Why Maven?

1. **Apache Maven Is** Powerful Java **project management** and **build management tool**. That means we can manage java project builds very easily using maven. Maven can helps you to **minimize**your project and build management**time and efforts**. It Is fine to manage project manually If It Is small. But If project Is very large or there are many projects then It Is very hard for developer to manage each of them manually.
2. Using maven build tool, You can setup all those things which required to run your project code Independently. Maven provides common platform to generate source code, compiling and packaging code.

Main objectives of maven build tool are as bellow.

* It makes project build process easy.
* It provides easy and uniform build system.
* It provides quality project document Information.
* Managing project dependencies.
* Provides guild lines for better project management practices.
* Facilitate easy and transparent migration to new features.
* It allows to build project using project object model (POM).
* It downloads required dependency's jar files automatically from Maven central repositories.

**Why To Use Maven In Selenium Project?**

As Maven Is build management tool, It will helps you to manage your selenium project's build easily. If you are thinking that you will get some extra functions from maven to write selenium automate test cases then you are thinking wrong. It Is build management tool and It will manage your selenium test project's build compilation, documentation and other related project tasks It self. It will helps you to create right project structure, add and manage jar files In project's build path etc..

**Maven uses POM.xml configuration file which kept all project configuration related Information**. For selenium, You need to provide selenium webdriver version related Information In POM.xml file and then It will download all required jar files automatically and store It In local repository called m2. Later on If you wants to change version of selenium webdriver then you need to modify version In POM.xml file only. Then maven will download new version jar files automatically and store In local repository. That means If you upgrade any dependency's version In POM.xml file, First It will check that version's jar files are available or not In local repository. If available then fine else It will download them from maven central repository.