. Dev-Prod-Vergleichbarkeit

Entwicklung, Staging und Produktion so ähnlich wie möglich halten

! Das Thema wird im nächsten Semester behandelt.

11. Logs

Logs als Strom von Ereignissen behandeln

□ SLF4J o. Ä. nutzen korrekt Streams, die man nach Bedarf umleiten kann.

12. Admin-Prozesse

Admin/Management-Aufgaben als einmalige Vorgänge behandeln

□ Separate Skripte sind auf ein Minimum beschränkt.

From:

<https://wiki.bzz.ch/> - BZZ - Modulwiki

Permanent link:

<https://wiki.bzz.ch/de/modul/ffit/3-jahr/java/learningunits/lu16/a?rev=1767567896>

Last update: 2026/01/05 00:04



