Week 4 - Spring Web Project Setup using Maven

# 1. Project Generation via Spring Initializr

- URL: https://start.spring.io/  
- Group: com.cognizant  
- Artifact: spring-learn  
- Dependencies: Spring Web, Spring Boot DevTools  
- Click "Generate" to download the zip.

# 2. Import into Eclipse

- Extract the zip into Eclipse workspace root folder.  
- Open Eclipse > File > Import > Maven > Existing Maven Projects.  
- Browse to extracted folder and click Finish.

# 3. Build using Maven (Command Line)

Command:  
mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456  
  
(Note: Replace '123456' with your actual Cognizant user ID)

# 4. Logging in SpringLearnApplication.java

```java  
package com.cognizant.springlearn;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);  
  
 public static void main(String[] args) {  
 LOGGER.info("Start");  
 SpringApplication.run(SpringLearnApplication.class, args);  
 LOGGER.info("End");  
 }  
}  
```

# 5. Project Structure Overview

- src/main/java: Application source code (controllers, models, etc.)  
- src/main/resources: Configuration files like application.properties  
- src/test/java: Test classes  
- SpringLearnApplication.java: Main class to bootstrap the Spring Boot app

# 6. @SpringBootApplication Annotation

- A meta-annotation that combines:  
 - @Configuration  
 - @EnableAutoConfiguration  
 - @ComponentScan  
- Marks the main class and enables Spring Boot functionality

# 7. pom.xml and Dependency Hierarchy

- Defines project dependencies and metadata  
- Dependencies include spring-boot-starter-web and spring-boot-devtools  
  
Sample snippet:  
```xml  
<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>  
```  
- View full dependency tree using 'Dependency Hierarchy' tab in Eclipse.

# 8. Application Execution

- Run SpringLearnApplication.java  
- Check logs to verify application starts and main method logs appear