MAVEN

JAVA PROJECT MANAGEMENT
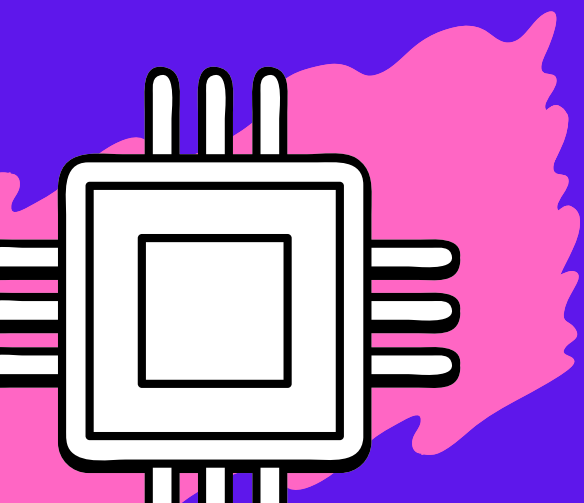
Regina Rodriguez

What is maven?

Maven is a project management and comprehension tool primarily used in Java software development. It was developed by the Apache Software Foundation. Here's a summary of what Maven is and what it's used for:
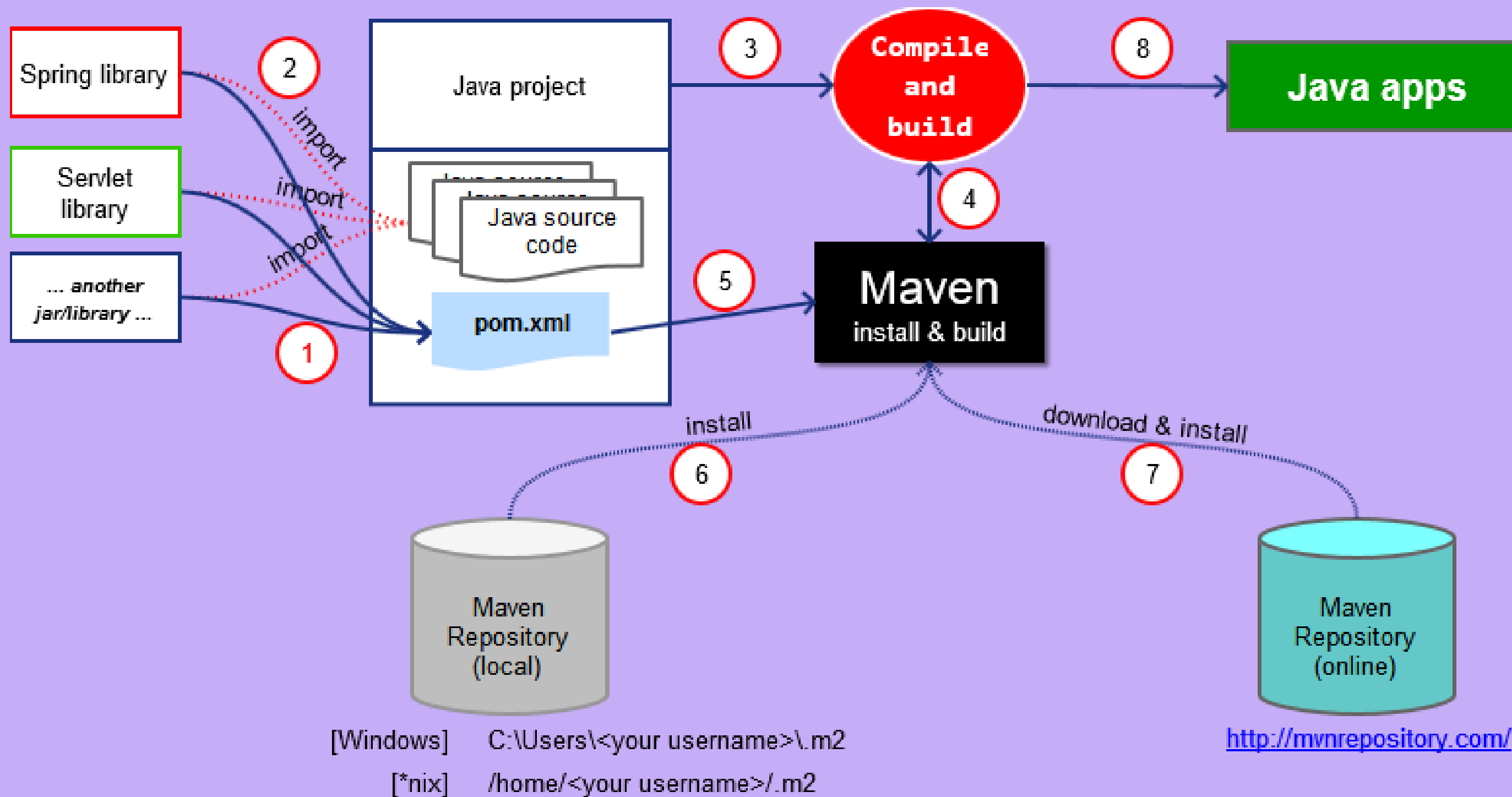
A Dependency Management.

B Task Automation.

C Project Structure.

D Multi-Project Management.

How it works

Spring library

Servlet library

... another jar/library ...

