1. Create a Project from Maven Template

In a terminal (\*uix or Mac) or command prompt (Windows), navigate to the folder you want to create the Java project. Type this command :

mvn archetype:generate

-DgroupId={project-packaging}

-DartifactId={project-name}

-DarchetypeArtifactId={maven-template}

-DinteractiveMode=false

This tells Maven to generate a Java project from a Maven template. For example,

D:\>mvn archetype:generate -DgroupId=com.mkyong.hashing -DartifactId=java-project -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

...

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 3.992 s

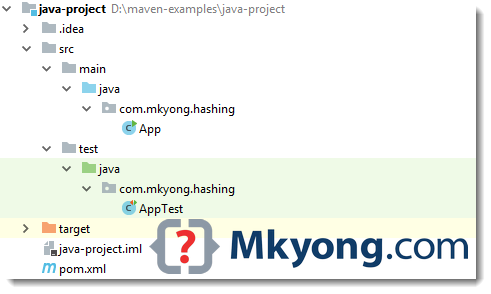
[INFO] Finished at: 2018-09-27T17:15:57+08:00

[INFO] ------------------------------------------------------------------------

Above command will generate a Java project from maven-archetype-quickstart template.

2. Maven Directory Layout

The following project directory structure will be created. In short, source code puts in folder /src/main/java/, unit test code puts in /src/test/java/.



*P.S Above figure is captured from IntelliJ IDEA, just ignore those .idea folder.*

**Note**  
Read this [Maven standard directory layout](https://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html).

3. POM file

Review the generated pom.xml. It’s quite empty, just a single jUnit dependency.

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>com.mkyong.hashing</groupId>

<artifactId>java-project3</artifactId>

<packaging>jar</packaging>

<version>1.0-SNAPSHOT</version>

<name>java-project</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

This POM file is like the Ant build.xml file, it describes the entire project information, everything from directory structure, project plugins, project dependencies, how to build this project and etc, read this [official POM guide](http://maven.apache.org/guides/introduction/introduction-to-the-pom.html).

4. Update POM

4.1 Add compiler properties to tell Maven use a specified JDK version to compile the source code.

<properties>

*<!-- https://maven.apache.org/general.html#encoding-warning -->*

<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>

4.2 Update jUnit to 4.12

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

4.3 Add commons-codec for SHA hashing.

*<!-- Dependency for hashing -->*

*<!-- https://search.maven.org/artifact/commons-codec/commons-codec/1.11/jar -->*

<dependency>

<groupId>commons-codec</groupId>

<artifactId>commons-codec</artifactId>

<version>1.11</version>

</dependency>

4.4 Complete updated version.

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>com.mkyong.hashing</groupId>

<artifactId>java-project</artifactId>

<packaging>jar</packaging>

<version>1.0-SNAPSHOT</version>

<name>java-project</name>

<url>http://maven.apache.org</url>

<properties>

*<!-- https://maven.apache.org/general.html#encoding-warning -->*

<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>

<dependency>

<groupId>commons-codec</groupId>

<artifactId>commons-codec</artifactId>

<version>1.11</version>

</dependency>

</dependencies>

</project>

5. Write Code

5.1 Update the App.java to accept an input and hash it with a SHA-256 algorithm.

App.java

package com.mkyong.hashing;

import org.apache.commons.codec.digest.DigestUtils;

public class App {

public static void main(String[] args) {

if (args.length < 1) {

System.err.println("Please provide an input!");

System.exit(0);

}

System.out.println(sha256hex(args[0]));

}

public static String sha256hex(String input) {

return DigestUtils.sha256Hex(input);

}

}

5.2 Unit Test.

AppTest.java

package com.mkyong.hashing;

import org.junit.Assert;

import org.junit.Test;

public class AppTest {

private String INPUT = "123456";

@Test

public void testLength() {

Assert.assertEquals(64, App.sha256hex(INPUT).length());

}

@Test

public void testHex() {

String expected = "8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92";

Assert.assertEquals(expected, App.sha256hex(INPUT));

}

}

Done.

6. Maven Build

6.1 Let build it with mvn package

D:\java-project>mvn package

[INFO] Scanning for projects...

[INFO]

[INFO] ------------------< com.mkyong.hashing:java-project >-------------------

[INFO] Building java-project 1.0-SNAPSHOT

[INFO] --------------------------------[ jar ]---------------------------------

[INFO]

......

