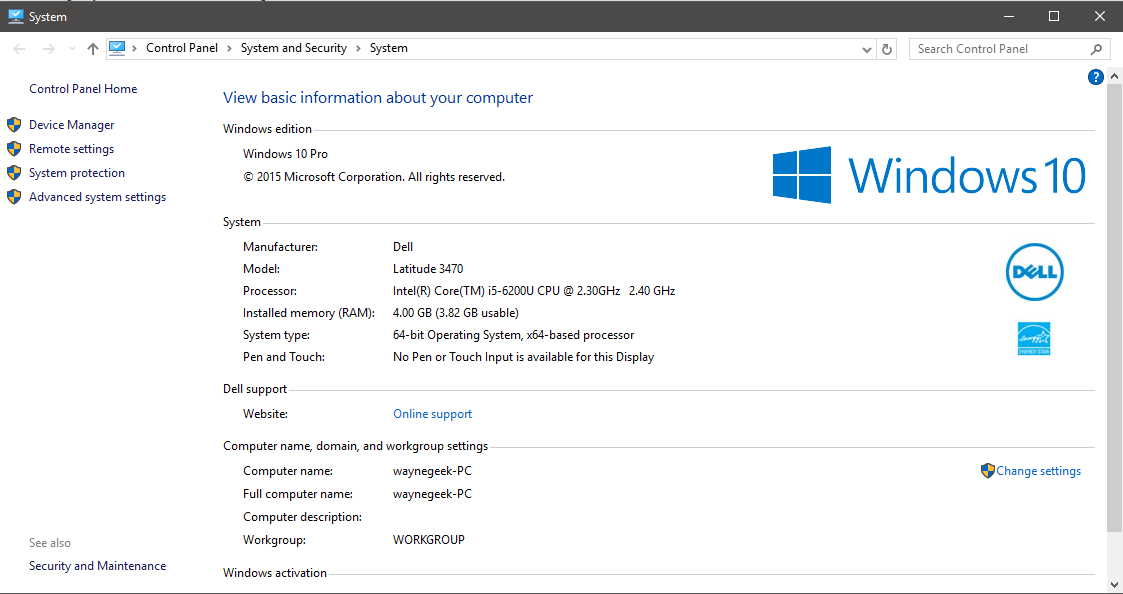
**JAVA INSTALLATION**

**1. Install Java JDK and JRE**

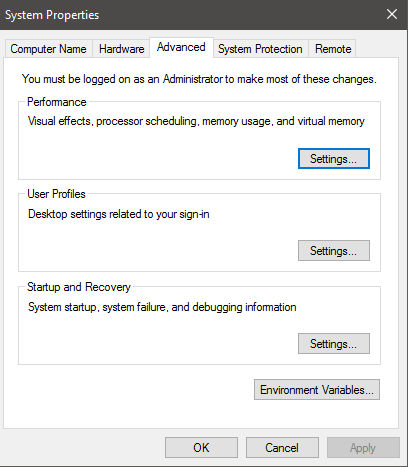
**a. Run the downloaded installer** (e.g., "jdk-8u{xx}-windows-x64.exe"), which installs both the JDK and JRE. By default, the JDK will be installed in directory "C:\Program Files\Java\jdk1.8.0\_xx", wherexx denotes the latest upgrade number; and JRE in "C:\Program Files\Java\jre1.8.0\_xx".

