

Hello World RESTful Web Service:

SpringLearnApplication.java:

```
package com.cognizant.spring_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

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

    public static void main(String[] args) {

        LOGGER.info("START");

        SpringApplication.run(SpringLearnApplication.class, args);

        LOGGER.info("END");}

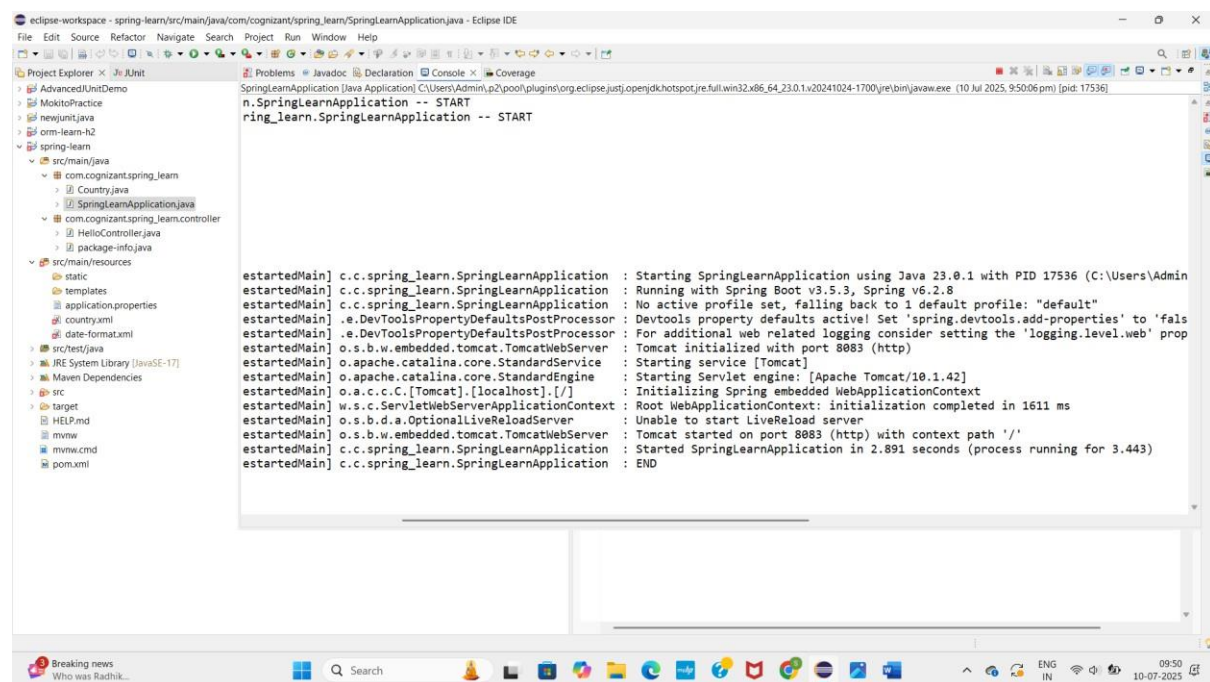
}
```

application.properties:

server.port=8083

logging.level.com.cognizant.spring_learn=debug

OUTPUT:



REST - Country Web Service

SpringLearnApplication.java:

```
package com.cognizant.spring_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

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

    public static void main(String[] args) {

        LOGGER.info("START");

        SpringApplication.run(SpringLearnApplication.class, args);

        LOGGER.info("END");

    }

}
```

CountryController.java:

```
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.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 = context.getBean("country", Country.class);

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

        LOGGER.info("END");

        return country;
    }
}

```

Country.java:

```

package com.cognizant.spring_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class Country {

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

    private String code;

    private String name;

    public Country() {

        LOGGER.debug("Inside Country Constructor");

    }

    public String getCode() {

        LOGGER.debug("Getting Code: {}", code);

        return code;

    }

    public void setCode(String code) {

        LOGGER.debug("Setting Code: {}", code);

        this.code = code;

    }

    public String getName() {

        LOGGER.debug("Getting Name: {}", name);

        return name;

    }

    public void setName(String name) {

```

```

    LOGGER.debug("Setting Name: {}", name);

    this.name = name;

}

@Override

public String toString() {

    return "Country [code=" + code + ", name=" + name + "];" ;
}

