[Maven](https://maven.apache.org/) is an open-source build automation and project management tool widely used for Java applications. As a build automation tool, it automates the source code compilation and dependency management, assembles binary codes into packages, and executes test scripts. Maven translates and packages your source code so that it becomes an executable application. Maven can manage a project's build, reporting and documentation from a central piece of information.

Using Maven, you can create Java deliverables like JAR, EAR, and WAR files. The pom.xml file helps you to do these tasks. These XML files contain your Java project’s name, group ID, and other valuable information. Based on this information, Maven creates the JAR (Java archives) files and transfers them to the Maven repository.

**1. Project Object Model (POM)**

A POM is the basement of the Maven framework. It’s a type of XML file that accommodates data from your project and the configuration details. It includes the project, group ID, POM model version, artifact ID (project ID), and version. The project is the key element of your XML file. Group ID means the ID of the group to which your project belongs. Here, the version informs you about the number of your project releases.

**2. Maven repositories and dependencies**

The Maven repository stores all your projects’ jars, plugins, library jars, other artifacts, and the dependencies are the third-party software required by your project. You will find three types of repositories in Maven – local, central, and remote. The local repository stores all the Maven dependencies.

The Maven community handles the central repository. If you don’t find a dependency in your local repo, you can find it in the central repo. Again the developers can utilize their customized repository. That is known as the remote repository.

**Build lifecycle is made up of different phases. The default lifecycle consists of the following phases:**

* **Validate**: It authorizes your project’s correctness and ensures necessary data are available.
* **Compile**: It compiles the source code of your project.
* **Test**: It tests your compiled source codes with the unit testing frameworks.
* **Package**: In this phase, your source code will be packaged as a deliverable.
* **Verify**: It ensures your code’s quality with an integration test.
* **Install**: Your code will be installed in the local repository.
* **Deploy**: Finally, the code is ready to share with the other developers.

An archetype is a model, pattern, or symbol that is universally understood and used as a prototype for others to copy, emulate, or pattern.

Maven Archetype is a Maven plugin that makes it possible to create a project structure based on a template. These archetypes are essentially project templates that Maven generates when you create a new project. Archetype is a Maven project templating toolkit