**2. Include JDK’s Directory in JAVA\_HOME**

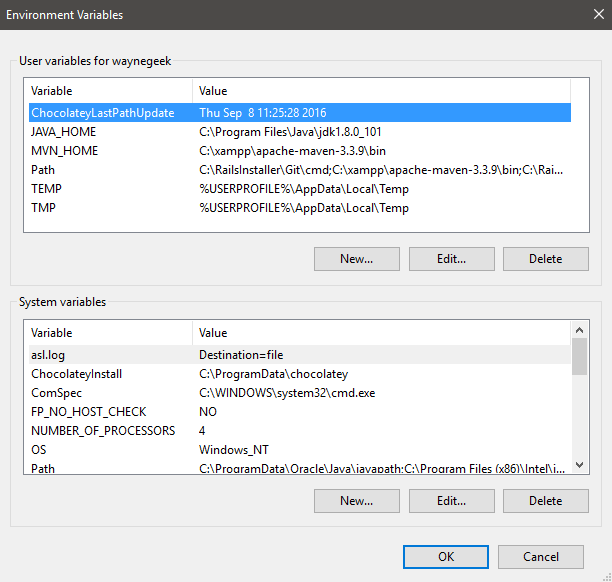
**Go to ->This PC -> Right Click then select Properties, a window will pop up**



Click **Advanced System Settings**, then switch to **Advance Tab** and click **Environment Variables**.

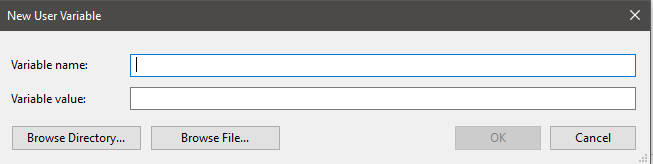
****

Create **New** Variables for your PC.

****

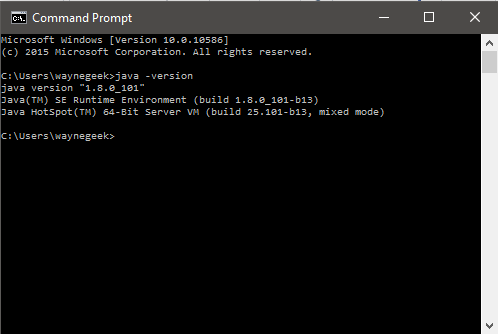
**VARIABLE NAME – JAVA\_HOME**

**VARIABLE VALUE – Java JDK Directory**

****

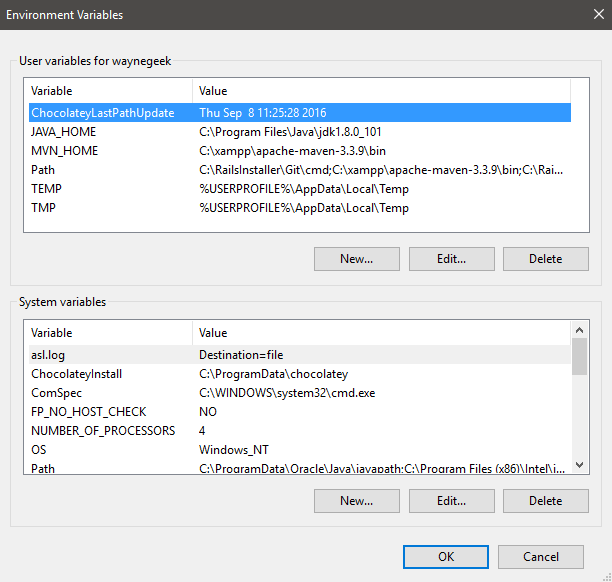
**3. Verify Installation**

Go to CMD, type **java –version**, if JRE and JDK version shows up, you have successfully installed JAVA.



**MAVEN INSTALLATION**

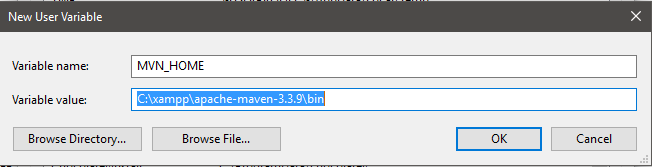
1. Make sure JDK is installed, and “**JAVA\_HOME**” variable is added as Windows environment variable.



