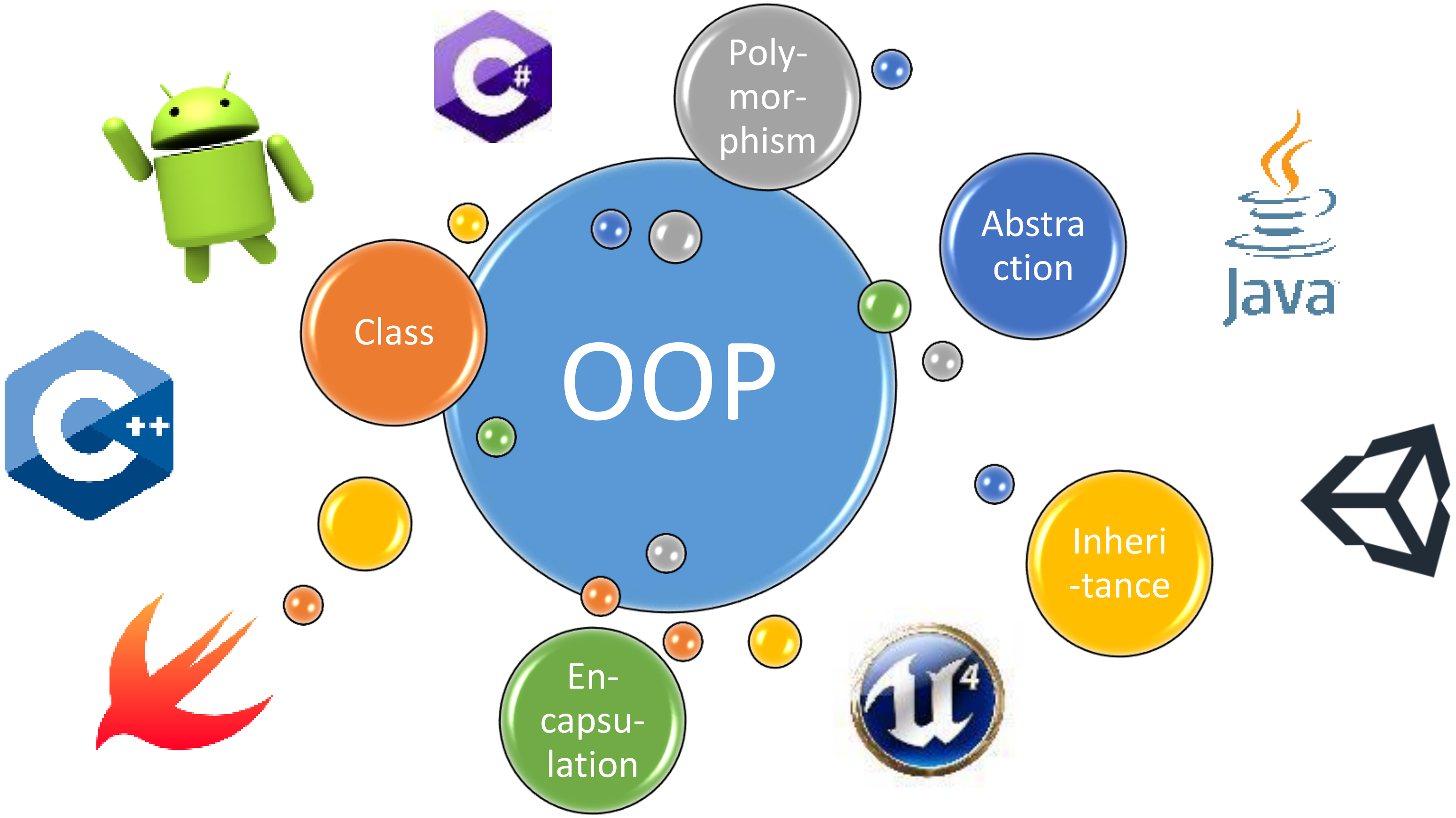


OOP

Class

Poly-morphism

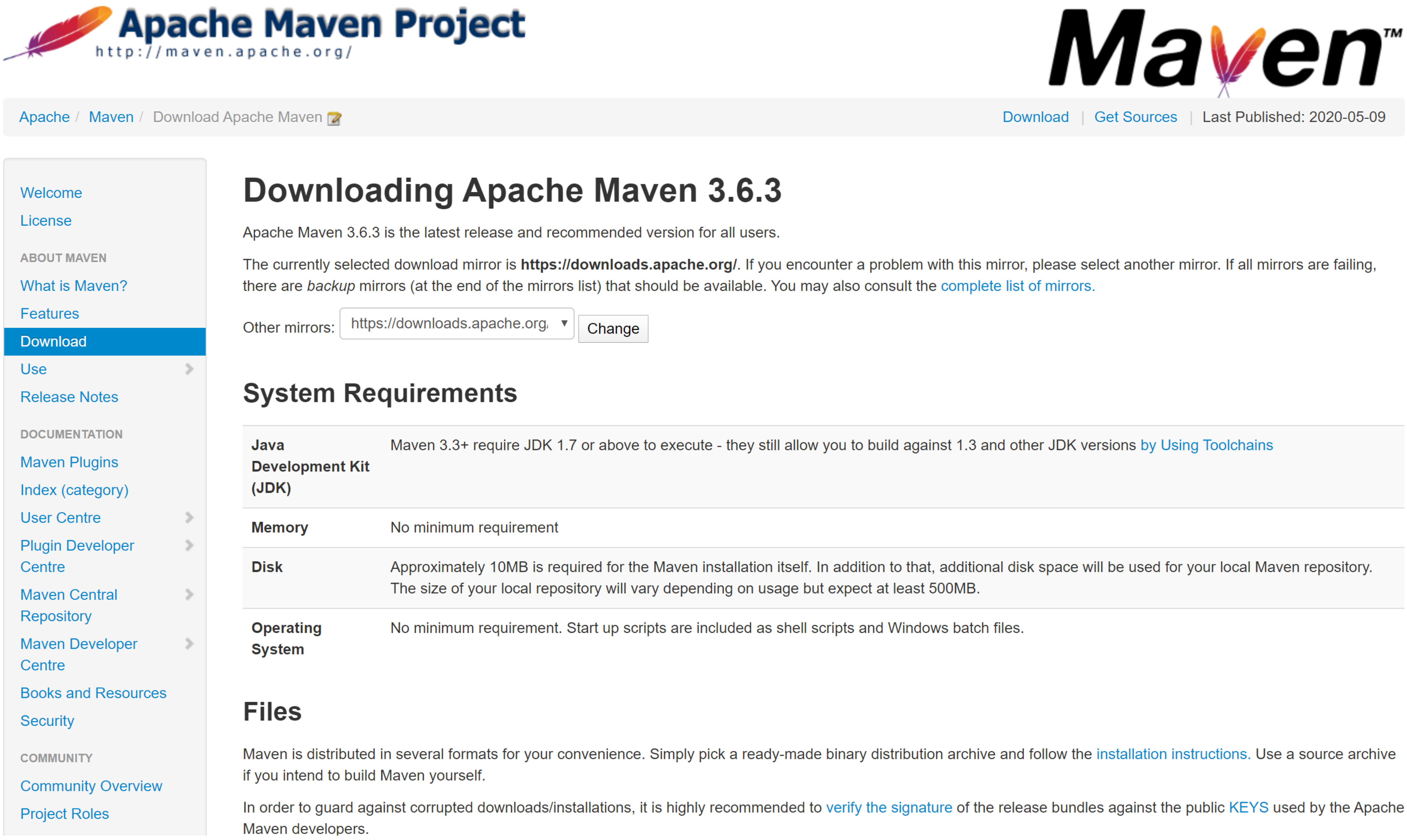
Abstraction

Inheri-tance

En-capsu-lation

Maven

Kuan-Ting Lai
2020/5/11



Downloading Apache Maven 3.6.3

Apache Maven 3.6.3 is the latest release and recommended version for all users.

The currently selected download mirror is **https://downloads.apache.org/**. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available. You may also consult the complete list of mirrors.

Other mirrors: https://downloads.apache.org/ ▾ Change

System Requirements

Java Development Kit (JDK)	Maven 3.3+ require JDK 1.7 or above to execute - they still allow you to build against 1.3 and other JDK versions by Using Toolchains
Memory	No minimum requirement
Disk	Approximately 10MB is required for the Maven installation itself. In addition to that, additional disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.
Operating System	No minimum requirement. Start up scripts are included as shell scripts and Windows batch files.