2

import

import

import

1

Java project

Java source code

pom.xml

3

5

Compile and build

8

Java apps

4

Maven install & build

6

install

7

download & install

Maven Repository (local)

Maven Repository (online)

[Windows]   C:\Users\<your username>\.m2

[*nix]   /home/<your username>/.m2

http://mvnrepository.com/

Maven Project Configuration

Maven projects are configured using a pom.xml file, which is located at the root of the project. This file includes:

1. Basic Structure:
- \<groupId\>: Identifies the project's group (e.g., com.example).
- \<artifactId\>: Unique identifier for the project.
- \<version\>: Project version (e.g., 1.0-SNAPSHOT).

The pom.xml file is essential for managing dependencies, automating the build process, and configuring project settings in a Maven project.

2. Dependencies:
- Defined within \<dependencies\>. For example:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    <version>2.7.0</version>
</dependency>
```

3. Plugins:
- Configured within \<plugins\> to extend Maven's functionality. For example, the compiler plugin:

4. Repositories:
- Optional, defined within \<repositories\> to specify where Maven should look for dependencies.

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.8.1</version>
    <configuration>
        <source>1.8</source>
        <target>1.8</target>
    </configuration>
</plugin>
```
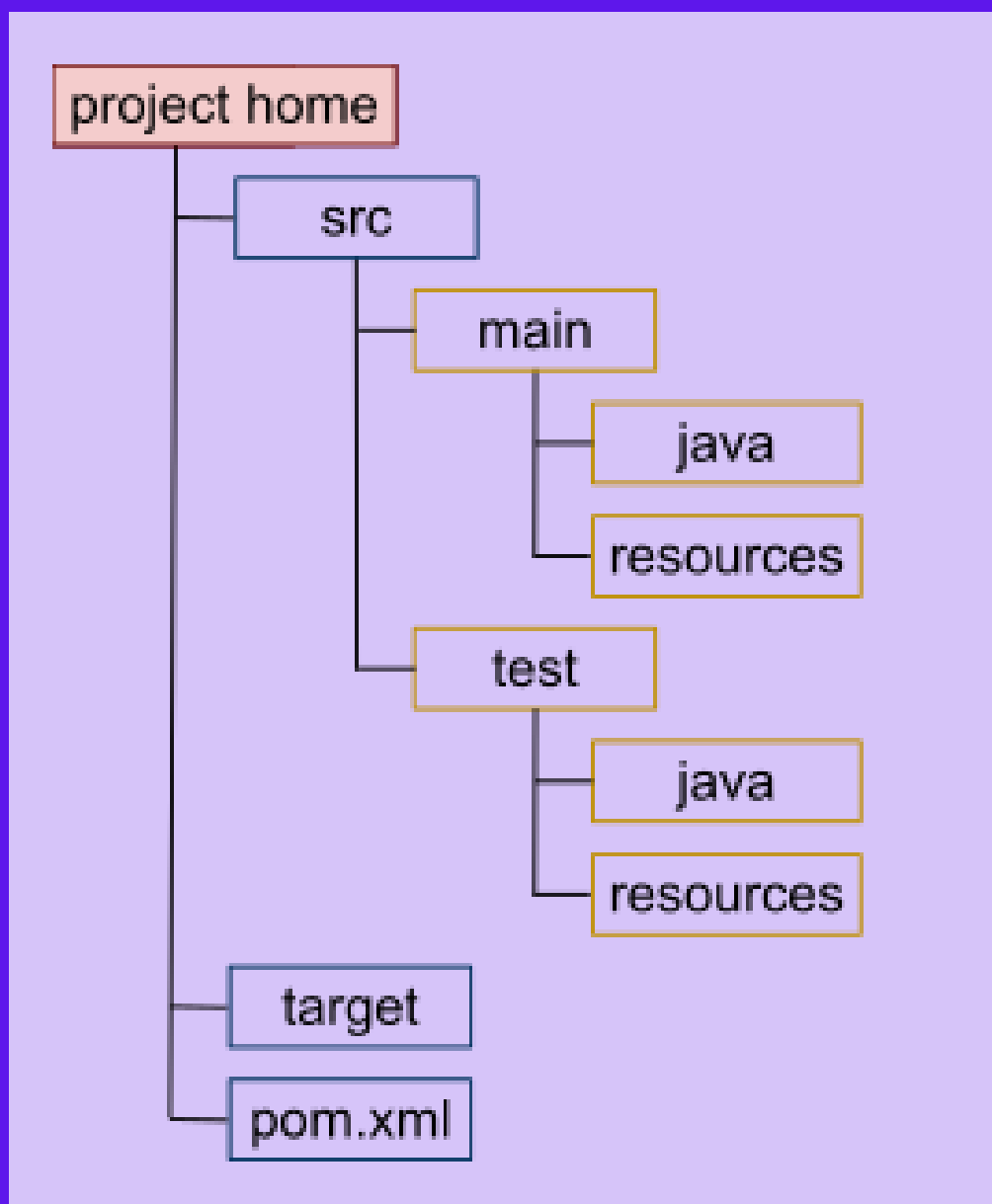
Maven Project Structure

The typical project structure in Maven follows a standardized layout, which makes it easier to manage and understand projects, especially in larger teams. Here's a brief overview:

1. **pom.xml:** Main configuration file, located in the root.

2. **src/main/java:** Contains the main Java code.

3. **src/main/resources:** Holds configuration files and resources used by the application.

4. **src/test/java:** Stores test code.

5. **src/test/resources:** Includes test-specific resources.

6. **target/:** Where Maven puts the compiled code and build artifacts.