



Noël Vaes

Java Trainer & Consultant



**Java Advanced 2**  
**PXL-versie**  
**Academiejaar 2019-2020**

Roode Roosstraat 5  
3500 Hasselt  
België

+32 474 38 23 94  
noel@noelvaes.eu  
www.noelvaes.eu

Vrijwel alle namen van software- en hardwareproducten die in deze cursus worden genoemd, zijn tegelijkertijd ook handelsmerken en dienen dienovereenkomstig te worden behandeld.

Alle rechten voorbehouden. Niets uit deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand of openbaar worden gemaakt in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of op enige andere manier, zonder voorafgaande schriftelijke toestemming van de auteur. De enige uitzondering die hierop bestaat, is dat eventuele programma's en door de gebruiker te typen voorbeelden mogen worden ingevoerd opgeslagen en uitgevoerd op een computersysteem, zolang deze voor privédoeleinden worden gebruikt, en niet bestemd zijn voor reproductie of publicatie.

Correspondentie inzake overnemen of reproductie kunt u richten aan:

Noël Vaes  
Roode Roosstraat 5  
3500 Hasselt  
België

Tel: +32 474 38 23 94

noel@noelvaes.eu  
www.noelvaes.eu

Ondanks alle aan de samenstelling van deze tekst bestede zorg, kan de auteur geen aansprakelijkheid aanvaarden voor eventuele schade die zou kunnen voortvloeien uit enige fout, die in deze uitgave zou kunnen voorkomen.

08/02/2020

Copyright© 2020 Noël Vaes



# Inhoudsopgave

<b>Hoofdstuk 1: Maven</b>	<b>8</b>
1.1. Inleiding	8
1.2. <i>Maven</i> installeren en configureren	8
1.3. Mijn eerste <i>Maven</i> -project	8
1.4. Project Object Model	13
1.5. Dependencies	15
1.5.1. Scope	19
1.5.2. Transitiviteit van afhankelijkheden	22
1.5.3. Versiereeksen en conflictoplossing	23
1.6. <i>Phases</i> en <i>Goals</i>	24
1.6.1. Default lifecycle	25
1.6.2. Clean lifecycle	29
1.6.3. Site lifecycle	30
1.7. Repositories	32
1.7.1. Globale Repository	32
1.7.2. Lokale Repository	32
1.7.3. Bedrijfs-repository	32
1.8. <i>Properties</i> en <i>resource filtering</i>	36
1.8.1. <i>Maven properties</i>	36
1.8.2. Omgevingsvariabelen	37
1.8.3. <i>Systeem-properties</i>	37
1.8.4. Zelf gedefinieerde <i>properties</i>	37
1.8.5. Resource filtering	38
1.9. Uitvoerbare JAR-bestanden	40
1.10. JAR-bestanden voor broncode en documentatie	42
1.11. Integratie in een IDE	44
1.11.1. Integratie in Eclipse	44
1.11.2. Integratie in IntelliJ IDEA	48
<b>Hoofdstuk 2: JUnit</b>	<b>53</b>
2.1. Inleiding	53
2.2. Mijn eerste test	53
2.3. Integratie in de ontwikkelomgeving	54
2.3.1. Integratie met <i>Maven</i>	54
2.4. De levenscyclus van een testklasse	58
2.5. Weergavenaam	60
2.6. Parameters	61
2.7. Testen uitschakelen	62
2.8. Assert-methoden	62
2.9. Grenzen testen	63
2.10. Exceptions testen	64
2.11. Stub- en mock-objecten	65
2.12. Tijdsbeperking van testen	73
2.13. Testen met herhaling	74
<b>Hoofdstuk 3: JDBC</b>	<b>76</b>
3.1. Inleiding	76
3.2. De databaseserver	77
3.3. Een verbinding maken met een database	77
3.3.1. De database- <i>driver</i> laden	77
3.3.2. Een connectie maken	78
3.4. SQL-commando's gebruiken	81
3.4.1. Een statement creëren	81
3.4.2. Gegevens uit een database opvragen	82



