**1.RESTful Web Service resource naming guidelines**   
  
**CountryController**

package com.example.countryrest.controller;

import com.example.countryrest.model.Country;

import com.example.countryrest.service.CountryService;

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

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

import java.util.List;

@RestController

@RequestMapping("/countries")

public class CountryController {

@Autowired

private CountryService countryService;

@GetMapping

public List<Country> getAllCountries() {

return countryService.getAllCountries();

}

@GetMapping("/{code}")

public Country getCountryById(@PathVariable String code) {

return countryService.getCountryById(code);

}

@PostMapping

public Country createCountry(@RequestBody Country country) {

return countryService.createCountry(country);

}

@PutMapping

public Country updateCountry(@RequestBody Country country) {

return countryService.updateCountry(country);

}

@DeleteMapping("/{code}")

public void deleteCountry(@PathVariable String code) {

countryService.deleteCountry(code);

}

}

**Country**

package com.example.countryrest.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;

}

}

**CountryService**

package com.example.countryrest.service;

import com.example.countryrest.model.Country;

import org.springframework.stereotype.Service;

import java.util.\*;

@Service

public class CountryService {

private Map<String, Country> countryMap = new HashMap<>();

public CountryService() {

countryMap.put("IN", new Country("IN", "India"));

countryMap.put("US", new Country("US", "United States"));

}

public List<Country> getAllCountries() {

return new ArrayList<>(countryMap.values());

}

public Country getCountryById(String code) {

return countryMap.get(code);

}

public Country createCountry(Country country) {

countryMap.put(country.getCode(), country);

return country;

}

public Country updateCountry(Country country) {

countryMap.put(country.getCode(), country);

return country;

}

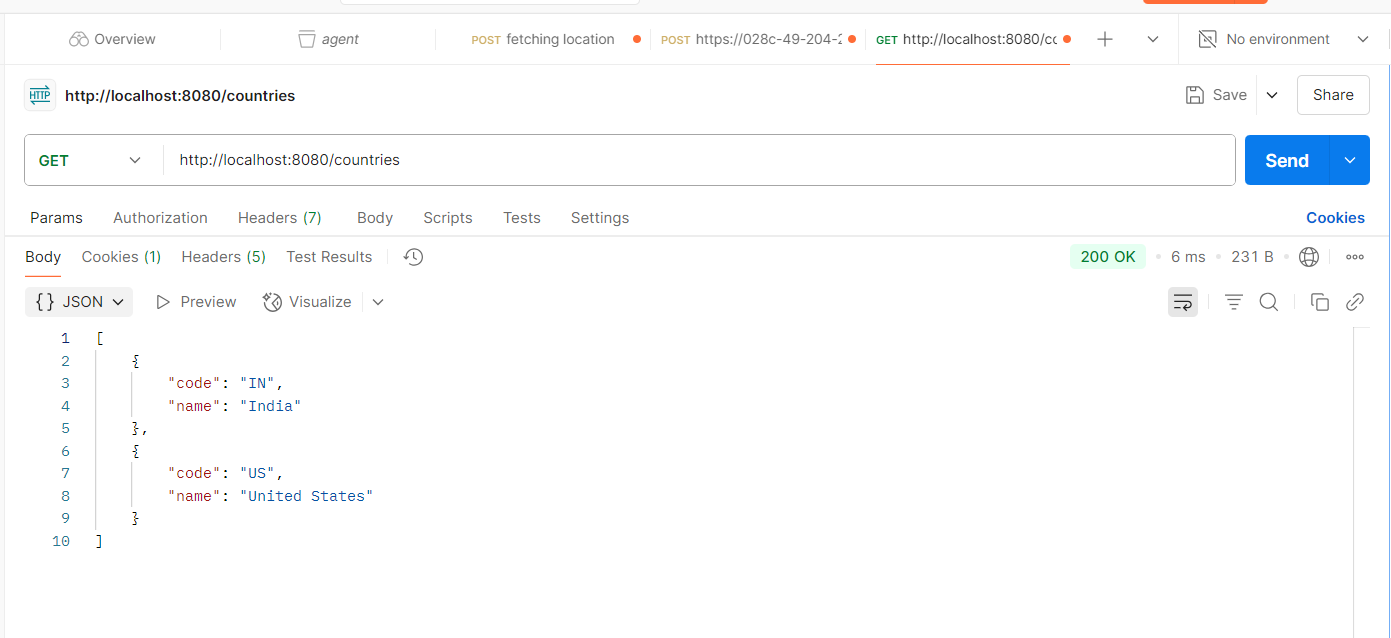
public void deleteCountry(String code) {

