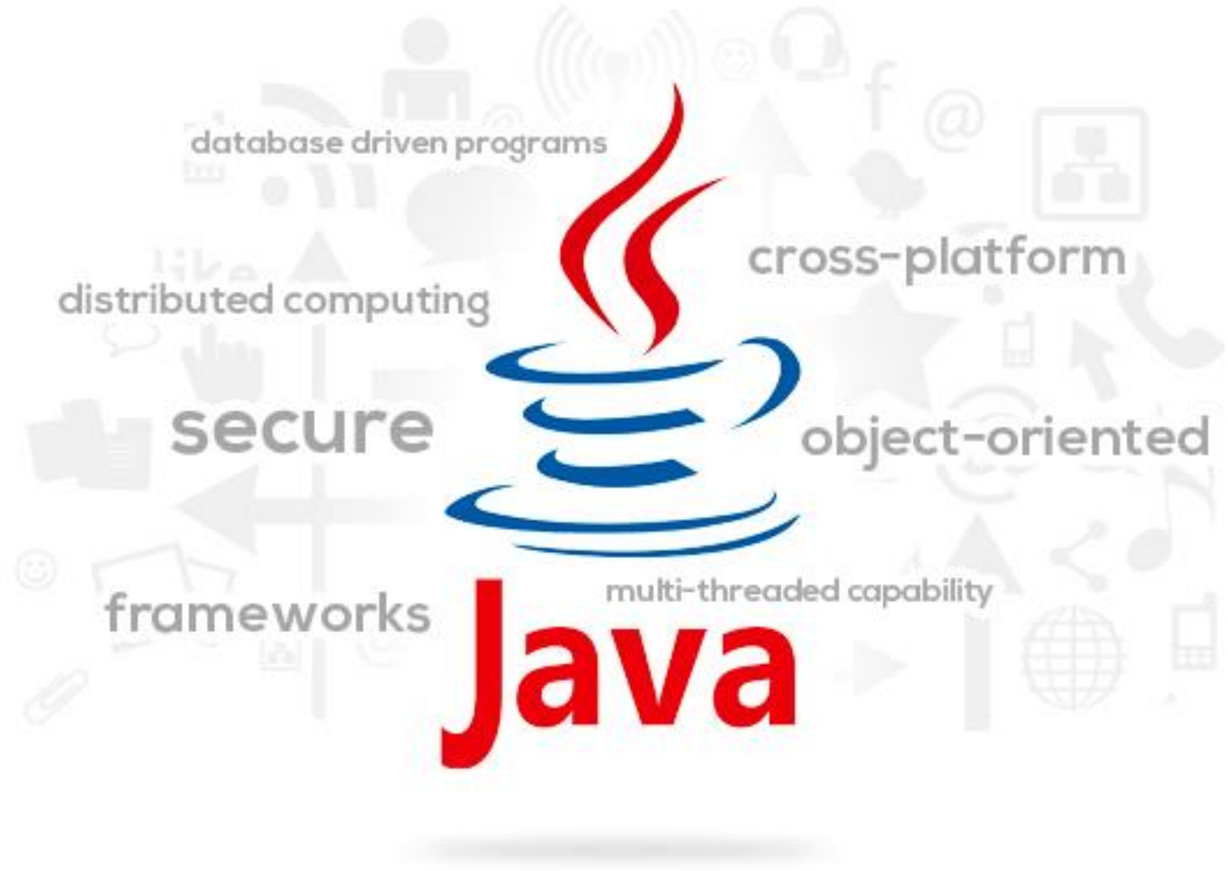


Yadab Raj Ojha

Sr. Java Developer / Lecture
Email: yadabrajojha@gmail.com
Blog: http://yro-tech.blogspot.com/
Java Tutorial: https://github.com/yrojha4ever/JavaStud
LinkedIn: https://www.linkedin.com/in/yrojha
Twitter: https://twitter.com/yrojha4ever
Website: http://javaenvagilist.com/

Maven:

- Introduction.
- Understanding problem without maven.
- Build Tool
- Install and verify Maven
- Maven Build Life cycle and Standard Directory structure
- POM
- Maven plugins and goal and phases.
- Repository
- Create Maven Projects
- Maven Commands

YROJHA: yadabrajoj

Maven

- Maven is a powerful *project management tool* that is based on POM (project object model). Current version of Maven is 3.
- It is used for projects build, dependency and documentation.
- Maven is more advanced than ant build.
- Maven provides a way to help with managing: Builds, Documentation, Reporting, Dependencies, and Distribution.
- Advantages of Maven: Reuse, Dependency Management, Build Life Cycle Management.

il.com

Understanding the problem without Maven

There are many problems that we face during the project development. They are discussed below:

1) Adding set of Jars in each project: In case of struts, spring, hibernate frameworks, we need to add set of jar files in each project. It must include all the dependencies of jars also.

2) Creating the right project structure: We must create the right project structure in servlet, struts etc, otherwise it will not be executed.

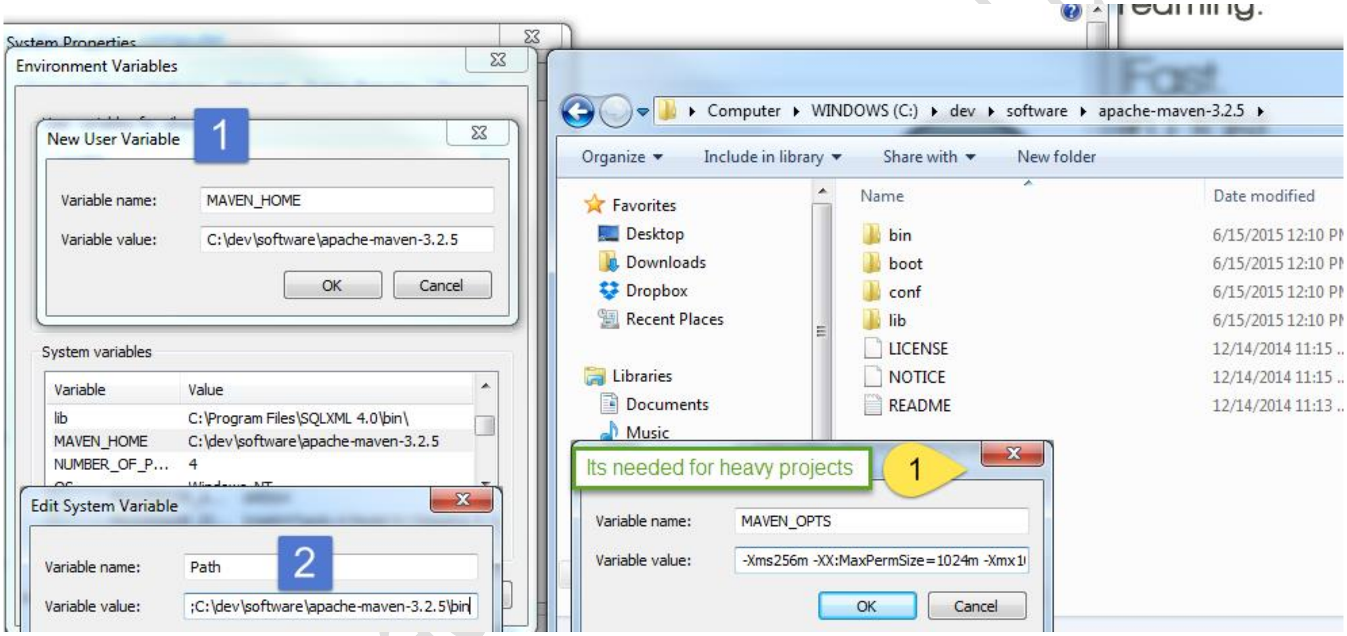
3) Building and Deploying the project: We must have to build and deploy the project so that it may work.