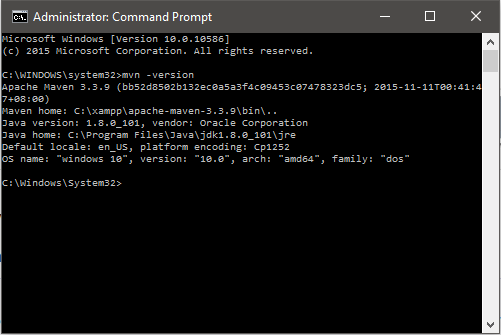
2. Visit [Maven official website](http://maven.apache.org/download.cgi), download the Maven zip file, for example : apache-maven-3.3.9-bin.zip.

Unzip to your xampp folder – C:\Program Files\xampp\apache-maven-3.3.9

3. Add MVN\_HOME variables in the windows environment, and point it to your Maven Folder.



4. Verify installation, in CMD, run mvn -version



**Create a Project in IntelliJ Idea.**

**https://github.com/kolorobot/spring-mvc-quickstart-archetype**

1. Run on CMD,

mvn archetype:generate

-DarchetypeGroupId=pl.codeleak

-DarchetypeArtifactId=spring-mvc-quickstart

-DarchetypeVersion=1.0.0

-DgroupId=com.up.skill

-DartifactId=test-maven-project (name of jar file for the project)

-Dversion=0.0.1

-DarchetypeRepository=http://kolorobot.github.io/spring-mvc-quickstart-archetype

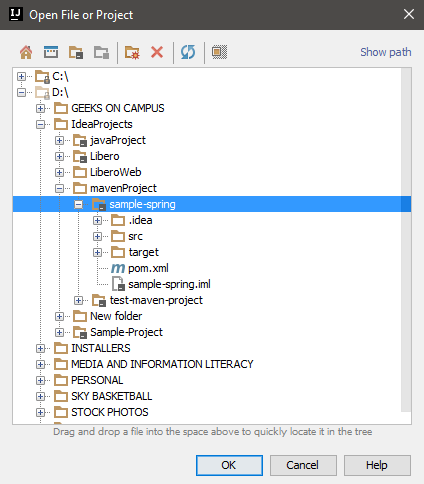
This downloads a spring project from **http://kolorobot.github.io/spring-mvc-quickstart-archetype** from remote library to generate Spring MVC archetype project.

**DgroupId** **->** (will identify your project uniquely across all projects, so we need to enforce a naming schema. It has to follow the package name rules, what means that has to be at least as a domain name you control,)

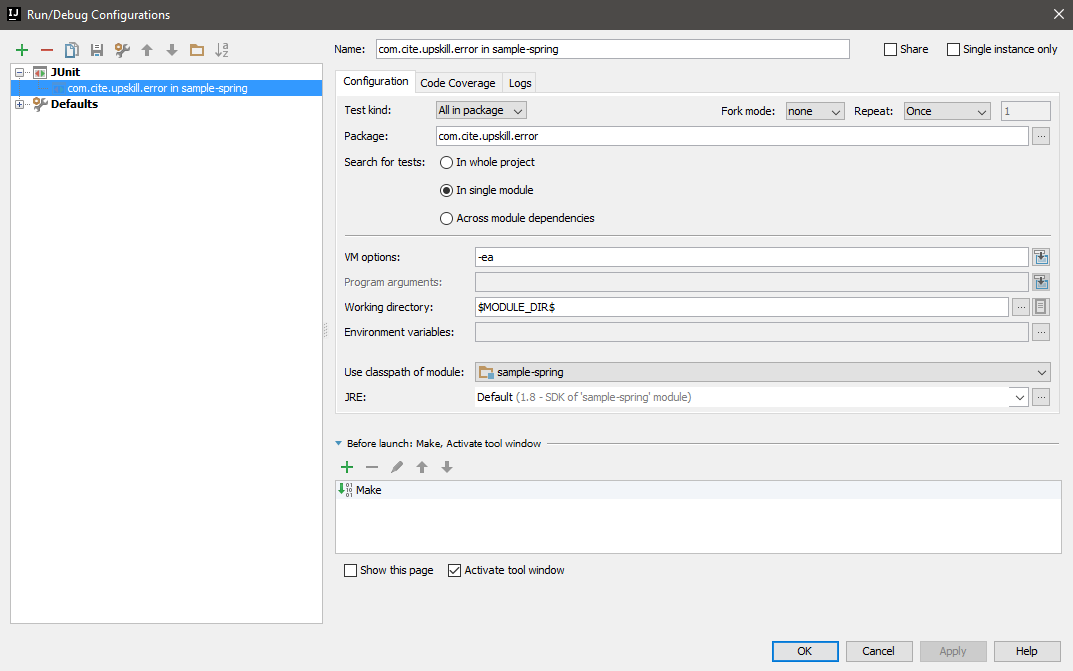
**DartifactId ->** is the name of the jar without version. If you created it then you can choose whatever name you want with lowercase letters and no strange symbols. If it's a third party jar you have to take the name of the jar as it's distributed.

