**46.1. Maven Overview**

**What is Maven**:

Maven is a powerful project management tool that is based on POM (project object model). It is used for projects build, dependency and documentation. It simplifies the build process like ANT. But it is too much advanced than ANT.

In short terms we can tell maven is a tool that can be used for building and managing any Java-based project. maven make the day-to-day work of Java developers easier and generally help with the comprehension of any Java-based project.

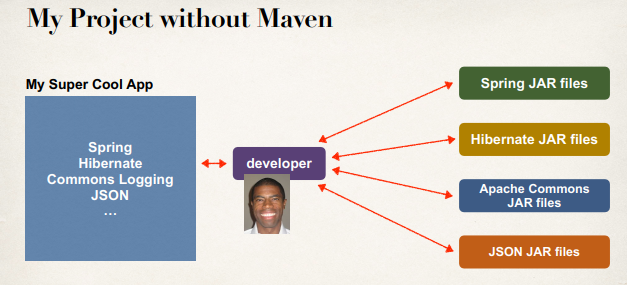
On a word maven is a project management tool for our application. The most popular use of maven is for build management and dependencies.

**What Problems Does Maven Solve**?

* When building your Java project, you may need additional JAR files
  + For example: Spring, Hibernate, Commons Logging, JSON etc.

* One approach is to download the JAR files from each project web site
* Manually add the JAR files to your build path / class-path

**Java project without Maven**:

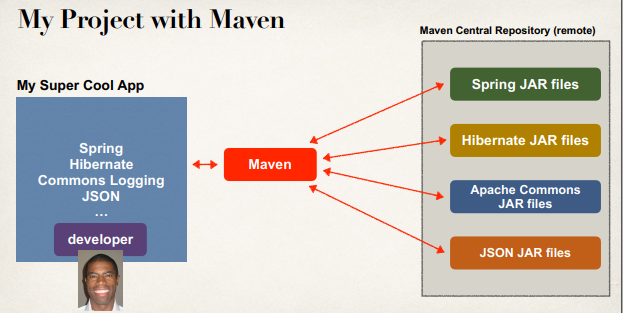


Maven can actually help us with this process and do a lot of works for us.

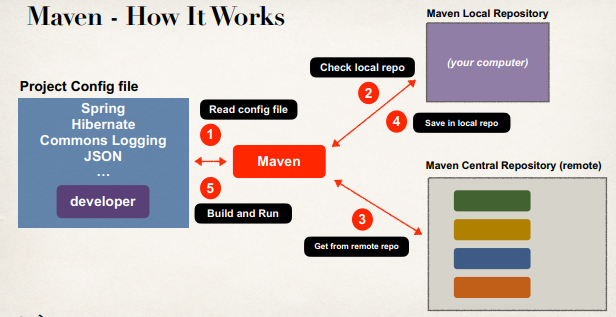
**Maven Solution**:

* Tell Maven the projects we are working with (dependencies)
  + Spring, Hibernate etc.
* Maven will go out and download the JAR files for those projects for us
* And Maven will make those JAR files available during compile/run
* Think of Maven as our friendly helper / personal shopper :-)

**My Project with Maven**:



**Maven - How It Works**:



**Handling JAR Dependencies**:

* When Maven retrieves a project dependency
  + It will also download supporting dependencies
  + For example: Spring depends on commons-logging …
* Maven will handle this for us automagically

**Building and Running**:

* When we build and run your app
* Maven will handle class / build path for us
* Based on config file, Maven will add JAR files accordingly

46.1. Maven Overview