

WEEK 4_3 SOLUTIONS

Securing RESTful Web Services with Spring Security

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);
    }
}
```

SecurityConfig.class:

```
package com.cognizant.spring_learn.security;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.web.SecurityFilterChain;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.Customizer;

@Configuration
@EnableWebSecurity
public class SecurityConfig {

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http
            .authorizeHttpRequests(auth -> auth
                .anyRequest().authenticated()
            )
    }
}
```

```

        .httpBasic(Customizer.withDefaults()) // Enables Basic Auth
        .csrf(csrf -> csrf.disable());      // Disable CSRF for simplicity

    return http.build();
}
}

```

Output:

```

MINGW64/c/Users/shaf1
shaf1@LAPTOP-BESMIVGT MINGW64 ~ (master)
$ curl -s -v -u user:60cdb7a6-2024-4ce8-b5fe-6bf442c974a4 http://localhost:8030/countries
* Could not resolve host: 60cdb7a6-2024-4ce8-b5fe-6bf442c974a4
* Closing connection
* Trying [::1]:8030...
* Trying 127.0.0.1:8030...
* connect to ::1 port 8030 failed: Connection refused
* connect to 127.0.0.1 port 8030 failed: Connection refused
* Failed to connect to localhost port 8030 after 2269 ms: Couldn't connect to server
* Closing connection
shaf1@LAPTOP-BESMIVGT MINGW64 ~ (master)
$ AC
shaf1@LAPTOP-BESMIVGT MINGW64 ~ (master)
$ curl -s -v -u user:60cdb7a6-2024-4ce8-b5fe-6bf442c974a4 http://localhost:8083/countries
[{"code":"IN","name":"India"}, {"code":"DE","name":"Germany"}, {"code":"US","name":"United States"}, {"code":"JP","name":"Japan"}]* Trying [::1]:8083...
* Connected to localhost (::1) port 8083
* Server auth using Basic with user 'user'
> GET /countries HTTP/1.1
> Host: localhost:8083
> Authorization: Basic dXNlcjo2MGNKYjdhN10yMDI0LTRjZTgtYjVmZS02YmY0NDJjoTcyOTQ=
> User-Agent: curl/8.4.0
> Accept: */*
<
< HTTP/1.1 200
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 0
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
< X-Frame-Options: DENY
< Content-Type: application/json
< Transfer-Encoding: chunked
< Date: Fri, 11 Jul 2025 18:23:47 GMT
<
{
  [133 bytes data]
* Connection #0 to host localhost left intact
shaf1@LAPTOP-BESMIVGT MINGW64 ~ (master)
$ =

```

Hands on 2

REST - Country Web Service

SpringLearnApplication.java:

```
// Shaik Shafana Shihaam
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);
    }
}
```

CountryController.class

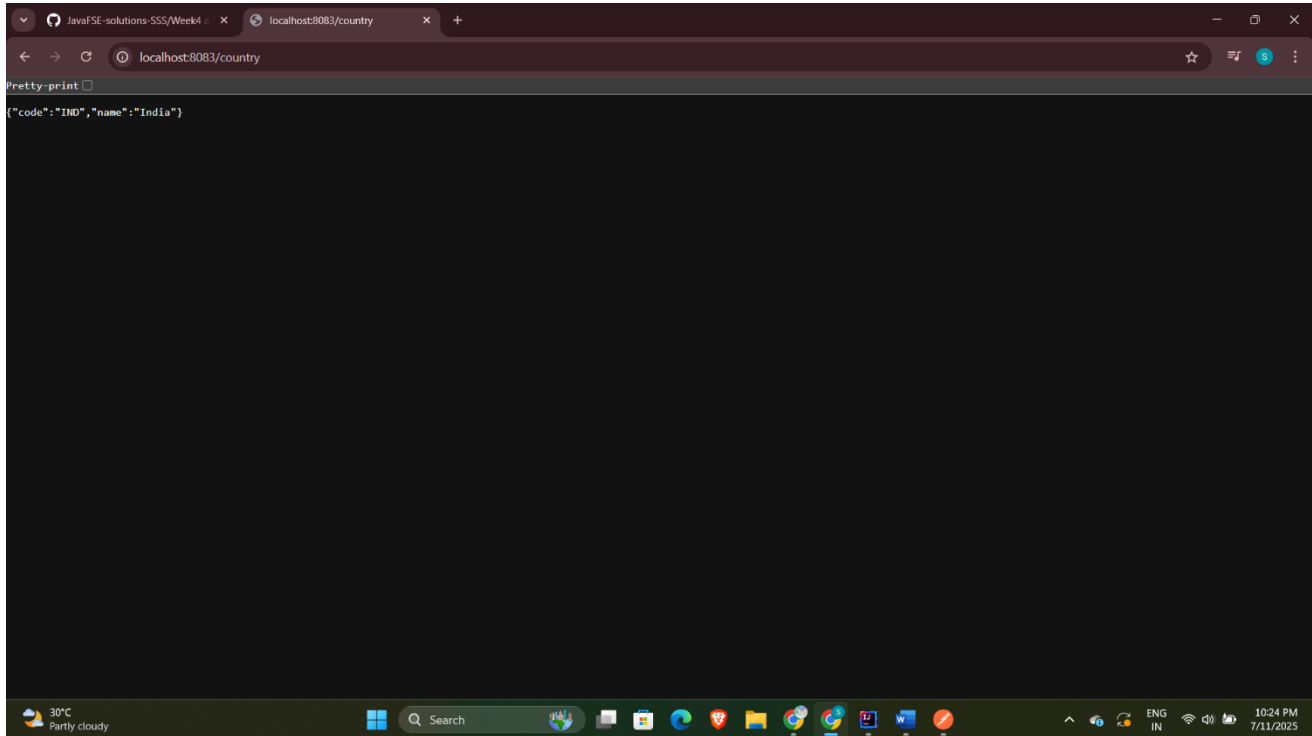
```
package com.cognizant.spring_learn.controller;

import com.cognizant.spring_learn.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.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