```

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

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

    <bean id="country" class="com.cognizant.spring_learn.Country">

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

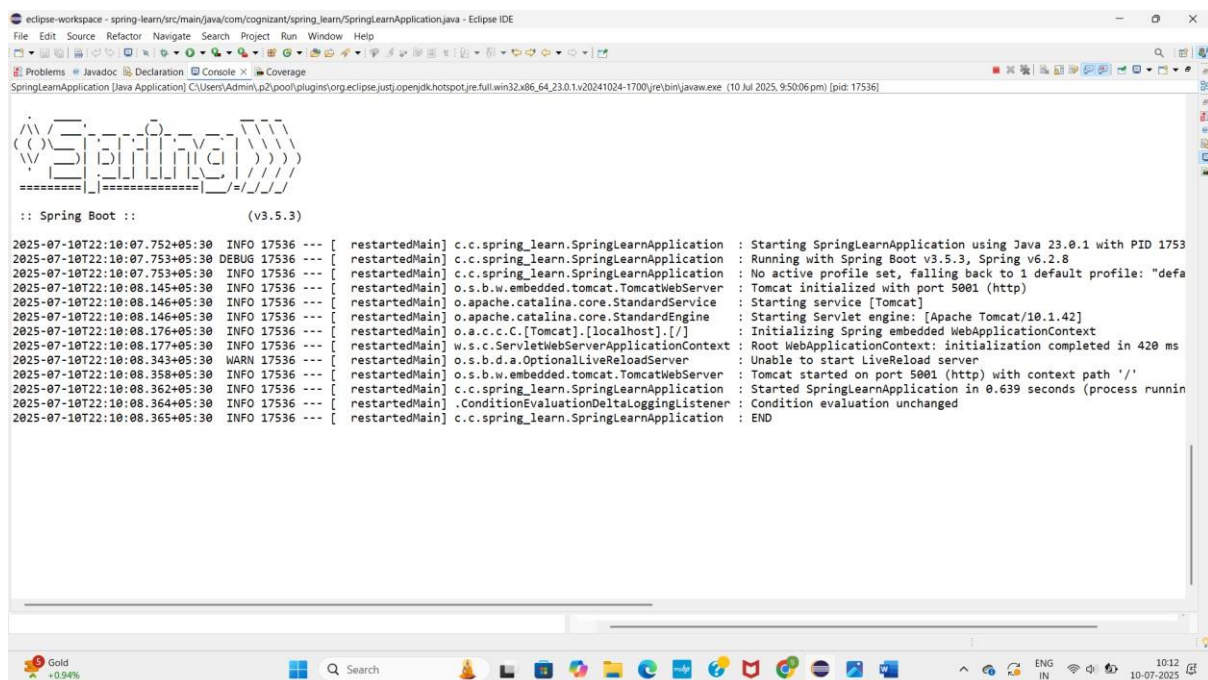
        <property name="name" value="India"/>

    </bean>

</beans>

```

OUTPUT:



The screenshot shows the Eclipse IDE interface with the console window open. The console displays the following logs:

```

:: Spring Boot ::                (v3.5.3)

2025-07-10T22:10:07.752+05:30 INFO 17536 --- [ restartedMain] c.c.spring_learn.SpringLearnApplication : Starting SpringLearnApplication using Java 23.0.1 with PID 1753
2025-07-10T22:10:07.753+05:30 DEBUG 17536 --- [ restartedMain] c.c.spring_learn.SpringLearnApplication : Running with Spring Boot v3.5.3, Spring v6.2.8
2025-07-10T22:10:08.145+05:30 INFO 17536 --- [ restartedMain] c.c.spring_learn.SpringLearnApplication : No active profile set, falling back to 1 default profile: "defa
2025-07-10T22:10:08.146+05:30 INFO 17536 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 5001 (http)
2025-07-10T22:10:08.146+05:30 INFO 17536 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-07-10T22:10:08.146+05:30 INFO 17536 --- [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.42]
2025-07-10T22:10:08.176+05:30 INFO 17536 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-07-10T22:10:08.177+05:30 INFO 17536 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 420 ms
2025-07-10T22:10:08.343+05:30 WARN 17536 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : Unable to start LiveReload server
2025-07-10T22:10:08.358+05:30 INFO 17536 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 5001 (http) with context path '/'
2025-07-10T22:10:08.362+05:30 INFO 17536 --- [ restartedMain] c.c.spring_learn.SpringLearnApplication : Started SpringLearnApplication in 0.639 seconds (process runnin
2025-07-10T22:10:08.364+05:30 INFO 17536 --- [ restartedMain] .ConditionEvaluationDeltaLoggingListener : Condition evaluation unchanged
2025-07-10T22:10:08.365+05:30 INFO 17536 --- [ restartedMain] c.c.spring_learn.SpringLearnApplication : END

```

REST - Get All Countries:

CountryNotFoundException.java:

```

package com.cognizant.spring_learn.service.exception;

import org.springframework.http.HttpStatus;

```

```
import org.springframework.web.bind.annotation.ResponseStatus;
@ResponseStatus(value = HttpStatus.NOT_FOUND, reason = "Country not found")
public class CountryNotFoundException extends Exception {
    public CountryNotFoundException(String message) {
        super(message);
    }
}
```

CountryService.java:

```
package com.cognizant.spring_learn.service;

import com.cognizant.spring_learn.Country;
import com.cognizant.spring_learn.service.exception.CountryNotFoundException;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class CountryService {
    private static final Logger LOGGER = LoggerFactory.getLogger(CountryService.class);
    public Country getCountry(String code) throws CountryNotFoundException {
        LOGGER.info("START - getCountry()");
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
        List<Country> countries = (List<Country>) context.getBean("countryList");
        for (Country country : countries) {
            if (country.getCode().equalsIgnoreCase(code)) {
                LOGGER.debug("Found Country: {}", country);
                LOGGER.info("END - getCountry()");
                return country;
            }
        }
        LOGGER.error("Country Not Found: {}", code);
        throw new CountryNotFoundException("Country not found");
    }
}
```

CountryController.java:

```
package com.cognizant.spring_learn.controller;
import com.cognizant.spring_learn.Country;
import com.cognizant.spring_learn.service.CountryService;
import com.cognizant.spring_learn.service.exception.CountryNotFoundException;
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("/countries/{code}")
    public Country getCountry(@PathVariable String code) throws CountryNotFoundException {
```

```

        LOGGER.info("START - getCountry()");
        Country country = countryService.getCountry(code);
        LOGGER.debug("Country Returned: {}", country);
        LOGGER.info("END - getCountry()");
        return country;
    }
}

```

CountryNotFoundException.java:

```

package com.cognizant.spring_learn.service.exception;
public class CountryNotFoundException extends Exception {
    public CountryNotFoundException(String message) {
        super(message);
    }
}

```

OUTPUT:



REST – Add New Country using POST /country:

CountryController.java:

```

package com.cognizant.spring_learn.controller;
import com.cognizant.spring_learn.Country;
import com.cognizant.spring_learn.service.CountryService;
import com.cognizant.spring_learn.service.exception.CountryNotFoundException;
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("/countries/{code}")
    public Country getCountry(@PathVariable String code) throws CountryNotFoundException {
        LOGGER.info("START - getCountry()");
        Country country = countryService.getCountry(code);
    }
}

```

```

        LOGGER.debug("Country Returned: {}", country);
        LOGGER.info("END - getCountry()");
        return country;
    }
    @PostMapping("/country")
    public Country addCountry(@RequestBody Country country) {
        LOGGER.info("START - addCountry()");
        LOGGER.debug("Country Received: {}", country);
        LOGGER.info("END - addCountry()");
        return country;
    }
}

```

Country.java:

