

Maven Example

We can create a simple maven example by executing the **archetype:generate** command of **mvn tool**.

To create a simple java project using maven, you need to open command prompt and run the **archetype:generate** command of mvn tool.

Syntax

The **syntax** to generate the project architecture is given below:

```
mvn archetype:generate -DgroupId=groupid -DartifactId=artifactid
-DarchetypeArtifactId=maven-archetype-quickstart -
DinteractiveMode=booleanValue
```

Example

The **example** to generate the project architecture is given below:

```
mvn archetype:generate -DgroupId=com.javatpoint -DartifactId=CubeGenerator
-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

Note: Here, we are using maven-archetype-quickstart to create simple maven core project. if you use maven-archetype-webapp, it will generate a simple maven web application.

Output

Now it will **generate following code in the command prompt**:

```
mvn archetype:generate -DgroupId=com.javatpoint -DartifactId=CubeGenerator
-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----
[INFO]
[INFO] >>> maven-archetype-plugin:2.2:generate (default-cli) @ standalone-pom >>
>
[INFO]
[INFO] <<< maven-archetype-plugin:2.2:generate (default-cli) @ standalone-pom <<
<
[INFO]
[INFO] --- maven-archetype-plugin:2.2:generate (default-cli) @ standalone-pom ---
-
[INFO] Generating project in Batch mode
Downloading: http://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.0/maven-archetype-quickstart-1.0.jar
Downloaded: http://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.0/maven-archetype-quickstart-1.0.jar (5 KB at 3.5 KB/sec)
Downloading: http://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.0/maven-archetype-quickstart-1.0.pom
Downloaded: http://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.0/maven-archetype-quickstart-1.0.pom (703 B at 0.9 KB/sec)
[INFO] -----
```

```
---
[INFO] Using following parameters for creating project from Old (1.x) Archetype:
maven-archetype-quickstart:1.0
[INFO] -----
---

[INFO] Parameter: groupId, Value: com.javatpoint
[INFO] Parameter: packageName, Value: com.javatpoint
[INFO] Parameter: package, Value: com.javatpoint
[INFO] Parameter: artifactId, Value: CubeGenerator
[INFO] Parameter: basedir, Value: C:\Users\SSS IT
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\Users\SSS IT\CubeGene
rator
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 10.913s
[INFO] Finished at: Thu Dec 26 16:45:18 IST 2013
[INFO] Final Memory: 9M/25M
[INFO] -----

'cmd' is not recognized as an internal or external command,
operable program or batch file.
```

Generated Directory Structure

Now go to the current directory from where you have executed the mvn command. For example: **C:\Users\SSS IT\CubeGenerator**. You will see that a simple java project is created that has the following directory:

```
CubeGenerator
-src
--main
---java
----com
-----javatpoint
-----App.java
--test
---java
----com
-----javatpoint
-----AppTest.java
-pom.xml
```

As you can see, there are created 3 files pom.xml, App.java and AppTest.java. Let's have a quick look at these files:

1) Automatically Generated pom.xml file

```
<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.javatpoint</groupId>
    <artifactId>CubeGenerator</artifactId>
    <packaging>jar</packaging>
    <version>1.0-SNAPSHOT</version>
    <name>CubeGenerator</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>
```

2) Automatically Generated App.java file

```
package com.javatpoint;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )
    {
        System.out.println( "Hello World!" );
    }
}
```

3) Automatically Generated AppTest.java file

```
package com.javatpoint;

import junit.framework.Test;
import junit.framework.TestCase;
import junit.framework.TestSuite;

/**
 * Unit test for simple App.
 *
 */
public class AppTest
    extends TestCase
{
    /**
     * Create the test case
     *
     * @param testName name of the test case
     */
    public AppTest( String testName )
    {
        super( testName );
    }

    /**
     * @return the suite of tests being tested
     */
    public static Test suite()
    {
        return new TestSuite( AppTest.class );
    }

    /**
     * Rigorous Test :-)
```

```
*/  
  
public void testApp()  
{  
    assertTrue( true );  
}  
}
```

Compile the Maven Java Project

To compile the project, go to the project directory,

for example: **C:\Users\SSS IT\CubeGenerator** and write the following command on the command prompt:

```
mvn clean compile
```

Now, you will see a lot of execution on the command prompt. If you check your project directory, **target directory** is created that contains the class files.

Run the Maven Java Project

To run the project, go to the project directory\target\classes,

for example: **C:\Users\SSS IT\CubeGenerator\target\classes** and write the following command on the command prompt:

```
java com.javatpoint.App
```

Now, you will see the output on the command prompt:

Output of the maven example

```
Hello World!
```

How to build the maven project or how to package maven project?

The **mvn package** command completes the build life cycle of the maven project such as:

- 1. validate
- 2. compile
- 3. test
- 4. package
- 5. integration-test
- 6. verify
- 7. install
- 8. deploy

Visit this link to know more about build life cycle <http://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>

You need to execute the following command on the command prompt to package the maven project:

```
mvn package
```

Now you will see that **a jar file is created** inside the project/target directory.

You can also run the maven project by the jar file. To do so, go to the maven project directory, for example: **C:\Users\SSS IT\CubeGenerator** and execute the following command on the cmd:

```
java -classpath target\CubeGenerator-1.0-SNAPSHOT.jar,.; com.javatpoint.App
```

Now you will see the following output:

```
Hello World!
```