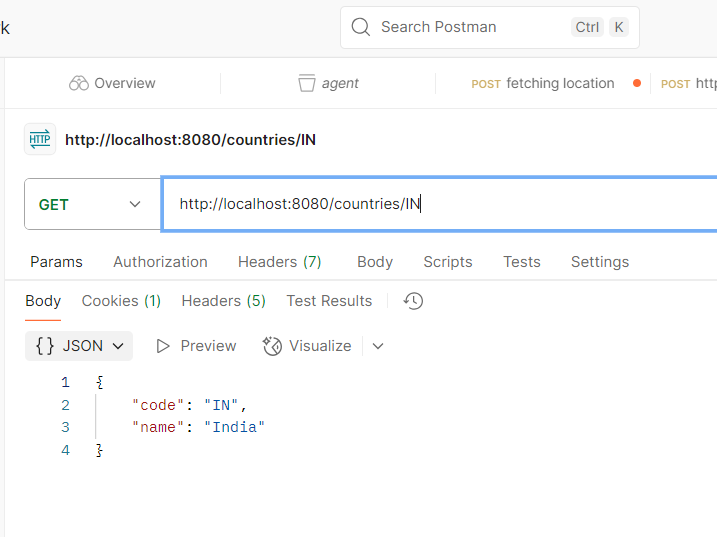
countryMap.remove(code);

}

}







**2.Create RESTful Web Service to handle POST request of Country**

**CountryController**

package com.example.countryrest.controller;

import com.example.countryrest.model.Country;

import com.example.countryrest.service.CountryService;

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

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

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import java.util.List;

@RestController

@RequestMapping("/countries")

public class CountryController {

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

@Autowired

private CountryService countryService;

@GetMapping

public List<Country> getAllCountries() {

return countryService.getAllCountries();

}

@GetMapping("/{code}")

public Country getCountryById(@PathVariable String code) {

return countryService.getCountryById(code);

}

@PostMapping("/log")

public void addCountry() {

logger.info("Start");

}

@PutMapping

public Country updateCountry(@RequestBody Country country) {

return countryService.updateCountry(country);

}

@DeleteMapping("/{code}")

public void deleteCountry(@PathVariable String code) {

countryService.deleteCountry(code);

}

}

**Country**

package com.example.countryrest.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;