**Dversion ->** if you distribute it then you can choose any typical version with numbers and dots (1.0, 1.1, 1.0.1, ...). Don't use dates as they are usually associated with SNAPSHOT (nightly) builds. If it's a third party artifact, you have to use their version number whatever it is, and as strange as it can look.

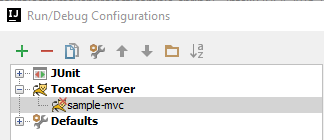
2. Open the path of the project archetype you created in IntelliJ.



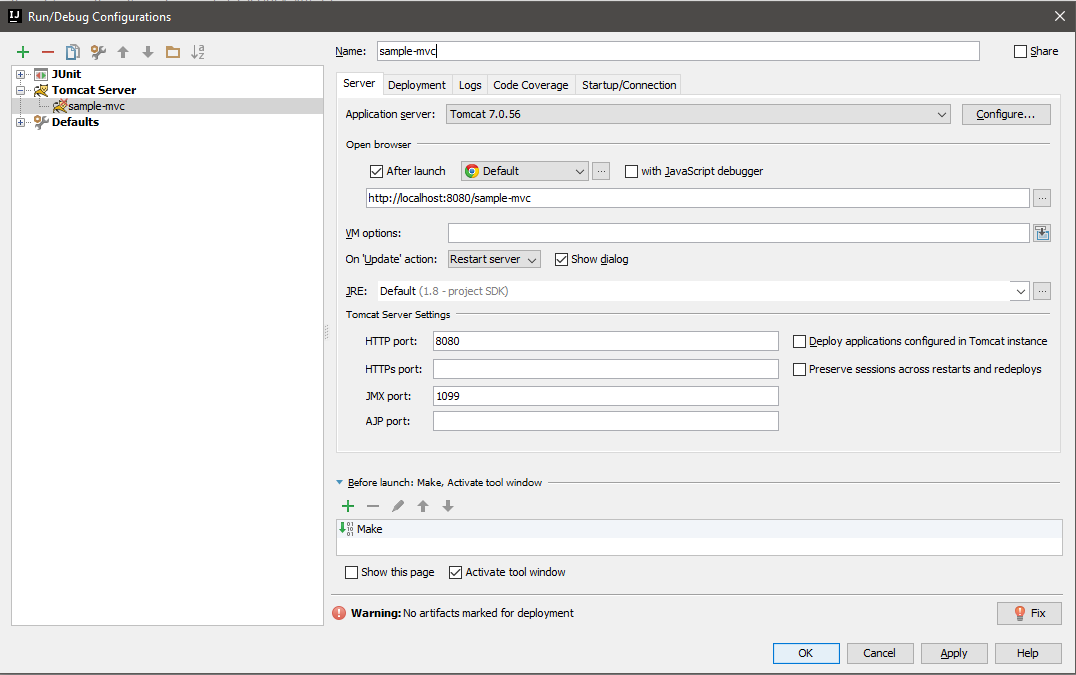
3. Run your first Spring Project. On the Menu Bar, click Run, then Edit Configurations.