```

package com.cognizant.spring_learn;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
public class Country {
    private static final Logger LOGGER = LoggerFactory.getLogger(Country.class);
    private String code;
    private String name;
    public Country() {
        LOGGER.debug("Inside Country Constructor");
    }
    public String getCode() {
        LOGGER.debug("Getting Code: {}", code);
        return code;
    }
    public void setCode(String code) {
        LOGGER.debug("Setting Code: {}", code);
        this.code = code;
    }
    public String getName() {
        LOGGER.debug("Getting Name: {}", name);
        return name;
    }
    public void setName(String name) {
        LOGGER.debug("Setting Name: {}", name);
        this.name = name;
    }
    @Override
    public String toString() {
        return "Country{" +
            "code='" + code + '\'' +
            ", name='" + name + '\'' +
            '}';
    }
}

```

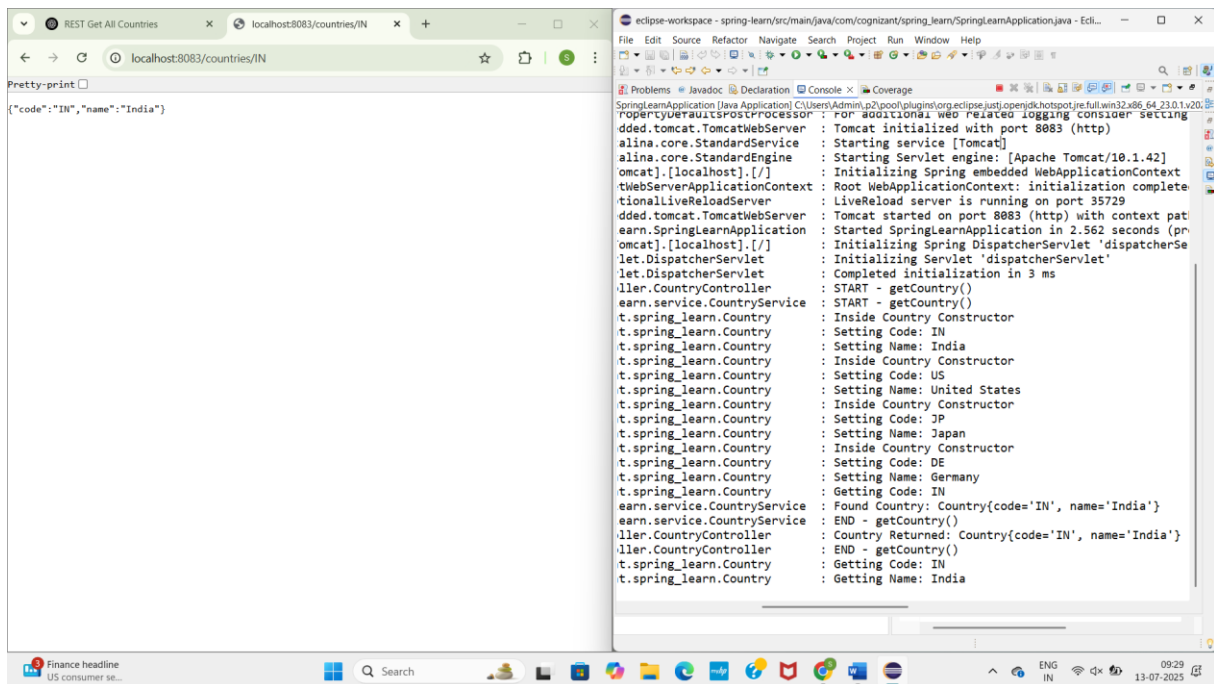
CountryNotFoundException.java:

```

package com.cognizant.spring_learn.service.exception;
public class CountryNotFoundException extends Exception {
    public CountryNotFoundException(String message) {
        super(message);
    }
}

```

OUTPUT:



Add New Country & Save in List:

CountryService.java:

```
package com.cognizant.spring_learn.service;
import com.cognizant.spring_learn.Country;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Service;
```

```
import java.util.ArrayList;
import java.util.List;
@Service
public class CountryService {
    private static final Logger LOGGER = LoggerFactory.getLogger(CountryService.class);
    private List<Country> countryList = new ArrayList<>();
    public Country addCountry(Country country) {
        LOGGER.info("START - addCountry()");
        countryList.add(country);
        LOGGER.debug("Country added: {}", country);
        LOGGER.info("END - addCountry()");
        return country;
    }
    public List<Country> getAllCountries() {
        return countryList;
    }
}
```

CountryController.java:

```
package com.cognizant.spring_learn.controller;
import com.cognizant.spring_learn.Country;
import com.cognizant.spring_learn.service.CountryService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
```



```

public class CountryController {
    private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);
    private CountryService countryService;
    @PostMapping("/country")
    public Country addCountry(@RequestBody Country country) {
        LOGGER.info("START - /country POST");
        Country saved = countryService.addCountry(country);
        LOGGER.info("END - /country POST");
        return saved;
    }
}

```

OUTPUT:

```

2025-07-13T09:54:42.956+05:30 INFO 21644 --- [ restartedMain] c.c.spring_learn.SpringLearnApplication : Starting SpringLearnApplication using Java 23.0.1 with PID
2025-07-13T09:54:42.960+05:30 DEBUG 21644 --- [ restartedMain] c.c.spring_learn.SpringLearnApplication : Running with Spring Boot v3.5.3, Spring v6.2.8
2025-07-13T09:54:42.962+05:30 INFO 21644 --- [ restartedMain] c.c.spring_learn.SpringLearnApplication : No active profile set, falling back to 1 default profile:
2025-07-13T09:54:43.057+05:30 INFO 21644 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.ad
2025-07-13T09:54:44.324+05:30 INFO 21644 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8085 (http)
2025-07-13T09:54:44.338+05:30 INFO 21644 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-07-13T09:54:44.338+05:30 INFO 21644 --- [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.42]
2025-07-13T09:54:44.382+05:30 INFO 21644 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Initializing Spring embedded WebApplicationContext
2025-07-13T09:54:44.828+05:30 INFO 21644 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2025-07-13T09:54:44.876+05:30 INFO 21644 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8085 (http) with context path '/'
2025-07-13T09:54:44.888+05:30 INFO 21644 --- [ restartedMain] c.c.spring_learn.SpringLearnApplication : Started SpringLearnApplication in 2.369 seconds (process r
2025-07-13T10:02:17.421+05:30 INFO 21644 --- [nio-8085-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-07-13T10:02:17.422+05:30 INFO 21644 --- [nio-8085-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-07-13T10:02:17.430+05:30 INFO 21644 --- [nio-8085-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 6 ms
2025-07-13T10:02:53.947+05:30 DEBUG 21644 --- [nio-8085-exec-3] com.cognizant.spring_learn.Country : Inside Country Constructor
2025-07-13T10:02:53.947+05:30 DEBUG 21644 --- [nio-8085-exec-3] com.cognizant.spring_learn.Country : Setting Code: FR
2025-07-13T10:02:53.949+05:30 DEBUG 21644 --- [nio-8085-exec-3] com.cognizant.spring_learn.Country : Setting Name: France
2025-07-13T10:02:53.961+05:30 INFO 21644 --- [nio-8085-exec-3] c.c.s.controller.CountryController : START - /country POST
2025-07-13T10:02:53.961+05:30 INFO 21644 --- [nio-8085-exec-3] c.c.spring_learn.service.CountryService : START - addCountry()
2025-07-13T10:02:53.961+05:30 DEBUG 21644 --- [nio-8085-exec-3] c.c.spring_learn.service.CountryService : Country added: Country{code='FR', name='France'}
2025-07-13T10:02:53.962+05:30 INFO 21644 --- [nio-8085-exec-3] c.c.spring_learn.service.CountryService : END - addCountry()
2025-07-13T10:02:53.963+05:30 INFO 21644 --- [nio-8085-exec-3] c.c.s.controller.CountryController : END - /country POST
2025-07-13T10:02:53.973+05:30 DEBUG 21644 --- [nio-8085-exec-3] com.cognizant.spring_learn.Country : Getting Code: FR
2025-07-13T10:02:53.973+05:30 DEBUG 21644 --- [nio-8085-exec-3] com.cognizant.spring_learn.Country : Getting Name: France
2025-07-13T10:05:04.481+05:30 DEBUG 21644 --- [nio-8085-exec-7] com.cognizant.spring_learn.Country : Inside Country Constructor
2025-07-13T10:05:04.482+05:30 DEBUG 21644 --- [nio-8085-exec-7] com.cognizant.spring_learn.Country : Setting Code: FR
2025-07-13T10:05:04.482+05:30 DEBUG 21644 --- [nio-8085-exec-7] com.cognizant.spring_learn.Country : Setting Name: France
2025-07-13T10:05:04.483+05:30 INFO 21644 --- [nio-8085-exec-7] c.c.s.controller.CountryController : START - /country POST
2025-07-13T10:05:04.483+05:30 INFO 21644 --- [nio-8085-exec-7] c.c.spring_learn.service.CountryService : START - addCountry()
2025-07-13T10:05:04.483+05:30 DEBUG 21644 --- [nio-8085-exec-7] c.c.spring_learn.service.CountryService : Country added: Country{code='FR', name='France'}
2025-07-13T10:05:04.483+05:30 INFO 21644 --- [nio-8085-exec-7] c.c.spring_learn.service.CountryService : END - addCountry()
2025-07-13T10:05:04.484+05:30 INFO 21644 --- [nio-8085-exec-7] c.c.s.controller.CountryController : END - /country POST
2025-07-13T10:05:04.485+05:30 DEBUG 21644 --- [nio-8085-exec-7] com.cognizant.spring_learn.Country : Getting Code: FR
2025-07-13T10:05:04.485+05:30 DEBUG 21644 --- [nio-8085-exec-7] com.cognizant.spring_learn.Country : Getting Name: France

```

REST - Get country:

CountryNotFoundException.java:

```

package com.cognizant.spring_learn.service.exception;
public class CountryNotFoundException extends Exception {
    public CountryNotFoundException(String message) {
        super(message);
    }
}

```

CountryService.java:

```

package com.cognizant.spring_learn.service;
import com.cognizant.spring_learn.Country;
import com.cognizant.spring_learn.service.exception.CountryNotFoundException;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Service;
import java.util.ArrayList;
import java.util.List;
@Service
public class CountryService {
    private static final Logger LOGGER = LoggerFactory.getLogger(CountryService.class);
    private List<Country> countryList = new ArrayList<>();
    public CountryService() {
        Country india = new Country();

```

```

        india.setCode("IN");
        india.setName("India");

        Country usa = new Country();
        usa.setCode("US");
        usa.setName("United States");
        countryList.add(india);
        countryList.add(usa);
    }

    public Country getCountry(String code) throws CountryNotFoundException {
        LOGGER.info("START - getCountry()");
        for (Country country : countryList) {
            if (country.getCode().equalsIgnoreCase(code)) {
                LOGGER.info("END - getCountry() → Found");
                return country;
            }
        }
        LOGGER.warn("Country with code {} not found", code);
        throw new CountryNotFoundException("Country not found");
    }

    public Country addCountry(Country country) {
        LOGGER.info("Adding country {}", country);
        countryList.add(country);
        return country;
    }

    public List<Country> getAllCountries() {
        return countryList;
    }
}

CountryController.java:
package com.cognizant.spring_learn.controller;
import com.cognizant.spring_learn.Country;
import com.cognizant.spring_learn.service.CountryService;
import com.cognizant.spring_learn.service.exception.CountryNotFoundException;
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("/countries/{code}")
    public Country getCountry(@PathVariable String code) throws CountryNotFoundException {
        LOGGER.info("START - /countries/{code}", code);
        Country country = countryService.getCountry(code);
        LOGGER.info("END - /countries/{code}", code);
        return country;
    }

    @PostMapping("/country")
    public Country addCountry(@RequestBody Country country) {
        return countryService.addCountry(country);
    }
}

```

```

    @GetMapping("/countries")
    public List<Country> getAllCountries() {
        return countryService.getAllCountries();
    }
}

```

GlobalExceptionHandler.java:

```

package com.cognizant.spring_learn.exception;
import com.cognizant.spring_learn.service.exception.CountryNotFoundException;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
@RestControllerAdvice
public class GlobalExceptionHandler {
    @ExceptionHandler(CountryNotFoundException.class)
    public ResponseEntity<String> handleCountryNotFound(CountryNotFoundException ex) {
        return new ResponseEntity<>("Country not found", HttpStatus.NOT_FOUND);
    }
}

```

CountryNotFoundException:

```

package com.cognizant.spring_learn.service.exception;
public class CountryNotFoundException extends Exception {
    public CountryNotFoundException(String message) {
        super(message);
    }
}

```

OUTPUT:

The screenshot displays the application's output in two windows. On the left, a web browser at `localhost:8085/countries/IN` shows a JSON response: `{"code": "IN", "name": "India"}`. On the right, the Eclipse IDE's console window shows the application's startup logs, including the initialization of Spring Boot, Tomcat, and the execution of the `getCountry()` endpoint for the code 'IN'.

```

SpringLearnApplication [Java Application] C:\Users\Admin\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.23.0.1\jre\bin\java.exe
nApplication : Starting SpringLearnApplication using Java 23.0.1 with PID 3048
nApplication : Running with Spring Boot v3.5.3, Spring v6.2.8
nApplication : No active profile set, falling back to 1 default profile: "default"
PostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'true' to enable the Spring Devtools property defaults.
PostProcessor : For additional web related logging consider setting the 'logging.level.org.springframework.web' property.
catWebServer : Tomcat initialized with port 8085 (http)
dardService : Starting service [Tomcat]
dardEngine : Starting Servlet engine: [Apache Tomcat/10.1.42]
st].[/] : Initializing Spring embedded WebApplicationContext
cationContext : Root WebApplicationContext: initialization completed in 1696 ms
Country : Inside Country Constructor
Country : Setting Code: IN
Country : Setting Name: India
Country : Inside Country Constructor
Country : Setting Code: US
Country : Setting Name: United States
dServer : LiveReload server is running on port 35729
catWebServer : Tomcat started on port 8085 (http) with context path '/'
nApplication : Started SpringLearnApplication in 3.095 seconds (process running)
st].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
ervlet : Initializing Servlet 'dispatcherServlet'
ervlet : Completed initialization in 2 ms
troller : START - /countries/IN
untryService : START - getCountry()
Country : Getting Code: IN
untryService : END - getCountry() ? Found
troller : END - /countries/IN
Country : Getting Code: IN
Country : Getting Name: India

```

