**Fundamentals**

**for**

**Maven and**

**Project Object Model**

**(POM)**

**Objectives**

* Introduction to Maven
* Features of Maven
* Architecture of Maven
* Need of Maven
* Introduction to POM
* Maven goals

**Introduction**

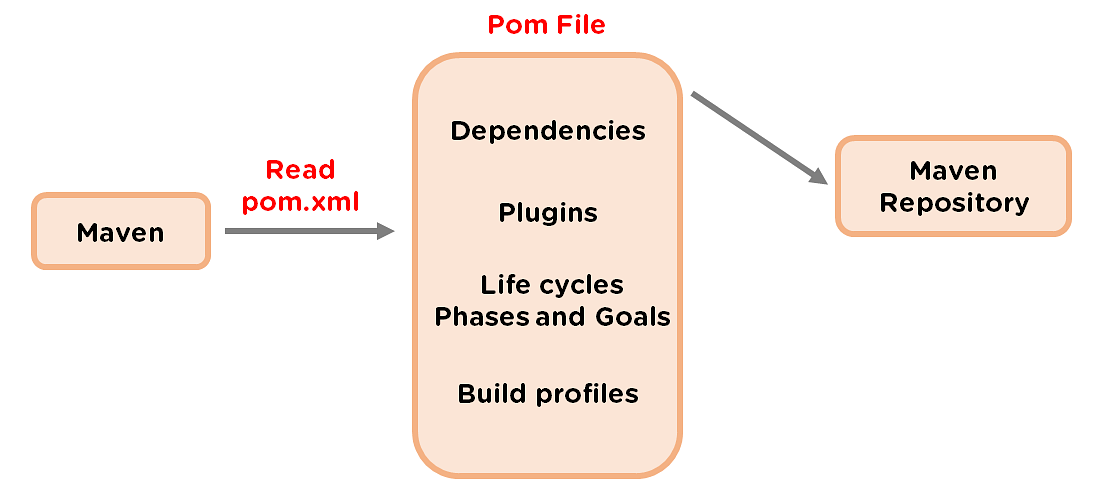
Maven, created by Jason van Zyl, began as a sub-project of [Apache Turbine](https://en.wikipedia.org/wiki/Apache_Turbine) in 2002. Maven is a build automation tool and it addresses two aspects of building software: how software is [built](https://en.wikipedia.org/wiki/Software_build), and its dependencies. Maven is built using a plugin-based architecture that allows it to make use of any application controllable through standard input

**Features of Maven:**

* Simple project setup that follows best practices.
* Consistent usage across all projects.
* Dependency management including automatic updating.
* A large and growing repository of libraries.
* Extensible, with the ability to easily write plugins in Java or scripting languages.
* Instant access to new features with little or no extra configuration.
* Maven is able to build any number of projects into predefined output types such as jar, war.
* Better Error and Integrity Reporting − Maven improved error reporting, and it provides you with a link to the Maven wiki page where you will get full description of the error.

Architecture of Maven:

The projects created in Maven contain POM files that describe the aspect of the project essentials. Maven architecture shows the process of creating and generating a report according to the requirements and executing lifecycles, phases, goals, plugins, and so on—from the first step.



* The first step refers to configuring Maven, which is stored in a Pom.xml. file.
* The POM file includes all of the configurations that Maven needs. The second step is to download the dependencies defined in pom.xml into the local repository from the central repository
* After the user starts working in Maven, the tool provides various default settings, so there is no need to add every configuration in the pom.xml

**Need of Maven:**

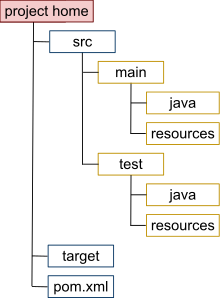
Maven is a software [project management build tool](https://www.simplilearn.com/tutorials/project-management-tutorial/project-management-tools) based on Project Object Model (POM). The tool is typically used for [Java](https://www.simplilearn.com/best-java-programs-article)-based projects. Every Java project requires certain dependencies, which are automatically downloaded when running a Maven build. This simplifies everyday tasks for Java developers and helps them with their projects.

Maven helps retrieve the correct JAR files for each project, as there may be different versions of separate packages. If you want to download dependencies, you no longer need to visit each software's official website. It can be quickly done now by visiting mvnrepository.com.

**Maven Folder Structure:**

Maven will require almost zero effort if convention is followed, there are three conventions that maven has:

* Standard layout
* Naming conventions
* One primary output per project



**Introduction to POM:**

* Project Object Model (POM) refers to the XML files with all the information regarding project and configuration details
* It contains the project description, as well as details regarding the versioning and configuration manage -ment of the project.
* The XML file is in the project home directory. Maven searches for the POM in the current directory when any given task needs to be executed.



* The **<project>** element is the root of the project descriptor.
* **<model Version>:** Declares to which version of project descriptor this POM conforms.
* **<group Id>**: A universally unique identifier for a project.
* **<artifact Id>**: The identifier for this artifact that is unique within the group given by the group ID. An artifact is something that is either produced or used by a project. Examples of artifacts produced by Maven for a project include: JARs, source and binary distributions, and WARs.
* **<packaging>**: The type of artifact this project produces, for example jar war ear pom. Plugins can create their own packaging, and therefore their own packaging types, so this list does not contain all possible types.
* Default value is: jar.
* **<Name>**: The full name of the project.
* **<Url>**: The URL to the project's homepage.
* **<Version>**: The current version of the artifact produced by this project.
* **<description>**: A detailed description of the project, used by Maven whenever it needs to describe the project, such as on the web site.
* **<dependencies>:** This element describes all of the dependencies associated with a project. These dependencies are used to construct a class path for your project during the build process.

**Maven Goals:**

There are three built-in build lifecycles.

1. **default**: handles project build and deployment.
2. **clean**: handles project cleaning
3. **site**: Handles the creation of project site documentation.

The Compiler Plug-in: This is used to test the code and compile the code, invokes Javac.

