**Spring Boot Course – Chad Darby**

Spring Boot runs on top of Spring Framework, it uses Spring framework underneath along with other Spring Projects Like Spring Web, Spring MVC, Spring Rest API like these there are many spring projects which we can use to create a Spring Boot App

* Minimizes manual configurations
* Handles the dependencies (JARs) using MAVEN or GRADLE
* Comes with Embedded Server (TOMCAT, JETTY etc.) can be run standalone because of this.

MYAPP.JAR

TOMCAT SERVER

CODE

Run Spring Boot Application in CMD

***java -jar myapp.jar***

How to create Starter Spring Boot Application – <https://start.spring.io>

* Here you can select the java version, build tool (Gradle or Maven)
* Add dependencies
* Add the project and package names

Creating Simple Rest Controller in Spring Boot (Hello world App)

@RestController  
public class MyController {  
 @RequestMapping("helloworld")  
 public String helloWorld() {  
 return "Hello, World!";  
 }  
}

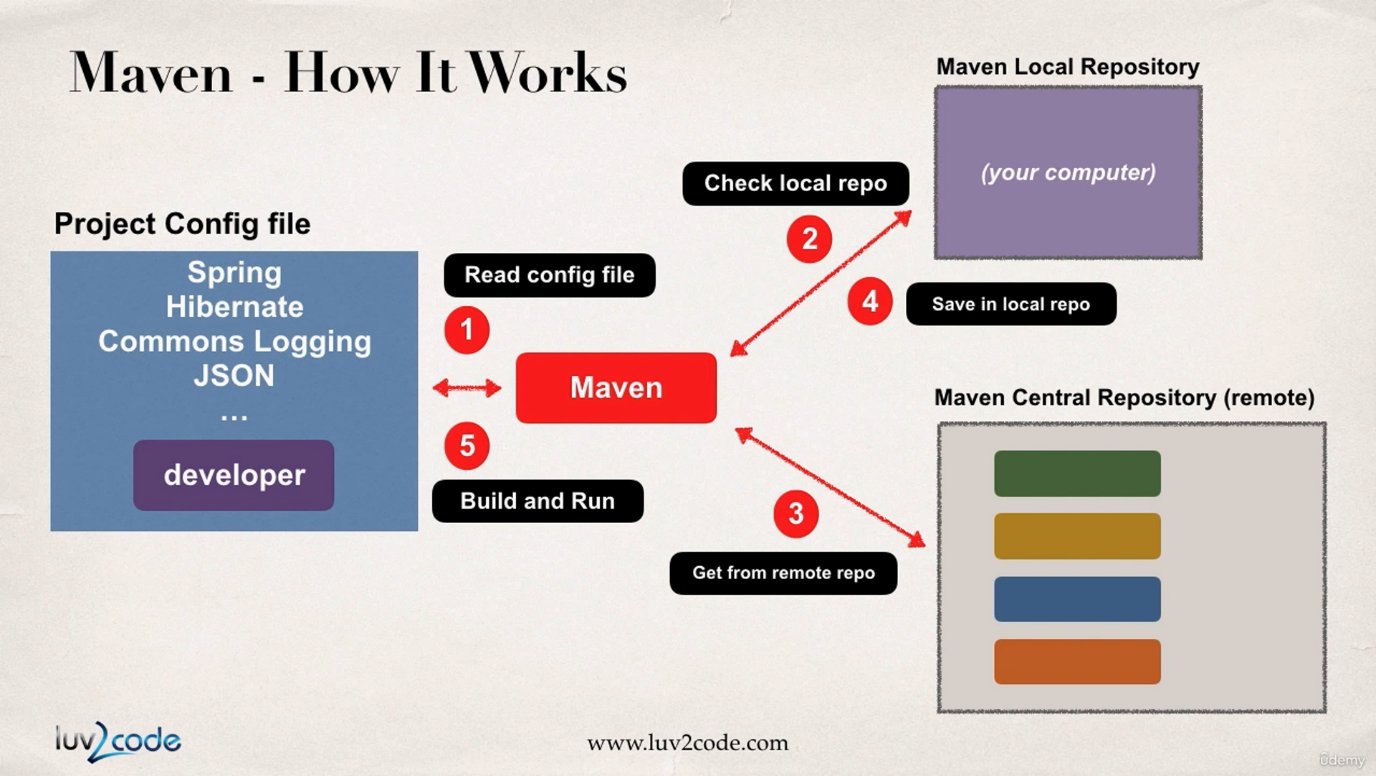
There are many spring projects which we can use in our application some of the examples being Spring Web, Spring Data JPA, Spring Security, Spring MVC etc.