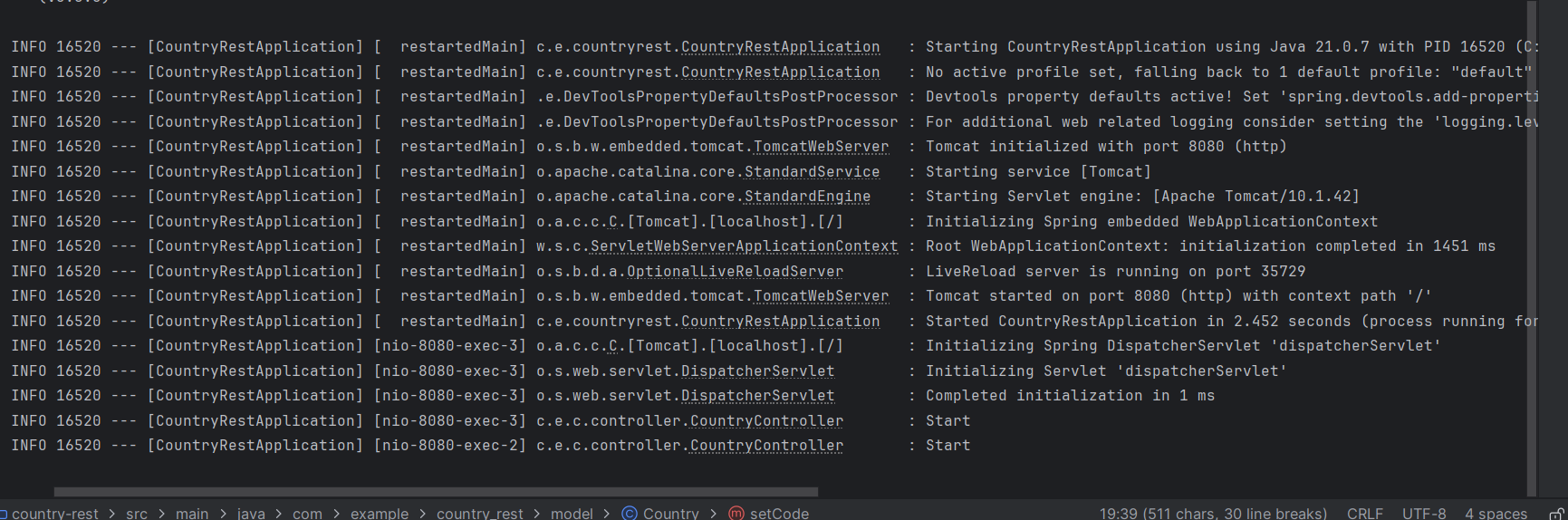
}

public void setName(String name) {

this.name = name;

}

}



**3.Read country data as a bean in RESTful Web Service** 

**CountryController**

package com.example.countryrest.controller;

import com.example.countryrest.model.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

@RestController

@RequestMapping("/countries")

public class CountryController {

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

@PostMapping

public Country addCountry(@RequestBody Country country) {

logger.info("Received Country: code={}, name={}", country.getCode(), country.getName());

return country;

}

}

**Country**

package com.example.countryrest.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;