What is Build Tool

A build tool takes care of everything for building a process. It does following:

- Generates source code (if auto-generated code is used)
- Generates documentation from source code
- Compiles source code
- Packages compiled code into JAR of ZIP file
- Installs the packaged code in local repository, server repository, or central repository

- http://maven.apache.org/download.html
- **Add MAVEN_HOME environment in Environment Variables:(Installation folder/apache_maven_2.2.1)**
- **Add M2 environment variable in User variable with value %MAVEN_HOME%/bin**
- **Update “Path” environment variable and prepend the value %M2% to add Maven available in command line**
- **JAVA_HOME to location of JDK and %JAVA_HOME%\bin in PATH environment variable**
- **Open a new cmd and run mvn –version to verify the installation.**



New User Variable

Variable name: MAVEN_HOME

Variable value: C:\dev\software\apache-maven-3.2.5

Variable	Value
lib	C:\Program Files\SQLXML 4.0\bin\
MAVEN_HOME	C:\dev\software\apache-maven-3.2.5
NUMBER_OF_P...	4

Edit System Variable

Variable name: Path

Variable value: ;C:\dev\software\apache-maven-3.2.5\bin

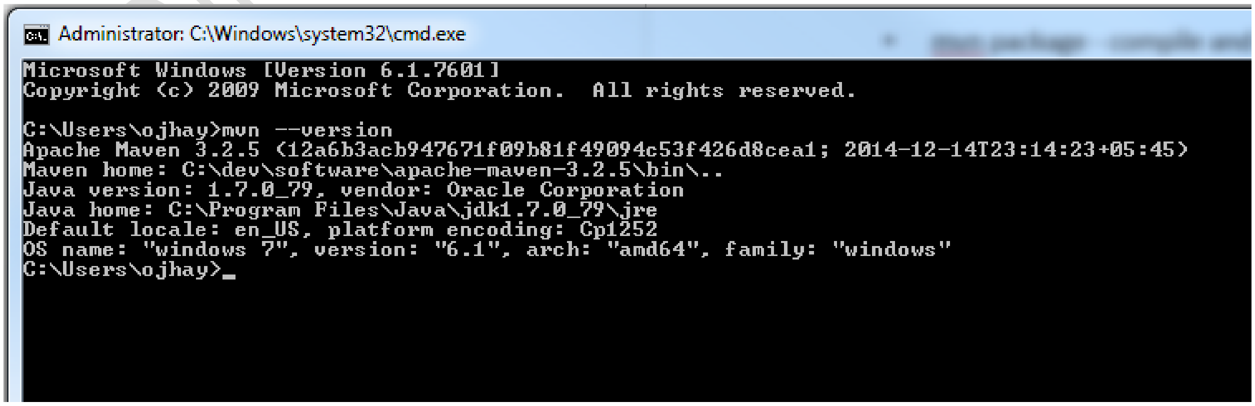
Name	Date modified
bin	6/15/2015 12:10 PM
boot	6/15/2015 12:10 PM
conf	6/15/2015 12:10 PM
lib	6/15/2015 12:10 PM
LICENSE	12/14/2014 11:15 ...
NOTICE	12/14/2014 11:15 ...
README	12/14/2014 11:13 ...

Variable name: MAVEN_OPTS

Variable value: -Xms256m -XX:MaxPermSize=1024m -Xmx1

-Xms256m -XX:MaxPermSize=1024m -Xmx1024m

Verify Maven:



Administrator: C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\ojhay>mvn --version
Apache Maven 3.2.5 (12a6b3acb947671f09b81f49094c53f426d8cea1; 2014-12-14T23:14:23+05:45)
Maven home: C:\dev\software\apache-maven-3.2.5\bin\..
Java version: 1.7.0_79, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.7.0_79\jre
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 7", version: "6.1", arch: "amd64", family: "windows"
C:\Users\ojhay>_
```

Maven Build Life Cycle

Default Life Cycle Phases

- **Validate**
- **Compile**
- **Test**
- **Package**
- **Integration-test**
- **Verify**
- **Install**
- **Deploy**

Two other Maven Life Cycle

- **Clean**
- **Site**

Maven Standard Directory Layout

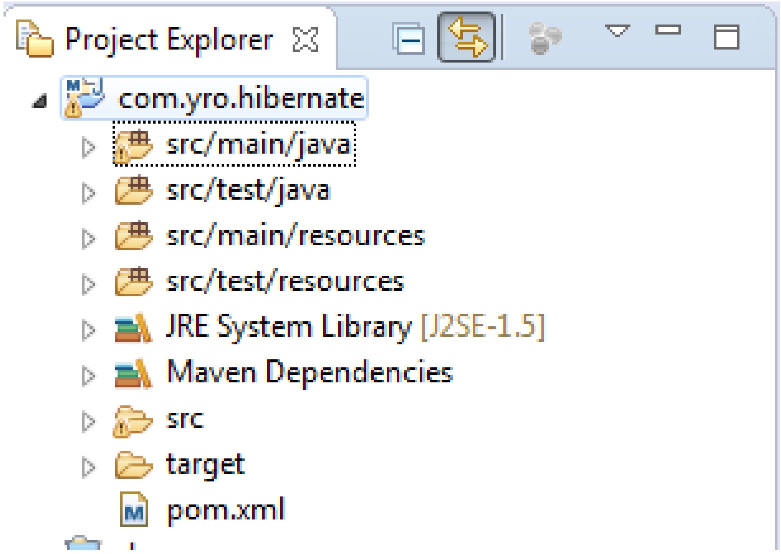
Src

- **src/main/java**
- **src/main/resources**
- **src/main/webapp**
- **src/test/java**
- **src/test/filter**
- **src/site**
- **src/main/filters_resource**
- **src/main/assembly**
- **src/main/config**

Target

Sample Maven Project Structure

YROJHA: yadabrajojha@gmail.com

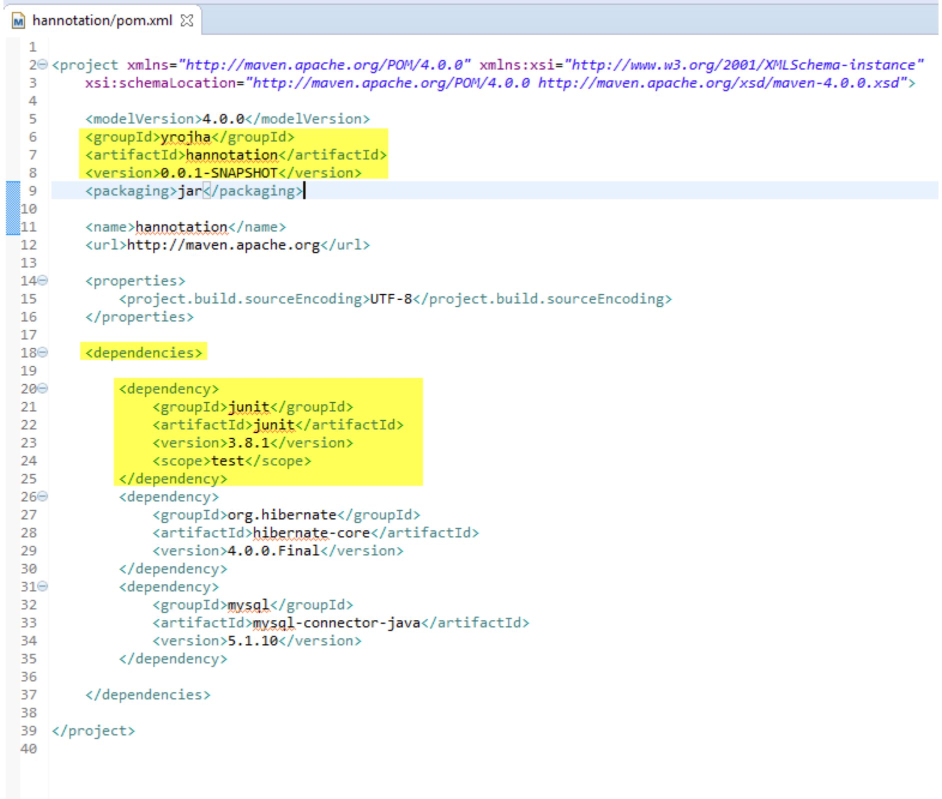


POM(Project Object Model)

- POM is an xml file which is the core of project's configuration.
- It is a single configuration file that contains the majority of information required to build a project.
- In short, POM contains every important piece of information about project.

rajojha@gmail.com

Sample POM



