**Step 1: Hello World REST API**

**Step 1: Set Server Port**

**File: application.properties**

server.port=8083

**Step 2: Create Controller Package**

Package name: com.cognizant.spring\_learn.controller

**Step 3: Created HelloController.java**

package com.cognizant.spring\_learn.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");

String message = "Hello World!!";

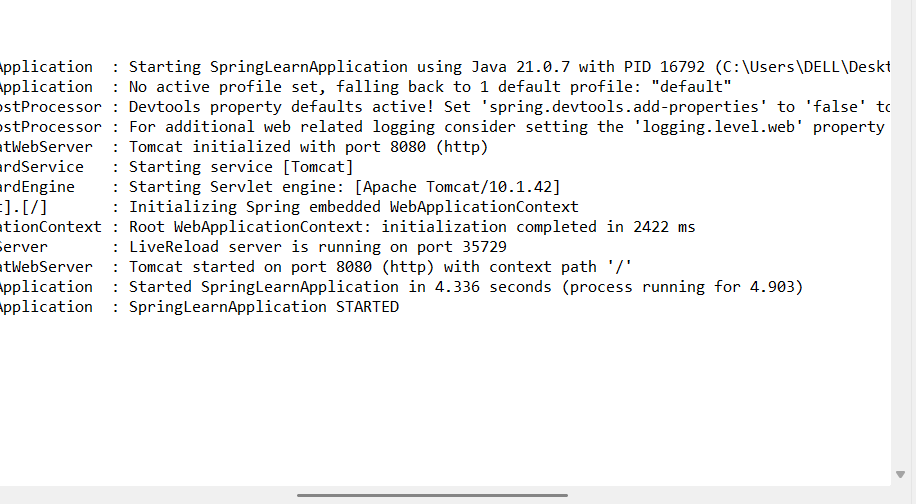
LOGGER.info("END");

return message;

}

}

**Step 4: Run the Application**

****

**Step 2: Create /country Endpoint to Return a JSON Object**

**Code**

**Country.java**

package com.cognizant.spring\_learn;

public class Country {

private String code;

private String name;

public Country() {}

public String getCode() { return code; }

public void setCode(String code) { this.code = code; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

@Override

public String toString() {

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

}

}

HelloController.java

@GetMapping("/country")

public Country getCountryIndia() {

Country country = new Country();

country.setCode("IN");

country.setName("India");

return country;

}

**HelloController.java**

@GetMapping("/country")

public Country getCountryIndia() {

Country country = new Country();

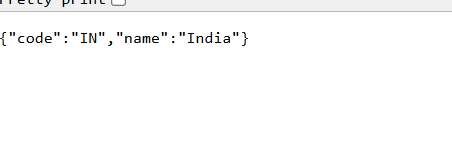
country.setCode("IN");

country.setName("India");

return country;

}

**Output:**



**Step 3: Return a List of Countries using /countries Endpoint**

**STEP 1: Update HelloController.java**

@GetMapping("/countries")

public List<Country> getAllCountries() {

List<Country> countries = new ArrayList<>();

Country india = new Country();

india.setCode("IN");

india.setName("India");

Country usa = new Country();

usa.setCode("US");

usa.setName("United States");

Country japan = new Country();

japan.setCode("JP");

japan.setName("Japan");

countries.add(india);

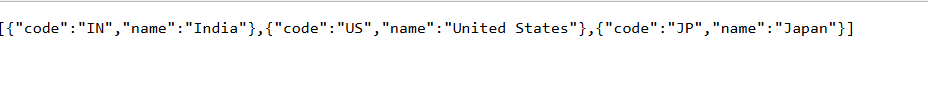
countries.add(usa);

countries.add(japan);

return countries;

}

**Output:**

****

**Step 4: Return a Country by Code (Using /countries/{code})**

**STEP 1: Create a Custom Exception Class**

**CountryNotFoundException.java**

package com.cognizant.spring\_learn.exception;

public class CountryNotFoundException extends Exception {

public CountryNotFoundException(String message) {

super(message);

}

}

**STEP 2: Add /countries/{code} Endpoint in HelloController.java**

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

public Country getCountryByCode(@PathVariable String code) throws CountryNotFoundException {

List<Country> countries = new ArrayList<>();

Country india = new Country(); india.setCode("IN"); india.setName("India");

Country usa = new Country(); usa.setCode("US"); usa.setName("United States");

Country japan = new Country(); japan.setCode("JP"); japan.setName("Japan");

countries.add(india);

countries.add(usa);

countries.add(japan);

for (Country c : countries) {

if (c.getCode().equalsIgnoreCase(code)) {

return c;

}

}

throw new CountryNotFoundException("Country not found");

}

**STEP 3: Add Global Exception Handler**

package com.cognizant.spring\_learn.controller;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

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

import java.util.HashMap;

import java.util.Map;

@ControllerAdvice

public class GlobalExceptionHandler {

@ExceptionHandler(CountryNotFoundException.class)

public ResponseEntity<?> handleCountryNotFound(CountryNotFoundException ex) {

Map<String, Object> body = new HashMap<>();

body.put("status", HttpStatus.NOT\_FOUND.value());

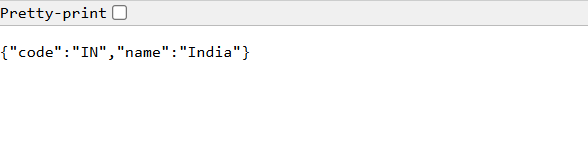
body.put("message", ex.getMessage());

return new ResponseEntity<>(body, HttpStatus.NOT\_FOUND);

}

}

**Output:**

****

**Step 5: Unit Testing with MockMVC**

**STEP 1: Add Dependencies**

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

**STEP 2: Create Test Class**

package com.cognizant.spring\_learn.controller;

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

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

import com.cognizant.spring\_learn.SpringLearnApplication;

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;

@SpringBootTest(classes = SpringLearnApplication.class)

@AutoConfigureMockMvc

public class HelloControllerTest {

@Autowired

private MockMvc mockMvc;

@Test

public void testHelloEndpoint() throws Exception {

mockMvc.perform(get("/hello"))

.andExpect(status().isOk())

.andExpect(content().string("Hello World!!"));

}

@Test

public void testGetCountryIndia() throws Exception {

mockMvc.perform(get("/country"))

.andExpect(status().isOk())

.andExpect(jsonPath("$.code").value("IN"))

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

}

@Test

public void testGetAllCountries() throws Exception {

mockMvc.perform(get("/countries"))

.andExpect(status().isOk())

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

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

.andExpect(jsonPath("$[2].code").value("JP"));

}

@Test

public void testGetCountryByCodeSuccess() throws Exception {

mockMvc.perform(get("/countries/IN"))

.andExpect(status().isOk())

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

}

@Test

public void testGetCountryByCodeNotFound() throws Exception {

mockMvc.perform(get("/countries/XX"))

.andExpect(status().isNotFound())

.andExpect(jsonPath("$.message").value("Country not found"));

}

}

**STEP 3: Run the Tests**

