Bộ môn: Kỹ Thuật Phần Mềm

Lập trình WWW *Java*

.

Maven Project

Overview

How to?

- Create Maven projects with Eclipse
- Add dependencies to Maven pom.xml file
- Build and run Maven projects

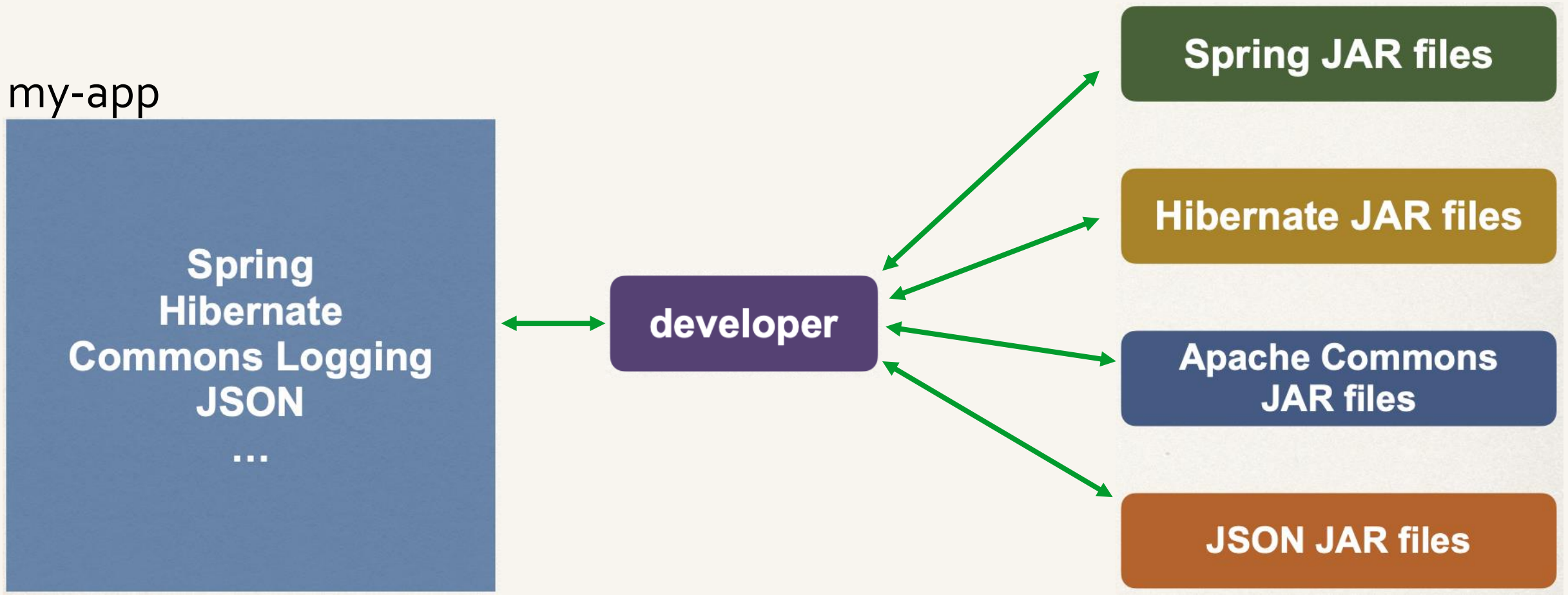
What is Maven?

- Maven is a **Project Management tool**
- Most popular use of Maven is for **build management** and **dependencies**

What Problems Does Maven Solve?

- When building your Java project, you may need additional JAR files
  - For example: Spring, Hibernate, Commons Logging, JSON etc...
- One approach is to download the JAR files from each project web site
- Manually add the JAR files to your build path / classpath