```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>
    <groupId>yrojha</groupId>
    <artifactId>hannotation</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>hannotation</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>

        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>4.0.0.Final</version>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>5.1.10</version>
        </dependency>

    </dependencies>

</project>
```

Mavens Goal and Phases

- A goal is a task in Maven terminology.
- Usually, goals are specified as an argument to Maven on the command line, but there are some default goals that may be called by other goals.
- A build phase is made up of goals. A goal represents a specific task which contributes to the building and managing of a project.
- Example:
- resources:resources
- surefire:test

Maven Plug-ins

- Maven Plugin is a collection of one or more goals.
- Examples of Maven Plugin can be simple core plugins like Jar plugin which contains goal for creating Jar File.
- For Instance:

Goal	**Plugin**
compile	Compiler
create	archetype

ojha@gmail.com

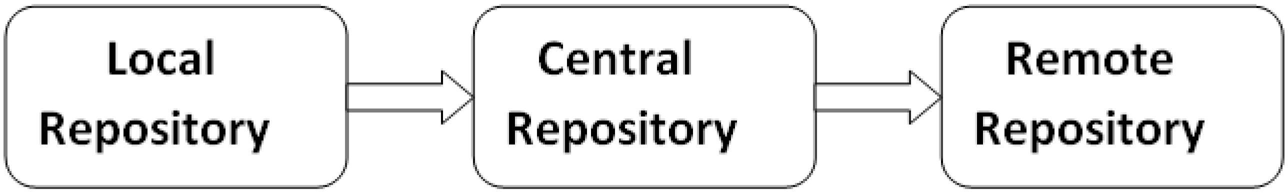
Maven Repository

A **maven repository** is a directory of packaged JAR file with pom.xml file. Maven searches for dependencies in the repositories. There are 3 types of maven repository:

1. Local Repository
2. Central Repository
3. Remote Repository

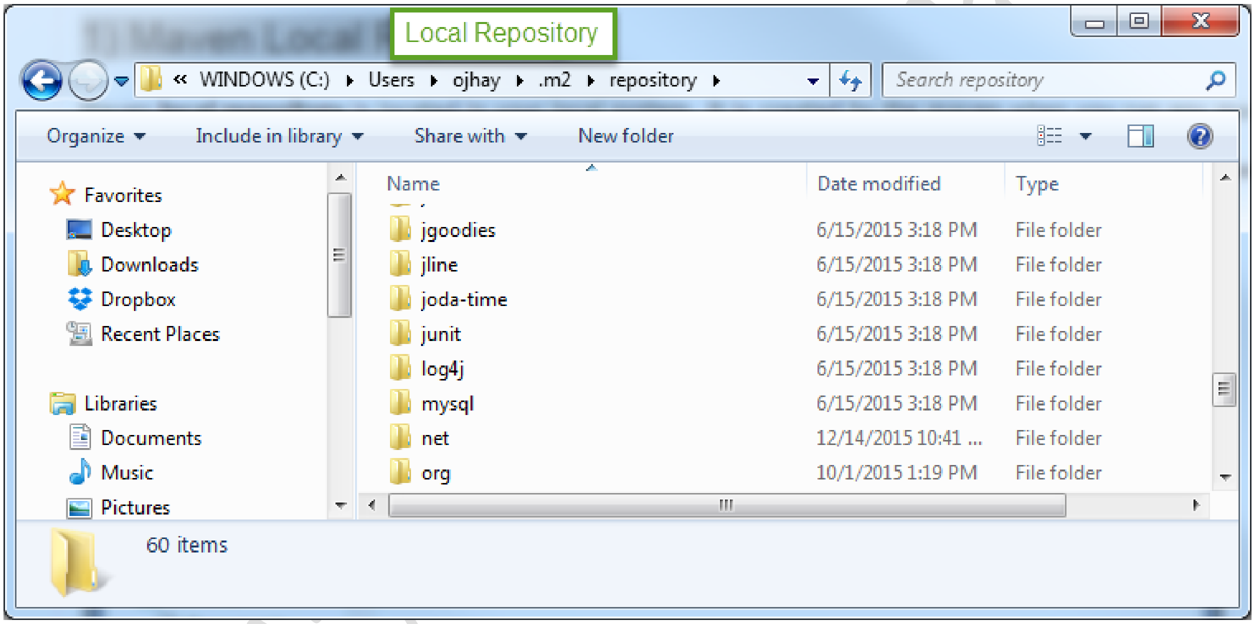
Maven searches for the dependencies in the following order:

Local repository then **Central repository** then **Remote repository**.



Local Repository → Central Repository → Remote Repository

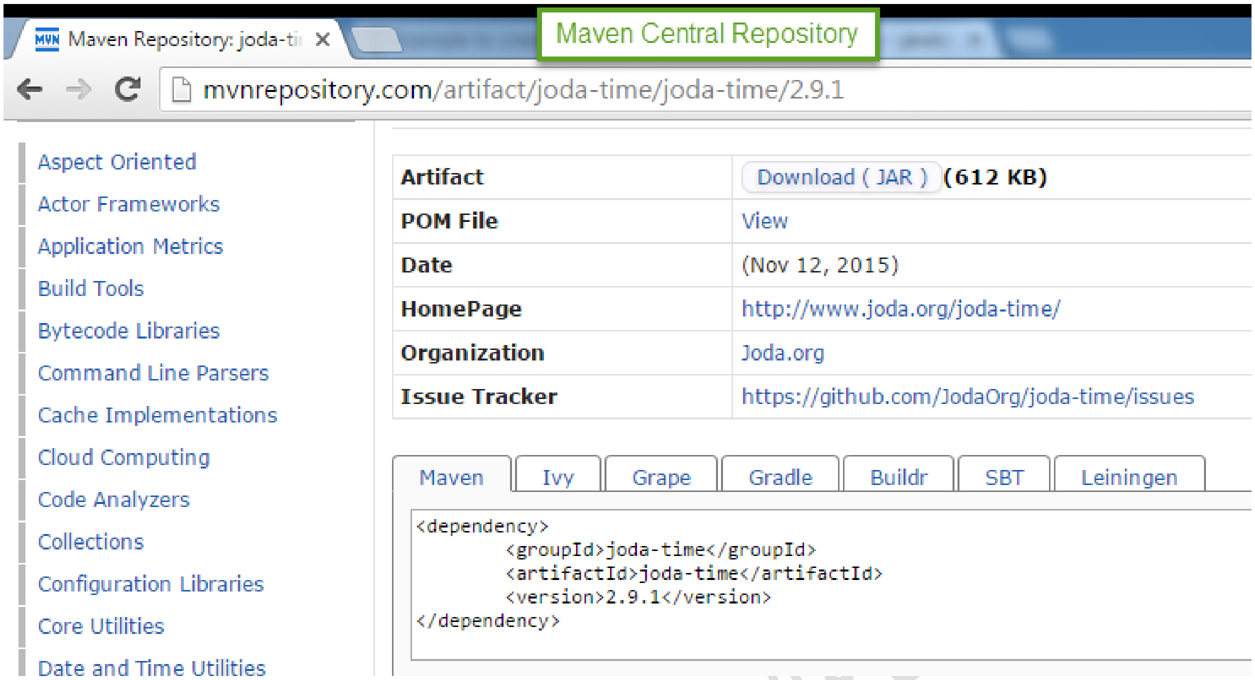
1) Maven Local Repository



Name	Date modified	Type
jgoodies	6/15/2015 3:18 PM	File folder
jline	6/15/2015 3:18 PM	File folder
joda-time	6/15/2015 3:18 PM	File folder
junit	6/15/2015 3:18 PM	File folder
log4j	6/15/2015 3:18 PM	File folder
mysql	6/15/2015 3:18 PM	File folder
net	12/14/2015 10:41 ...	File folder
org	10/1/2015 1:19 PM	File folder

60 items

2) Maven Central Repository

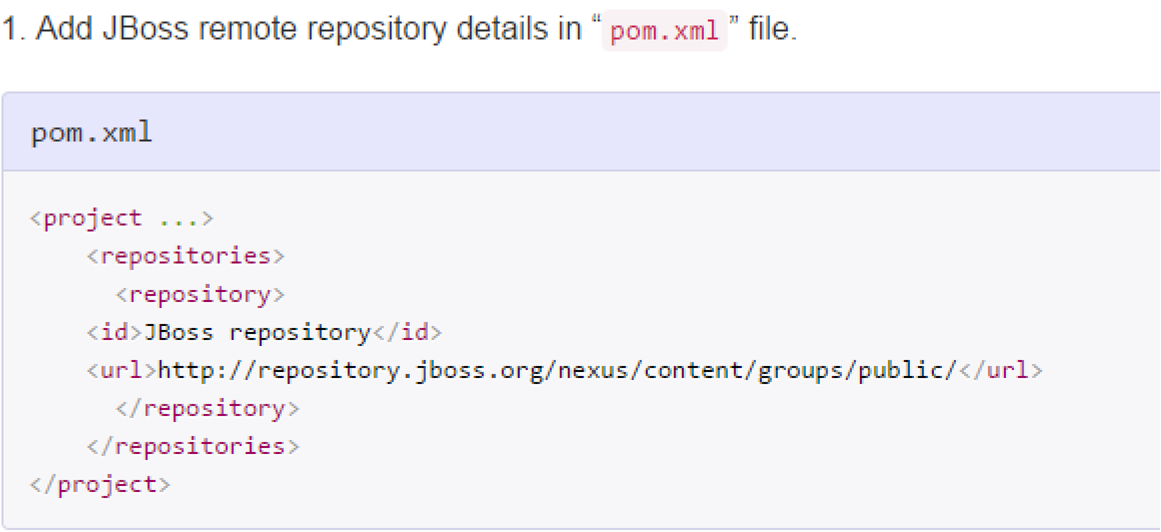


Artifact	Download (JAR) **(612 KB)**
POM File	View
Date	(Nov 12, 2015)
HomePage	http://www.joda.org/joda-time/
Organization	Joda.org
Issue Tracker	https://github.com/JodaOrg/joda-time/issues

