SPRING REST HANDS ON Solutions

1. Hello World RESTful Web Service

Write a REST service in the spring learn application created earlier, that returns the text "Hello World!!" using Spring Web Framework. Refer details below:

Method: GET
URL: /hello
Controller: com.cognizant.springlearn.controller.HelloController
Method Signature: public String sayHello()
Method Implementation: return hard coded string "Hello World!!"
Sample Request: http://localhost:8083/hello
Sample 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

CODE : HelloContoller.java file

```
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: sayHello()");
```

```java
        String response = "Hello World!!";
        LOGGER.info("END: sayHello()");
        return response;
    }
}
```

CONSOLE:


```
Inside main method...
Inside main method...

  /\\ /___'_ __ _ __ (_)_ __  __ _ \ \ \ \
 ( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
  \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
   '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.5.3)

2025-07-13T12:51:54.629+05:30  INFO 15960 --- [spring-learn] [  restartedMain] c.c.springlearn.SpringLearnApplication   : Starting SpringLearnApplication using Java 24.0.1 with PID 1596
0 (C:\Users\nehar\OneDrive\spring tools\spring-learn\spring-learn\target\classes started by nehar in C:\Users\nehar\OneDrive\spring tools\spring-learn)
2025-07-13T12:51:54.638+05:30  INFO 15960 --- [spring-learn] [  restartedMain] c.c.springlearn.SpringLearnApplication   : No active profile set, falling back to 1 default profile: "defa
ult"
2025-07-13T12:51:54.716+05:30  INFO 15960 --- [spring-learn] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-pro
perties' to 'false' to disable
2025-07-13T12:51:54.717+05:30  INFO 15960 --- [spring-learn] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'loggin
g.level.web' property to 'DEBUG'
2025-07-13T12:51:56.333+05:30  INFO 15960 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8083 (http)
2025-07-13T12:51:56.365+05:30  INFO 15960 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2025-07-13T12:51:56.366+05:30  INFO 15960 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.42]
2025-07-13T12:51:56.427+05:30  INFO 15960 --- [spring-learn] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2025-07-13T12:51:56.428+05:30  INFO 15960 --- [spring-learn] [  restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1709 ms
2025-07-13T12:51:57.014+05:30  INFO 15960 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2025-07-13T12:51:57.105+05:30  INFO 15960 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8083 (http) with context path '/'
2025-07-13T12:51:57.118+05:30  INFO 15960 --- [spring-learn] [  restartedMain] c.c.springlearn.SpringLearnApplication   : Started SpringLearnApptext: initialization completed in 1709 ms
2025-07-13T12:51:57.014+05:30  INFO 15960 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2025-07-13T12:51:57.105+05:30  INFO 15960 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8083 (http) with context path '/'
text: initialization completed in 1709 ms
2025-07-13T12:51:57.014+05:30  INFO 15960 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
text: initialization completed in 1709 ms
text: initialization completed in 1709 ms
2025-07-13T12:51:57.014+05:30  INFO 15960 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2025-07-13T12:51:57.105+05:30  INFO 15960 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8083 (http) with context path '/'
2025-07-13T12:51:57.118+05:30  INFO 15960 --- [spring-learn] [  restartedMain] c.c.springlearn.SpringLearnApplication   : Started SpringLearnApplication in 3.222 seconds (process runnin
g for 3.82)
Spring Boot Application started!
```
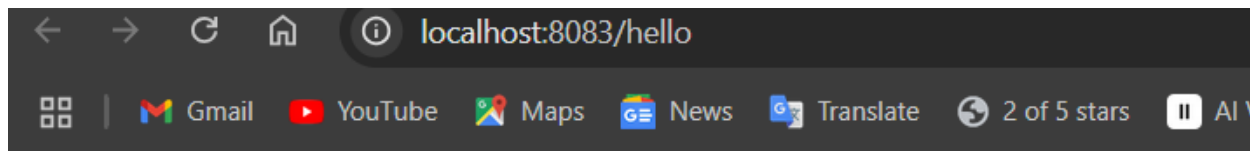
```
2025-07-13T12:52:29.089+05:30  INFO 15960 --- [spring-learn] [nio-8083-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-07-13T12:52:29.090+05:30  INFO 15960 --- [spring-learn] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2025-07-13T12:52:29.091+05:30  INFO 15960 --- [spring-learn] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 1 ms
2025-07-13T12:52:29.181+05:30  INFO 15960 --- [spring-learn] [nio-8083-exec-1] c.c.s.controller.HelloController         : START: sayHello()
2025-07-13T12:52:29.181+05:30  INFO 15960 --- [spring-learn] [nio-8083-exec-1] c.c.s.controller.HelloController         : END: sayHello()
```
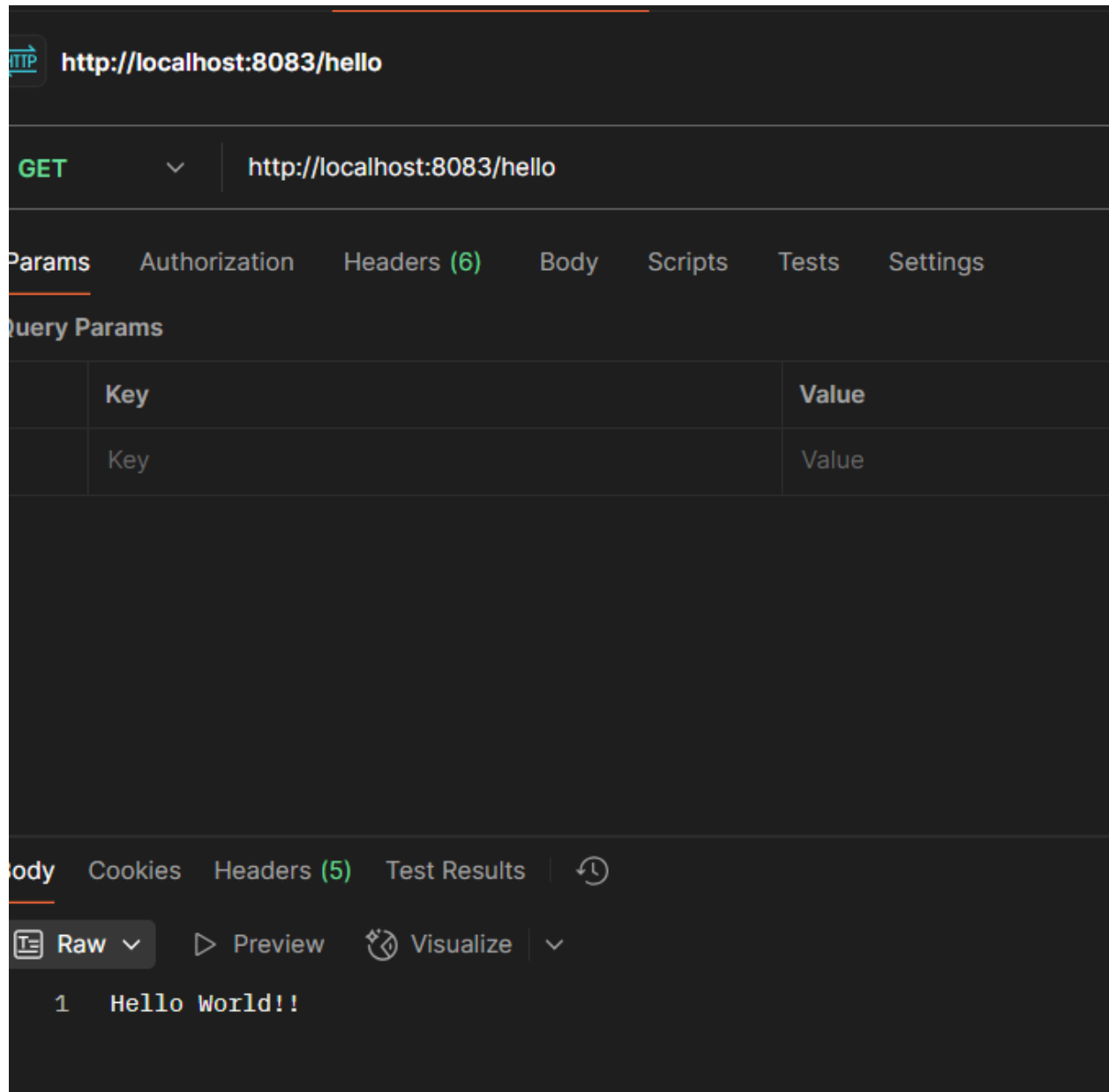
WEB BROWSER:

localhost:8083/hello

Hello World!!

Postman Console, Header, Network tabs:

Key	Value
Content-Type	text/plain;charset=UTF-8
Content-Length	13
Date	Sun, 13 Jul 2025 07:46:17 GMT
Keep-Alive	timeout=60
Connection	keep-alive

200 OK • 160 ms • 177 B • ⊕ | ∘∘∘

⊕ **Network**

HTTP Version	1.1
Local Address	::1
Remote Address	::1

2. REST - Country Web Service

Write a REST service that returns India country details in the earlier created spring learn application.

URL: /country
Controller: com.cognizant.spring-learn.controller.CountryController
Method Annotation: @RequestMapping
Method Name: getCountryIndia()
Method Implementation: Load India bean from spring xml configuration and return
Sample Request: http://localhost:8083/country
Sample 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

CODE:

```
package com.cognizant.springlearn.controller;
import com.cognizant.springlearn.country;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.ApplicationContext;
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);

    @Autowired
    private ApplicationContext context;

    @RequestMapping("/country")
    public country getCountryIndia() {
        LOGGER.info("START: getCountryIndia()");
        country country = (country) context.getBean("country");
        LOGGER.info("END: getCountryIndia()");
        return country;
    }
}
```

CONSOLE:

2025-07-13T14:33:54.715+05:30 INFO 24820 --- [spring-learn] [restartedMain] c.c.springlearn.SpringLearnApplication : Started SpringLearnApplication in 2.782 seconds (process runn
g for 3.33)
Spring Boot Application started!
2025-07-13T14:33:59.578+05:30 INFO 24820 --- [spring-learn] [nio-8084-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-07-13T14:33:59.579+05:30 INFO 24820 --- [spring-learn] [nio-8084-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-07-13T14:33:59.580+05:30 INFO 24820 --- [spring-learn] [nio-8084-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
2025-07-13T14:33:59.654+05:30 INFO 24820 --- [spring-learn] [nio-8084-exec-1] c.c.s.controller.countrycontroller : START: getCountryIndia()
2025-07-13T14:33:59.655+05:30 INFO 24820 --- [spring-learn] [nio-8084-exec-1] c.c.s.controller.countrycontroller : END: getCountryIndia()

localhost:8084/country

Gmail   YouTube   Maps   News

Pretty-print ☐

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