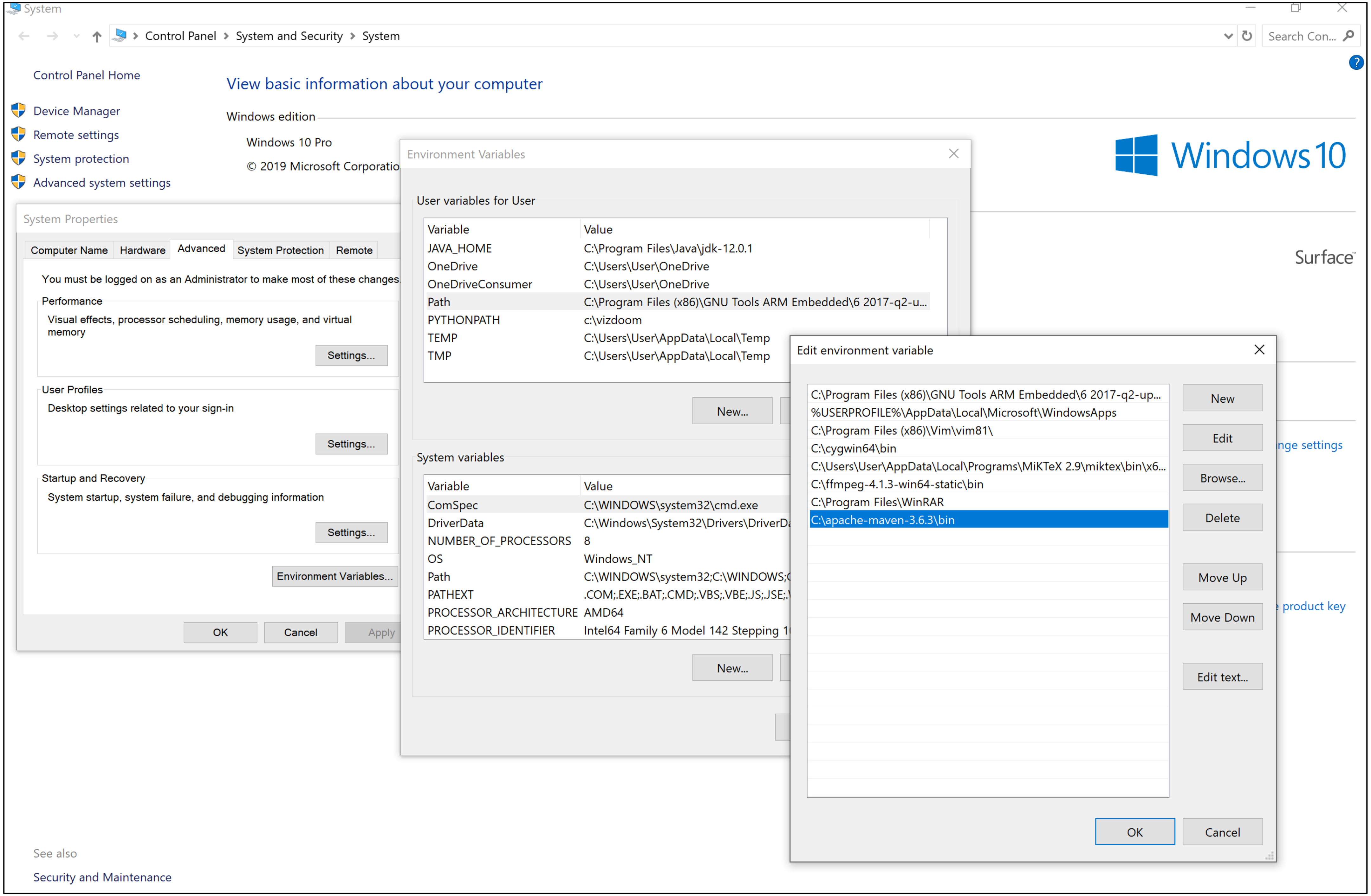
Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the installation instructions. Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public KEYS used by the Apache Maven developers.