```
<dependency>
        <groupId>joda-time</groupId>
        <artifactId>joda-time</artifactId>
        <version>2.9.1</version>
</dependency>
```

3) Maven Remote Repository

Maven **remote repository** is located on the web. Most of libraries can be missing from the central repository such as JBoss library etc, so we need to define remote repository in pom.xml file.



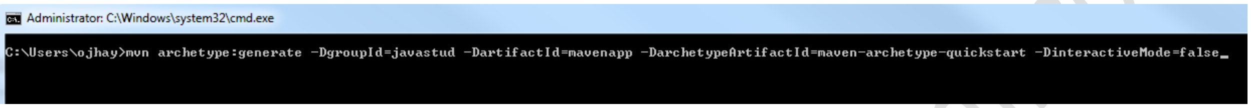
1. Add JBoss remote repository details in "pom.xml" file.

pom.xml

```
<project ...>
    <repositories>
      <repository>
    <id>JBoss repository</id>
    <url>http://repository.jboss.org/nexus/content/groups/public/</url>
      </repository>
    </repositories>
</project>
```

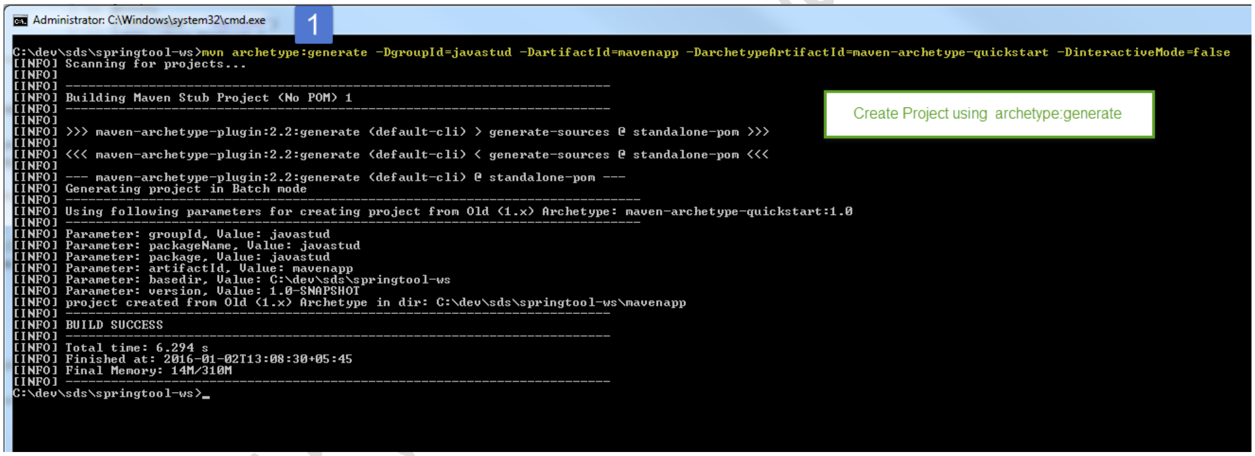
Creating Maven Project

- Archetype Plugin is used. Plugin goal is archetype:create
- It takes parameter in form of -Dname=value
- Creates a simple Maven Project.
- GroupId: can be name of the organization or dns name in reverse format. Used as package name if no explicit package name is defined.
- ArtifactId: Unique name of the project



```
C:\Users\ojhay>mvn archetype:generate -DgroupId=javastud -DartifactId=mavenapp -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false_
```

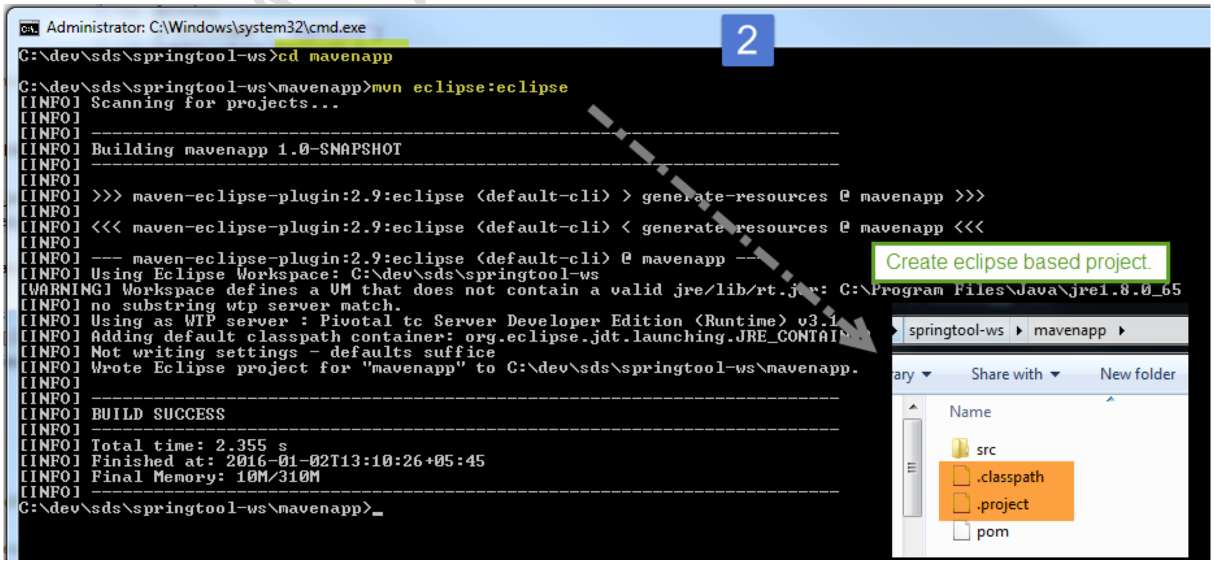
```
mvn archetype:generate -DgroupId=javastud -DartifactId=mavenapp
-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```



