

MAVEN - PROJECT TEMPLATES

http://www.tutorialspoint.com/maven/maven_project_templates.htm

Copyright © tutorialspoint.com

Maven provides users, a very large list of different types of project templates 614 in numbers using concept of **Archetype**. Maven helps users to quickly start a new java project using following command

```
mvn archetype:generate
```

What is Archetype?

Archetype is a Maven plugin whose task is to create a project structure as per its template. We are going to use *quickstart* archetype plugin to create a simple java application here.

Using Project Template

Let's open command console, go the **C:\ > MVN** directory and execute the following **mvn** command

```
C:\MVN>mvn archetype:generate
```

Maven will start processing and will ask to choose required archetype

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----
[INFO]
[INFO] >>> maven-archetype-plugin:2.4:generate (default-cli) > generate-sources@
standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:2.4:generate (default-cli) < generate-sources@
standalone-pom <<<
[INFO]
[INFO] --- maven-archetype-plugin:2.4:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Interactive mode
[WARNING] No archetype found in remote catalog. Defaulting to internal catalog
[INFO] No archetype defined. Using maven-archetype-quickstart (org.apache.maven.
archetypes:maven-archetype-quickstart:1.0)
Choose archetype:
1: internal -> org.apache.maven.archetypes:maven-archetype-archetype (An archetype which
contains a sample archetype.)
2: internal -> org.apache.maven.archetypes:maven-archetype-j2ee-simple (An archetype
which contains a simplified sample J2EE application.)
3: internal -> org.apache.maven.archetypes:maven-archetype-plugin (An archtypewhich
contains a sample Maven plugin.)
4: internal -> org.apache.maven.archetypes:maven-archetype-plugin-site (An archetype
which contains a sample Maven plugin site.
    This archetype can be layered upon an existing Maven plugin project.)
5: internal -> org.apache.maven.archetypes:maven-archetype-portlet (An archetype which
contains a sample JSR-268 Portlet.)
6: internal -> org.apache.maven.archetypes:maven-archetype-profiles ()
7: internal -> org.apache.maven.archetypes:maven-archetype-quickstart (An archetype which
contains a sample Maven project.)
8: internal -> org.apache.maven.archetypes:maven-archetype-site (An archetype which
contains a sample Maven site which demonstrates
    some of the supported document types like APT, XDoc, and FML and demonstrates how
    to i18n your site. This archetype can be layered upon an existing Maven project.)
9: internal -> org.apache.maven.archetypes:maven-archetype-site-simple (An archetype
which contains a sample Maven site.)
10: internal -> org.apache.maven.archetypes:maven-archetype-webapp (An archetype
which contains a sample Maven Webapp project.)
Choose a number or apply filter (format: [groupId:]artifactId, case sensitive contains):
```

7:

Press Enter to choose to default option7: *maven - archetype - quickstart*. Maven will ask for project detail. Enter project detail as asked. Press Enter if default value is provided. You can override them by entering your own value.

```
Define value for property 'groupId': : com.companyname.insurance
Define value for property 'artifactId': : health
Define value for property 'version': 1.0-SNAPSHOT:
Define value for property 'package': com.companyname.insurance:
```

Maven will ask for project detail confirmation. Press enter or press Y

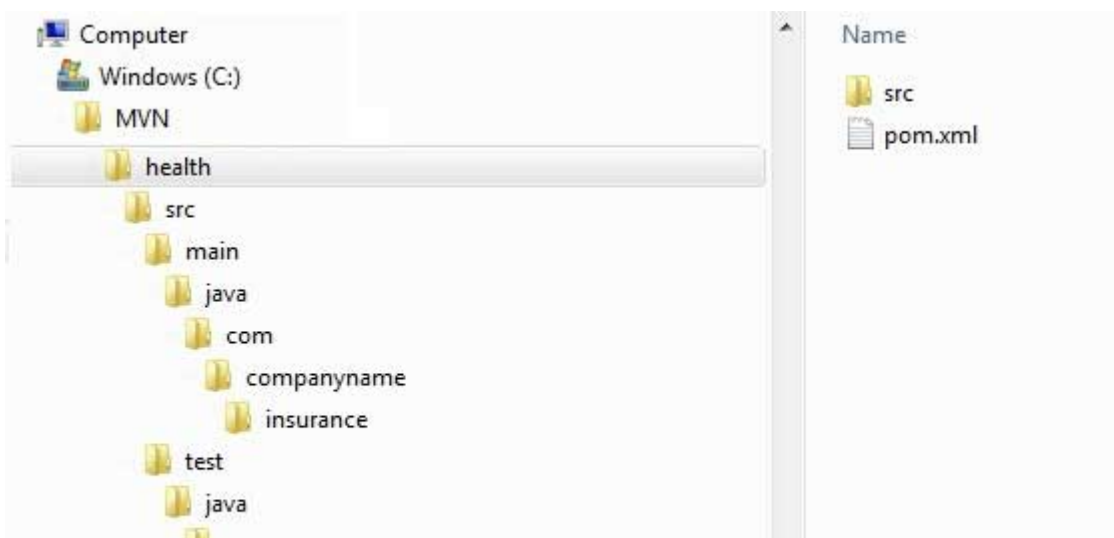
```
Confirm properties configuration:
groupId: com.companyname.insurance
artifactId: health
version: 1.0-SNAPSHOT
package: com.companyname.insurance
Y:
```

Now Maven will start creating project structure and will display the following:

```
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype:
maven-archetype-quickstart:1.1
[INFO] -----
[INFO] Parameter: groupId, Value: com.companyname.insurance
[INFO] Parameter: packageName, Value: com.companyname.insurance
[INFO] Parameter: package, Value: com.companyname.insurance
[INFO] Parameter: artifactId, Value: health
[INFO] Parameter: basedir, Value: C:\MVN
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\MVN\health
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 02:29 min
[INFO] Finished at: 2015-09-27T12:18:02+05:30
[INFO] Final Memory: 16M/247M
[INFO] -----
```

Created Project

Now go to **C:\ > MVN** directory. You'll see a java application project created named **health** which was given as *artifactId* at the time of project creation. Maven will create a standard directory layout for the project as shown below:





Created POM.xml

Maven generates a POM.xml file for the project as listed below:

```
<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/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.companyname.insurance</groupId>
  <artifactId>health</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>health</name>
  <url>http://maven.apache.org</url>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Created App.java

Maven generates sample java source file, App.java for the project as listed below:

Location: **C:\ > MVN > health > src > main > java > com > companyname > insurance > App.java**

```
package com.companyname.insurance;

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

Created AppTest.java

Maven generates sample java source test file, AppTest.java for the project as listed below:

Location: **C:\ > MVN > health > src > test > java > com > companyname > insurance > AppTest.java**

```
package com.companyname.insurance;

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 );
    }
}

```

That's it. Now you can see the power of Maven. You can create any kind of project using single command in maven and can kick start your development.

Loading [MathJax]/jax/output/HTML-CSS/jax.js