Hands-On 1 – Spring Boot Web Project Setup Documentation

# 1. Objective

To create a Spring Boot Maven Web application named 'spring-learn' using Spring Initializr and understand the structure, configuration, and build lifecycle.

# 2. Steps to Create the Project

## Step 1: Generate Spring Boot Project

- Go to: https://start.spring.io  
- Group: com.cognizant  
- Artifact: spring-learn  
- Dependencies:  
 \* Spring Boot DevTools  
 \* Spring Web  
- Click Generate and download the ZIP file.

## Step 2: Import into Eclipse

- Extract the ZIP into a folder in your Eclipse workspace.  
- Open Eclipse → File → Import → Maven → Existing Maven Projects → Browse to folder → Finish.

## Step 3: Build Project via Command Line

mvn clean package \  
 -Dhttp.proxyHost=proxy.cognizant.com \  
 -Dhttp.proxyPort=6050 \  
 -Dhttps.proxyHost=proxy.cognizant.com \  
 -Dhttps.proxyPort=6050 \  
 -Dhttp.proxyUser=123456

# 3. Project Structure Walkthrough

✅ src/main/java  
Contains all source Java classes, including main application class: SpringLearnApplication.java

✅ src/main/resources  
Contains application configuration files such as application.properties, etc.

✅ src/test/java  
Used for writing JUnit test cases and integration tests.

# 4. SpringLearnApplication.java

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

# 5. Purpose of @SpringBootApplication

The @SpringBootApplication annotation is a combination of:  
- @Configuration: Marks the class as a source of bean definitions.  
- @EnableAutoConfiguration: Enables Spring Boot's auto-configuration.  
- @ComponentScan: Scans the package for components, configurations, and services.  
  
This annotation simplifies Spring application configuration.

# 6. pom.xml

Key dependencies include:  
- spring-boot-starter-web  
- spring-boot-devtools  
  
<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
</dependency>

# 7. Dependency Hierarchy (in Eclipse)

Right-click the project → Maven → Show Dependency Hierarchy.  
Example:  
spring-boot-starter-web  
 └── spring-boot-starter  
 └── spring-boot  
 └── spring-web  
 └── spring-webmvc

# 8. Output (Upon Running)

:: Spring Boot :: (vX.X.X)  
SpringLearnApplication started.

# ✅ Conclusion

The Spring Boot project has been successfully created, imported, and run. Key concepts like application structure, dependency management, and @SpringBootApplication annotation have been demonstrated.