



UNIwersytet Gdański

Wprowadzenie do narzędzia MAVEN

Mateusz Miotk

26 marzec 2018

Instytut Informatyki UG

Maven - co to?

Maven to narzędzie do budowania kodu. Umożliwia on zarządzanie kodem, który obejmuje zbiór standardów, cykl życia projektu oraz zarządzanie zależnościami. Główną zaletą używania narzędzia **Maven** dla nas będzie fakt, że projekt utworzony w tym narzędziu będzie działał w dowolnym środowisku IDE (Eclipse, NetBeans).

Projekty w **Maven** można tworzyć przez środowisko IDE lub przez wiersz poleceń (przy pomocy polecenia **mvn**).

Maven można zainstalować na każdym systemie operacyjnym. Warunkiem jest mieć oprogramowanie **JAVA**.

Ponadto narzędzia Maven można używać do projektów w innych językach programowania np. Scala

Utworzenie projektu w Maven - wiersz poleceń

Maven charakteryzuje się kilkunastoma szablonami projektu. Można na przykład utworzyć aplikację webową (JAVA EE) lub zwykłą i wiele wiele innych.

Aby utworzyć projekt w Maven w wierszu poleceń należy użyć polecenia:

```
mvn archetype:generate
```

mvn archetype:generate - przykład

```
1772: remote => us:raton1:schemacrawler-archetype-plugin-1.0.1 (*)
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains): 926: 1635
Choose org.spilth:java8-junit4-quickstart version:
1: 1.0.1
2: 1.0.2
Choose a number: 2: 1
Downloading: https://repo.maven.apache.org/maven2/org/spilth/java8-junit4-quickstart/1.0.1/java8-junit4-quickstart-1.0.1.jar
Downloaded: https://repo.maven.apache.org/maven2/org/spilth/java8-junit4-quickstart/1.0.1/java8-junit4-quickstart-1.0.1.jar
at 8.2 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/spilth/java8-junit4-quickstart/1.0.1/java8-junit4-quickstart-1.0.1.jar
Downloaded: https://repo.maven.apache.org/maven2/org/spilth/java8-junit4-quickstart/1.0.1/java8-junit4-quickstart-1.0.1.jar
at 8.2 KB/sec)
Define value for property 'groupId': : mavenTest
Define value for property 'artifactId': : mavenTest
Define value for property 'version': : 1.0-SNAPSHOT
Define value for property 'package': : mavenTest
Confirm properties configuration:
groupId: mavenTest
artifactId: mavenTest
version: 1.0-SNAPSHOT
package: mavenTest
Y: : y
```

Rysunek 1:

Co powstało?

Utworzył nam się folder **mavenTest** i jego zawartość jest następująca:

```
-rw-rw-r-- 1 mateusz mateusz 1128 lut  6 11:07 pom.xml  
drwxrwxr-x 4 mateusz mateusz 4096 lut  6 10:08 src
```

Można też utworzyć projekt nie wchodząc w tryb interaktywny:

```
mvn archetype:generate -DgroupId=mavenTest2  
-DartifactId=myApp  
-DarchetypeArtifactId=maven-archetype-quickstart  
-DinteractiveMode=false
```

Plik **pom.xml** jest plikiem konfiguracyjnym projektu. W nim dokonujemy zmian, gdy chcemy użyć wtyczek, pakietów, innych wbudowanych narzędzi (np. Hamcrest, AssertJ, itd.)

Reszta wygląda tak jak w projekcie utworzonym w środowisku IDE (w tym przypadku).

Fragment zawartości pliku pom.xml

```
<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/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>mavenTest</groupId>
  <artifactId>mavenTest</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
```

...

Uruchomienie projektu

Aby skompilować projekt wystarczy w jego katalogu użyć polecenia:

```
mvn compile
```

Wówczas w projekcie utworzy się katalog **target** z skompilowanym projektem.

Aby wyczyścić projekt wystarczy użyć polecenia:

```
mvn clean
```


Aby uruchomić testy w projekcie Maven należy użyć polecenia:

```
mvn test
```

Uwaga

Skompilowany projekt nie oznacza, że projekt da się uruchomić!

Uwaga

Nazwy klas testów w Maven muszą się kończyć słowem **Test**!

Aby spakować projekt (w naszym przypadku stworzyć format .jar) należy użyć polecenia:

```
mvn package
```

Wówczas pojawi nam się plik z rozszerzeniem **.jar**.

Inne przydatne komendy w Maven

```
mvn install
```

Instaluje projekt w lokalnym repozytorium (.m2/repository/).

```
mvn verify
```

Sprawdza czy skompilowana paczka jest poprawna.

```
mvn site
```

Generuje dokumentacje do projektu,

```
mvn deploy
```

Instaluje projekt na zdalnym repozytorium

Instalowanie bibliotek, zależności

Aby na przykład dodać bibliotekę **Hamcrest** należy dodać do pliku **pom.xml** następującą wartość:

```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.hamcrest</groupId>
    <artifactId>hamcrest-all</artifactId>
    <version>1.3</version>
  </dependency>
</dependencies>
```

Wyszukiwarka pakietów jest dostępna na stronie:
<https://mvnrepository.com/>

Środowisko eclipse pozwala na przekonwertowanie projektu z Javy na Maven. Wystarczy prawym przyciskiem myszy kliknąć

Configure->Convert to Maven Project.

Oczywiście w samym środowisku również możemy utworzyć projekt w Maven. Wtedy opuszczamy całą konfigurację przy pomocy wiersza poleceń.






Alternatywa do narzędzia Maven - Gradle

Innym dość popularnym narzędziem jest **Gradle**. Jedną z jego zalet jest niewielka ilość szablonowego kodu, co pozwala tworzyć krótkie pliki konfiguracyjne. Narzędzie to używane jest w Google.

Jego plikiem konfiguracyjnym jest **build.gradle**, który zawiera informacje wyrażone za pomocą języka **Groovy**.

Przykład:

```
apply plugin: 'java'  
sourceCompatibility = 1.7  
version = '1.0'
```

-  V. Farcic, A. Garcia, *TDD. Programowanie w Javie. Sterowanie testami*, Wydawnictwo Helion, 2016.
-  *Maven in 5 minutes*, <https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>
-  T. O'Brien, J. van Zyl, B. Fox, J. Casey, J. Xu, T. Locher ,*Maven by Example. An Introduction to Apache Maven*, Sonatype, 2010
-  Srirangan, *Maven 3 Cookbook*, Packt Publishing, 2011
-  <https://gradle.org/>