1

Create Project using archetype:generate

```
C:\dev\sds\springtool-ws>mvn archetype:generate -DgroupId=javastud -DartifactId=mavenapp -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building Maven Stub Project (No POM) 1
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] >>> maven-archetype-plugin:2.2:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:2.2:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO] --- maven-archetype-plugin:2.2:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-quickstart:1.0
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: groupId, Value: javastud
[INFO] Parameter: packageName, Value: javastud
[INFO] Parameter: package, Value: javastud
[INFO] Parameter: artifactId, Value: mavenapp
[INFO] Parameter: basedir, Value: C:\dev\sds\springtool-ws
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\dev\sds\springtool-ws\mavenapp
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 6.294 s
[INFO] Finished at: 2016-01-02T13:08:30+05:45
[INFO] Final Memory: 14M/310M
[INFO] ------------------------------------------------------------------------
C:\dev\sds\springtool-ws>_
```



2

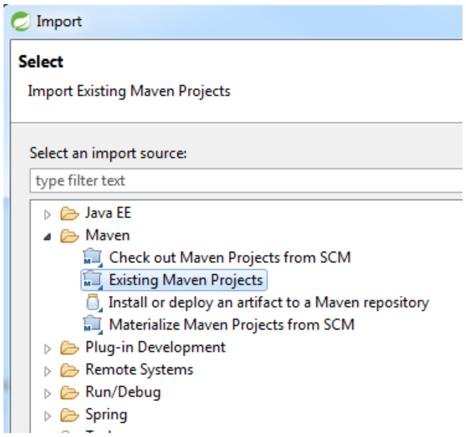
Create eclipse based project.

```
C:\dev\sds\springtool-ws>cd mavenapp

C:\dev\sds\springtool-ws\mavenapp>mvn eclipse:eclipse
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building mavenapp 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] >>> maven-eclipse-plugin:2.9:eclipse (default-cli) > generate-resources @ mavenapp >>>
[INFO]
[INFO] <<< maven-eclipse-plugin:2.9:eclipse (default-cli) < generate-resources @ mavenapp <<<
[INFO]
[INFO] --- maven-eclipse-plugin:2.9:eclipse (default-cli) @ mavenapp ---
[INFO] Using Eclipse Workspace: C:\dev\sds\springtool-ws
[WARNING] Workspace defines a VM that does not contain a valid jre/lib/rt.jar: C:\Program Files\Java\jre1.8.0_65
[INFO] no substring wtp server match.
[INFO] Using as WTP server : Pivotal tc Server Developer Edition (Runtime) v3.1
[INFO] Adding default classpath container: org.eclipse.jdt.launching.JRE_CONTAI
[INFO] Not writing settings - defaults suffice
[INFO] Wrote Eclipse project for "mavenapp" to C:\dev\sds\springtool-ws\mavenapp.
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2.355 s
[INFO] Finished at: 2016-01-02T13:10:26+05:45
[INFO] Final Memory: 10M/310M
[INFO] ------------------------------------------------------------------------
C:\dev\sds\springtool-ws\mavenapp>_
```

Now you can this maven project in eclipse using File>Import>General> Existing Project into Workspace>

Browse project.

Or you can do this:



Import

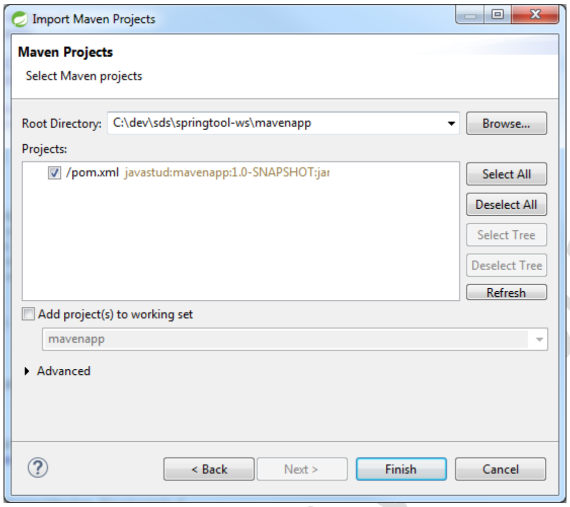
Select

Import Existing Maven Projects

Select an import source:

type filter text

- ▷ Java EE
- ◢ Maven
  - Check out Maven Projects from SCM
  - Existing Maven Projects
  - Install or deploy an artifact to a Maven repository
  - Materialize Maven Projects from SCM
- ▷ Plug-in Development
- ▷ Remote Systems
- ▷ Run/Debug
- ▷ Spring



Import Maven Projects

Maven Projects

Select Maven projects

Root Directory: C:\dev\sds\springtool-ws\mavenapp [Browse...]

Projects:

☑ /pom.xml javastud:mavenapp:1.0-SNAPSHOT:jar

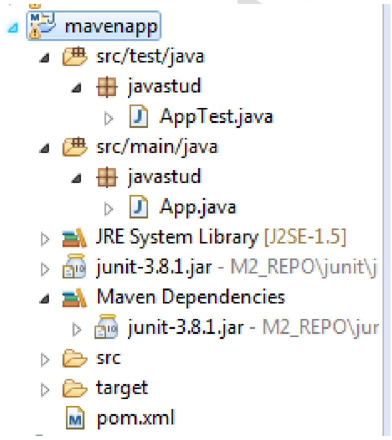
[Select All] [Deselect All] [Select Tree] [Deselect Tree] [Refresh]

☐ Add project(s) to working set

mavenapp

▸ Advanced

[< Back] [Next >] [Finish] [Cancel]

