

Apache Maven

Prof. dr Miloš Milić

Sadržaj

- Uvod
- Karakteristike
- Prednosti
- Struktura projekta
- POM datoteka
- Način rada
- Životni ciklus

Uvod

- Upravljanje i organizacija projekta
- Životni ciklus projekta
- Jedan od alata koji se može koristiti za lakšu izgradnju i upravljanje projektom

Funkcije

- Izrada programskog koda i generisanje dokumentacije iz programskog koda
- Testiranje softvera
- Upravljanje zavisnostima
- Podrška različite vrste projekata
- Pakovanje kompajliranog softvera u odgovarajuće datoteke (npr. jar, war, zip)
- Mogućnost postavljanja artefakta u serverski ili lokalni repozitorijum

Prednosti

- Veoma često korišćene funkcionalnosti
- Jednostavnije upravljanje projektom
- Zasnovan na standardima
- Omogućava visok stepen *automatizacije* procesa razvoja softvera
 - Softver inženjer se „oslobađa“ velikog broja koraka
- Korišćenje različitih dodataka
- Velika podrška od strane industrije

Struktura projekta

- my-app
 - |-- pom.xml
 - `-- src
 - |-- main
 - | `-- java
 - | `-- com
 - | `-- mycompany
 - | `-- app
 - | `-- App.java
 - `-- test
 - `-- java
 - `-- com
 - `-- mycompany
 - `-- app
 - `-- AppTest.java

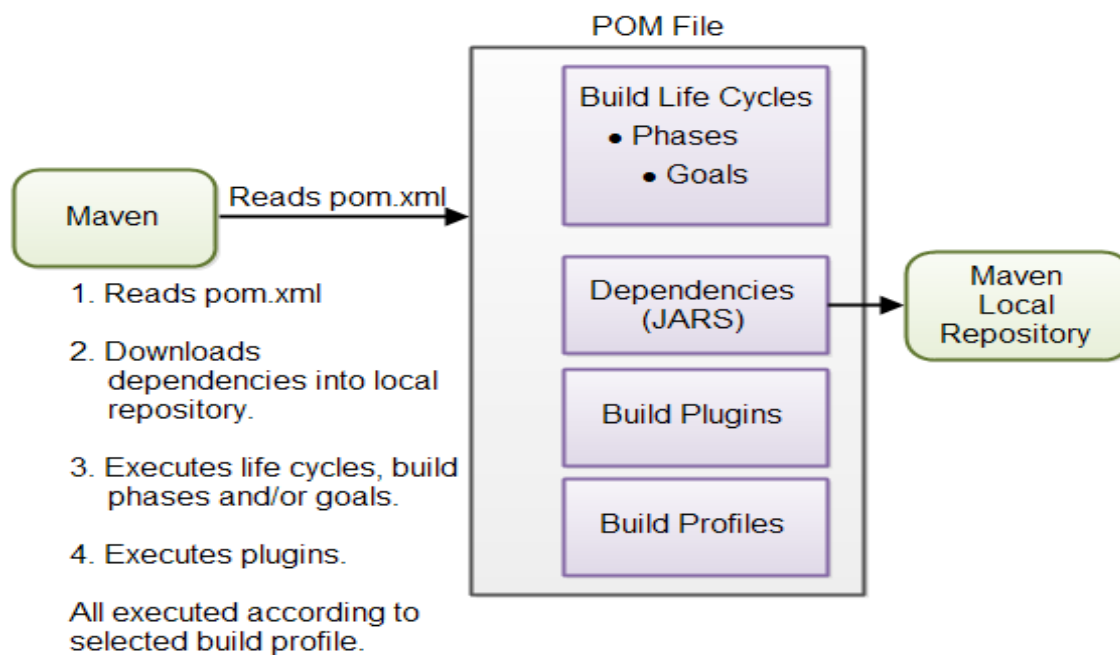
POM datoteka

- POM (engl. Project Object Model) datoteka
- Sadrži reference na programski kod, testove, resurse programa, zavisnosti uključene u projekat i sl.
- Nalazi se u osnovnom direktorijumu projekta

POM datoteka

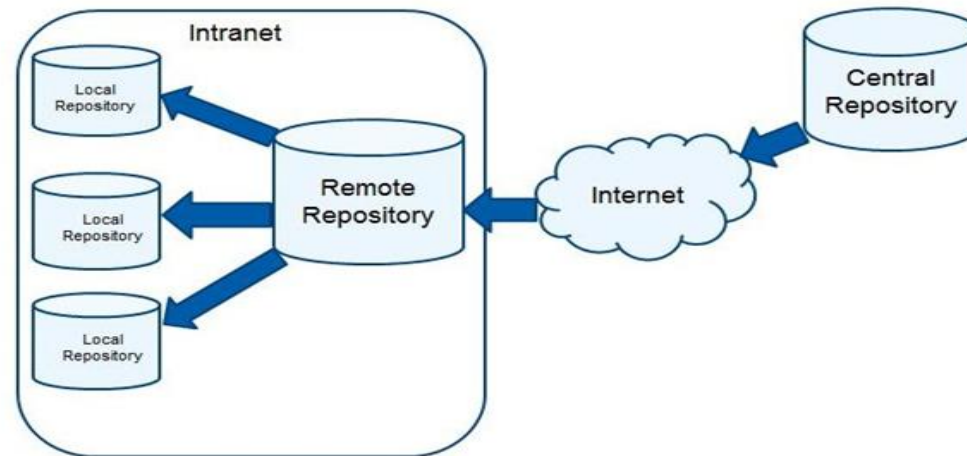
- `<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"`
- `xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">`
- `<modelVersion>4.0.0</modelVersion>`
- `<groupId>com.mycompany.app</groupId>`
- `<artifactId>my-app</artifactId>`
- `<version>1.0-SNAPSHOT</version>`
- `<name>my-app</name>`
- `<!-- FIXME change it to the project's website -->`
- `<url>http://www.example.com</url>`
- `<properties>`
- `<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>`
- `<maven.compiler.source>1.7</maven.compiler.source>`
- `<maven.compiler.target>1.7</maven.compiler.target>`
- `</properties>`
- `<dependencies>`
- `</dependencies>`
- `</project>`

Način rada



Način rada

- Provera i upravljanje zavisnostima
 - Tranzitivne zavisnosti
- Preuzimanje zavisnosti i smeštanje u odgovarajući repozitorijum



Životni ciklus

- Životni ciklus izgradnje je lista imenovanih faza koje se koriste kako bi se napravio redosled izvršavanja zadataka
- Pojednostavljeni prikaz standardnog životnog ciklusa obuhvata sledeće faze
 - process-resources
 - compile
 - process-test-resources
 - test-compile
 - test
 - package
 - install
 - deploy

Životni ciklus

- Zadaci dodataka se mogu vezati za različite faze životnog ciklusa
- Na primer, podrazumevano, zadatak “*compiler:compiler*” je vezan za fazu kompajliranja, dok je zadatak “*surefire:test*” vezan za test fazu
- Kada se izvrši komanda **mvn test**, Maven alat će izvršiti sve zadatke vezane za svaku fazu do faze *test*
- Maven, takođe ima standardni životni ciklus za osvežavanje (tzv. *clean*) projekta i generisanje sajta projekta
- Zahvaljujući standardnom životnom ciklusu, moguće je izgraditi, testirati i instalirati svaki Maven projekat koristeći komandu **mvn install**

Apache Maven

Prof. dr Miloš Milić