We can learn more about spring project here <https://spring.io/projects>

**What is Maven?**

Maven is a project management tool, it manages dependencies.

Dependencies of our application requires several JAR files which need to be downloaded from the internet Central MAVEN repository, this is handled by Maven project management tool, Gradle is also same as Maven.



Working of MAVEN, it will read config file first and checks local maven repo, and If required JARs are not found in local it will get them from Remote Maven central Repository.

Maven will build and handle your app

Maven will add class paths

Based on Config file Maven will add JAR files

**MAVEN Concepts**

**groupId** – name of company (com.google)

**artifactId** – name of the project (maps)

**version** - version of the project (1.2.2) (com.google.maps – 1.2.2)

**Adding Dependencies in pom.xml file**

<dependencies>

<dependency>

<groupId></groupId>

<artifactId></artifactId>

<version></version>

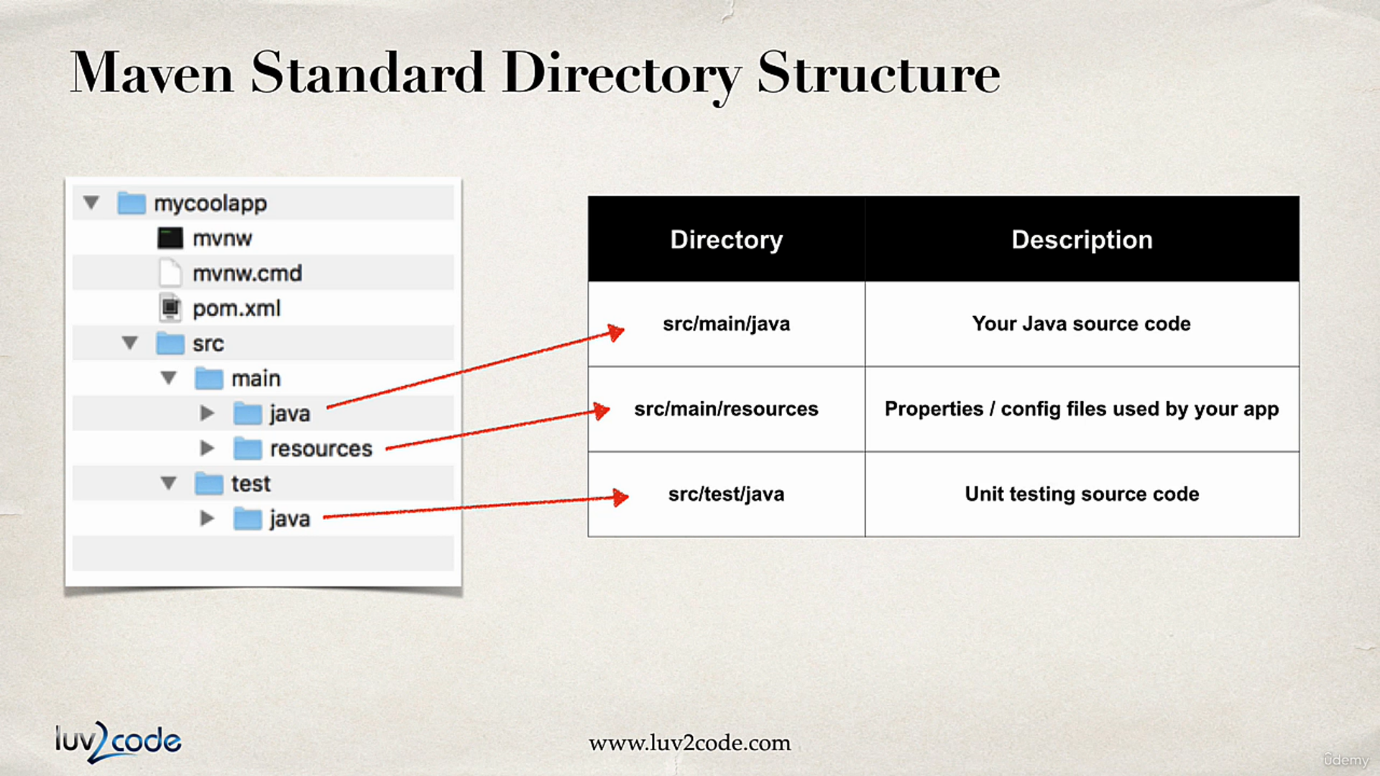
</dependency>

…

</dependencies>

You will get this in MVN Central website, and reading this config file Maven will build your app by importing required JARs

