**SPRINGREST4**

**PROJECT STRUCTURE:**

springrest4/

├── mvnw

├── mvnw.cmd

├── pom.xml

├── src

│ ├── main

│ │ ├── java

│ │ │ └── com

│ │ │ └── cognizant

│ │ │ └── springlearn

│ │ │ ├── Springrest4Application.java

│ │ │ ├── controller

│ │ │ │ └── CountryController.java

│ │ │ ├── exception

│ │ │ │ └── GlobalExceptionHandler.java

│ │ │ └── model

│ │ │ └── Country.java

│ │ └── resources

│ │ ├── application.properties

│ │ └── static/

│ │ └── templates/

└── └── test

└── java

└── com

└── cognizant

└── springlearn

└── Springrest4ApplicationTests.java

**CountryController.java**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.model.Country;

import jakarta.validation.Valid;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import java.util.ArrayList;

import java.util.List;

@RestController

@RequestMapping("/countries")

public class CountryController {

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

    private List<Country> countryList = new ArrayList<>();

    @GetMapping

    public List<Country> getAllCountries() {

        LOGGER.info("Start getAllCountries");

        return countryList;

    }

    @PostMapping

    public Country addCountry(@RequestBody @Valid Country country) {

        LOGGER.info("Start addCountry");

        countryList.add(country);

        return country;

    }

}

**GlobalExceptionHandler.java**

package com.cognizant.springlearn.exception;

import com.fasterxml.jackson.databind.exc.InvalidFormatException;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.http.HttpHeaders;

import org.springframework.http.HttpStatusCode;

import org.springframework.http.ResponseEntity;

import org.springframework.http.converter.HttpMessageNotReadableException;

import org.springframework.web.bind.MethodArgumentNotValidException;

import org.springframework.web.bind.annotation.ControllerAdvice;

import org.springframework.web.context.request.WebRequest;

import org.springframework.web.servlet.mvc.method.annotation.ResponseEntityExceptionHandler;

import java.util.\*;

import java.util.stream.Collectors;

@ControllerAdvice

public class GlobalExceptionHandler extends ResponseEntityExceptionHandler {

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

    @Override

    protected ResponseEntity<Object> handleMethodArgumentNotValid(

            MethodArgumentNotValidException ex,

            HttpHeaders headers,

            HttpStatusCode status,

            WebRequest request) {

        LOGGER.info("Start - Validation Error");

        Map<String, Object> body = new LinkedHashMap<>();

        body.put("timestamp", new Date());

        body.put("status", status.value());

        List<String> errors = ex.getBindingResult().getFieldErrors()

                .stream()

                .map(x -> x.getDefaultMessage())

                .collect(Collectors.toList());

        body.put("errors", errors);

        return new ResponseEntity<>(body, headers, status);

    }

    @Override

    protected ResponseEntity<Object> handleHttpMessageNotReadable(

            HttpMessageNotReadableException ex,

            HttpHeaders headers,

            HttpStatusCode status,

            WebRequest request) {

        Map<String, Object> body = new LinkedHashMap<>();

        body.put("timestamp", new Date());

        body.put("status", status.value());

        body.put("error", "Bad Request");

        if (ex.getCause() instanceof InvalidFormatException cause) {

            for (InvalidFormatException.Reference ref : cause.getPath()) {

                body.put("message", "Incorrect format for field '" + ref.getFieldName() + "'");

            }

        }

        return new ResponseEntity<>(body, headers, status);

    }

}

**Country.java**

package com.cognizant.springlearn.model;

import jakarta.validation.constraints.NotNull;

import jakarta.validation.constraints.Size;

public class Country {

    @NotNull

    @Size(min = 2, max = 2, message = "Country code should be 2 characters")

    private String code;

    @NotNull

    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;

    }

    @Override

    public String toString() {

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

    }

}

Springrest4Application.java

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class Springrest4Application {

    public static void main(String[] args) {

        SpringApplication.run(Springrest4Application.class, args);

    }

}

**Pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"

         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.cognizant</groupId>

    <artifactId>springrest4</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <packaging>jar</packaging>

    <name>springrest4</name>

    <description>Spring REST Project</description>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.2.5</version>

        <relativePath/>

    </parent>

    <properties>

        <java.version>17</java.version>

    </properties>

    <dependencies>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-web</artifactId>

        </dependency>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-validation</artifactId>

        </dependency>

        <dependency>

            <groupId>com.fasterxml.jackson.core</groupId>

            <artifactId>jackson-databind</artifactId>

        </dependency>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-test</artifactId>

            <scope>test</scope>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <plugin>

                <groupId>org.springframework.boot</groupId>

                <artifactId>spring-boot-maven-plugin</artifactId>

            </plugin>

        </plugins>

    </build>

</project>

**OUTPUT:**





