# WEEK 4

## Spring REST using Spring Boot 3

### Create a Spring Web Project using Maven

SpringLearnApplication.java

package com.example;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(SpringLearnApplication.class, args);  
 System.out.println("Spring Boot Application started...");  
 }  
}

OUTPUT:

Spring Boot Application started...

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

Country.java

package com.example.model;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class Country {  
 private String code;  
 private String name;  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(Country.class);  
  
 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 + "]";  
 }  
}

SpringLearnApplication.java

package com.example;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
import com.example.model.Country;  
  
public class SpringLearnApplication {  
 private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);  
  
 public static void main(String[] args) {  
 LOGGER.debug("Inside main()");  
 displayCountry();  
 }  
  
 public static void displayCountry() {  
 LOGGER.debug("Inside displayCountry()");  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = context.getBean("country", Country.class);  
 System.out.println("Country: " + country);  
 }  
}

OUTPUT:

Country [code=IN, name=India]

### Hello World RESTful Web Service

HelloController.java

package com.example.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.debug("Start - sayHello()");  
 String message = "Hello World!!";  
 LOGGER.debug("End - sayHello()");  
 return message;  
 }  
}

OUTPUT:

GET http://localhost:9090/hello  
Response: "Hello World!!"

### REST - Country Web Service

CountryController.java

package com.example.controller;  
  
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 com.example.model.Country;  
  
@RestController  
public class CountryController {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);  
  
 @RequestMapping("/country")  
 public Country getCountryIndia() {  
 LOGGER.debug("Start - getCountryIndia()");  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = context.getBean("country", Country.class);  
 LOGGER.debug("End - getCountryIndia()");  
 return country;  
 }  
}

SpringLearnApplication.java

package com.example;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
 public static void main(String[] args) {  
 System.out.println("Inside main()");  
 SpringApplication.run(SpringLearnApplication.class, args);  
 }  
}

OUTPUT:

{  
 "code": "IN",  
 "name": "India"  
}

What happens in the controller method?

@RequestMapping("/country")  
public Country getCountryIndia() {  
 ...  
 return country;  
}

When a client sends a GET request to /country, this method runs.  
  
It:  
- Loads the Spring XML file (country.xml)  
- Fetches the bean with ID "country" — which is the India object  
- Returns the Country Java object as a response

How is the Java bean converted to JSON?  
  
Spring Boot uses Jackson (a JSON processor) behind the scenes.  
  
When a method in a controller annotated with @RestController returns an object like Country, Spring Boot:  
- Converts the object to JSON using Jackson  
- Sets the HTTP response’s Content-Type to application/json  
- Sends it as the response body

In network tab of developer tools show the HTTP header details received:  
1. Right-click on page → Inspect  
2. Go to Network tab  
3. Refresh http://localhost:9090/country  
4. Click the /country request  
  
Check:  
- Status Code: 200 OK  
- Content-Type: application/json  
- Request Method: GET  
- Response: Should show {"code":"IN", "name":"India"}

### REST - Get country based on country code

CountryController.java

package com.example.controller;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import com.example.model.Country;  
import com.example.service.CountryService;  
  
@RestController  
public class CountryController {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);  
  
 @Autowired  
 private CountryService countryService;  
  
 @GetMapping("/countries/{code}")  
 public Country getCountry(@PathVariable String code) {  
 LOGGER.debug("Start - getCountry() with code: {}", code);  
 Country country = countryService.getCountry(code);  
 LOGGER.debug("End - getCountry(): {}", country);  
 return country;  
 }  
}

OUTPUT:

GET http://localhost:9090/countries/IN  
Response: {"code": "IN", "name": "India"}

### Creating users and roles in Spring Security

Country.java

package com.example.model;  
  
public class Country {  
 private String code;  
 private String name;  
  
 public Country() {}  
  
 public Country(String code, String name) {  
 this.code = code;  
 this.name = name;  
 }  
  
 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; }  
}

CountryService.java

package com.example.service;  
  
import com.example.model.Country;  
import org.springframework.stereotype.Service;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
import java.util.List;  
  
@Service  
public class CountryService {  
  
 public List<Country> getAllCountries() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countries = (List<Country>) context.getBean("countryList");  
 return countries;  
 }  
  
 public Country getCountry(String code) {  
 for (Country country : getAllCountries()) {  
 if (country.getCode().equalsIgnoreCase(code)) {  
 return country;  
 }  
 }  
 return null;  
 }  
}

OUTPUT:

curl -u user:pwd http://localhost:9090/countries