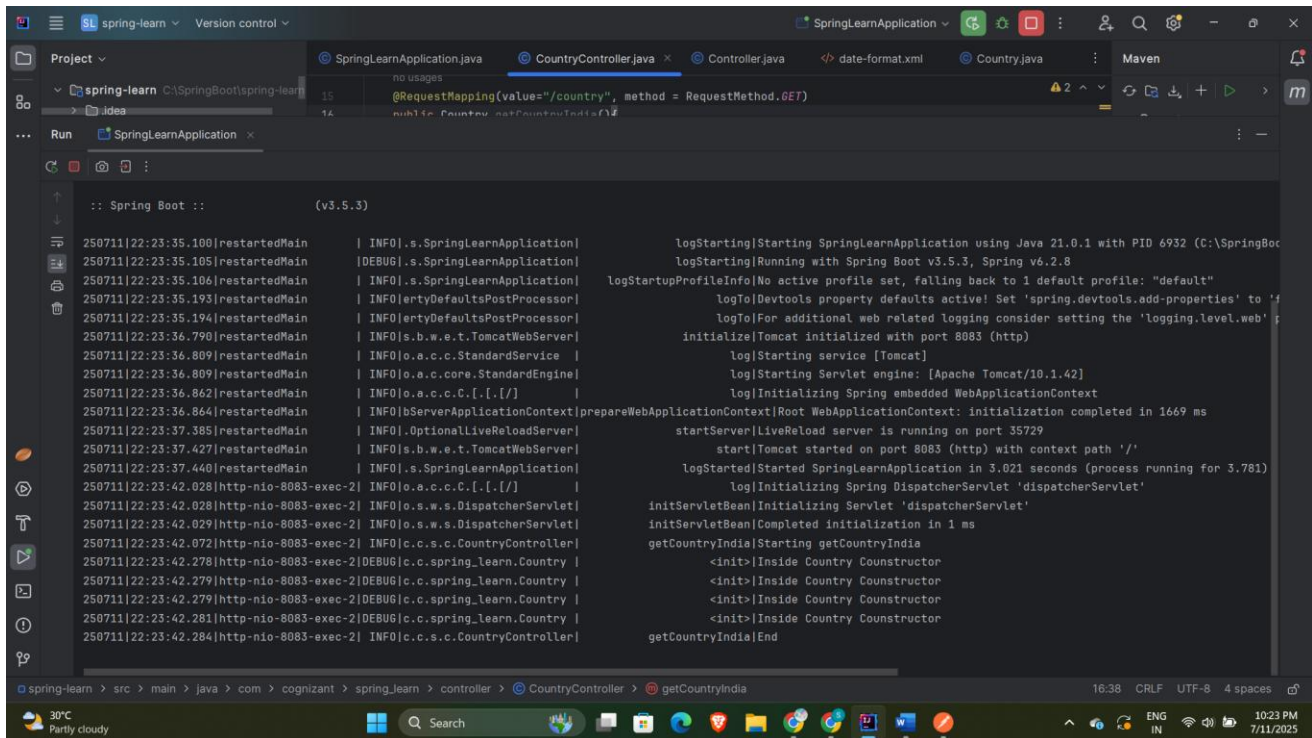
@RestController
public class CountryController {
    private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);
    @RequestMapping(value="/country", method = RequestMethod.GET)
    public Country getCountryIndia(){
        LOGGER.info("Starting getCountryIndia");
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
        Country country = context.getBean("in",Country.class);
        LOGGER.info("End");
        return country;
    }
}
```

```
}  
}
```

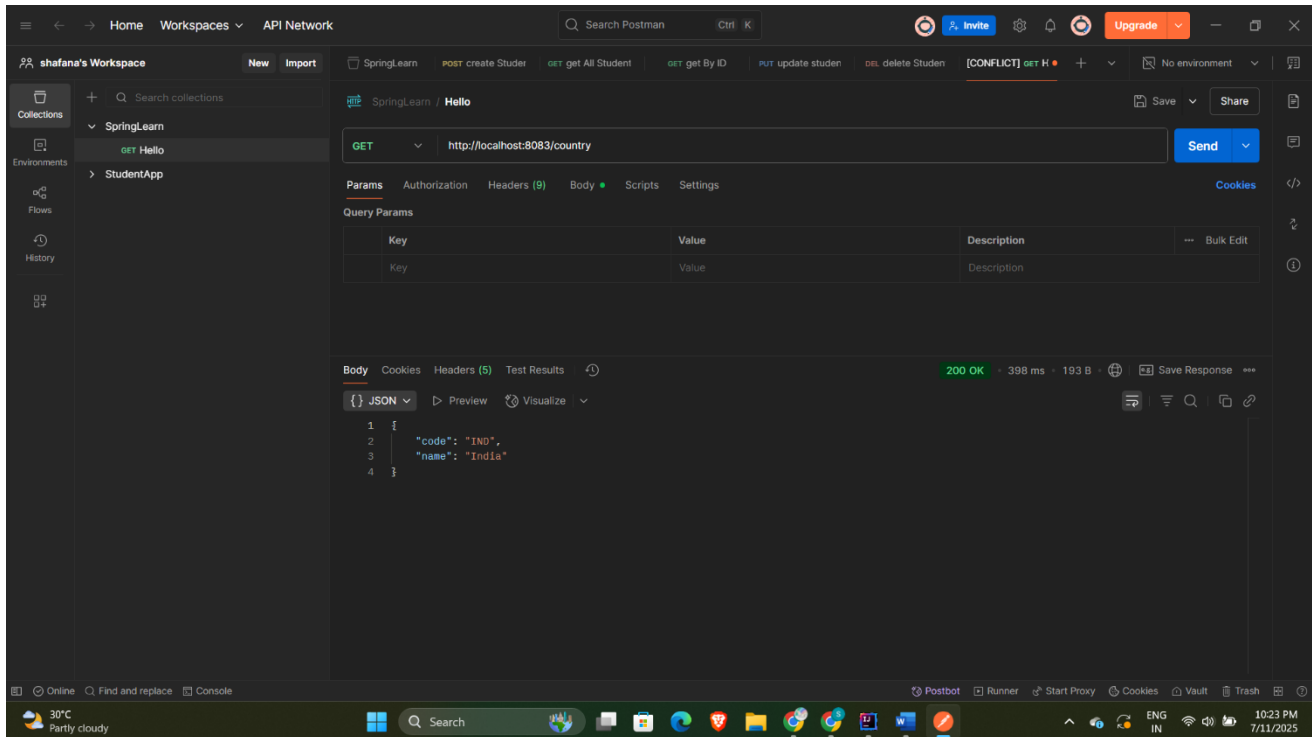
Output:



```
localhost:8083/country  
Pretty print  
{"code": "IND", "name": "India"}
```



```
Project  
spring-learn  
C:\SpringBoot\spring-learn  
Run  
SpringLearnApplication  
SpringLearnApplication.java  
CountryController.java  
Controller.java  
date-format.xml  
Country.java  
Maven  
@RequestMapping(value="/country", method = RequestMethod.GET)  
public Country getCountryIndia()  
:: Spring Boot :: (v3.5.3)  
250711|22:23:35.100|restartedMain | INFO |s.SpringLearnApplication| logStarting|Starting SpringLearnApplication using Java 21.0.1 with PID 6932 (C:\SpringBoc  
250711|22:23:35.105|restartedMain | DEBUG |s.SpringLearnApplication| logStarting|Running with Spring Boot v3.5.3, Spring v6.2.8  
250711|22:23:35.106|restartedMain | INFO |s.SpringLearnApplication| logStartupProfileInfo|No active profile set, falling back to 1 default profile: "default"  
250711|22:23:35.193|restartedMain | INFO |ertyDefaultsPostProcessor| logTo|Devtools property defaults active! Set 'spring.devtools.add-properties' to 'f  
250711|22:23:35.194|restartedMain | INFO |ertyDefaultsPostProcessor| logTo|For additional web related logging consider setting the 'logging.level.web' ;  
250711|22:23:36.790|restartedMain | INFO |s.b.w.e.t.TomcatWebServer| initialize|Tomcat initialized with port 8083 (http)  
250711|22:23:36.809|restartedMain | INFO |o.a.c.c.StandardService | log|Starting service [Tomcat]  
250711|22:23:36.809|restartedMain | INFO |o.a.c.c.core.StandardEngine| log|Starting Servlet engine: [Apache Tomcat/10.1.42]  
250711|22:23:36.862|restartedMain | INFO |o.a.c.c.C.[.[/]] | log|Initializing Spring embedded WebApplicationContext  
250711|22:23:36.864|restartedMain | INFO |bServerApplicationContext|prepareWebApplicationContext|Root WebApplicationContext: initialization completed in 1669 ms  
250711|22:23:37.385|restartedMain | INFO |OptionalLiveReloadServer| startServer|LiveReload server is running on port 35729  
250711|22:23:37.427|restartedMain | INFO |s.b.w.e.t.TomcatWebServer| start|Tomcat started on port 8083 (http) with context path '/'  
250711|22:23:37.440|restartedMain | INFO |s.SpringLearnApplication| logStarted|Started SpringLearnApplication in 3.021 seconds (process running for 3.781)  
250711|22:23:37.440|restartedMain | INFO |s.SpringLearnApplication| log|Initializing Spring DispatcherServlet 'dispatcherServlet'  
250711|22:23:42.028|http-nio-8083-exec-2| INFO |o.a.c.c.C.[.[/]] | initServletBean|Initializing Servlet 'dispatcherServlet'  
250711|22:23:42.029|http-nio-8083-exec-2| INFO |o.s.w.s.DispatcherServlet| initServletBean|Completed initialization in 1 ms  
250711|22:23:42.072|http-nio-8083-exec-2| INFO |c.c.s.c.CountryController| getCountryIndia|Starting getCountryIndia  
250711|22:23:42.278|http-nio-8083-exec-2| DEBUG |c.c.s.c.spring_learn.Country | <init>|Inside Country Counstructor  
250711|22:23:42.279|http-nio-8083-exec-2| DEBUG |c.c.s.c.spring_learn.Country | <init>|Inside Country Counstructor  
250711|22:23:42.279|http-nio-8083-exec-2| DEBUG |c.c.s.c.spring_learn.Country | <init>|Inside Country Counstructor  
250711|22:23:42.281|http-nio-8083-exec-2| DEBUG |c.c.s.c.spring_learn.Country | <init>|Inside Country Counstructor  
250711|22:23:42.284|http-nio-8083-exec-2| INFO |c.c.s.c.CountryController| getCountryIndia|End
```



Hands on 3

REST - Get all countries

CountryController.java

```
// Shaik Shafana Shihaam
```

```
package com.cognizant.spring_learn.controller;
```

```
import com.cognizant.spring_learn.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.GetMapping;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController
public class CountryController {
    private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);
    @RequestMapping(value="/country", method = RequestMethod.GET)
    public Country getCountryIndia(){
        LOGGER.info("Starting getCountryIndia");
        ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");
        Country country = context.getBean("in",Country.class);
        LOGGER.info("End");
        return country;
    }

    @GetMapping("/countries")
    public List<Country> getAllCountries(){
        LOGGER.info("starting getAllCountries..");
        ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");
        List<Country> countries = context.getBean("countryList",List.class);
        LOGGER.info("End//");
        return countries;
    }
}
```

Output :

The screenshot shows an IDE with the following components:

- Project View:** Shows the project structure with `spring-learn` and `SpringLearnApplication`.
- Run Console:** Displays logs for `SpringLearnApplication`. Key messages include:
 - `INFO|ertyDefaultsPostProcessor| logTo|Devtools property defaults active! Set 'spring.devtools.add-properties' to 'f`
 - `INFO|ertyDefaultsPostProcessor| logTo|For additional web related logging consider setting the 'logging.level.web' p`
 - `INFO|s.b.w.e.t.TomcatWebServer| initialize|Tomcat initialized with port 8083 (http)`
 - `INFO|o.a.c.c.StandardService| log|Starting service [Tomcat]`
 - `INFO|o.a.c.c.core.StandardEngine| log|Starting Servlet engine: [Apache Tomcat/10.1.42]`
 - `INFO|o.a.c.c.C.[./]| log|Initializing Spring embedded WebApplicationContext`
 - `INFO|bServerApplicationContext|prepareWebApplicationContext|Root WebApplicationContext: initialization completed in 1639 ms`
 - `INFO|OptionalLiveReloadServer| startServer|LiveReload server is running on port 35729`
 - `INFO|s.b.w.e.t.TomcatWebServer| start|Tomcat started on port 8083 (http) with context path '/'`
 - `logStarted|Started SpringLearnApplication in 2.994 seconds (process running for 3.62)`
 - `INFO|s.SpringLearnApplication| log|Initializing Spring DispatcherServlet 'dispatcherServlet'`
 - `INFO|o.a.c.c.C.[./]| initServletBean|Initializing Servlet 'dispatcherServlet'`
 - `INFO|o.s.w.s.DispatcherServlet| initServletBean|Completed initialization in 4 ms`
 - `INFO|c.c.s.c.CountryController| getAllCountries|starting getAllCountries..`
 - `DEBUG|c.c.spring_learn.Country | <init>|Inside Country Counstructor`
 - `DEBUG|c.c.spring_learn.Country | <init>|Inside Country Counstructor`
 - `DEBUG|c.c.spring_learn.Country | <init>|Inside Country Counstructor`
 - `DEBUG|c.c.spring_learn.Country | <init>|Inside Country Counstructor`
 - `INFO|c.c.s.c.CountryController| getAllCountries|End//`
 - `INFO|c.c.s.c.CountryController| getAllCountries|starting getAllCountries..`
 - `DEBUG|c.c.spring_learn.Country | <init>|Inside Country Counstructor`
 - `DEBUG|c.c.spring_learn.Country | <init>|Inside Country Counstructor`
 - `DEBUG|c.c.spring_learn.Country | <init>|Inside Country Counstructor`
 - `DEBUG|c.c.spring_learn.Country | <init>|Inside Country Counstructor`
 - `INFO|c.c.s.c.CountryController| getAllCountries|End//`

The screenshot shows a web browser with the following components:

- Address Bar:** `localhost:8083/countries`
- Page Title:** `Pretty-print`
- Content:** A JSON array of country objects:

```
[{"code": "IND", "name": "India"}, {"code": "US", "name": "United States"}, {"code": "DE", "name": "Germany"}, {"code": "DE", "name": "Germany"}]
```

HomeWorkspacesAPI Network

Search PostmanCtrl K

Invite

Upgrade

shafana's Workspace

NewImport

SpringLearn

GET Hello

StudentApp

SpringLearnHello

GEThttp://localhost:8083/countries

Send

ParamsAuthorizationHeaders (9)BodyScriptsSettings

Cookies

Query Params

Key	Value	Description
-----	-------	-------------

BodyCookiesHeaders (5)Test Results

200 OK · 568 ms · 294 B · Save Response

JSONPreviewVisualize

```
1 [
2   {
3     "code": "IND",
4     "name": "India"
5   },
6   {
7     "code": "US",
8     "name": "United States"
9   },
10  {
11    "code": "DE",
12    "name": "Germany"
13  },
14  {
15    "code": "DE",
16    "name": "Germany"
17  }
18 ]
```

OnlineFind and replaceConsole

PostbotRunnerStart ProxyCookiesVaultTrash

30°CPartly cloudySearch

ENG IN10:35 PM7/11/2025

REST - Get country based on country code

CountryController.class:

```
package com.cognizant.spring_learn.controller;

import com.cognizant.spring_learn.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.*;

import java.util.List;

@RestController
public class CountryController {
    private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);
    @RequestMapping(value="/country", method = RequestMethod.GET)
    public Country getCountryIndia(){
        LOGGER.info("Starting getCountryIndia");
        ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");
        Country country = context.getBean("in",Country.class);
        LOGGER.info("End");
        return country;
    }

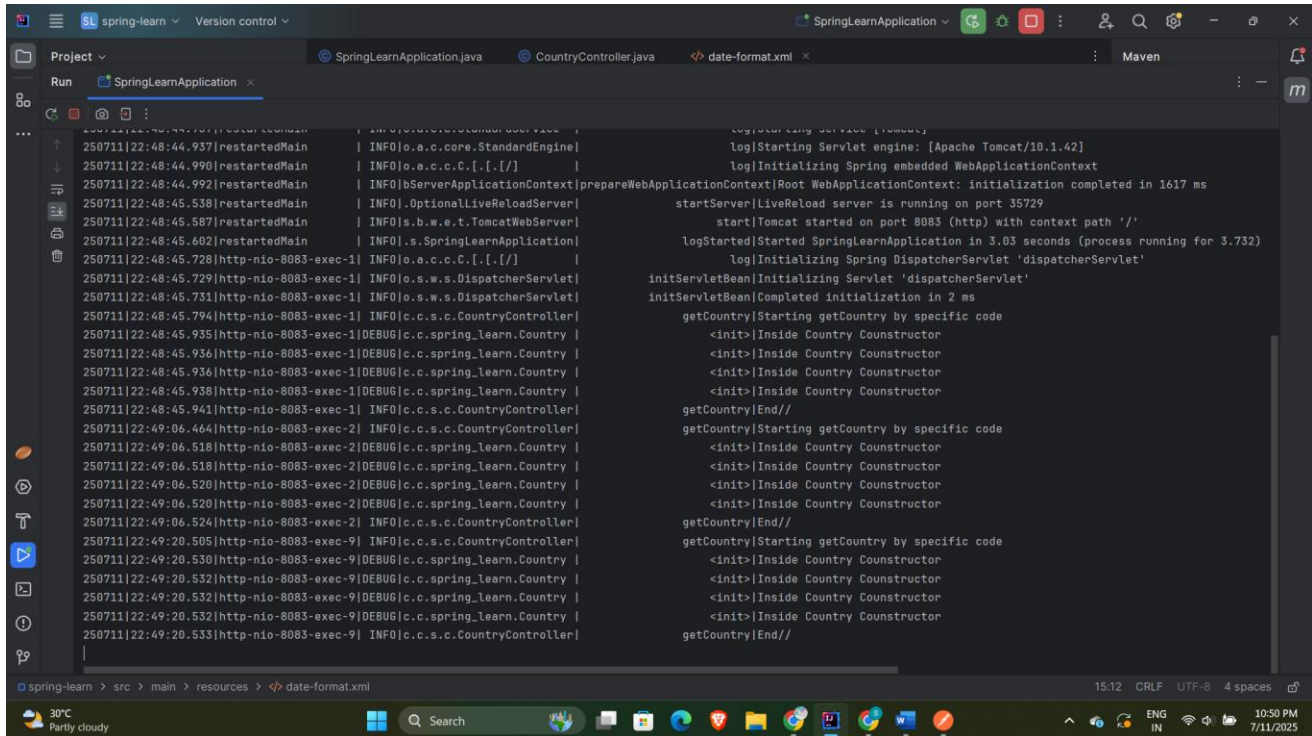
    @GetMapping("/countries/{code}")
    public Country getCountry(@PathVariable String code){
        LOGGER.info("Starting getCountry by specific code");
        ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");
        Country country = context.getBean(code,Country.class);
        LOGGER.info("End//");
        return country;
    }
    @GetMapping("/countries")
```

```

public List<Country> getAllCountries(){
    LOGGER.info("starting getAllCountries..");
    ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");
    List<Country> countries = context.getBean("countryList",List.class);
    LOGGER.info("End//");
    return countries;
}
}

```

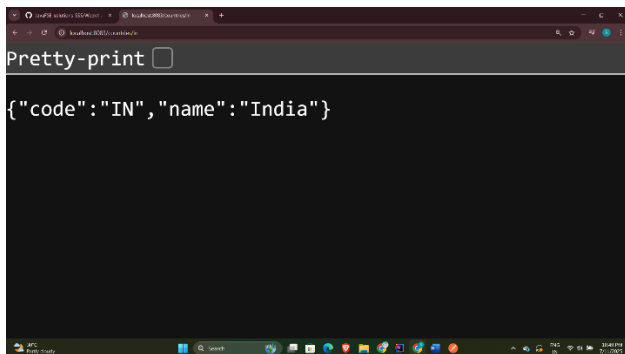
Output :



```

250711|22:48:44.937|restartedMain|INFO|o.a.c.c.o.StandardEngine|logStarting Servlet engine: [Apache Tomcat/10.1.42]
250711|22:48:44.990|restartedMain|INFO|o.a.c.c.C.[.][.]|log|Initializing Spring embedded WebApplicationContext
250711|22:48:45.538|restartedMain|INFO|b.ServerApplicationContext|prepareWebApplicationContext|Root WebApplicationContext: initialization completed in 1617 ms
250711|22:48:45.538|restartedMain|INFO|.OptionalLiveReloadServer|startServer|LiveReload server is running on port 35729
250711|22:48:45.587|restartedMain|INFO|.s.w.e.t.TomcatWebServer|start|Tomcat started on port 8083 (http) with context path '/'
250711|22:48:45.602|restartedMain|INFO|.SpringLearnApplication|logStarted|Started SpringLearnApplication in 3.03 seconds (process running for 3.732)
250711|22:48:45.728|http-nio-8083-exec-1|INFO|o.a.c.c.C.[.][.]|log|Initializing Spring DispatcherServlet 'dispatcherServlet'
250711|22:48:45.729|http-nio-8083-exec-1|INFO|.s.w.s.DispatcherServlet|initServletBean|Initializing Servlet 'dispatcherServlet'
250711|22:48:45.731|http-nio-8083-exec-1|INFO|.s.w.s.DispatcherServlet|initServletBean|Completed initialization in 2 ms
250711|22:48:45.794|http-nio-8083-exec-1|INFO|.c.s.c.CountryController|getCountry|Starting getCountry by specific code
250711|22:48:45.935|http-nio-8083-exec-1|DEBUG|.c.s.c.spring_learn.Country|<init>|Inside Country Counstructor
250711|22:48:45.936|http-nio-8083-exec-1|DEBUG|.c.s.c.spring_learn.Country|<init>|Inside Country Counstructor
250711|22:48:45.936|http-nio-8083-exec-1|DEBUG|.c.s.c.spring_learn.Country|<init>|Inside Country Counstructor
250711|22:48:45.938|http-nio-8083-exec-1|DEBUG|.c.s.c.spring_learn.Country|<init>|Inside Country Counstructor
250711|22:48:45.941|http-nio-8083-exec-1|INFO|.c.s.c.CountryController|getCountry|End//
250711|22:49:06.464|http-nio-8083-exec-2|INFO|.c.s.c.CountryController|getCountry|Starting getCountry by specific code
250711|22:49:06.518|http-nio-8083-exec-2|DEBUG|.c.s.c.spring_learn.Country|<init>|Inside Country Counstructor
250711|22:49:06.518|http-nio-8083-exec-2|DEBUG|.c.s.c.spring_learn.Country|<init>|Inside Country Counstructor
250711|22:49:06.520|http-nio-8083-exec-2|DEBUG|.c.s.c.spring_learn.Country|<init>|Inside Country Counstructor
250711|22:49:06.520|http-nio-8083-exec-2|DEBUG|.c.s.c.spring_learn.Country|<init>|Inside Country Counstructor
250711|22:49:06.524|http-nio-8083-exec-2|INFO|.c.s.c.CountryController|getCountry|End//
250711|22:49:20.505|http-nio-8083-exec-9|INFO|.c.s.c.CountryController|getCountry|Starting getCountry by specific code
250711|22:49:20.530|http-nio-8083-exec-9|DEBUG|.c.s.c.spring_learn.Country|<init>|Inside Country Counstructor
250711|22:49:20.532|http-nio-8083-exec-9|DEBUG|.c.s.c.spring_learn.Country|<init>|Inside Country Counstructor
250711|22:49:20.532|http-nio-8083-exec-9|DEBUG|.c.s.c.spring_learn.Country|<init>|Inside Country Counstructor
250711|22:49:20.532|http-nio-8083-exec-9|DEBUG|.c.s.c.spring_learn.Country|<init>|Inside Country Counstructor
250711|22:49:20.533|http-nio-8083-exec-9|INFO|.c.s.c.CountryController|getCountry|End//

```



```

Pretty-print ☐
{"code": "IN", "name": "India"}

```