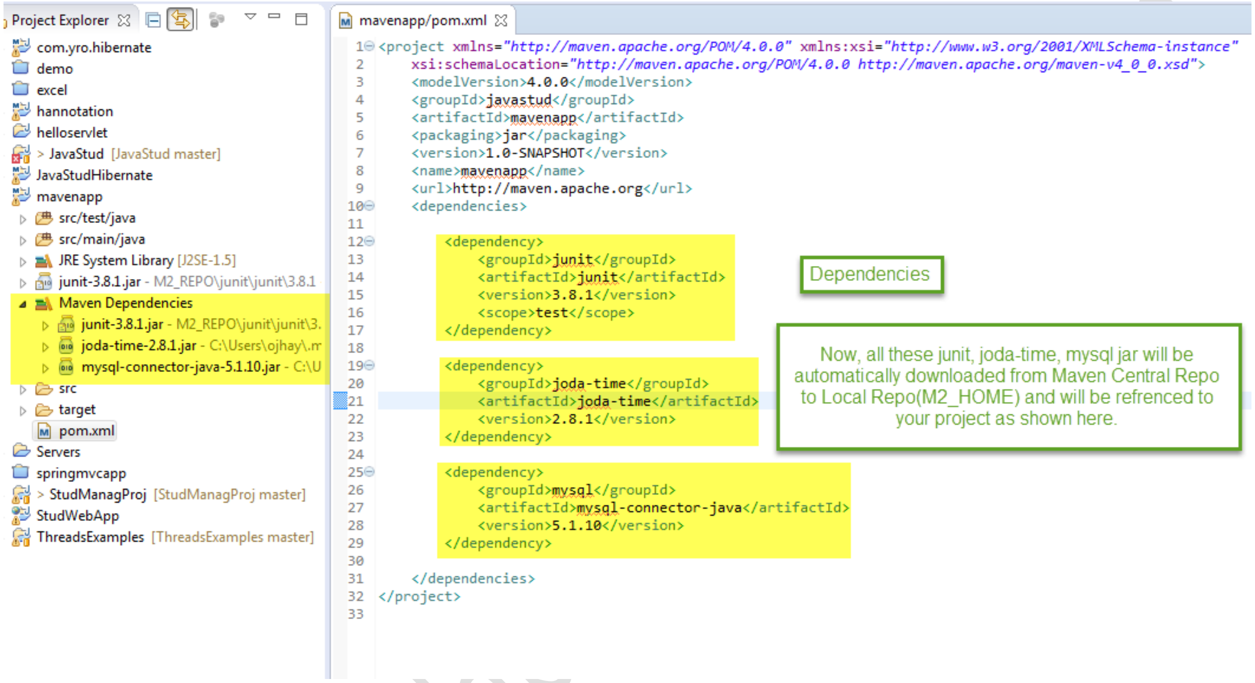


- ◢ mavenapp
  - ◢ src/test/java
    - ◢ javastud
      - ▷ AppTest.java
  - ◢ src/main/java
    - ◢ javastud
      - ▷ App.java
  - ▷ JRE System Library [J2SE-1.5]
  - ▷ junit-3.8.1.jar - M2_REPO\junit\j
  - ◢ Maven Dependencies
    - ▷ junit-3.8.1.jar - M2_REPO\jur
  - ▷ src
  - ▷ target
  - pom.xml

rajojha@gmail.com

Dependency

- Project have dependencies which are defined in pom.xml .
- For adding dependency, following should be defined.
- GroupId, ArtifactId, Version, Scope
- Maven supports transitive dependencies.


Project Explorer
com.yro.hibernate
demo
excel
hannotation
helloservlet
> JavaStud [JavaStud master]
JavaStudHibernate
mavenapp
src/test/java
src/main/java
JRE System Library [J2SE-1.5]
junit-3.8.1.jar - M2_REPO\junit\junit\3.8.1
Maven Dependencies
junit-3.8.1.jar - M2_REPO\junit\junit\3.
joda-time-2.8.1.jar - C:\Users\ojhay\.m
mysql-connector-java-5.1.10.jar - C:\U
src
target
pom.xml
Servers
springmvcapp
> StudManagProj [StudManagProj master]
StudWebApp
ThreadsExamples [ThreadsExamples master]
mavenapp/pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>javastud</groupId>
    <artifactId>mavenapp</artifactId>
    <packaging>jar</packaging>
    <version>1.0-SNAPSHOT</version>
    <name>mavenapp</name>
    <url>http://maven.apache.org</url>
    <dependencies>

        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>joda-time</groupId>
            <artifactId>joda-time</artifactId>
            <version>2.8.1</version>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>5.1.10</version>
        </dependency>

    </dependencies>
</project>
```

Dependencies

Now, all these junit, joda-time, mysql jar will be automatically downloaded from Maven Central Repo to Local Repo(M2_HOME) and will be refrenced to your project as shown here.

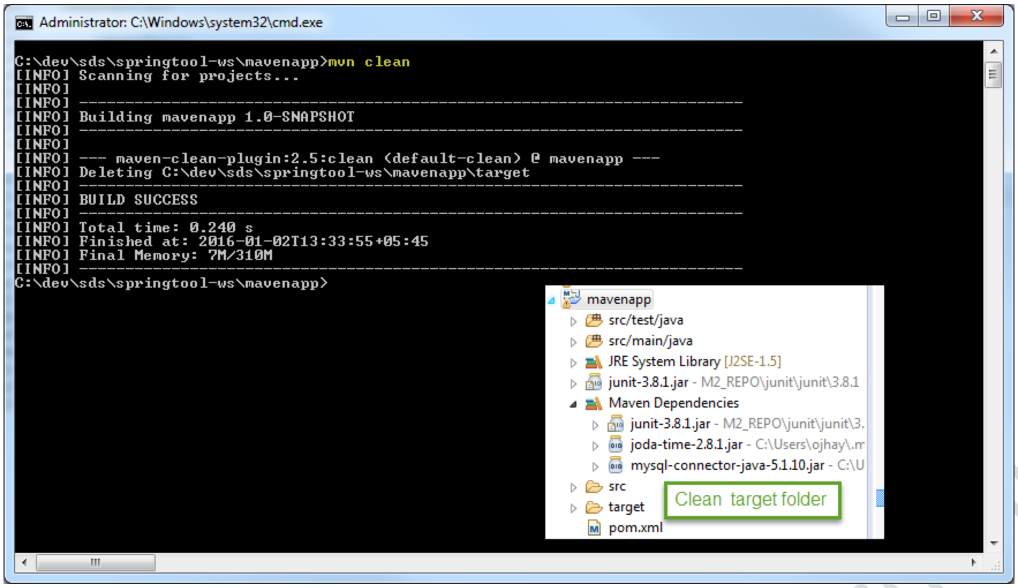
Adding Test-Scoped Dependencies

- A test scoped dependency is available only during test-compilation & test-execution.
- If the project has war or ear packaging, a test-scoped dependency would not be included in the projects output archive.

YROJHA.yada

Common Maven commands:

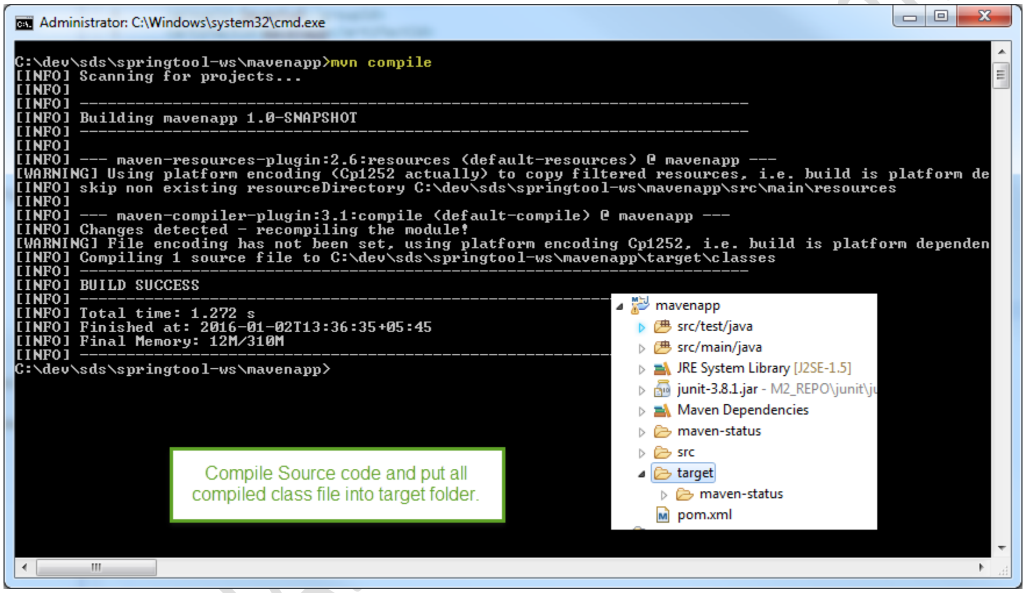
- mvn package - compile and create jars/wars
- mvn install - package + copy to local repo
- mvn clean - remove target directory
- mvn test - run unit tests
- mvn eclipse:eclipse - create Eclipse project files