Apache Maven

- Build and manage Java project
- Structure and contents are declared in Project Object Model (POM) file “pom.xml”
- Requiring JDK
- Download and extract apache-maven-xxx-bin.zip
- Add maven/bin to PATH
- Running maven
  - mvn compile



User variables for User

Variable	Value
JAVA_HOME	C:\Program Files\Java\jdk-12.0.1
OneDrive	C:\Users\User\OneDrive
OneDriveConsumer	C:\Users\User\OneDrive
Path	C:\Program Files (x86)\GNU Tools ARM Embedded\6 2017-q2-u...
PYTHONPATH	c:\vizdoom
TEMP	C:\Users\User\AppData\Local\Temp
TMP	C:\Users\User\AppData\Local\Temp

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverD
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\WINDOWS\system32;C:\WINDOWS;(
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.\
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 142 Stepping 1(

Edit environment variable

```
C:\Program Files (x86)\GNU Tools ARM Embedded\6 2017-q2-up...
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Program Files (x86)\Vim\vim81\
C:\cygwin64\bin
C:\Users\User\AppData\Local\Programs\MiKTeX 2.9\miktex\bin\x6...
C:\ffmpeg-4.1.3-win64-static\bin
C:\Program Files\WinRAR
C:\apache-maven-3.6.3\bin
```

Download and Set Maven Environment Variables

Download Maven from https://maven.apache.org/download.cgi

OS	Output

Windows	Set the environment variables using system properties.
	M2_HOME=C:\Program Files\Apache Software Foundation\apache-maven-3.3.1
	M2=%M2_HOME%\bin MAVEN_OPTS=-Xms256m -Xmx512m

Linux	Open command terminal and set environment variables. export M2_HOME=/usr/local/apache-maven/apache-maven-3.3.1 export M2=$M2_HOME/bin export MAVEN_OPTS=-Xms256m -Xmx512m
Mac	Open command terminal and set environment variables. export M2_HOME=/usr/local/apache-maven/apache-maven-3.3.1 export M2=$M2_HOME/bin export MAVEN_OPTS=-Xms256m -Xmx512m

Configuration

Item	Default
source code	${basedir}/src/main/java
Resources	${basedir}/src/main/resources
Tests	${basedir}/src/test
Complied byte code	${basedir}/target
distributable JAR	${basedir}/target/classes

POM Example

```
<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>com.companyname.project-group</groupId>
      <artifactId>project</artifactId>
      <version>1.0</version>
</project>
```

Build Lifecycle

Phase	Handles	Description
prepare-resources	resource copying	Resource copying can be customized in this phase.
validate	Validating the information	Validates if the project is correct and if all necessary information is available.
compile	compilation	Source code compilation is done in this phase.
Test	Testing	Tests the compiled source code suitable for testing framework.
package	packaging	This phase creates the JAR/WAR package as mentioned in the packaging in POM.xml.
install	installation	This phase installs the package in local/remote maven repository.
Deploy	Deploying	Copies the final package to the remote repository.

Build Lifecycle

- There are always **pre** and **post** phases to register **goals**, which must run prior to, or after a particular phase
- Maven has the following three standard lifecycles
  - clean
  - default(or build)
  - site