3.4.3. Gegevens wijzigen in een database.....	87
3.4.4. Gegevens toevoegen aan de database.....	88
3.4.5. Prepared statements.....	89
3.5. Transacties.....	90
3.6. Grote objecten.....	93
3.6.1. Het lezen van grote objecten.....	93
3.6.2. Het schrijven van grote objecten.....	93
3.7. Data Access Objects (DAO).....	94
<b>Hoofdstuk 4: Webcomponenten.....</b>	<b>98</b>
4.1. Webcontainers.....	98
4.1.1. Inleiding.....	98
4.1.2. Statische versus dynamische webpagina's.....	98
4.1.3. Java Enterprise Edition.....	99
4.1.4. Java-webcontainers.....	100
4.1.5. <i>Apache Tomcat/TomEE</i> .....	101
4.1.5.1. Installatie.....	101
4.1.5.2. Integratie in Eclipse.....	101
4.1.5.3. Integratie in <i>IntelliJ</i> .....	104
4.1.6. Het HTTP-protocol.....	105
4.1.6.1. Request message.....	106
4.1.6.2. Response message.....	107
4.1.6.3. HTTP/2.....	108
4.2. Java-webapplicaties.....	108
4.2.1. Inleiding.....	108
4.2.2. Webapplicatie-mappenstructuur.....	108
4.2.3. Webapplicatie-configuratie.....	111
4.2.4. WAR-bestanden.....	112
4.2.5. De context van een webapplicatie.....	113
4.3. Servlets.....	113
4.3.1. Inleiding.....	113
4.3.2. Klassenhierarchie voor servlets.....	114
4.3.3. Mijn eerste <i>servlet</i> : "Hello World".....	116
4.3.3.1. De <i>servlet</i> -code schrijven en compileren.....	116
4.3.3.2. De <i>servlet</i> configureren.....	117
4.3.3.3. URL-patronen.....	119
4.3.4. De levenscyclus van een <i>servlet</i> .....	120
4.3.4.1. De methode <i>init()</i> en de initialisatieparameters.....	123
4.3.4.2. De methode <i>destroy()</i> .....	125
4.3.4.3. Service-methoden.....	126
4.3.4.3.1. De methode <i>doGet()</i> .....	127
4.3.4.3.2. De methode <i>doPost()</i> .....	129
4.3.4.3.3. Karaktercodering.....	131
4.3.4.4. Overige methoden.....	131
4.3.5. Scope-objecten.....	132
4.3.5.1. <i>Request</i> en <i>Response</i> .....	133
4.3.5.2. Sessies.....	135
4.3.5.2.1. De sessiestatus bijhouden.....	135
4.3.5.2.2. De implementatie van sessies.....	137
4.3.5.2.3. Levensduur van een sessie.....	139
4.3.5.2.4. Session event handling.....	140
4.3.5.3. De <i>servlet context</i> .....	141
4.3.5.3.1. Attributen van de <i>servlet context</i> .....	143
4.3.5.3.2. Parameters van de <i>servlet context</i> .....	144
4.3.5.3.3. <i>Events</i> van de <i>servlet context</i> .....	145
4.3.5.3.4. Resources uit de webapplicatie gebruiken.....	146
4.3.6. Insluiten, doorsturen en omleiden.....	148
4.3.6.1. Dynamisch insluiten ( <i>include</i> ).....	148