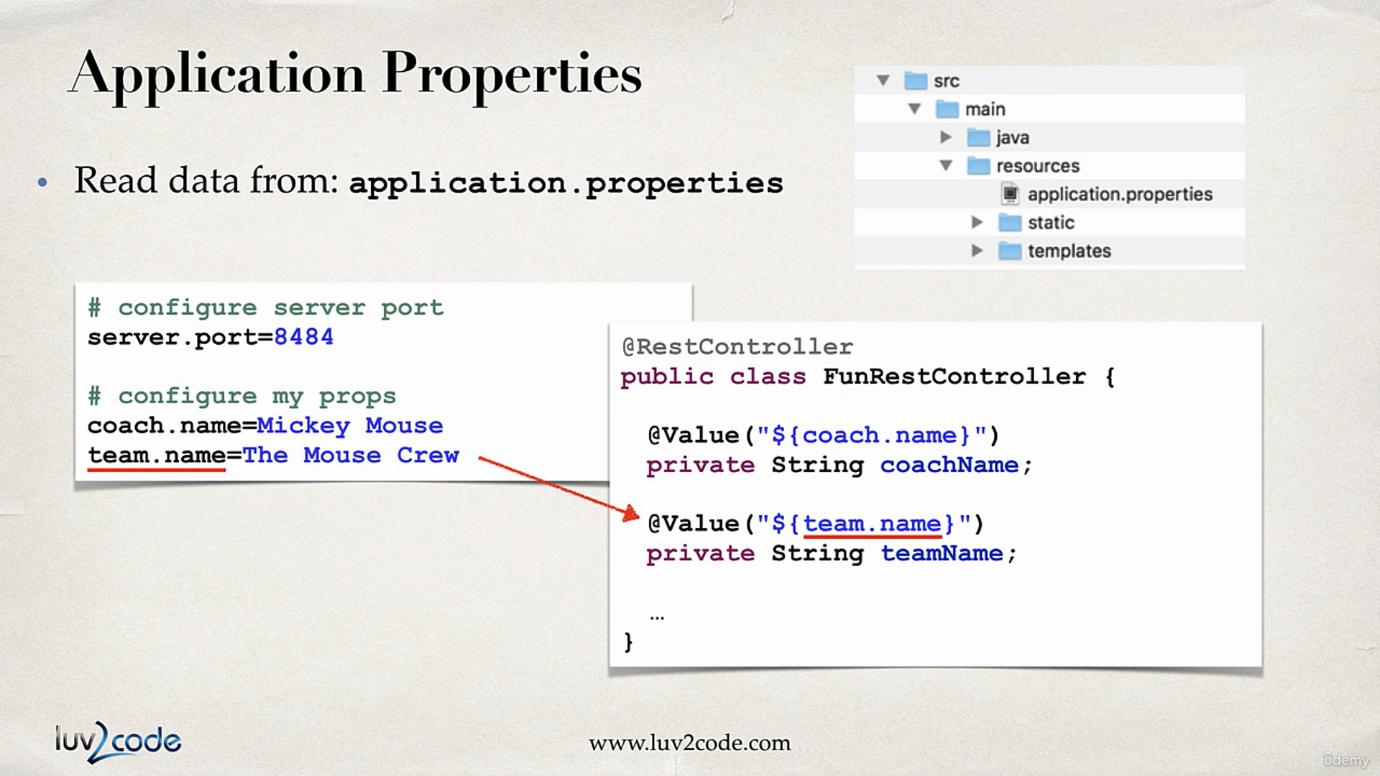
**Spring Boot project structure**

****

**mvnw.cmd(win) and mvnw (linux/mac)** are maven wrapper files which are used to automatically download the correct version of maven and run the maven for the application.

**pom.xml** is used to add dependencies and plugins required for the project application.

**application.properties** : we can add the properties for application in this file



**Spring boot starters:** A dependency which is a collection of other dependencies which are bare minimum requirement to build application.

For e.g. **spring-boot-starter-web**, this contains -> Spring Web, Spring Web MVC, Hibernate-validator, json, Tomcat. (Add Spring Web in Spring initializer)

<https://www.javatpoint.com/spring-boot-starters>

**Spring boot starter parent:** These are maven defaults, this contains default compiler version, UTF-8 source encoding, others…

In <properties> it contains java.version which we are using for the project

<plugins> is used to configure application to run in command line using a command *mvn spring-boot:run*

**Spring Boot Devtools:** By using this dependency we no need to restart the application after every change, whenever we save the changes to file the spring app automatically restarts.

IntelliJ configuration of Devtools:

Settings>Build, Execution, Deployment>Compiler -> Build project automatically

Advanced settings > allow auto-make