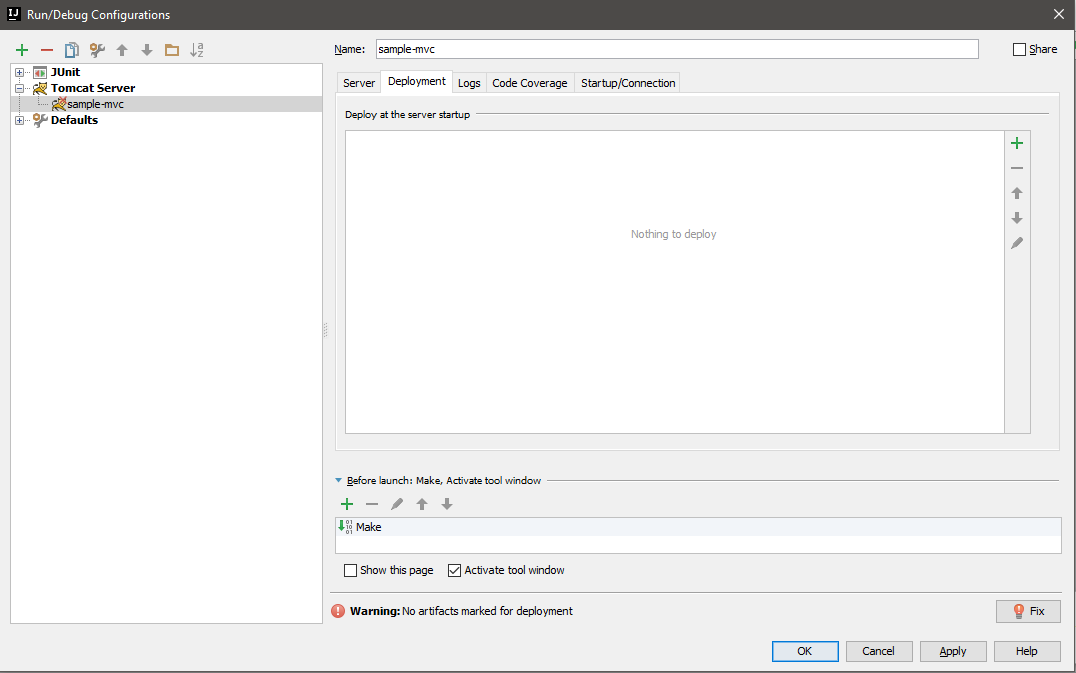
4. On the upper left corner, click the green “**+**” button then click **Tomcat > localhost**



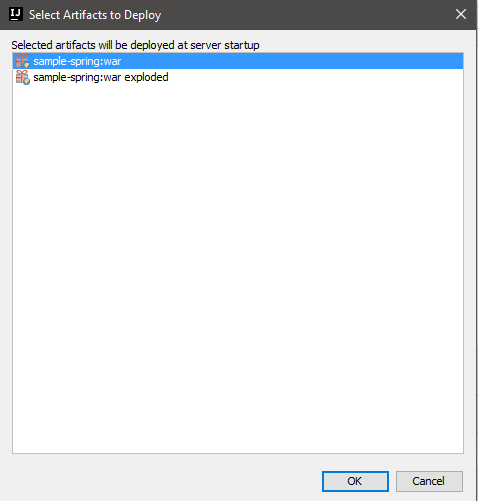
5. On the Server Tab, make sure address and port numbers are the same.



6. On the Deployment Tab, click the green “**+**” button on the right side, then click **Artifact**.



7. Select project-name:war exploded



**WAR** vs **WAR Exploded**

Project ready for server deployment Used when you want to run and modify content.

Makes it easy to for you to modify content on .war

Used for debugging