4.3.6.2. Dynamisch doorsturen ( <i>forward</i> ).....	150
4.3.6.3. Omleiden ( <i>redirect</i> ).....	151
4.3.7. File upload.....	152
4.3.8. Multithreading.....	154
4.3.9. Cookies.....	154
4.3.10. Filters.....	157
4.3.11. Beveiliging van webapplicaties.....	162
4.3.11.1. Authenticatie.....	162
4.3.11.1.1. Basic Authentication.....	164
4.3.11.1.2. Digest authentication.....	164
4.3.11.1.3. Formulier-gebaseerde authenticatie.....	165
4.3.11.1.4. HTTPS Client Certificate.....	166
4.3.11.2. Autorisatie.....	166
4.3.11.2.1. Configuratie via web.xml.....	167
4.3.11.2.2. Configuratie via annotaties.....	168
4.3.11.2.3. Programmatische beveiliging.....	170
4.3.11.3. Encryptie.....	172
4.3.12. Foutafhandeling.....	174
4.3.13. RESTful Web Services.....	175
4.3.13.1. Inleiding: <i>Web API</i> .....	175
4.3.13.2. REST.....	177
4.3.13.3. <i>Web Services</i> volgens de REST-architectuur.....	178
4.3.13.3.1. URL's.....	178
4.3.13.3.2. URL templates.....	179
4.3.13.3.3. Methoden.....	179
4.3.13.3.4. Representaties van resources.....	182
4.3.13.3.5. Status-codes.....	183
4.3.13.3.6. Verwijzingen.....	185
4.3.13.3.7. Request parameters.....	186
4.3.13.3.8. Documentatie van een REST API.....	186
4.4. <i>RESTful Web Services</i> met JAX-RS.....	187
4.4.1. Configuratie van JAX-RS.....	187
4.4.2. Domeinmodel en DAO.....	188
4.4.3. Rest Endpoint.....	191
4.4.4. Client-toepassingen.....	195
4.4.4.1. Browser.....	195
4.4.4.2. Java.....	196
4.4.4.3. HTML en Ajax.....	198
4.4.5. URL's.....	201
4.4.5.1. URL-templates en padvariabelen.....	201
4.4.6. HTTP-methoden.....	202
4.4.6.1. GET.....	203
4.4.6.2. POST.....	205
4.4.6.3. PUT.....	206
4.4.6.4. PATCH.....	207
4.4.6.5. DELETE.....	209
<b>Hoofdstuk 5: JPA - Hibernate.....</b>	<b>210</b>
5.1. Object-Relational Mapping.....	210
5.1.1. Inleiding.....	210
5.1.2. Enterprise JavaBeans 2.0.....	211
5.1.3. Hibernate – TopLink.....	212
5.1.4. Java Persistence API.....	212
5.2. Installatie & configuratie.....	213
5.2.1. Installatie van JPA met <i>Hibernate</i> .....	213
5.2.2. Configuratie van JPA.....	214
5.3. Mijn eerste <i>entity</i> -klasse.....	217
5.3.1. Daar komt hij weer .....	217



5.3.2. De <i>entity</i> -klasse.....	218
5.3.3. De <i>Persistence Unit</i> .....	219
5.3.4. Het hoofdprogramma.....	220
5.4. Persistente objecten.....	222
5.4.1. <i>Entity</i> -klassen.....	222
5.4.2. Primary keys.....	223
5.4.2.1. Enkelvoudige <i>primary key</i> .....	224
5.4.2.2. Samengestelde <i>primary key</i> .....	224
5.4.2.3. Autogenerated primary keys.....	227
5.4.3. De identiteit van objecten.....	229
5.4.3.1. Object identity.....	229
5.4.3.2. Object equality.....	229
5.4.3.3. Database identity.....	229
5.4.4. <i>Field access</i> versus <i>property access</i> .....	229
5.5. Werken met <i>entity</i> -objecten.....	231
5.5.1. Inleiding.....	231
5.5.2. De <i>Entity Manager</i> .....	232
5.5.3. De <i>Persistence Context</i> .....	233
5.5.4. Transacties.....	236
5.5.4.1. ACID transacties.....	236
5.5.4.2. JDBC transacties.....	236
5.5.4.3. JPA-transacties.....	237
5.5.4.4. Transacties en de <i>persistence context</i> .....	238
5.5.5. Mogelijkheden van de <i>Entity Manager</i> .....	240
5.5.5.1. Objecten bewaren.....	240
5.5.5.2. Objecten opzoeken.....	240
5.5.5.3. Objecten aanpassen.....	241
5.5.5.4. Objecten verwijderen.....	241
5.5.5.5. Objecten verversen.....	241
5.5.5.6. Objecten wegschrijven.....	241
5.5.5.7. Objecten verwijderen uit de <i>persistence context</i> .....	242
5.5.5.8. Overige methoden.....	242
5.6. Het domeinmodel.....	245
5.6.1. Inleiding.....	245
5.6.2. Configuratie van tabellen en kolommen.....	246
5.6.2.1. Tabelnamen.....	246
5.6.2.2. Kolommen.....	247
5.6.2.3. Secundaire tabellen.....	248
5.6.3. Mapping van datatypes.....	249
5.6.3.1. Primary keys.....	249
5.6.3.2. Enkelvoudige datatypes.....	249
5.6.3.2.1. Primitieve datatypes.....	250
5.6.3.2.2. Datums en tijden.....	250
5.6.3.2.3. Het opsommingstype.....	251
5.6.3.2.4. Grote objecten.....	252
5.6.3.2.5. Speciale datatypes.....	252
5.6.3.2.6. Eigen conversies.....	252
5.6.3.2.7. Velden uitsluiten en virtuele velden.....	253
5.6.4. Relaties tussen entiteiten.....	254
5.6.4.1. Soorten relaties.....	254
5.6.4.2. One to one.....	255
5.6.4.2.1. Het <i>cascadetype</i> .....	258
5.6.4.2.2. Het <i>fetch</i> -type.....	258
5.6.4.2.3. Bidirectionele mapping.....	259
5.6.4.2.4. Orphan Removal.....	260
5.6.4.3. <i>One to many</i> en <i>many to one</i> .....	261
5.6.4.3.1. <i>Fetch type</i> in een to-many-relatie (1+n-probleem).....	264
5.6.4.3.2. Geordende en gesorteerde verzamelingen.....	264



