SPRING REST USING SPRINGBOOT:

2.spring-Rest:

Hands-on-1:

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() {

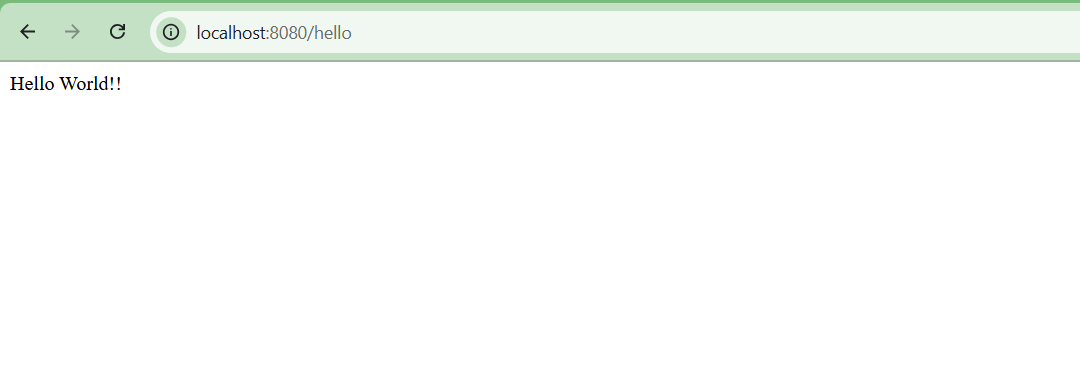
LOGGER.info("START");

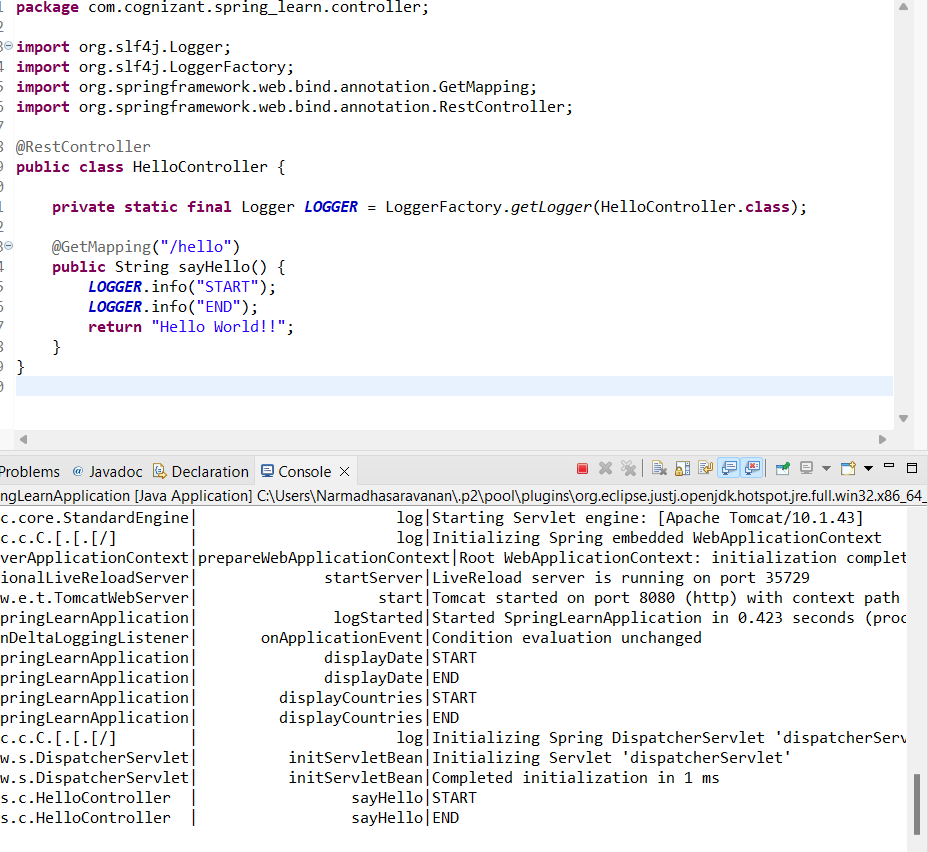
LOGGER.info("END");

return "Hello World!!";

}

}





Hands-on-2:

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

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

@RestController

public class CountryController {

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

@RequestMapping("/country")

public Country getCountryIndia() {

*LOGGER*.info("START");

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

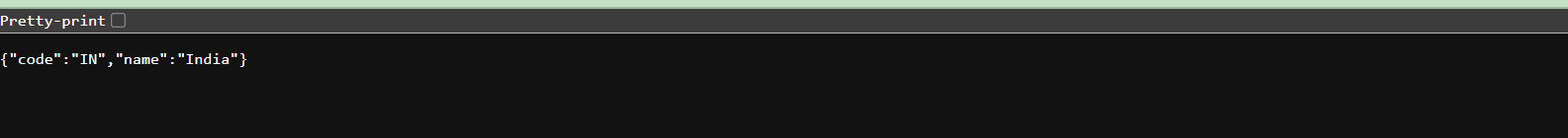
Country country = context.getBean("in", Country.class);

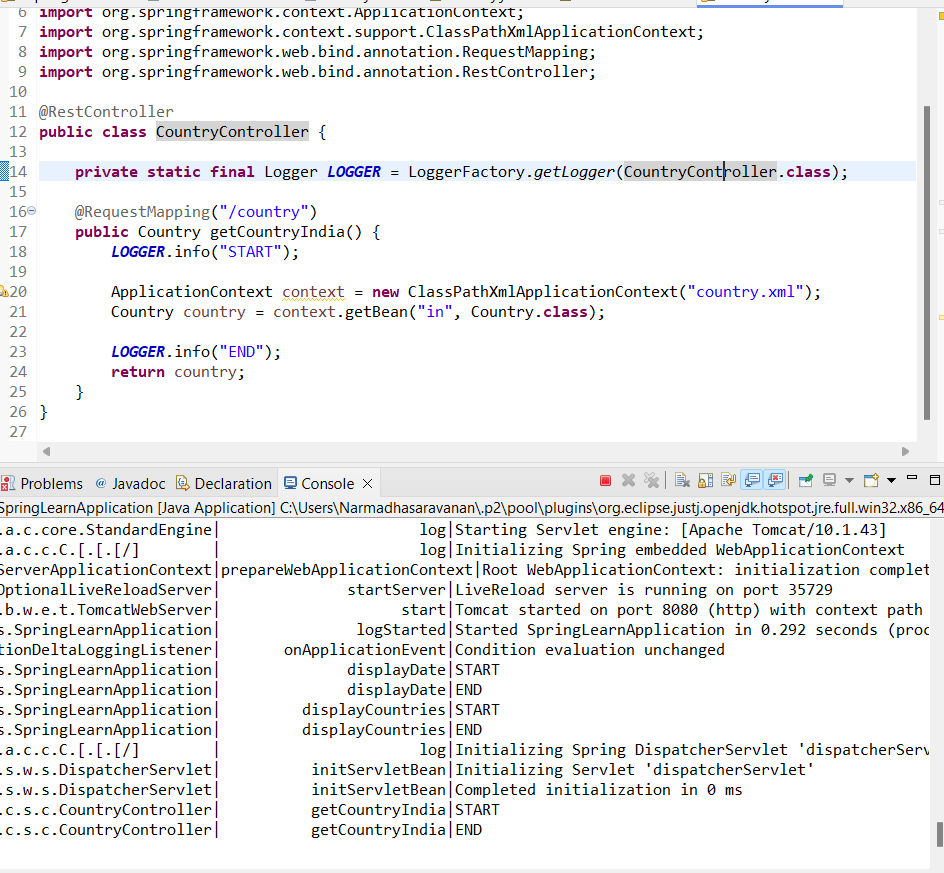
*LOGGER*.info("END");

return country;

}

}





Hands-on-3:

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

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

import java.util.List;

@RestController

public class CountryController {

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

@RequestMapping("/country")

public Country getCountryIndia() {

*LOGGER*.info("START");

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

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

*LOGGER*.info("END");

return country;

}

@RequestMapping("/countries")

public List<Country> getAllCountries() {

*LOGGER*.info("START");

