**Week 4: Spring Boot 3 – REST API with Hands-on Implementation (Project: spring-rest-handson)**

**Steps to Set Up the Spring Web Application Using Maven:**

1. Visit [https://start.spring.io](https://start.spring.io/)
2. Set **Group** to: com.cognizant
3. Set **Artifact** to: spring-learn
4. Choose the dependencies: **Spring Boot DevTools** and **Spring Web**
5. Generate and download the zip file for your project.
6. Extract the downloaded archive into your Eclipse Workspace root directory.
7. Open the terminal and run the following Maven command to build the project:

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

1. In Eclipse, go to File > Import > Maven > Existing Maven Projects, browse to your extracted folder, and finish the import.
2. Add a debug log in the main() method to verify application launch.
3. Execute the SpringLearnApplication class to start the Spring Boot app.

**Key Project Components & Concepts**

* **src/main/java**: Contains the source code of the application.
* **src/main/resources**: Holds application-level configuration files (like application.properties).
* **src/test/java**: Meant for unit and integration test code.
* **SpringLearnApplication.java**: Acts as the entry point of the application with the main() method.

**@SpringBootApplication Annotation:**

This annotation is a combination of:

* @Configuration: Marks the class as a source of bean definitions.
* @EnableAutoConfiguration: Enables Spring Boot’s auto-configuration feature.
* @ComponentScan: Automatically scans the current package for Spring components.

**Overview of pom.xml:**

* Defines all dependencies and plugins for the project.
* Under the **Dependency Hierarchy**, you can view the full tree of imported dependencies (direct and transitive).

**Sample Code Explanation:**

@SpringBootApplication

public class SpringLearnApplication {

private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(SpringLearnApplication.class, args);

displayCountry(context);

}

public static void displayCountry(ApplicationContext context) {

Country country = context.getBean(Country.class);

LOGGER.debug("Country : {}", country);

}

@Configuration

static class AppConfig {

@Bean

public Country country() {

Country country = new Country();

country.setCode("IN");

country.setName("India");

return country;

}

}

}

Country class:

class Country {

private static final Logger LOGGER = LoggerFactory.getLogger(Country.class);

private String code;

private String name;

public Country() {

LOGGER.debug("Country object instantiated.");

}

public String getCode() {

LOGGER.debug("Fetching code");

return code;

}

public void setCode(String code) {

LOGGER.debug("Setting code");

this.code = code;

}

public String getName() {

LOGGER.debug("Fetching name");

return name;

}

public void setName(String name) {

LOGGER.debug("Setting name");

this.name = name;

}

@Override

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}