5.6.4.4. Many to many.....	265
5.7. Zoekopdrachten.....	271
5.7.1. Inleiding.....	271
5.7.2. Query API en JPQL.....	272
5.7.2.1. De Query API.....	272
5.7.2.2. Named queries.....	273
5.7.2.3. JPQL.....	275
5.7.2.3.1. Onderdelen van een zoekopdrachten.....	275
5.7.2.3.2. Selectie van de entiteiten (FROM-clausule).....	276
5.7.2.3.3. Projectie van de resultaten (SELECT-clausule).....	276
5.7.2.3.3.1. Selectie van entity-objecten.....	276
5.7.2.3.3.2. Selectie van attributen.....	277
5.7.2.3.3.3. Selectie van ingesloten objecten.....	277
5.7.2.3.3.4. Selectie van gerelateerde objecten.....	278
5.7.2.3.3.5. Aggregatiefuncties.....	278
5.7.2.3.3.6. Scalaire uitdrukkingen.....	279
5.7.2.3.4. De restrictie van de zoekoperatie (WHERE-clausule).....	279
5.7.2.3.4.1. Parameters in de zoekopdracht.....	279
5.7.2.3.4.2. Letterlijke waarden.....	279
5.7.2.3.4.3. Operatoren.....	280
5.7.2.3.4.4. Functies.....	281
5.7.2.3.4.5. Sortering van resultaten.....	282
5.7.2.3.5. Binnen relaties zoeken: joins.....	282
5.7.2.3.5.1. INNER JOIN.....	283
5.7.2.3.5.2. FETCH JOIN.....	283



## Hoofdstuk 1: Maven

### 1.1. Inleiding

*Maven* is een projectmanagement-*tool*. Het is een *tool* waarmee men een Java-project op een gestandaardiseerde wijze kan vormgeven en beheren. Dit behelst onder andere het *builden* (compileren, JAR maken enzovoort) van een project maar ook het genereren van rapporten en documentatie, het maken van een website enzovoort. *Maven* is dus meer dan een *build tool* zoals *ANT*, maar alle mogelijkheden van een *build tool* zijn wel voorzien.

In de volgende paragrafen zullen we stap voor stap de mogelijkheden van *Maven* verkennen aan de hand van praktische voorbeelden.

We zullen *Maven* hier behandelen in combinatie met JDK 11.

### 1.2. Maven installeren en configureren

*Maven* is een *open-source*-project van *Apache* en is te vinden op volgende website: <http://maven.apache.org>. *Maven* kan geïnstalleerd worden door het bestand **apache-maven-3.x.y-bin.zip** af te halen en uit te pakken op het lokale systeem.