```
C:\dev\sds\springtool-ws\mavenapp>mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building mavenapp 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ mavenapp ---
[INFO] Deleting C:\dev\sds\springtool-ws\mavenapp\target
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 0.240 s
[INFO] Finished at: 2016-01-02T13:33:55+05:45
[INFO] Final Memory: 7M/310M
[INFO] ------------------------------------------------------------------------
C:\dev\sds\springtool-ws\mavenapp>
```

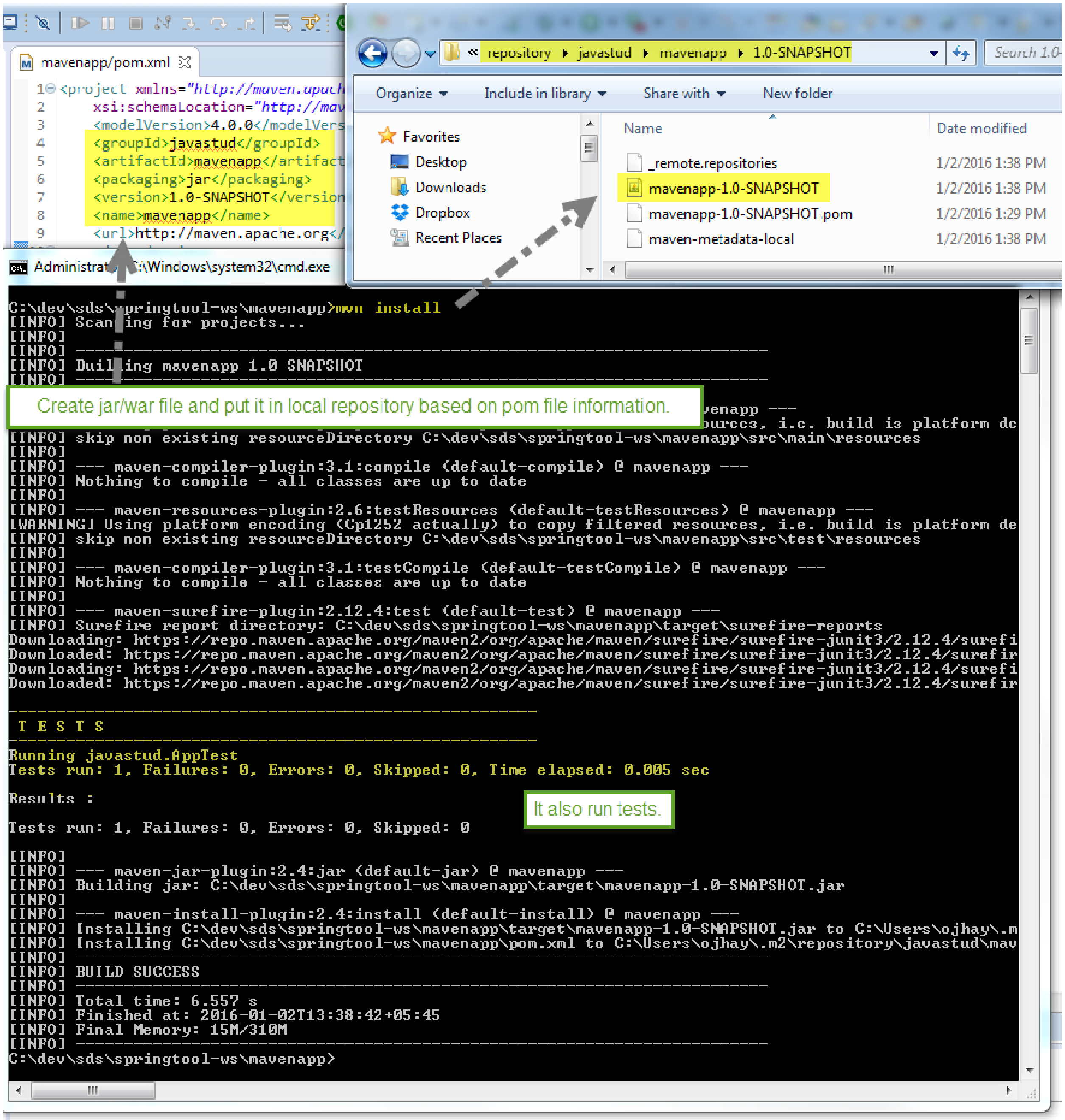
mavenapp
src/test/java
src/main/java
JRE System Library [J2SE-1.5]
junit-3.8.1.jar - M2_REPO\junit\junit\3.8.1
Maven Dependencies
junit-3.8.1.jar - M2_REPO\junit\junit\3.
joda-time-2.8.1.jar - C:\Users\ojhay\.m
mysql-connector-java-5.1.10.jar - C:\U
src
target
pom.xml

Clean target folder



```
C:\dev\sds\springtool-ws\mavenapp>mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building mavenapp 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ mavenapp ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform de
[INFO] skip non existing resourceDirectory C:\dev\sds\springtool-ws\mavenapp\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ mavenapp ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependen
[INFO] Compiling 1 source file to C:\dev\sds\springtool-ws\mavenapp\target\classes
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ---------------------------------------------------------------
[INFO] Total time: 1.272 s
[INFO] Finished at: 2016-01-02T13:36:35+05:45
[INFO] Final Memory: 12M/310M
[INFO] ---------------------------------------------------------------
C:\dev\sds\springtool-ws\mavenapp>
```

Compile Source code and put all compiled class file into target folder.



mavenapp/pom.xml

```
<project xmlns="http://maven.apach
    xsi:schemaLocation="http://mav
    <modelVersion>4.0.0</modelVers
    <groupId>javastud</groupId>
    <artifactId>mavenapp</artifact
    <packaging>jar</packaging>
    <version>1.0-SNAPSHOT</version
    <name>mavenapp</name>
    <url>http://maven.apache.org</
```

Name	Date modified
_remote.repositories	1/2/2016 1:38 PM
mavenapp-1.0-SNAPSHOT	1/2/2016 1:38 PM
mavenapp-1.0-SNAPSHOT.pom	1/2/2016 1:29 PM
maven-metadata-local	1/2/2016 1:38 PM

Administrator: C:\Windows\system32\cmd.exe

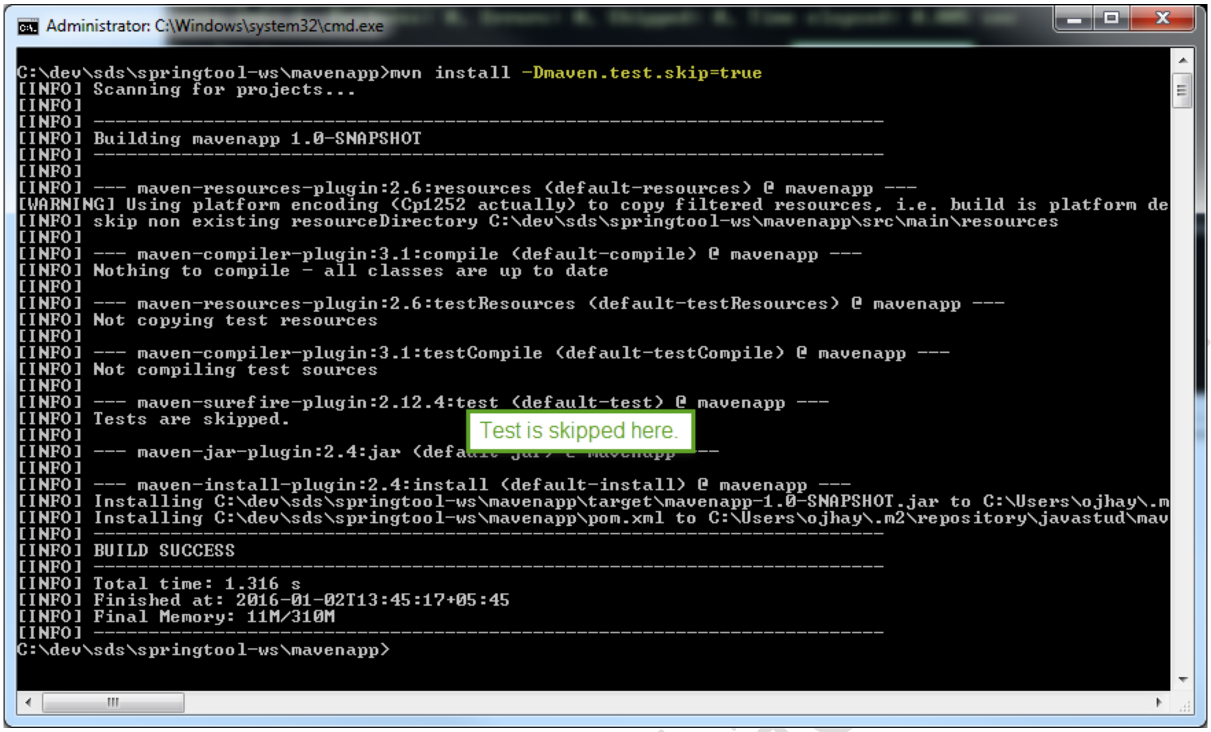
```
C:\dev\sds\springtool-ws\mavenapp>mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building mavenapp 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
                                                                        enapp ---
                                                                        ources, i.e. build is platform de
[INFO] skip non existing resourceDirectory C:\dev\sds\springtool-ws\mavenapp\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ mavenapp ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ mavenapp ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform de
[INFO] skip non existing resourceDirectory C:\dev\sds\springtool-ws\mavenapp\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ mavenapp ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ mavenapp ---
[INFO] Surefire report directory: C:\dev\sds\springtool-ws\mavenapp\target\surefire-reports
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit3/2.12.4/surefi
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit3/2.12.4/surefir
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit3/2.12.4/surefi
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit3/2.12.4/surefir

