

Maven:

Apache Maven – Build Management Tool (wie make für C)

Maven unterstützt die Softwareerstellung:

- Abhängigkeiten auflösen
- Projektstrukturen prüfen
- Kompilieren
- Testen
- Verpacken (Kompilate, nicht Kompilate → .jar)
- Install
- ...

Folgt man den Maven vorgegebenen Standards, braucht man für die meisten Aufgaben des Build-Managements nur sehr wenige Konfigurationseinstellungen zu hinterlegen!

Konfigurationsdatei **pom.xml** **zwingend** nötig für: *mvn spring-boot:run*

Spring Framework

Applikationsframework mit umfassender Infrastruktur, das das entwickeln von Java Applikationen erleichtert.

Ziele:

- Abhängigkeiten im Code reduzieren (loose coupling)
- Fokus der Entwickler auf Anwendungsentwicklung
- Entwicklung basiert auf POJO'S (Plain Old Java Objects)

POJO's: Plain Old Java Objects

- Java Klassen ohne komplexe Strukturen
- Erben von keiner Elternklasse
- Implementieren kein Interface

Pattern: MVC – Model View Controller

- Präsentationsschicht – Java Server Faces, Spring MVC
- Anwendungsschicht – Java Beans
- Persistenzschicht – Java Persistence API

Präsentationsschicht:

Java Server Faces – Framework zum erstellen einer komponentenbasierten GUI, für Webapplikationen

Anwendungsschicht:

Java Beans – Serialisierbare Objekte mit Eigenschaften (private member variablen, getter, setter)
Auch für GUI Elemente (Buttons, Checkboxes...)
Haben public non argument constructor

Persistenzschicht:

JPA – Java Persistence API (Spring Data JPA)
Übertragung von Java Objekten zu Datenbankeinträgen
Laufzeitobjekte über Sitzungen hinaus speichern
Speichern und laden von Daten in/aus relationalen Datenbanken
Speichern von POJO'S in memory based Datenbanken