#### Opdracht 1: Maven installeren

- Haal het bestand **apache-maven-3.x.y-bin.zip** van de website <http://maven.apache.org>.
- Pak dit bestand uit op je lokale systeem, bijvoorbeeld in **C:\Program Files\**
- Voeg de volgende omgevingsvariabele toe aan het besturingssysteem:

```
MAVEN_HOME="C:\Program Files\apache-maven-3.x.y"
```

- Voeg tevens de plaats van *Maven* toe aan de variabele `PATH` zodat we *Maven* kunnen uitvoeren vanop de commandolijn:

```
PATH=...;%MAVEN_HOME%\bin
```

- Zorg er tevens voor dat de omgevingsvariabele `JAVA_HOME` verwijst naar de installatie van de JDK (en niet naar de JRE).
- Open een commandowindow en voer het volgende commando uit:

```
mvn -version
```

```
Opdrachtprompt
C:\Users\info>mvn -version
Apache Maven 3.6.0 (97c98ec64a1fdfee7767ce5fffb20918da4f719f3; 2018-10-24T20:41:47+02:00)
Maven home: C:\Java\apache-maven-3.6.0\bin\..
Java version: 11.0.1, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-11.0.1
Default locale: nl_BE, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
C:\Users\info>
```

### 1.3. Mijn eerste Maven-project

Na een succesvolle installatie kunnen we *Maven* beginnen te gebruiken voor het beheer van





## Java-projecten.

Een project wordt beschreven in een *Project Object Model*. Dit is een XML-bestand met de naam **pom.xml** dat zich in de hoofdmap van het project dient te bevinden.

Java-projecten bevatten doorgaans afzonderlijke mappen voor de broncode, bibliotheken (JAR-bestanden), gecompileerde klassen enzovoort. *Maven* maakt het beheer van projecten eenvoudiger door deze mappenstructuur te standaardiseren. Het is mogelijk, maar niet aangewezen om hiervan af te wijken. Het gebruik van de *Maven*-standaardinstellingen heeft als voordeel dat de configuratie minimaal is en dat men makkelijk inzicht krijgt in nieuwe projecten die ook deze standaard volgen.

Voor een eenvoudig Java-project ziet deze mappenstructuur er als volgt uit:

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

Map/bestand	Inhoud
src/main/java	De Java-broncode van het project.
src/main/resources	Andere bestanden die we nodig hebben, zoals <i>property</i> -bestanden, configuratiebestanden, afbeeldingen enzovoort.
src/test/java	De Java-broncode van de testklassen ( <i>JUnit</i> of <i>TestNG</i> )
src/test/resources	Andere bestanden die we enkel nodig hebben in de testklassen.
target	Gegenereerde <i>artifacts</i> : JAR - WAR - EAR - ZIP.
target/classes	De gecompileerde klassen.
target/test-classes	De gecompileerde testklassen.
pom.xml	Dit bestand bevat het <i>Project Object Model</i> .

Terwijl Java-broncodebestanden uit de map **src/main/java** gecompileerd worden naar de map **target/classes** worden de *resource*-bestanden uit de map **src/main/resources** gekopieerd naar de map **target/classes**. Dit impliceert dat deze bestanden in het finale *modulepath* beschikbaar worden gesteld.

We zullen dit alles illustreren met een voorbeeld. Stel dat we een project willen maken voor de alom bekende "Hello World". We maken hiervoor de volgende mappenstructuur:

```
project
```



```

+---src
  +---main
    +---java
      +---module-info.java
      +---eu
        +---noelvaes
          +---hello
            +---App.java
pom.xml

```

De broncode:

### ***App.java***

```

package eu.noelvaes.hello;

/**My own application class.
 *
 * @author No&euml;l Vaes
 *
 */
public class App {
    /**This method says hello to the world.
     * @return "Hello World"
     */
    public String sayHello() {
        return "Hello World";
    }

    public static void main(String[] args) {
        App app = new App();
        System.out.println(app.sayHello());
    }
}

```

De modulebeschrijving:

### ***module-info.java***

```

module eu.noelvaes.hello {
}

```

Onze module krijgt hier de naam ***eu.noelvaes.hello***. We exporteren geen pakketten aangezien het hier om een hoofdprogramma gaat.

Het POM-bestand:

### ***pom.xml***

```

<?xml version="1.0" encoding="UTF-8"?>
<project
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd"
  xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <modelVersion>4.0.0</modelVersion>
  <groupId>eu.noelvaes.hello</groupId>
  <artifactId>Hello</artifactId>

```