

Programming Advanced java

MAVEN



Overzicht

1. Wat is maven?
2. Installeren en configureren
3. Dependencies
4. Lifecycle
5. Repositories
6. Executable package
7. IDE integratie

Wat is maven?

- Projectmanagement-tool
 - Gestandaardiseerde wijze om project op te zetten en te beheren
 - Plugin framework
- Meer dan een build tool (~ ANT)
 - compile
 - test
 - package in jar
 - ...
- Rapportage / documentatie
- Ontstaan uit nood van Dependency management

Overzicht

1. Wat is maven?
- 2. Installeren en configureren**
3. Dependencies
4. Lifecycle
5. Repositories
6. Executable package
7. IDE integratie

Install

- CLI
 - java -version
 - Javac -version
 - mvn –version

Vraag

Hoe zet ik een java project op?

1ste project

Via Maven Archetypes (~templates)

<https://maven.apache.org/guides/introduction/introduction-to-archetypes.html>

- mvn archetype:generate
 - DarchetypeGroupId=org.apache.maven.archetypes
 - DarchetypeArtifactId=maven-archetype-quickstart
 - groupId: be.pxl.student
 - artifactId: HelloWorld
- Pom.xml
- Folder structuur
- 1st commands
 - mvn package
 - mvn clean

```
project
  +-src
    +-main
      +-java
      +-resources
    +-test
      +-java
      +-resources
  +-target
    +-classes
    +-test-classes
pom.xml
```

Logging framework

- SLF4J
 - <https://www.slf4j.org/manual.html>
- LOG4J
 - <https://logging.apache.org/log4j/2.x/articles.html>
 - <https://logging.apache.org/log4j/2.x/manual/configuration.html>

Logging framework

			x: Visible				
	FATAL	ERROR	WARN	INFO	DEBUG	TRACE	ALL
OFF							
FATAL	x						
ERROR	x	x					
WARN	x	x	x				
INFO	x	x	x	x			
DEBUG	x	x	x	x	x		
TRACE	x	x	x	x	x	x	
ALL	x	x	x	x	x	x	x

Overzicht

1. Wat is maven?
2. Installeren en configureren
- 3. Dependencies**
4. Lifecycle
5. Repositories
6. Executable package
7. IDE integratie

Dependencies

- Maven Central Repository
 - <http://search.maven.org/>
- Scope
- Transitiviteit
 - <optional>
- Versiereeksen

Dependencies - scope

Scope	Omschrijving
compile	De afhankelijkheid wordt in alle omstandigheden aan het <i>classpath</i> toegevoegd en kunnen mee verpakt worden met het <i>artifact</i> . Dit is tevens de standaardinstelling indien de <i>tag</i> achterwege wordt gelaten.
provided	De afhankelijkheid wordt toegevoegd aan het <i>classpath</i> tijdens de compilatie maar er wordt verondersteld dat ze tijdens de uitvoering standaard aanwezig zijn. De JAR-bestanden worden niet mee verpakt. Deze <i>scope</i> wordt bijvoorbeeld gebruikt bij afhankelijkheid van de <i>servlet API</i> .
runtime	De afhankelijkheid is niet vereist voor de compilatie maar is wel nodig tijdens het testen en de uitvoering. Voorbeeld is een JDBC-driver die dynamisch geladen wordt.
test	De afhankelijkheid is enkel nodig tijdens het compileren en uitvoeren van de testen.
system	Hetzelfde als <code>runtime</code> maar hier dient men ook een pad op te geven naar het JAR-bestand op het lokale systeem. Dit wordt enkel gebruikt voor afhankelijkheden van <i>native bibliotheken</i> .

Dependencies - oefeningen

- Opdracht 3 – p16
 - Start van archetype quickstart
 - Voeg log4j dependency toe
 - Gebruik logger.debug

Overzicht

1. Wat is maven?
2. Installeren en configureren
3. Dependencies
- 4. Lifecycle**
5. Repositories
6. Executable package
7. IDE integratie

Lifecycles

- default (p23)
 - mvn [goals] [phase]
- clean (p27)
 - mvn clean
- site (p27)
 - mvn site
- Lifecycle → x aantal Phases → x aantal goals op basis van packaging type
 - Default: <packaging>jar</packaging>
- Phase
 - *{plugin name}* : *{plugin goal}*

Lifecycles - oefening

- Opdracht 6 – javadoc:Javadoc (p28)
- Maven plugins: <https://maven.apache.org/plugins/>
- Javadoc plugin: <https://maven.apache.org/plugins/maven-javadoc-plugin/>

Overzicht

1. Wat is maven?
2. Installeren en configureren
3. Dependencies
4. Lifecycle
- 5. Repositories**
6. Executable package
7. IDE integratie

Repositories

- global repository
 - remote repository / maven central repository
 - <http://repo1.maven.org/maven2>
 - <http://search.maven.org>
 - <https://mvnrepository.com/>
- internal repository
 - nexus / artifactory
 - mvn deploy
- local repository
 - `~/.m2/repository`
 - mvn install

Repositories - internal

- pom.xml
 - <distributionManagement>
 - <repository>
 - <**id**>
 - ...
- ~/.m2/settings.xml
 - <servers>
 - <server>
 - <**id**>
 - ...

Properties

- maven properties
 \${project.artifactId}
- omgevingsvariabelen
 \${env.USERNAME}
- systeem properties
 \${java.version}
 \${file.separator}
- zelf gedefinieerde properties

```
<properties>
    <my.own.property>something</my.own.property>
</properties>
```

 \${my.own.property}

Properties - Resource filtering

resource filtering = properties gebruiken buiten de pom.xml

- Phase: process-resources

```
<build>
  <resources>
    <resource>
      <directory>src/main/resources</directory>
      <filtering>true</filtering>
    </resource>
  </resources>
</build>
```

Properties

- Opdracht 8 p35

Overzicht

1. Wat is maven?
2. Installeren en configureren
3. Dependencies
4. Lifecycle
5. Repositories
- 6. Executable package**
7. IDE integratie

Vraag

Hoe start ik mijn java applicatie op buiten mijn IDE?

Hoe verdeel ik mijn applicatie naar derden?

Executable package

Run on command line

- Via classes
- Via jar op classpath
- Via mvn command
 - mvn exec:java -Dexec.mainClass="be.pxl.student.App"
- Via executable jar file
 - maven-jar-plugin
 - opdracht 9 p37

Overzicht

1. Wat is maven?
2. Installeren en configureren
3. Dependencies
4. Lifecycle
5. Repositories
6. Executable package
7. IDE integratie