1. **Spring-Rest-Handson**

## Hands On 1

## Create a Spring Web Project using Maven

## 1. SpringLearnApplication.java

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

2. CLI Build Command

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

3. pom.xml

<parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.3.0</version>  
</parent>  
  
<dependencies>  
 <!-- Spring MVC + embedded Tomcat -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <!-- Hot reload & LiveReload -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
  
 <!-- Testing support -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
</dependencies>  
  
<build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
</build>

## Hands On 4

# **Spring Core – Load Country from Spring Configuration XML**

## Project Structure

*src  
└─ main  
 ├─ java  
 │ └─ com  
 │ └─ cognizant  
 │ └─ springlearn  
 │ ├─ Country.java  
 │ └─ SpringLearnApplication.java  
 └─ resources  
 └─ country.xml*

## country.xml

*<?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>*

## Country.java

*package com.cognizant.springlearn;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class Country {  
 private static final Logger LOGGER = LoggerFactory.getLogger(Country.class);  
  
 private String code;  
 private String name;  
  
 public Country() {  
 LOGGER.debug("Inside Country Constructor.");  
 }  
  
 public String getCode() {  
 LOGGER.debug("getCode() called.");  
 return code;  
 }  
  
 public void setCode(String code) {  
 LOGGER.debug("setCode() called.");  
 this.code = code;  
 }  
  
 public String getName() {  
 LOGGER.debug("getName() called.");  
 return name;  
 }  
  
 public void setName(String name) {  
 LOGGER.debug("setName() called.");  
 this.name = name;  
 }  
  
 @Override  
 public String toString() {  
 return "Country [code=" + code + ", name=" + name + "]";  
 }  
}*

## SpringLearnApplication.java

*package com.cognizant.springlearn;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class SpringLearnApplication {  
 private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);  
  
 public static void main(String[] args) {  
 displayCountry();  
 }  
  
 public static void displayCountry() {  
 ApplicationContext context =  
 new ClassPathXmlApplicationContext("country.xml");  
  
 Country country = context.getBean("country", Country.class);  
 LOGGER.debug("Country : {}", country);  
 ((ClassPathXmlApplicationContext) context).close();  
 }  
}*