**Title: Hands-On Exercises for Spring Web Project using Maven**

### Hands-On 1: Create a Spring Web Project using Maven

**Objective:** Set up a basic Spring Boot Web Project using Maven.

**Steps:**

1. **Navigate to Spring Initializr:**
   * URL: <https://start.spring.io>
   * Group: com.cognizant
   * Artifact: spring-learn
   * Dependencies: Spring Web, Spring Boot DevTools
2. **Download the project ZIP** and extract it in your Eclipse workspace.
3. **Build the project using Maven (command line):**

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

1. **Import Project in Eclipse:**
   * File > Import > Maven > Existing Maven Projects > Select folder > Finish
2. **Inspect and Run the Application:**
   * Open SpringLearnApplication.java
   * Add logging to main()
   * Run the application

**Key Concepts Covered:**

* @SpringBootApplication annotation
* SpringApplication.run() method
* Folder structure: src/main/java, src/main/resources, src/test/java
* POM.xml: Dependencies and dependency tree

### Hands-On 2: Spring Core – Load Country from Spring Configuration XML

**Objective:** Load a Country bean from an XML configuration file.

**Steps:**

1. **Create XML Configuration file** `\*\* in\*\*src/main/resources`

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 https://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="country" class="com.cognizant.springlearn.Country">  
 <property name="code" value="IN" />  
 <property name="name" value="India" />  
 </bean>  
  
</beans>

1. **Create** ``\*\* class with:\*\*
   * code and name properties
   * Constructor with debug log: “Inside Country Constructor”
   * Getter and Setter with logs
   * toString() method
2. **In** ``**, add method:**

public void displayCountry() {  
 LOGGER.info("START");  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = (Country) context.getBean("country", Country.class);  
 LOGGER.debug("Country : {}", country.toString());  
 LOGGER.info("END");  
}

1. **Invoke** \*\* in \*\*\*\* method.\*\*

**Concepts Covered:**

* ApplicationContext, ClassPathXmlApplicationContext
* Spring Bean definition and loading
* Logging instead of System.out.println()

This document covers two core hands-on implementations using Spring and Maven as described in the shared assignment document. Subsequent tasks like logging and scope will be added in future documents.