**Cognizant Deep Nurture 4.0 Hands-on Exercise - Week 4**

Exersice 1:   
Hello World RESTful Web ServiceWrite a REST service in the spring learn application created earlier, thatreturns the text "Hello World!!" using Spring Web Framework. Refer detailsbelow:Method: GETURL: /helloController: com.cognizant.spring-learn.controller.HelloControllerMethod Signature: public String sayHello()Method Implementation: return hard coded string "Hello World!!"Sample Request: http://localhost:8083/helloSample Response: Hello World!!IMPORTANT NOTE: Don't forget to include start and end log in the sayHello()method.Try the URL http://localhost:8083/hello in both chrome browser and postman.SME to explain the following aspects:In network tab of developer tools show the HTTP header details receivedIn postman click on "Headers" tab to view the HTTP header details received    
   
   
1. Controller Creation

HelloController.java

**Location**: src/main/java/com/cognizant/springlearn/controller/

java

CopyEdit

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");

String message = "Hello World!!";

LOGGER.info("End");

return message;

}

}

2. Configure application.properties

src/main/resources/application.properties

properties

CopyEdit

server.port=8083

3. Run Application

Right-click SpringLearnApplication.java → Run As → Java Application   
 App will start at: http://localhost:8083

4. Test the Endpoint

Chrome Browser or Postman

Visit:

bash

CopyEdit

http://localhost:8083/hello

OUTPUT: 

614fdf641b022fc472f7911c91ee223a.png

Exersice 2   
Write a REST service that returns India country details in the earlier createdspring learn application.URL: /countryController: com.cognizant.spring-learn.controller.CountryControllerMethod Annotation: @RequestMappingMethod Name: getCountryIndia()Method Implementation: Load India bean from spring xml configuration andreturnSample Request: http://localhost:8083/countrySample Response:{"code": "IN","name": "India"}SME to explain the following aspects:What happens in the controller method?How the bean is converted into JSON reponse?In network tab of developer tools show the HTTP header details receivedIn postman click on "Headers" tab to view the HTTP header details received    
   
   
1. Step-by-Step Implementation

Country.java

(If not already created)

**Location**: src/main/java/com/cognizant/springlearn/model/

java

CopyEdit

package com.cognizant.springlearn.model;

public class Country {

private String code;

private String name;

// Getters and Setters

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 + "]";

}

}

country.xml

**Location**: src/main/resources/country.xml

xml

CopyEdit

<?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

https://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="in" class="com.cognizant.springlearn.model.Country">

<property name="code" value="IN"/>

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

</bean>

</beans>

CountryController.java

**Location**: src/main/java/com/cognizant/springlearn/controller/

java

CopyEdit

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.model.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 = (Country) context.getBean("in");

LOGGER.info("End");

return country;

}

}

application.properties

properties

CopyEdit

server.port=8083

   
OUTPUT:   
 



Exercise 3:   
REST - Get country based on country codeWrite a REST service that returns a specific country based on country code.The country code should be case insensitive.Controller: com.cognizant.spring-learn.controller.CountryControllerMethod Annotation: @GetMapping("/countries/{code}")Method Name: getCountry(String code)Method Implemetation: Invoke countryService.getCountry(code)Service Method: com.cognizant.spring-learn.service.CountryService.getCountry(String code)Service Method Implementation:Get the country code using @PathVariableGet country list from country.xmlIterate through the country listMake a case insensitive matching of country code and return the country.Lambda expression can also be used instead of iterating the country listSample Request: http://localhost:8083/country/inSample Response:{"code": "IN","name": "India"}    
   
   
   
Step 1: country.xml – Add List of Countries

**Location**: src/main/resources/country.xml

xml

CopyEdit

<?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

https://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<bean class="com.cognizant.springlearn.model.Country">

<property name="code" value="IN"/>

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

</bean>

<bean class="com.cognizant.springlearn.model.Country">

<property name="code" value="US"/>

<property name="name" value="United States"/>

</bean>

</list>

</constructor-arg>

</bean>

</beans>

Step 2: CountryService.java

**Location**: src/main/java/com/cognizant/springlearn/service/

java

CopyEdit

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.model.Country;

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) {

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

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

return countryList.stream()

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

.findFirst()

.orElse(null); // or throw exception if needed

}

}

Step 3: CountryController.java

**Location**: src/main/java/com/cognizant/springlearn/controller/

java

CopyEdit

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.model.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.web.bind.annotation.\*;

@RestController

public class CountryController {

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

@Autowired

private CountryService countryService;

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

public Country getCountry(@PathVariable String code) {

LOGGER.info("Start");

Country country = countryService.getCountry(code);

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

LOGGER.info("End");

return country;

}

}

OUTPUT: 