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

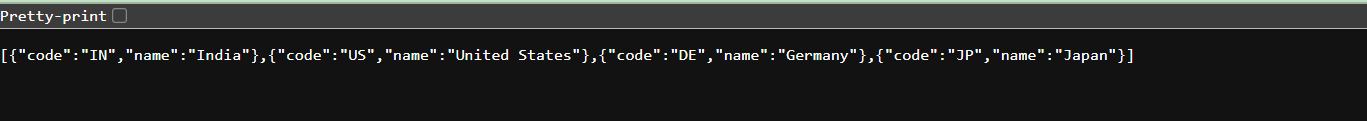
List<Country> countries = context.getBean("countryList", List.class);

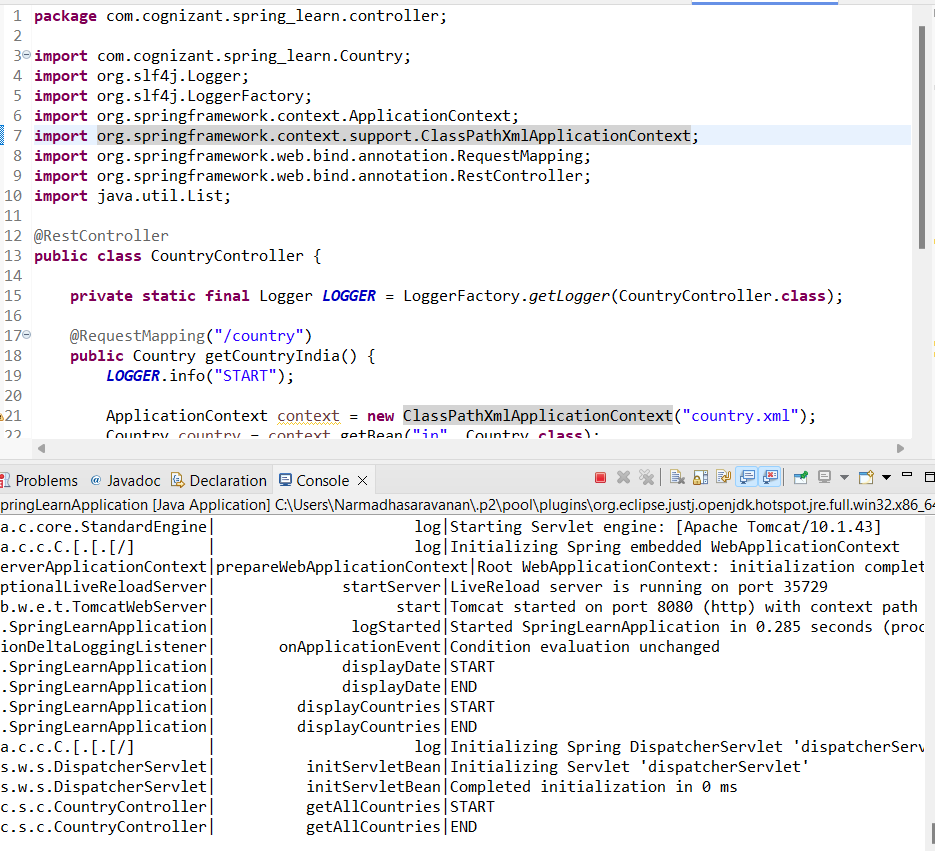
*LOGGER*.info("END");

return countries;

}

}





Hands-on-4:

package com.cognizant.spring\_learn.service;

import com.cognizant.spring\_learn.Country;

import com.cognizant.spring\_learn.service.exception.CountryNotFoundException;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

public Country getCountry(String code) throws CountryNotFoundException {

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

List<Country> countries = context.getBean("countryList", List.class);

return countries.stream()

