**Hands-on 1 Summary: Create Spring Web Project**

**Step-by-Step Project Creation**

1. **Go to:** <https://start.spring.io/>
2. **Group:** com.cognizant
3. **Artifact Id:** spring-learn
4. **Dependencies:** Spring Boot DevTools, Spring Web
5. **Download ZIP**, extract to workspace
6. **Build via CLI:** mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 \-Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456
7. **Import in Eclipse:** File > Import > Maven > Existing Maven Projects

**SME Walkthrough Points**

**1. Folder Structure**

| **Folder** | **Purpose** |
| --- | --- |
| src/main/java | Java source code (business logic) |
| src/main/resources | Configuration files (application.properties, XML configs) |
| src/test/java | Test cases (JUnit) |

**2. SpringLearnApplication.java**

* Contains the main() method.
* Launches the application.
* Annotated with @SpringBootApplication.

**3. Purpose of @SpringBootApplication**

* Combines @Configuration, @EnableAutoConfiguration, and @ComponentScan.
* Reduces configuration clutter.

**4. pom.xml**

* Manages dependencies using Maven.

**Key Tags to Walk Through:**

<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>

</dependency>

</dependencies>

**5. Dependency Hierarchy**

* In Eclipse: Go to pom.xml → Dependency Hierarchy tab.
* Helps identify:
  + Transitive dependencies,
  + Conflicts,
  + Dependency tree levels.

**Output:**