-------------------------------------------------------

T E S T S

-------------------------------------------------------

Running com.mkyong.hashing.AppTest

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.067 sec

Results :

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

[INFO]

[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ java-project ---

[INFO] Building jar: D:\java-project\target\java-project-1.0-SNAPSHOT.jar

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 1.956 s

[INFO] Finished at: 2018-09-28T12:40:18+08:00

[INFO] ------------------------------------------------------------------------

It compiles, run unit test and package the project into a jar file and puts it into the project/target folder.

7. Run#1

7.1 Run it. Oops… By default, Maven didn’t add the project dependencies commons-codec into the jar file.

D:\java-project>java -cp target/java-project-1.0-SNAPSHOT.jar com.mkyong.hashing.App 123456

Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/commons/codec/digest/DigestUtils

at com.mkyong.hashing.App.sha256hex(App.java:18)

at com.mkyong.hashing.App.main(App.java:13)

Caused by: java.lang.ClassNotFoundException: org.apache.commons.codec.digest.DigestUtils

at java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(Unknown Source)

at java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(Unknown Source)

at java.base/java.lang.ClassLoader.loadClass(Unknown Source)

... 2 more

7.2 To solve it, we can use this maven-shade-plugin to create an uber/fat-jar – group everything into a single jar file.

pom.xml

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-shade-plugin</artifactId>

<version>3.2.0</version>

<executions>

*<!-- Attach the shade goal into the package phase -->*

<execution>

<phase>package</phase>

<goals>

<goal>shade</goal>

</goals>

</execution>

</executions>

</plugin>

</plugins>

</build>

7.3 Package it again!

D:\java-project>mvn clean package

[INFO] Scanning for projects...

[...

[INFO] --- maven-shade-plugin:3.2.0:shade (default) @ java-project ---

[INFO] Including commons-codec:commons-codec:jar:1.11 in the shaded jar.

[INFO] Replacing original artifact with shaded artifact.

[INFO] Replacing D:\java-project\target\java-project-1.0-SNAPSHOT.jar

with D:\java-project\target\java-project-1.0-SNAPSHOT-shaded.jar

...

Two jars will be generated, review the file size :

D:\java-project>dir target

Volume in drive D is Samsung970

Volume Serial Number is 10DF-E63D

Directory of D:\java-project\target

28/09/2018 12:57 PM 335,643 java-project-1.0-SNAPSHOT.jar

28/09/2018 12:57 PM 3,053 original-java-project-1.0-SNAPSHOT.jar

...

8. Run#2

8.1 Run it again. Good, the result is expected.

D:\java-project>java -cp target/java-project-1.0-SNAPSHOT.jar com.mkyong.hashing.App 123456

8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92

8.2 Can we run it as Jar? No, there is no main class.

D:\java-project>java -jar target/java-project-1.0-SNAPSHOT.jar 123456

no main manifest attribute, in target/java-project-1.0-SNAPSHOT.jar

8.3 To solve it, add the main class in maven-shade-plugin like this.

pom.xml

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-shade-plugin</artifactId>

<version>3.2.0</version>

<executions>

*<!-- Attach the shade into the package phase -->*

<execution>

<phase>package</phase>

<goals>

<goal>shade</goal>

</goals>

<configuration>

<transformers>

<transformer implementation="org.apache.maven.plugins.shade.resource.ManifestResourceTransformer">

<mainClass>com.mkyong.hashing.App</mainClass>

</transformer>

</transformers>

</configuration>

</execution>

</executions>

</plugin>

9. Run#3

9.1 Package it again!

D:\java-project>mvn clean package

9.2 Run it as Jar.

D:\java-project>java -jar target/java-project-1.0-SNAPSHOT.jar 123456

8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92

Done.

10. POM

Final POM file.

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>com.mkyong.hashing</groupId>

<artifactId>java-project</artifactId>

<packaging>jar</packaging>

<version>1.0-SNAPSHOT</version>

<name>java-project</name>

<url>http://maven.apache.org</url>

<properties>

*<!-- https://maven.apache.org/general.html#encoding-warning -->*

<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>

<dependency>

<groupId>commons-codec</groupId>

<artifactId>commons-codec</artifactId>

<version>1.11</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-shade-plugin</artifactId>

<version>3.2.0</version>

<executions>

*<!-- Attach the shade into the package phase -->*

<execution>

<phase>package</phase>

<goals>

<goal>shade</goal>

</goals>

<configuration>

<transformers>

<transformer

implementation="org.apache.maven.plugins.shade.resource.ManifestResourceTransformer">

<mainClass>com.mkyong.hashing.App</mainClass>

</transformer>

</transformers>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</project>