**Create a Spring Web Project using Maven**

**SME Walkthrough**

**1. src/main/java - Application Code Folder**

* Contains all Java source code
* Includes main application class: SpringLearnApplication.java
* Future controllers, services, and models will go here

**2. src/main/resources - Application Configuration Folder**

* Stores configuration files
* Includes application.properties
* Also used for static resources (like HTML, CSS) and templates (if using Thymeleaf)

**3. src/test/java - Test Code Folder**

* Used for writing JUnit or integration test cases
* Mirrors the structure of src/main/java

**4. SpringLearnApplication.java - Main Class Walkthrough**

* Located in com.cognizant.spring\_learn package
* Entry point of Spring Boot application

Code:

@SpringBootApplication  
public class SpringLearnApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(SpringLearnApplication.class, args);  
 }  
}

* This starts the embedded Tomcat server

**5. Purpose of @SpringBootApplication Annotation**

* Combination of three annotations:
  + @Configuration: Marks the class as a configuration class
  + @EnableAutoConfiguration: Enables auto-configuration of Spring Boot
  + @ComponentScan: Scans the package for components (@Component, @RestController, etc.)

**6. pom.xml - Maven Configuration File**

* Manages dependencies and build lifecycle
* Contains project metadata

Key configurations:

<dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
</dependencies>

**6.1. XML Configuration Walkthrough**

* Each dependency block declares a component required for the project
* spring-boot-starter-web brings in Tomcat, REST controllers, JSON handling, etc.
* spring-boot-devtools allows live reloading during development

**6.2. Show Dependency Hierarchy**

