**Spring REST using Spring Boot 3**

**1.spring-rest-handson**

**Hands on 1**

**Create a Spring Web Project using Maven**

**pom.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>

<optional>true</optional>

</dependency>

</dependencies>

**SpringLearnApplication.java**

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

System.out.println("Main Method Execution Started...");

SpringApplication.run(SpringLearnApplication.class, args);

System.out.println("Spring Boot Application Started Successfully...");

}

}

**OUTPUT:**

Main Method Execution Started...

... [Spring Boot logs]

Spring Boot Application Started Successfully...

**Logging Setup (application.properties)**

# Set server port

server.port=8082

# Set logging level

logging.level.root=INFO

# Customize log format

logging.pattern.console=%d{yyyy-MM-dd HH:mm:ss} - %msg%n

**Added logger manually in Code:**

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);

LOGGER.info("Info level log example");

**Console Output:**

2025-07-13 20:32:01 - Main Method Execution Started...

2025-07-13 20:32:01 - Spring Boot Application Started Successfully...

2025-07-13 20:32:01 - Info level log example

**Hands on 4**

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

**STEP 1: Country Class**

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("Inside getCode()");

return code;

}

public void setCode(String code) {

LOGGER.debug("Inside setCode()");

this.code = code;

}

public String getName() {

LOGGER.debug("Inside getName()");

return name;

}

public void setName(String name) {

LOGGER.debug("Inside setName()");

this.name = name;

}

@Override

public String toString() {

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

}

}

**STEP 2: Spring XML Configuration File**

<?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

http://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>

**STEP 3: Added displayCountry() Method in Main Class**

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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);

displayCountry();

LOGGER.info("END");

}

public static void displayCountry() {

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

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

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

}

}

**application.properties**

logging.level.root=DEBUG

logging.pattern.console=%d{yyyy-MM-dd HH:mm:ss} - %msg%n

**Console Output**:

2025-07-13 21:30:00 - START

2025-07-13 21:30:00 - Inside Country Constructor.

2025-07-13 21:30:00 - Inside setCode()

2025-07-13 21:30:00 - Inside setName()

2025-07-13 21:30:00 - Inside getCode()

2025-07-13 21:30:00 - Inside getName()

2025-07-13 21:30:00 - Country : Country [code=IN, name=India]

2025-07-13 21:30:00 – END