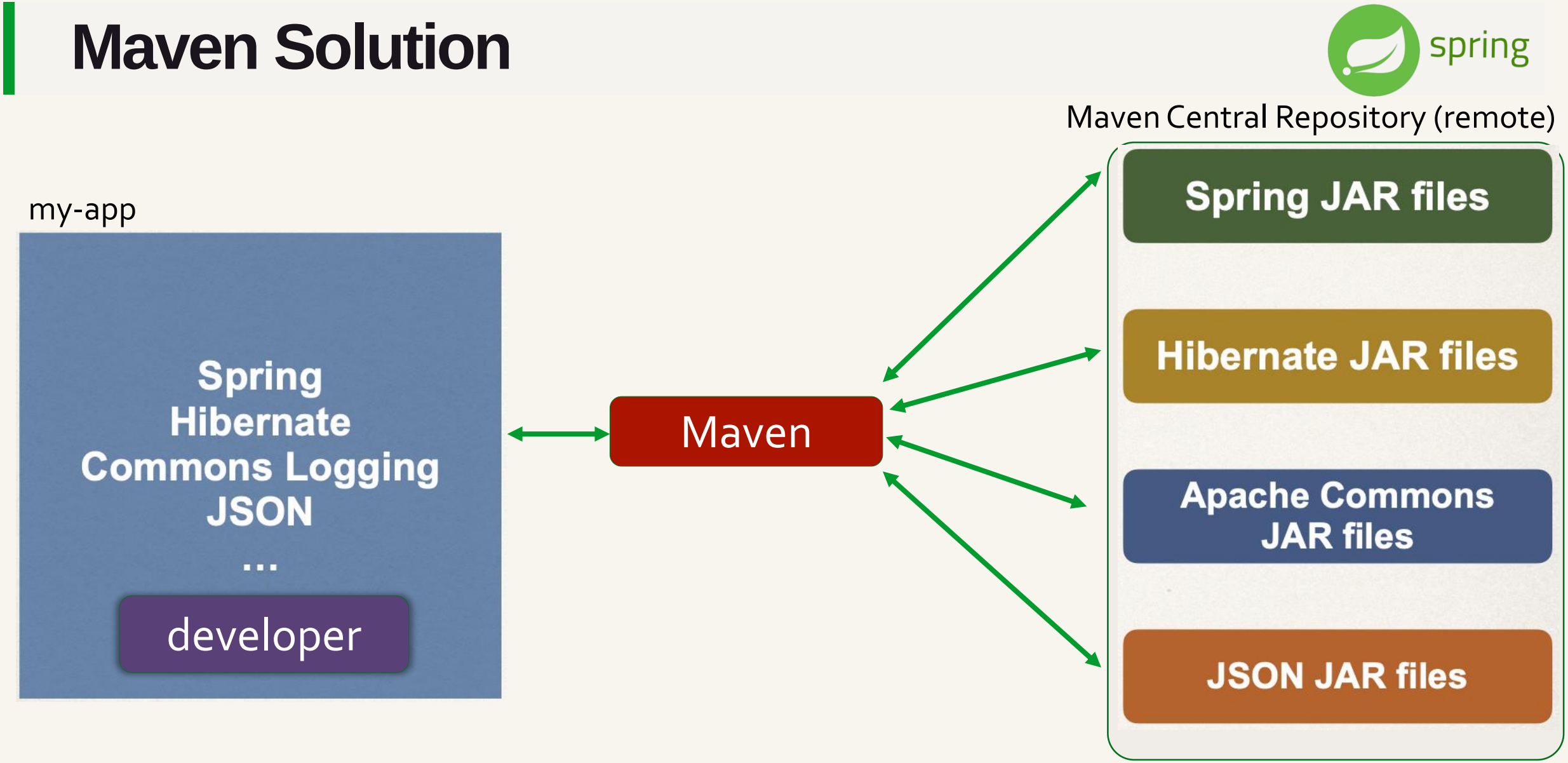
Project without Maven



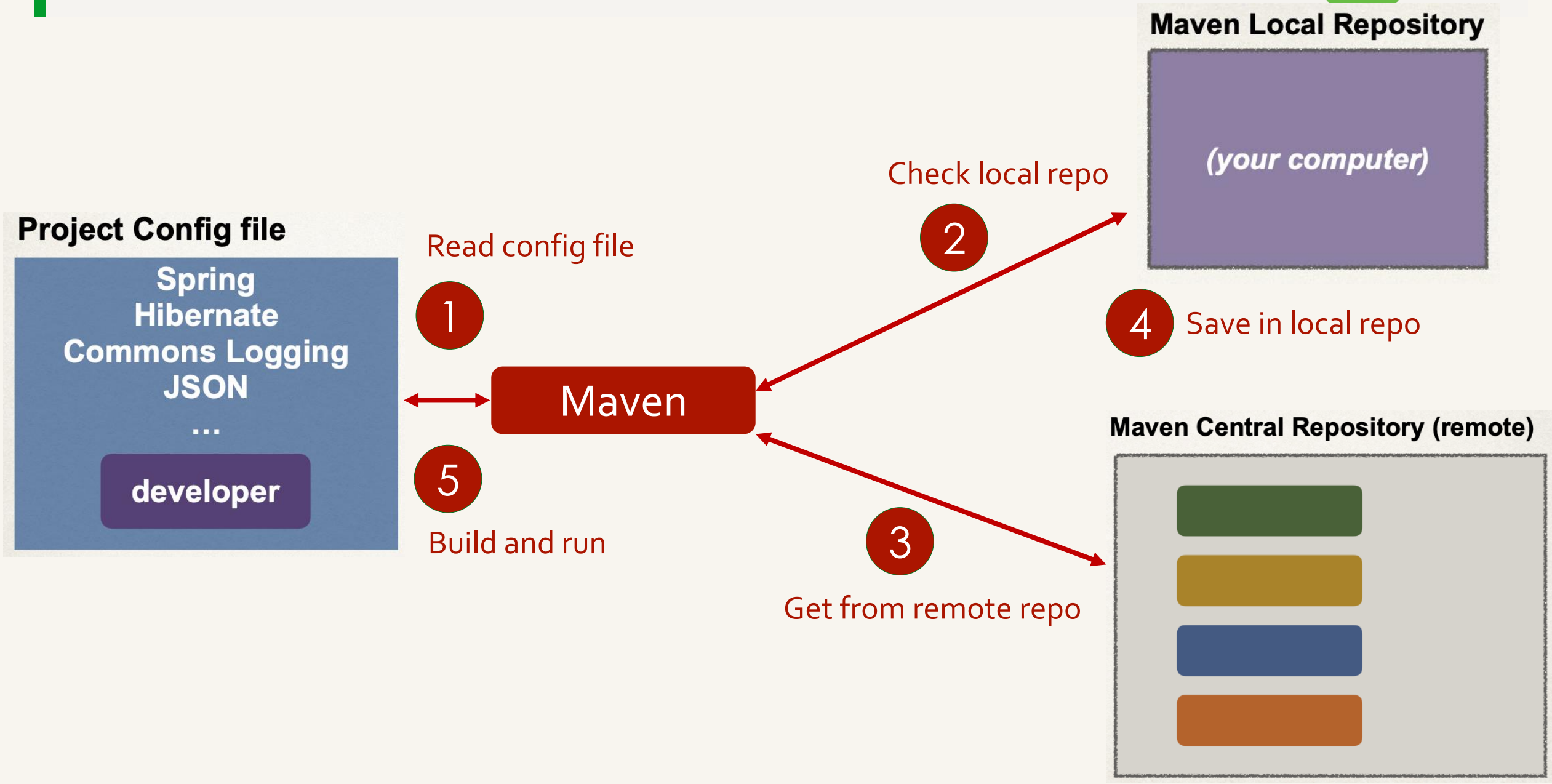
Maven Solution

- Tell Maven the projects you are working with (dependencies)
  - Spring, Hibernate etc...
- Maven will go out and download the JAR files for those projects for you
- And Maven will make those JAR files available during compile/run
- Think of Maven as your **friendly helper**

