Spring REST Hands-On

**Create a Spring Web Project using Maven**

Code:

private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);

public static void main(String[] args) { SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("SpringLearnApplication started...");}

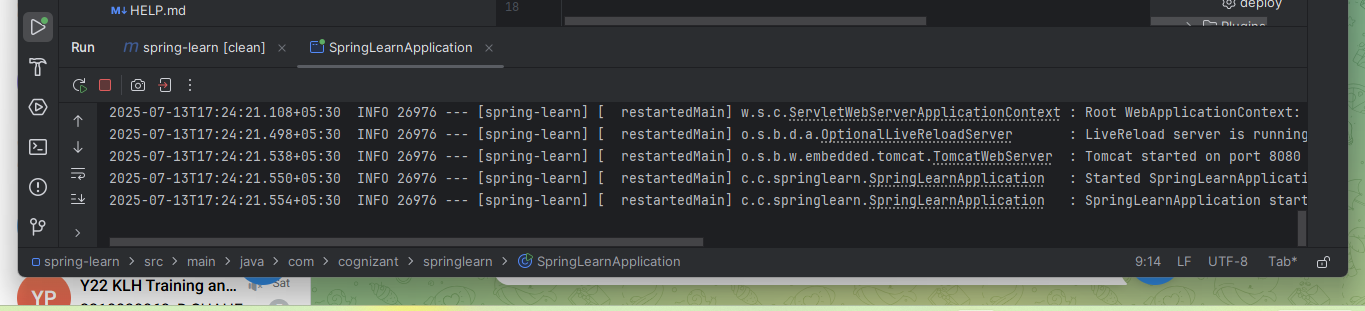
SME Walkthrough

1. 📂 src/main/java

Contains all the main application source code

Folder structure:

com.cognizant.springlearn └── SpringLearnApplication.java

Output: 

**Hands-On -Spring Core – Load Country from Spring Configuration XML**

country.xml

<bean id="country" class="com.cognizant.springlearn.Country"> <property name="code" value="IN" />

<property name="name" value="India" /></bean>

Country.java Class

public Country() {

LOGGER.debug("Inside Country Constructor.");}

public void setCode(String code) {

LOGGER.debug("Inside setCode()"); this.code = code;

}

3 Method displayCountry() in SpringLearnApplication.java

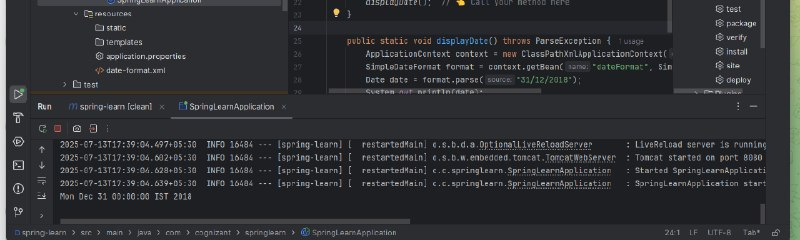
public static void displayCountry() { ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

Country country = context.getBean("country", Country.class); LOGGER.debug("Country : {}", country.toString());

}

Called displayCountry() from main() Method

displayDate();displayCountry();

output: 

**Hello World RESTful Web Service**

Code:

HelloController.java

package com.cognizant.springlearn.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class HelloController {

private static final Logger LOGGER = LoggerFactory.getLogger(HelloController.class);

@GetMapping("/hello")

public String sayHello() {

LOGGER.info("START - sayHello()");

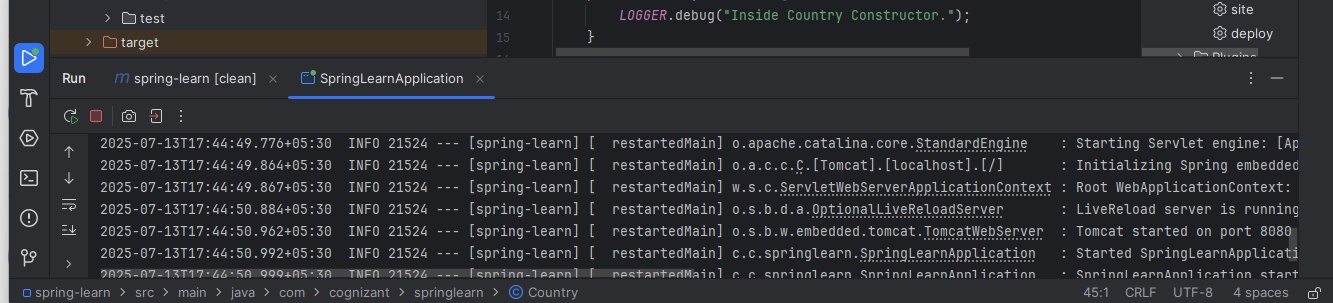
String message = "Hello World!!";

LOGGER.info("END - sayHello()");

return message;

}

}

Output: 

**2.Hello World & Country RESTful Web Service (Spring Boot)**

HTTP Request and Response Overview

HTTP Request Example:

GET /hello.txt HTTP/1.1

User-Agent: curl/7.16.3 Host: www.example.com

Accept-Language: en

Annotation: @RestController

Method:

@GetMapping("/hello")public String sayHello() {

LOGGER.info("START - sayHello()"); String message = "Hello World!!";

LOGGER.info("END - sayHello()"); return message;

}

Controller Class: CountryController.java

Method:

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START - getCountryIndia()");

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

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

LOGGER.info("END - getCountryIndia()");

return country;

}

Request: http://localhost:8080/country

Response:

{

"code": "IN",

"name": "India"

}

Testing the REST APIs🖥️ A. Using Browser (GET only)

URL: http://localhost:8080/hello

URL: http://localhost:8080/country

Output:



**REST - Get country based on country code**

Code:

CountryController.java

java

CopyEdit

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.Country;

import com.cognizant.springlearn.service.CountryService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.web.bind.annotation.\*;

@RestController

public class CountryController {

private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);

@Autowired

private CountryService countryService;

// Endpoint: http://localhost:8080/country

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START - getCountryIndia()");

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

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

LOGGER.info("END - getCountryIndia()");

return country;

}

// Endpoint: http://localhost:8080/countries/in

@GetMapping("/countries/{code}")

public Country getCountry(@PathVariable String code) {

LOGGER.info("START - getCountry() with code: {}", code);

Country country = countryService.getCountry(code);

LOGGER.info("END - getCountry()");

return country;

}

}

Output:

