**SPRING REST USING SPRING BOOT 3**

**2. Spring – rest – handson**

**Hello World RESTful Web Service**

1. Eclipse IDE -> File -> New -> Spring starter project

* Name: spring-learn
* Group: com.cognizant
* Artifact: spring-learn
* Type: Maven
* Packaging: Jar
* Click Next.
* Select dependencies: Spring web
* Click finish.

1. HelloController (controller class)

* Src/main/java -> package (com.cognizant.springlearn) -> new -> class (HelloController)
* HelloController.java:

package com.cognizant.springlearn;

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()");

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

return "Hello World!!";

}

}

1. Main application (created automatically)

* SpringLearnApplication.java

package com.cognizant.springlearn;

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);

}

}

1. Application.properties

server.port=8083

spring.application.name=spring-learn

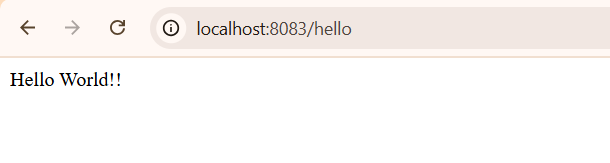
logging.level.org.springframework=info

logging.level.com.cognizant.springlearn=debug

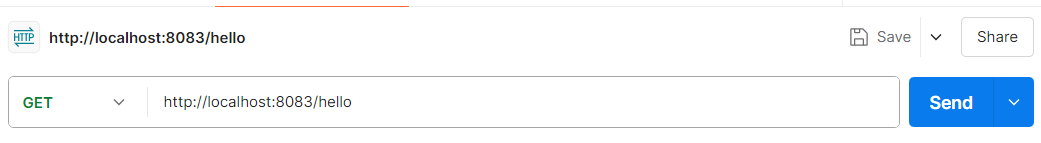
logging.pattern.console=%d{yyMMdd}|%d{HH:mm:ss.SSS}|%-20.20thread|%5p|%-25.25logger{25}|%25M|%m%n

1. SpringLearnApplication -> Run As -> Java Application
2. Testing

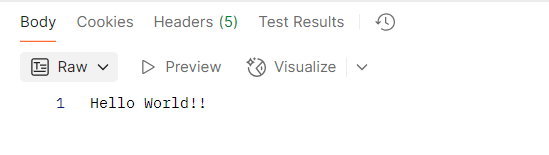
* Browser
  + <http://localhost:8083/hello>



* Postman
  + Set method to GET
  + <http://localhost:8083/hello>

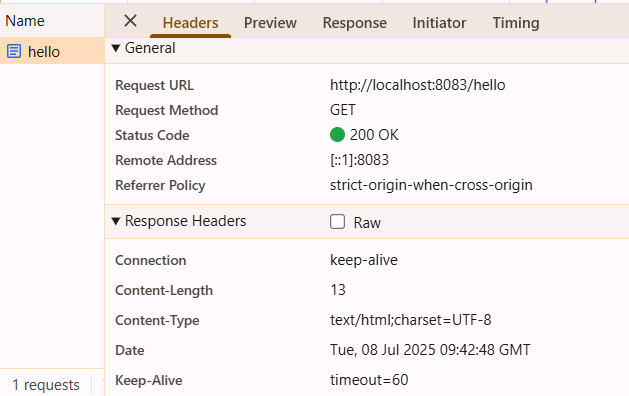


* + Click send

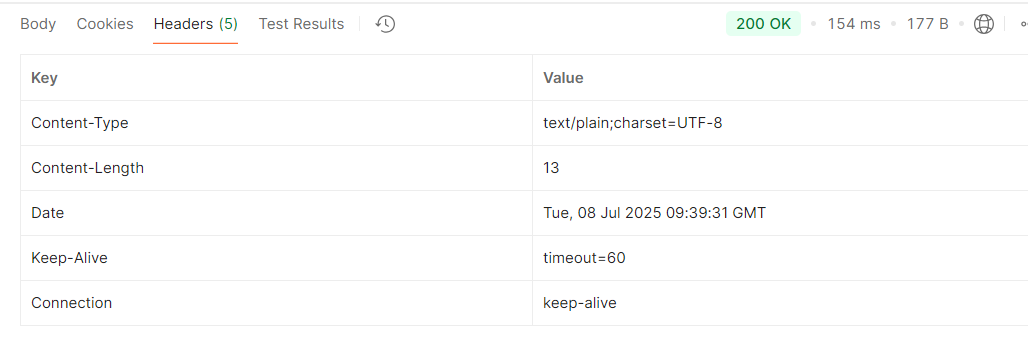


1. Verifying HTTP Headers

* Chrome
  + Press F12 or right click anywhere on page, choose ‘Inspect’.
  + Click ‘Network’ tab
  + Network tab must be active. You will see a row named hello
  + Click that row. You’ll see:



* Postman
* Click on Headers to see response headers



**REST - Country Web Service**

1. spring-learn.xml

* Src/main/resources -> New -> File -> springlearn.xml

<?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

http://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>

1. Country model class

* Src/main/java -> package (com.cognizant.springlearn.model) -> class (Country)
* Country.java

package com.cognizant.springlearn.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;

}

}

1. CountryController class

* Src/main/java -> package (com.cognizant.springlearn.controller) -> class (CountryController)
* CountryController.java

package com.cognizant.springlearn.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;

}

}

1. Src/main/java -> com.cognizant.springlearn -> SpringLearnApplication.java

package com.cognizant.spring-learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

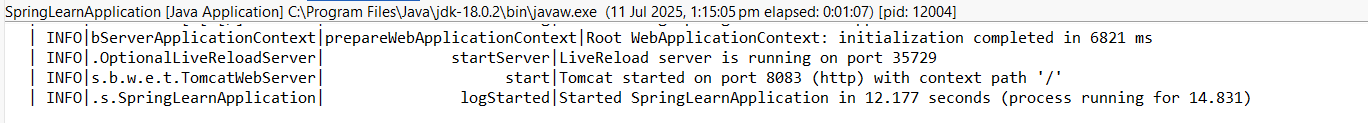
public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

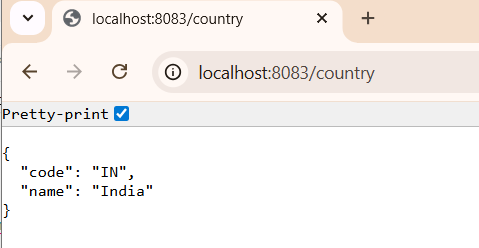
}

1. SpringLearnApplication -> Run As -> Java Application



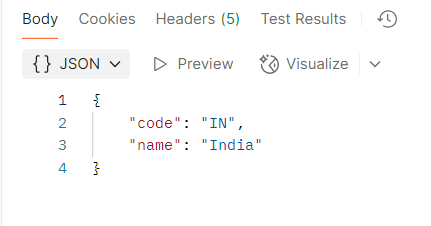
1. Test in browser ot Postman

* <http://localhost:8083/country>
* Browser:



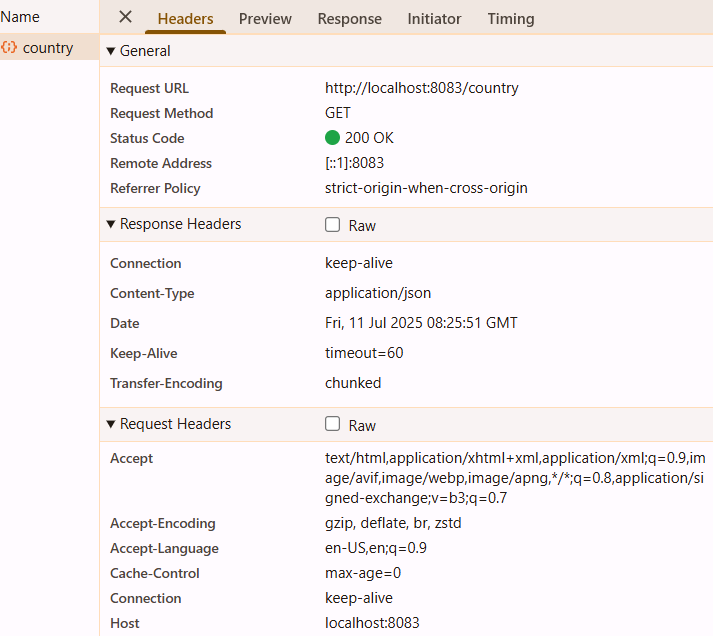
* Postman



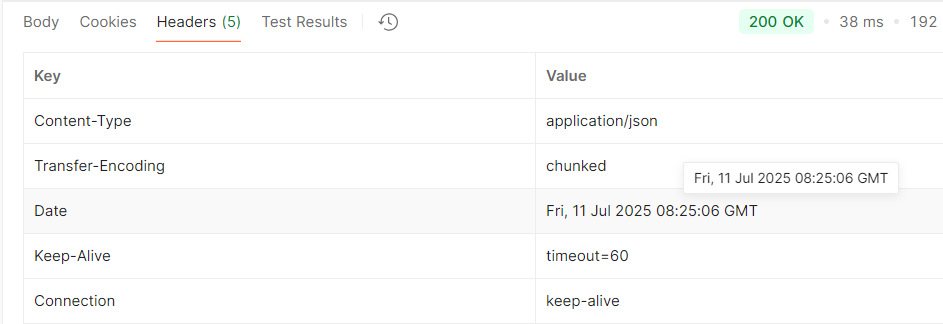


1. HTTP Headers

* Chrome -> Right click -> Inspect -> Network -> Country



* Postman -> Headers



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

.

1. Country.xml

* Src/main/resources -> New -> File
* country.xml

<?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

http://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>

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

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

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

</bean>

</list>

</constructor-arg>

</bean>

</beans>

1. CountryService.java

* Src/main/java -> package (com.cognizant.springlearn.service) -> class (CountryService)
* CountryService.java

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);

}

}

1. CountryController.java

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.service.CountryService;

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

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

@RestController

public class CountryController {

@Autowired

private CountryService countryService;

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

public Country getCountry(@PathVariable String code) {

return countryService.getCountry(code);

}

}

1. Country.java

package com.cognizant.springlearn.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) {

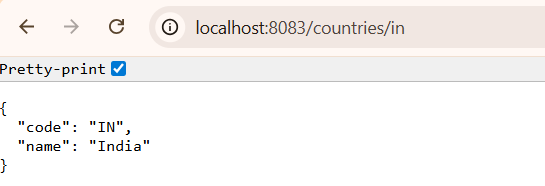
this.name = name;

}

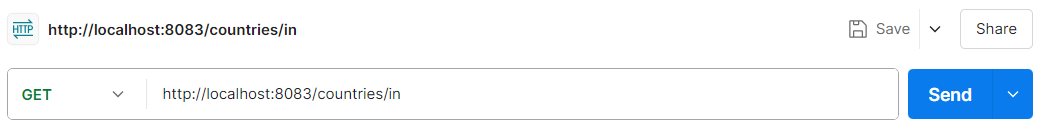
}

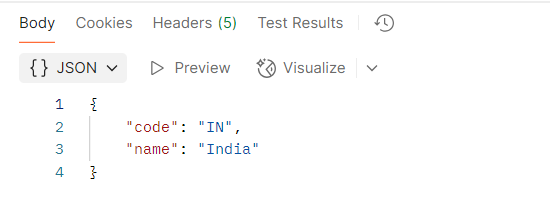
1. SpringLearnApplication -> Run As -> Java Application

* Browser:
* <http://localhost:8083/countries/in>



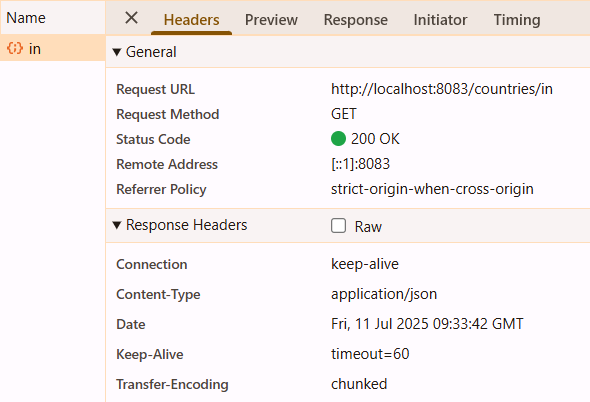
* Postman
* <http://localhost:8083/countries/in>





1. HTTP Headers

-Chrome



-Postman