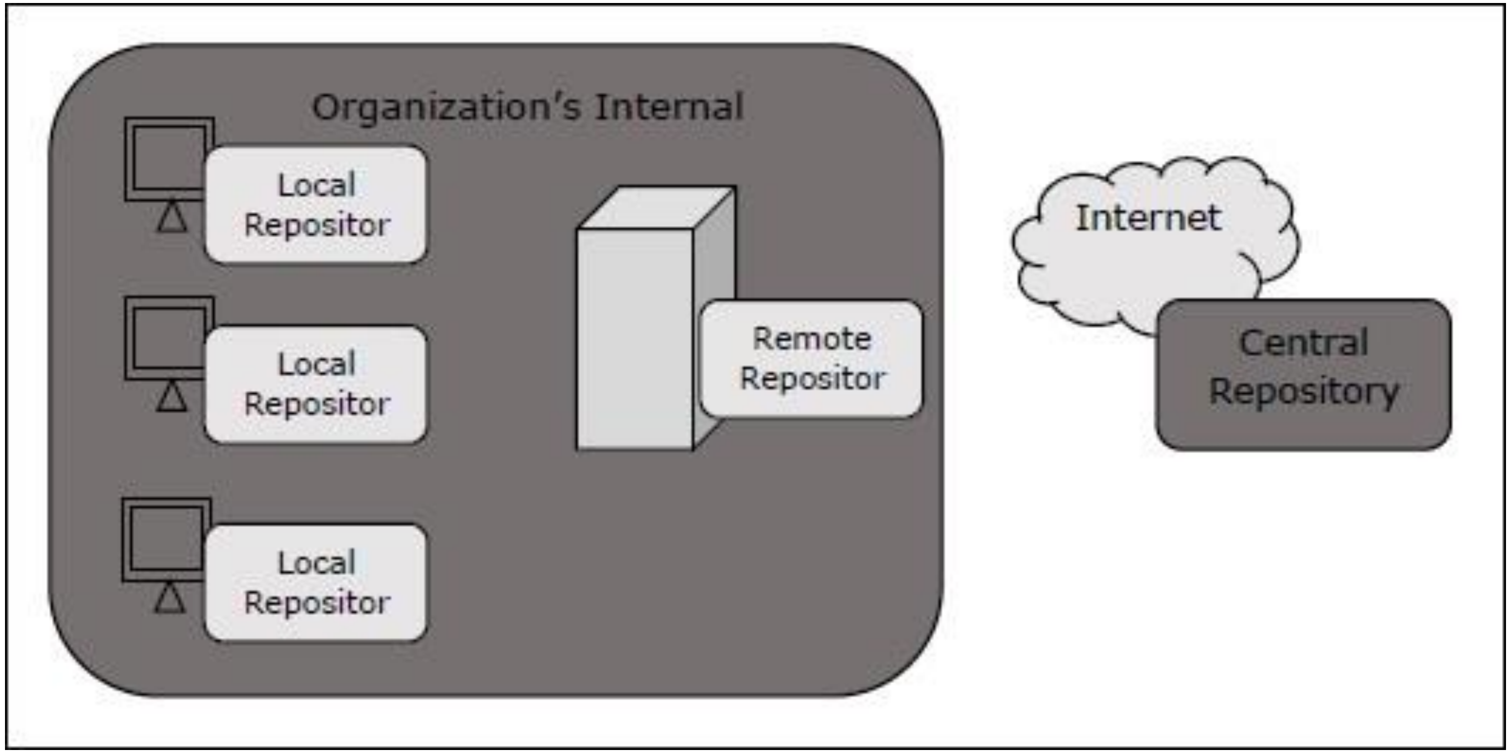
Build Profile

- Create different settings. Ex: Production vs Development

Type	Where it is defined
Per Project	Defined in the project POM file, pom.xml
Per User	Defined in Maven settings xml file (%USER_HOME%/.m2/settings.xml)
Global	Defined in Maven global settings xml file (%M2_HOME%/conf/settings.xml)

Repository

- Local
- Central
- Remote



```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/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.companyname.projectgroup</groupId>
    <artifactId>project</artifactId>
    <version>1.0</version>
    <dependencies>
        <dependency>
            <groupId>com.companyname.common-lib</groupId>
            <artifactId>common-lib</artifactId>
            <version>1.0.0</version>
        </dependency>
    <dependencies>
    <repositories>
        <repository>
            <id>companyname.lib1</id>
            <url>http://download.companyname.org/maven2/lib1</url>
        </repository>
        <repository>
            <id>companyname.lib2</id>
            <url>http://download.companyname.org/maven2/lib2</url>
        </repository>
    </repositories>
</project>
```

