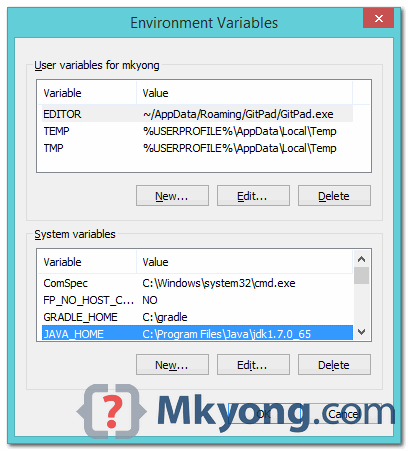
Install DockerToolbox-1.12.6.exe for Windows 8 not never version, otherwise “Docker Quickstart Terminal” gets certificate exceptions

Install Java JDK1.8

1. JDK and JAVA\_HOME

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



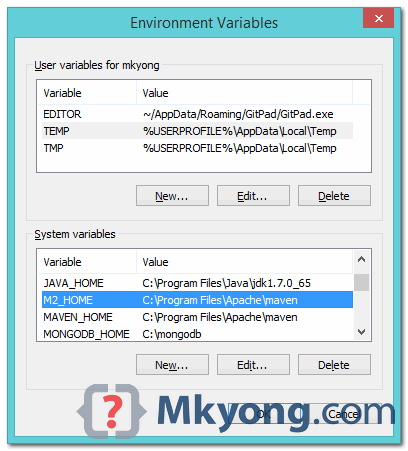
Need to add Spring Tools to Eclipse:

Visit [Maven official website](http://maven.apache.org/download.cgi), download the Maven zip file, for example : apache-maven-3.2.2-bin.zip. Unzip it to the folder you want to install Maven.

Assume you unzip to this folder – C:\Program Files\Apache\maven

3. Add M2\_HOME and MAVEN\_HOME

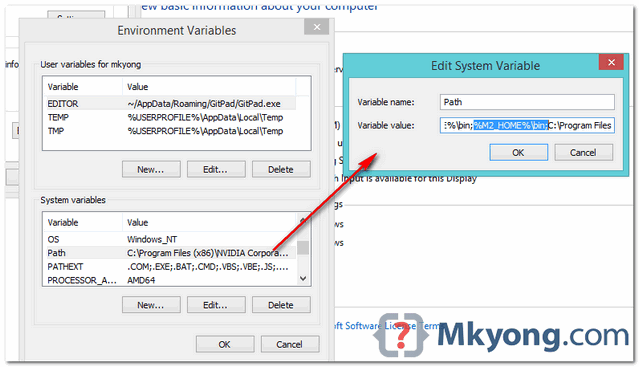
Add both M2\_HOME and MAVEN\_HOME variables in the Windows environment, and point it to your Maven folder.



**M2\_HOME or MAVEN\_HOME**  
Maven document said add M2\_HOME only, but some programs still reference Maven folder with MAVEN\_HOME, so, it’s safer to add both.

4. Add To PATH

Update PATH variable, append Maven bin folder – %M2\_HOME%\bin, so that you can run the Maven’s command everywhere.



5. Verification

Done, to verify it, run mvn –version in the command prompt.

C:\Users\mkyong>mvn -version

Apache Maven 3.2.2 (45f7c06d68e745d05611f7fd14efb6594181933e; 2014-06-17T21:51:42+08:00)

Maven home: C:\Program Files\Apache\maven

Java version: 1.7.0\_65, vendor: Oracle Corporation

Java home: C:\Program Files\Java\jdk1.7.0\_65\jre

Default locale: en\_US, platform encoding: Cp1252

OS name: "windows 8.1", version: "6.3", arch: "amd64", family: "windows"

C:\Users\mkyong>

View Video To Dockerize the Spring Boot Application:

<https://www.youtube.com/watch?v=Ojs9I-gnkc8>

Create new docker file in the root of the project called “Dockerfile” without an extension by just using the “Add file” functionality.

Add the following to the Dockerfile:

FROM openjdk:8

🡪 says which base image this docker imager will be created from

EXPOSE 8080

🡪says that the container listens on this port at runtime

ADD /target/directory-info-service.jar directory-info-service.jar

* ADD[“<src>”,”<dest>”]. Copies files URLs from <src> and adds them to the filesystem of the container at the path specified in <dest>
* This is done to copy the project’s compiled JAR file which mavern creates to the docker image

ENTRYPOINT ["java","-jar","directory-info-service.jar"]

🡪The ENTRYPOINT allows us to configure a container which will run as an executable

Install Mavern in windows:

<https://www.mkyong.com/maven/how-to-install-maven-in-windows/>

Then open “Docker Quickstart Terminal”

* Go to the source code directory (cd /c/Users/Guillaume/workspace/directory-info-service/)
* Run: mvn clean
  + This does a clean and downloads the required dependencies
* Run: mvn install
  + This installs the decencies for the project, builds the project and creates the directory-info-service.jar file which contains all **of the source code and references**
* Run: docker build -f <Dockerfile name> -t <Docker image name (has to be lowercase)> . (**notice the space dot at the end**)
* Thus run: docker build -f Dockerfile -t directoryinfoservice .
  + This builds the docker image
* Run: docker images
  + Gets a list of docker images
  + You should see the directoryinfoservice image in the list if it was successfully built
* Run: docker run -p <publish exposed port> <dockerimage>
* Thus Run: docker run -p 8080:8080 directoryinfoservice
  + This will run the docker image and expose port 8080 on the docker image as port 8080 to the “outside world”
* Press Ctrl + C (to allow you to type and see the input)
* Run: docker-machine ls
  + This will give you the IP address of the docker machine (192.168.99.100:2376 on my machine)
* Open a web browser and run: http://<docker machine ip>:8080/svc/v1/directoryinfo/getDirectoryInfo/?directory=C:/temp

Add this to the pom.xml ????

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-mavern-plugin</artifactId>

</plugin>

</plugins>

<finalName>directory-info-service</finalName>

</build>