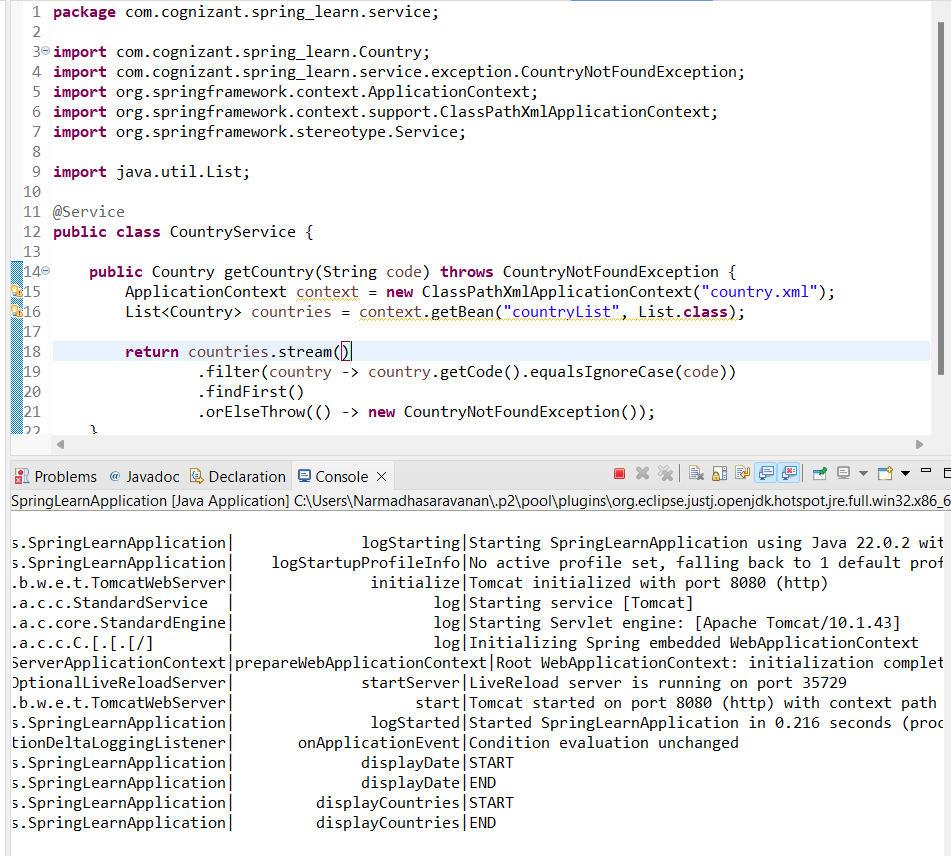
.filter(country -> country.getCode().equalsIgnoreCase(code))

.findFirst()

.orElseThrow(() -> new CountryNotFoundException());

}

}



Hands-on-5:

package com.cognizant.springlearn.service.exception;

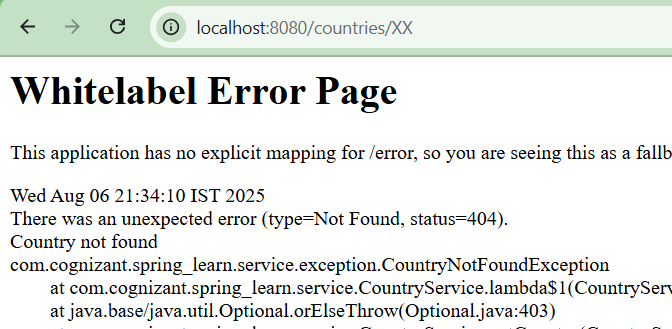
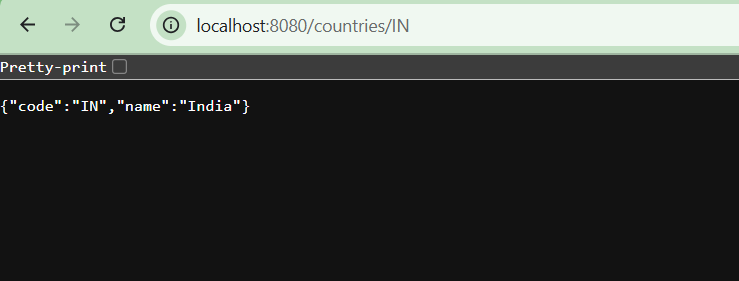
import org.springframework.http.HttpStatus;