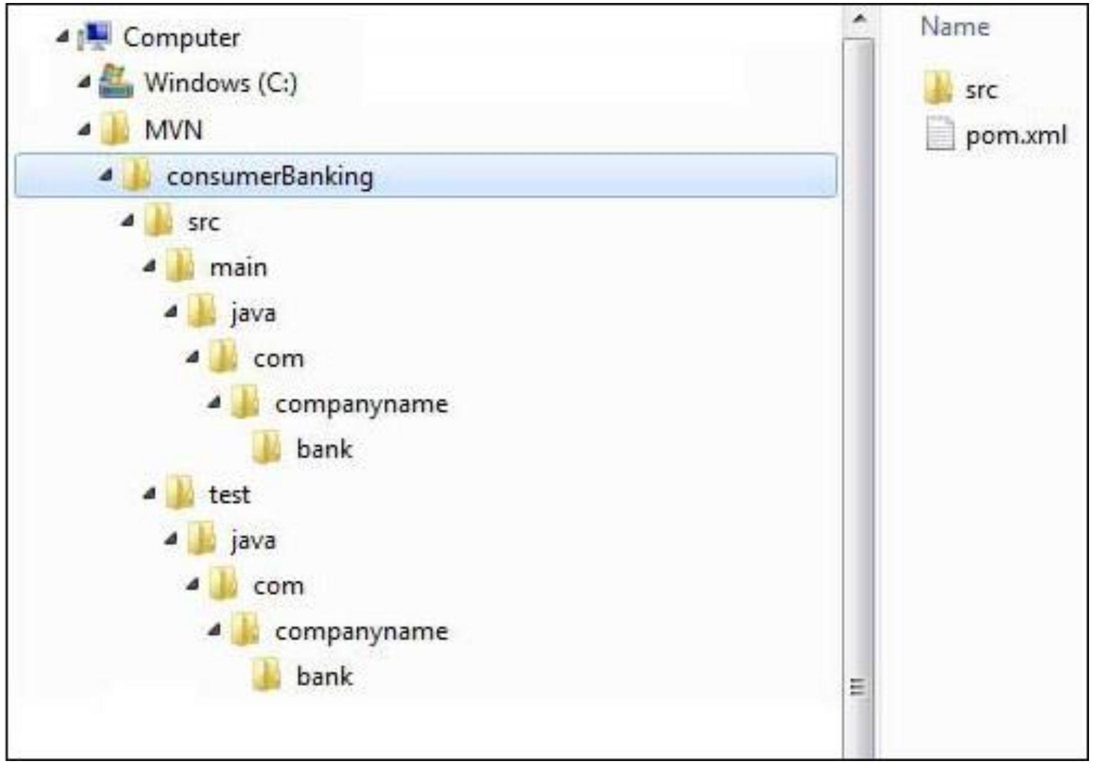
Maven Plugins

```
mvn [plugin-name]:[goal-name]
```

Sr.No.	Plugin & Description
1	**clean** Cleans up target after the build. Deletes the target directory.
2	**compiler** Compiles Java source files.
3	**surefire** Runs the JUnit unit tests. Creates test reports.
4	**jar** Builds a JAR file from the current project.
5	**war** Builds a WAR file from the current project.
6	**javadoc** Generates Javadoc for the project.
7	**antrun** Runs a set of ant tasks from any phase mentioned of the build.

Create a Project (archetype:generate)

```
C:\MVN>mvn archetype:generate
-DgroupId = com.companyname.bank
-DartifactId = consumerBanking
-DarchetypeArtifactId = maven-archetype-quickstart
-DinteractiveMode = false
```



Build the Example App “Hello World!”

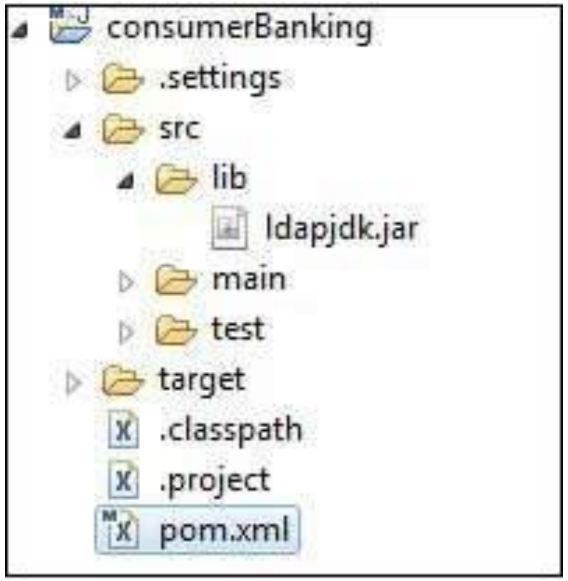
- C:\MVN\consumerBanking> mvn clean package
- cd C:\MVN\consumerBanking\target\classes>
- java com.companyname.bank.App

```
 Hello World!
```