Maven Solution



Maven – How It Works

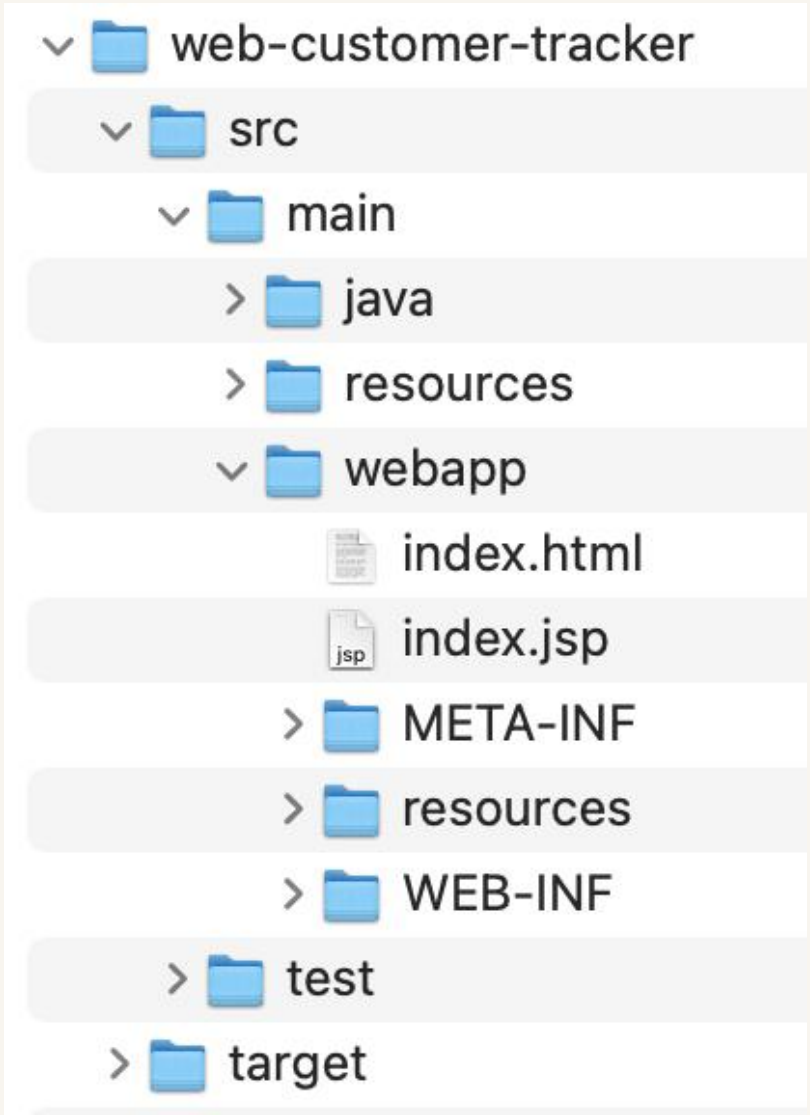


Building and Running

When you build and run your app:

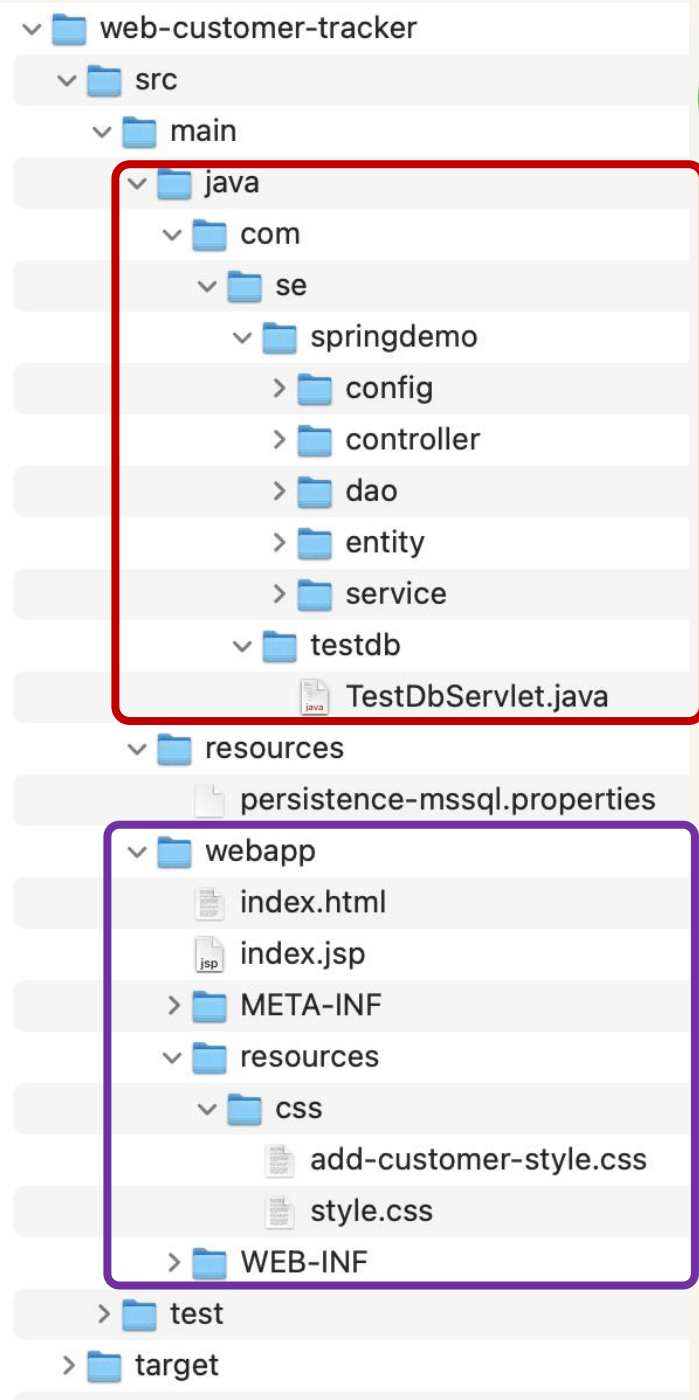
- Maven will handle class / build path for you
- Based on config file, Maven will add JAR files accordingly

