**1. src/main/java**

* **Purpose**: Holds all **business logic** and source code.
* Folder follows the group structure, e.g.:

src/main/java/com/cognizant/springlearn/SpringLearnApplication.java

**2. src/main/resources**

* application.properties or application.yml
* Static resources (HTML, CSS, JS)
* Templates for Thymeleaf/Freemarker (if used)

**3. src/test/java**

* Holds **unit tests** and **integration tests**
* Mirrors the main package structure
* Used by testing frameworks like **JUnit** or **Mockito**

**4. SpringLearnApplication.java**

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

System.out.println("SpringLearnApplication started...");

SpringApplication.run(SpringLearnApplication.class, args);

}

}

* SpringApplication.run() bootstraps the Spring Boot application.
* **Entry point** to the application.

**5. @SpringBootApplication Annotation**

This is a **meta-annotation** that includes:

| **Annotation** | **Purpose** |
| --- | --- |
| @Configuration | Marks class as a configuration source |
| @EnableAutoConfiguration | Enables Spring Boot's auto configuration |
| @ComponentScan | Enables component scanning in the base package |

**6. pom.xml - Walkthrough**

**Sample structure:**

xml

<groupId>com.cognizant</groupId>

<artifactId>spring-learn</artifactId>

<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>

<optional>true</optional>

</dependency>

</dependencies>

* **spring-boot-starter-web**: Brings in Spring MVC + Tomcat.
* **spring-boot-devtools**: Enables auto-restart on file changes.

**Dependency Hierarchy**

In Eclipse:

* Right-click pom.xml > **Dependency Hierarchy**
* Visualize **transitive dependencies**
* Helps trace **version conflicts**, exclusions, etc.