External Dependencies

- Add **lib** folder to the src folder
- Copy any jar (**ldapjdk.jar**) into the lib folder

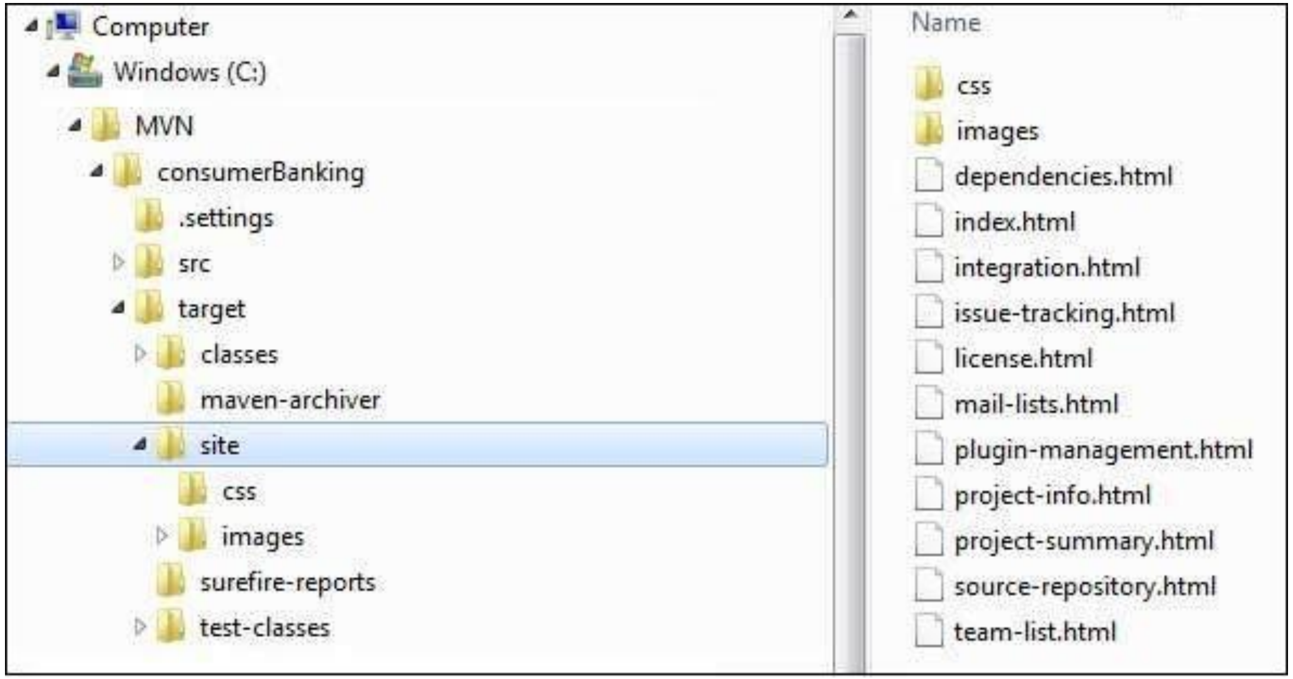
```
<dependency>
     <groupId>ldapjdk</groupId>
     <artifactId>ldapjdk</artifactId>
     <scope>system</scope>
     <version>1.0</version>
     <systemPath>${basedir}\src\lib\ldapjdk.jar</systemPath>
</dependency>
```

