**Archetype Maven**

Archetype Maven:

**Archetype** is a **Maven** project templating toolkit. An **archetype** is defined as an original pattern or model from which all other things of the same kind are made. The name fits as we are trying to provide a system that provides a consistent means of generating **Maven** projects.

Archetype will help authors create Maven project templates for users, and provides users with the means to generate parameterized versions of those project templates.

USES**:**

* Using archetypes provides a great way to enable developers quickly in a way consistent with best practices employed by your project or organization.
* Within the Maven project, we use archetypes to try and get our users up and running as quickly as possible by providing a sample project that demonstrates many of the features of Maven, while introducing new users to the best practices employed by Maven.
* In a matter of seconds, a new user can have a working Maven project to use as a jumping board for investigating more of the features in Maven.
* A good example of this is the Maven site archetype. If, for example, you have used the quick start archetype to generate a working project, you can then quickly create a site for that project by using the site archetype within that existing project. You can do anything like this with archetypes.

Using an Archetype:

To create a new project based on an Archetype, you need to call mvn archetype:generate goal, like the following:

1. mvn archetype:generate

## Guide to Creating Archetypes:

Creating an archetype is a pretty straight forward process. An archetype is a very simple artifact that contains the project prototype you wish to create. An archetype is made up of:

* An [archetype descriptor](https://maven.apache.org/archetype/archetype-models/archetype-descriptor/archetype-descriptor.html) (archetype-metadata.xml in directory: src/main/resources/META-INF/maven/). It lists all the files that will be contained in the archetype and categorizes them so they can be processed correctly by the archetype generation mechanism.
* the prototype files that are copied by the archetype plugin (directory: src/main/resources/archetype-resources/)
* the prototype pom (pom.xml in: src/main/resources/archetype-resources)
* A pom for the archetype (pom.xml in the archetype's root directory).

Step1: Add Archetype Plugin:

|  |
| --- |
|  |

 Step2: Create the archetype descriptor

The [archetype descriptor](https://maven.apache.org/archetype/archetype-models/archetype-descriptor/archetype-descriptor.html) is a file called archetype-metadata.xml which must be located in the src/main/resources/META-INF/maven/ directory. An example of an archetype descriptor can be found in the quickstart archetype:

1. <archetype-descriptor
2. xsi:schemaLocation="http://maven.apache.org/plugins/maven-archetype-plugin/archetype-descriptor/1.0.0
3. http://maven.apache.org/xsd/archetype-descriptor-1.0.0.xsd"
4. xmlns="http://maven.apache.org/plugins/maven-archetype-plugin/archetype-descriptor/1.0.0"
5. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6. name="quickstart">
7. <fileSets>
8. <fileSet filtered="true" packaged="true">
9. <directory>src/main/java</directory>
10. </fileSet>
11. <fileSet>
12. <directory>src/test/java</directory>
13. </fileSet>
14. </fileSets>
15. </archetype-descriptor>

The attribute name tag should be the same as the artifactId in the archetype pom.xml.

The Boolean attribute partial show if this archetype is representing a full Maven project or only parts.

The requiredProperties, fileSets and modules tags represent the different parts of the project:

* <requiredProperties> : List of required properties to generate a project from this archetype
* <fileSets> : File sets definition
* <modules> : Modules definition

At this point one can only specify individual files to be created but not empty directories.

The quick start archetype shown above defines the following directory structure:

1. archetype
2. |-- pom.xml
3. `-- src
4. `-- main
5. `-- resources
6. |-- META-INF
7. | `-- maven
8. | `--archetype-metadata.xml
9. `-- archetype-resources
10. |-- pom.xml
11. `-- src
12. |-- main
13. | `-- java
14. | `-- App.java
15. `-- test
16. `-- java
17. `-- AppTest.java

### step3. Create the prototype files and the prototype pom.xml

The next component of the archetype to be created is the prototype pom.xml. Any pom.xml will do, just don't forget to the set artifactId and groupId as variables (${artifactId} / ${groupId}). Both variables will be initialized from the command line when calling archetype: generate.

An example for a prototype pom.xml is:

1. <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2. xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3. <modelVersion>4.0.0</modelVersion>
5. <groupId>${groupId}</groupId>
6. <artifactId>${artifactId}</artifactId>
7. <version>${version}</version>
8. <packaging>jar</packaging>
10. <name>${artifactId}</name>
11. <url>http://www.myorganization.org</url>
13. <dependencies>
14. <dependency>
15. <groupId>junit</groupId>
16. <artifactId>junit</artifactId>
17. <version>4.12</version>
18. <scope>test</scope>
19. </dependency>
20. </dependencies>
21. </project>

### step4. Install the archetype and run the archetype plugin

Now you are ready to install the archetype:

1. mvn install

Now that you have created an archetype, you can try it on your local system by using the following command. In this command, you need to specify the full information about the archetype you want to use (its groupId, its artifactId, and its version) and the information about the new project you want to create (artifactId and groupId). Don't forget to include the version of your archetype (if you don't include the version, you archetype creation may fail with a message that version: RELEASE was not found)

1. mvn archetype:generate \
2. -DarchetypeGroupId=<archetype-groupId> \
3. -DarchetypeArtifactId=<archetype-artifactId> \
4. -DarchetypeVersion=<archetype-version> \
5. -DgroupId=<my.groupid> \
6. -DartifactId=<my-artifactId>

References:

* <https://www.luckyryan.com/2013/02/15/create-maven-archetype-from-existing-project/>
* <https://maven.apache.org/guides/mini/guide-creating-archetypes.html>