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running javastud.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.005 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ mavenapp ---
[INFO] Building jar: C:\dev\sds\springtool-ws\mavenapp\target\mavenapp-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ mavenapp ---
[INFO] Installing C:\dev\sds\springtool-ws\mavenapp\target\mavenapp-1.0-SNAPSHOT.jar to C:\Users\ojhay\.m
[INFO] Installing C:\dev\sds\springtool-ws\mavenapp\pom.xml to C:\Users\ojhay\.m2\repository\javastud\mav
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 6.557 s
[INFO] Finished at: 2016-01-02T13:38:42+05:45
[INFO] Final Memory: 15M/310M
[INFO] ------------------------------------------------------------------------
C:\dev\sds\springtool-ws\mavenapp>
```

Create jar/war file and put it in local repository based on pom file information.

It also run tests.

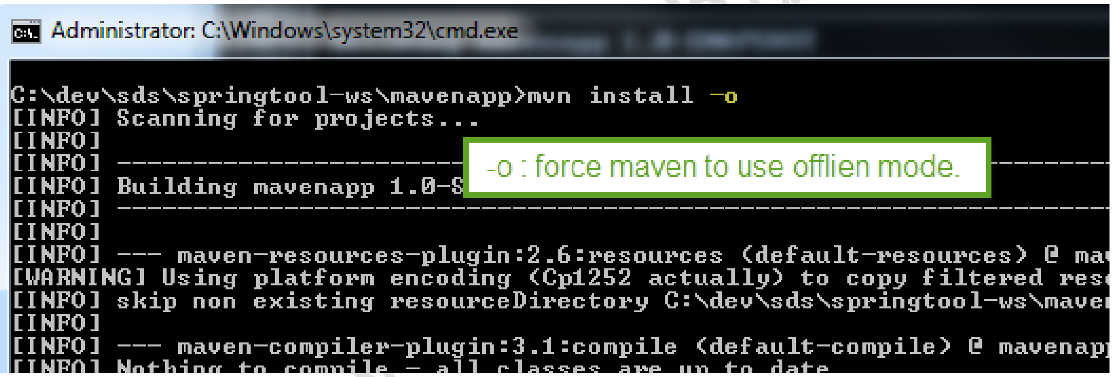
YRO

To Ignore Test: -Dmaven.test.skip=true



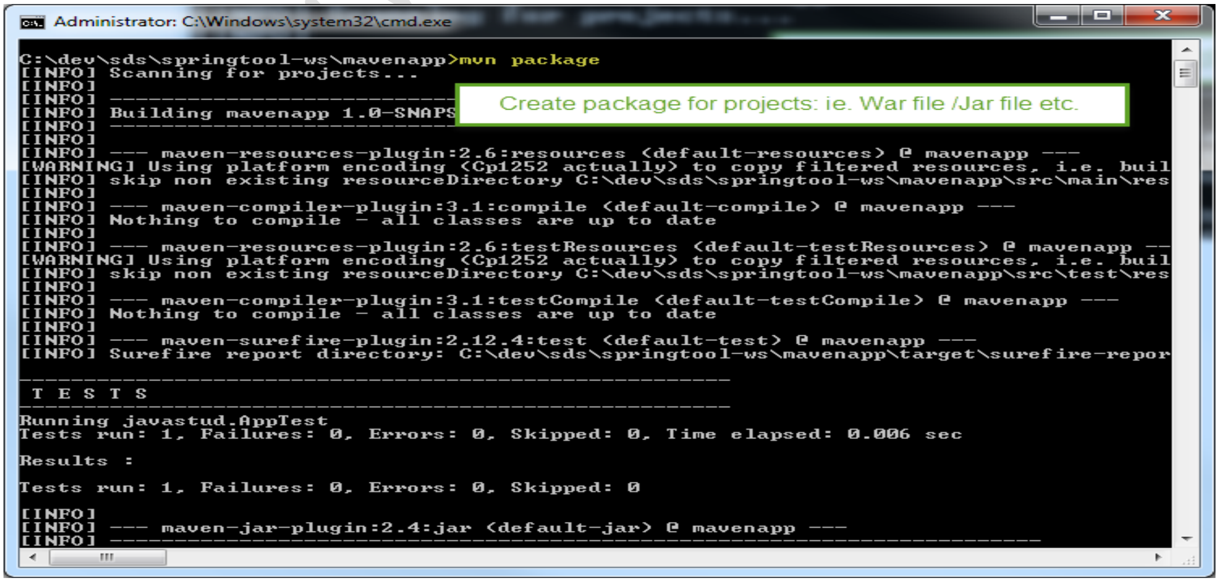
```
C:\dev\sds\springtool-ws\mavenapp>mvn install -Dmaven.test.skip=true
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building mavenapp 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ mavenapp ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform de
[INFO] skip non existing resourceDirectory C:\dev\sds\springtool-ws\mavenapp\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ mavenapp ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ mavenapp ---
[INFO] Not copying test resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ mavenapp ---
[INFO] Not compiling test sources
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ mavenapp ---
[INFO] Tests are skipped.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (defa
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ mavenapp ---
[INFO] Installing C:\dev\sds\springtool-ws\mavenapp\target\mavenapp-1.0-SNAPSHOT.jar to C:\Users\ojhay\.m
[INFO] Installing C:\dev\sds\springtool-ws\mavenapp\pom.xml to C:\Users\ojhay\.m2\repository\javastud\mav
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 1.316 s
[INFO] Finished at: 2016-01-02T13:45:17+05:45
[INFO] Final Memory: 11M/310M
[INFO] ------------------------------------------------------------------------
C:\dev\sds\springtool-ws\mavenapp>
```

Test is skipped here.



```
C:\dev\sds\springtool-ws\mavenapp>mvn install -o
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------                                          --------
[INFO] Building mavenapp 1.0-S
[INFO] ----------------------------------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ ma
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered res
[INFO] skip non existing resourceDirectory C:\dev\sds\springtool-ws\mave
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ mavenap
[INFO] Nothing to compile - all classes are up to date
```

-o : force maven to use offlien mode.



```
C:\dev\sds\springtool-ws\mavenapp>mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -------------------------
[INFO] Building mavenapp 1.0-SNAPS
[INFO] -------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ mavenapp ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. buil
[INFO] skip non existing resourceDirectory C:\dev\sds\springtool-ws\mavenapp\src\main\res
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ mavenapp ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ mavenapp --
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. buil
[INFO] skip non existing resourceDirectory C:\dev\sds\springtool-ws\mavenapp\src\test\res
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ mavenapp ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ mavenapp ---
[INFO] Surefire report directory: C:\dev\sds\springtool-ws\mavenapp\target\surefire-repor

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running javastud.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.006 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ mavenapp ---
[INFO] ------------------------------------------------------------------------
```

Create package for projects: ie. War file /Jar file etc.