Standard Directory Structure



Directory	Description
src/main/java	Your Java source code
src/main/resources	Properties / config files used by your app
src/main/webapp	JSP files and web config files other web assets (images, css, js, etc)
src/test	Unit testing code and properties
target	Destination directory for compiled code. Automatically created by Maven

Standard Directory Structure



Advantages of Maven

- Dependency Management
  - Maven will find JAR files for you
  - No more missing JARs
- Building and Running your Project
  - No more build path / classpath issues
- Standard directory structure

Maven Key Concepts

- POM File - pom.xml
- Project Coordinates

POM File - pom.xml

- Project Object Model file: POM file
- Configuration file for your project
  - Basically your “shopping list” for Maven
- Located in the root of your Maven project

Simple POM File

```
<modelVersion>4.0.0</modelVersion>

<groupId>iuh.fit.se</groupId>
<artifactId>mavenwebappdemo</artifactId>
<version>1.0</version>
<packaging>war</packaging>
```

Project name, version etc Output file type: JAR, WAR, ...

```
<dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.11</version>
        <scope>test</scope>
    </dependency>
    <!-- we nedd to add  the servlet api dependency: j
    <!-- goto search.maven.org -->
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>javax.servlet-api</artifactId>
        <version>4.0.1</version>
    </dependency>
    <dependency>
        <groupId>javax.servlet.jsp</groupId>
        <artifactId>javax.servlet.jsp-api</artifactId>
        <version>2.3.1</version>
    </dependency>
</dependencies>
```

List of projects we depend on Servlet, JSP, etc...

Project Coordinates

- Project Coordinates uniquely identify a project
  - Similar to GPS coordinates for your house: latitude / longitude
  - Precise information for finding your house (city, street, house #)

```
<groupId>iuh.fit.se</groupId>
<artifactId>mavenwebappdemo</artifactId>
<version>1.0</version>
```

City

Street

House number

Adding Dependencies


```
<project ...>
  ...
  <dependencies>

      <dependency>
         <groupId>org.springframework</groupId>
         <artifactId>spring-context</artifactId>
         <version>5.0.0.RELEASE</version>
      </dependency>

      <dependency>
         <groupId>org.hibernate</groupId>
         <artifactId>hibernate-core</artifactId>
         <version>5.2.11.Final</version>
      </dependency>
      ...

  </dependencies>

</project>
```

Dependency Coordinates

- To add given dependency project, we need
  - **Group ID**, **Artifact ID**
  - **Version** is optional ...
- Best practice is to include the version (repeatable builds)
- May see this referred to as: GAV
  - **G**roup ID, **A**rtifact ID and **V**ersion

How to Find Dependency Coordinates

- Option 1: Visit the project page (spring.io, hibernate.org etc)
- Option 2: Visit **http://search.maven.org** (easiest approach)

QUESTIONS