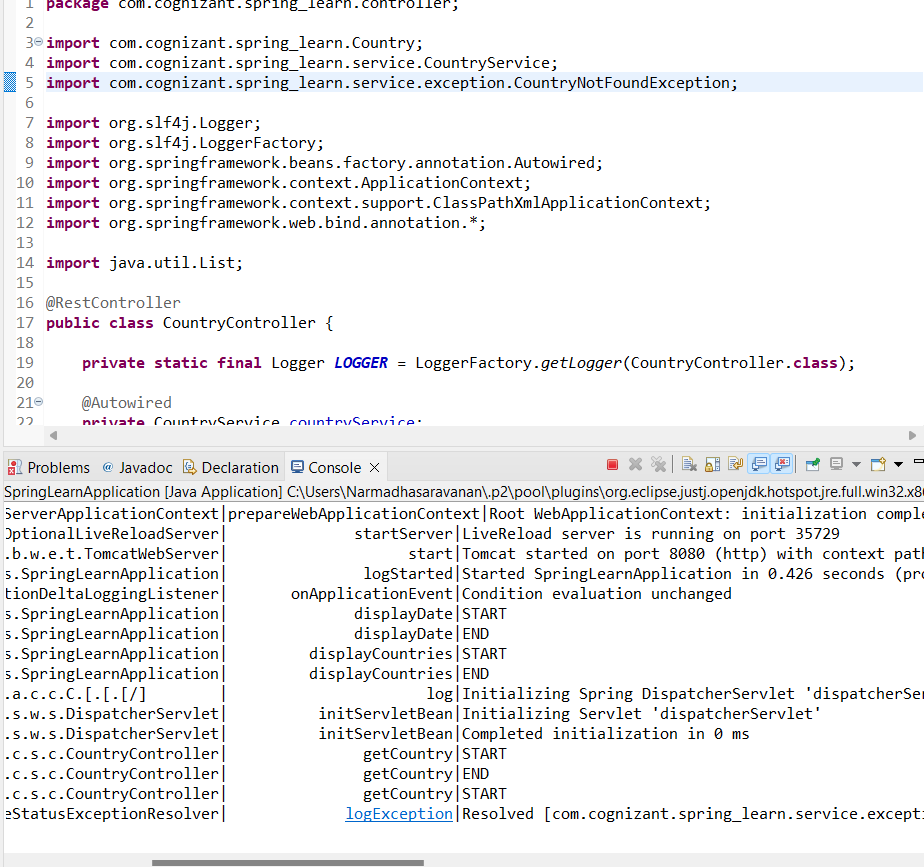
import org.springframework.web.bind.annotation.ResponseStatus;

@ResponseStatus(value = HttpStatus.NOT\_FOUND, reason = "Country not found")

public class CountryNotFoundException extends Exception {

}





Hands-on-6:

package com.cognizant.spring\_learn;

import static org.junit.jupiter.api.Assertions.*assertNotNull*;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.*get*;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.\*;

import com.cognizant.spring\_learn.controller.CountryController;

import org.junit.jupiter.api.Test;

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

import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.test.web.servlet.MockMvc;

import org.springframework.test.web.servlet.ResultActions;

@SpringBootTest

@AutoConfigureMockMvc

public class SpringLearnApplicationTests {

@Autowired

private CountryController countryController;

@Autowired

private MockMvc mvc;

// Test if CountryController is loaded into Spring Context

@Test

public void contextLoads() {

*assertNotNull*(countryController);

}

// Test GET /country (India details)

@Test

public void testGetCountryIndia() throws Exception {

ResultActions actions = mvc.perform(*get*("/country"));

actions.andExpect(*status*().isOk());

actions.andExpect(*jsonPath*("$.code").value("IN"));

actions.andExpect(*jsonPath*("$.name").value("India"));

}

// Test GET /countries (All countries list)

@Test

public void testGetAllCountries() throws Exception {

ResultActions actions = mvc.perform(*get*("/countries"));

actions.andExpect(*status*().isOk());

actions.andExpect(*jsonPath*("$[0].code").value("IN"));

actions.andExpect(*jsonPath*("$[1].code").value("US"));

actions.andExpect(*jsonPath*("$[2].code").value("DE"));

actions.andExpect(*jsonPath*("$[3].code").value("JP"));

}

// Test GET /countries/{code} - Valid Code

@Test

public void testGetCountryByCode() throws Exception {

ResultActions actions = mvc.perform(*get*("/countries/IN"));

actions.andExpect(*status*().isOk());

actions.andExpect(*jsonPath*("$.code").value("IN"));

actions.andExpect(*jsonPath*("$.name").value("India"));

}

// Test GET /countries/{code} - Invalid Code (Exception Scenario)

@Test

public void testGetCountryByInvalidCode() throws Exception {

ResultActions actions = mvc.perform(*get*("/countries/XX"));

actions.andExpect(*status*().isNotFound());

actions.andExpect(*status*().reason("Country not found"));

}

}