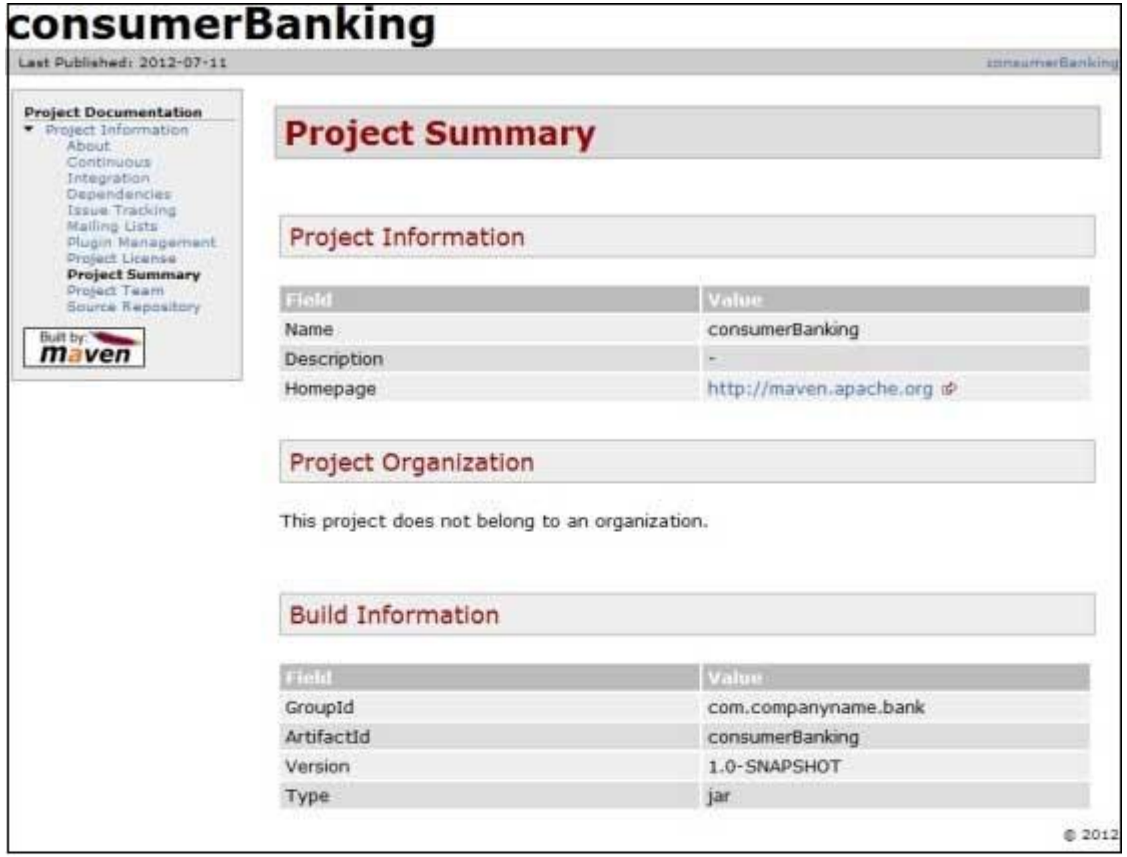


```
consumerBanking
  .settings
  src
    lib
      ldapjdk.jar
    main
    test
  target
  .classpath
  .project
  pom.xml
```

Project Documents

- mvn site





Last Published: 2012-07-11

Project Summary

Project Information

Field	Value
Name	consumerBanking
Description	-
Homepage	http://maven.apache.org

Project Organization

This project does not belong to an organization.

Build Information

Field	Value
GroupId	com.companyname.bank
ArtifactId	consumerBanking
Version	1.0-SNAPSHOT
Type	jar

© 2012

Snapshots (Development Copy)

- Snapshot: Maven will download every time
- Version: Maven will only download this version once

Manage Dependencies

Sr.No.	Scope & Description
1	**compile** This scope indicates that dependency is available in classpath of project. It is default scope.
2	**provided** This scope indicates that dependency is to be provided by JDK or web-Server/Container at runtime.
3	**runtime** This scope indicates that dependency is not required for compilation, but is required during execution.
4	**test** This scope indicates that the dependency is only available for the test compilation and execution phases.
5	**system** This scope indicates that you have to provide the system path.
6	**import** This scope is only used when dependency is of type pom. This scope indicates that the specified POM should be replaced with the dependencies in that POM's <dependencyManagement> section.

Create Web Application

```
mvn archetype:generate -DgroupId=org.aiotlab.oop -DartifactId=myjsp -
DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false
```

	Folder Structure & Description
1	**myjsp** contains src folder and pom.xml.
2	**src/main/webapp** contains index.jsp and WEB-INF folder.
3	**src/main/webapp/WEB-INF** contains web.xml
4	**src/main/resources** it contains images/properties files.

References

- https://www.tutorialspoint.com/maven/
- https://www.tutorialspoint.com/maven/maven_intellij_idea.htm
- https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html