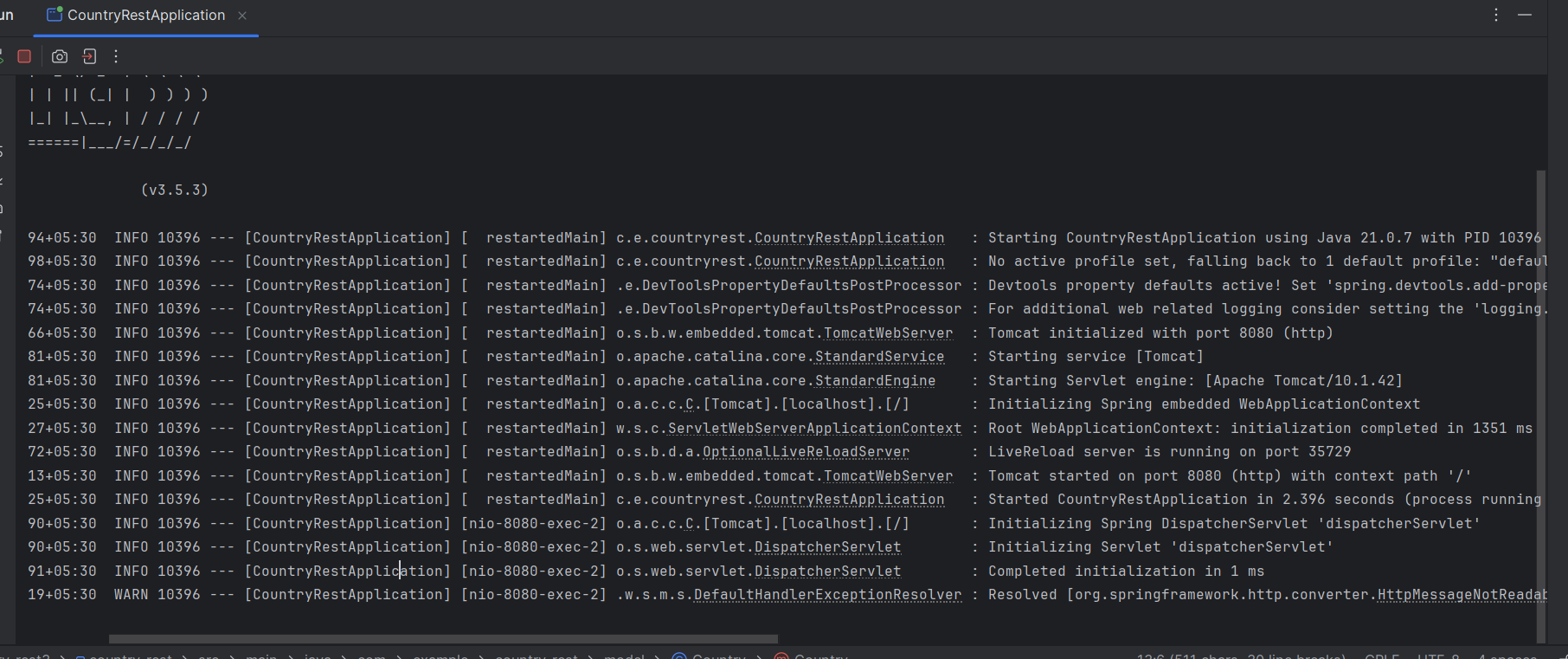
}

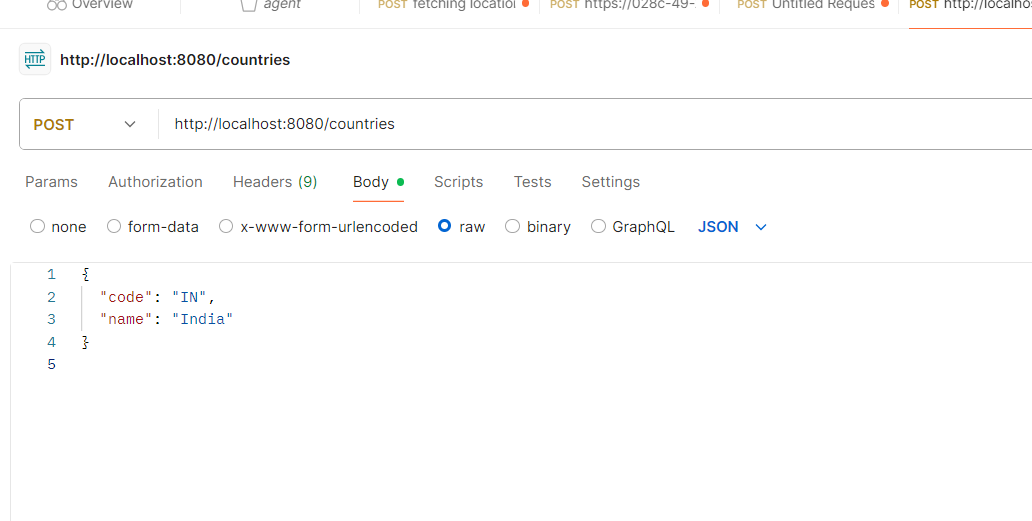
public void setName(String name) {

this.name = name;

}

}





**4.Validating country code** 

package com.example.countryrest.controller;

import com.example.countryrest.model.Country;

import org.springframework.http.HttpStatus;

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

import org.springframework.web.server.ResponseStatusException;

import jakarta.validation.Validation;

import jakarta.validation.Validator;

import jakarta.validation.ValidatorFactory;

import jakarta.validation.ConstraintViolation;

import java.util.\*;

@RestController

@RequestMapping("/countries")

public class CountryController {

@PostMapping

public Country addCountry(@RequestBody Country country) {

ValidatorFactory factory = Validation.buildDefaultValidatorFactory();

Validator validator = factory.getValidator();

Set<ConstraintViolation<Country>> violations = validator.validate(country);

List<String> errors = new ArrayList<>();

for (ConstraintViolation<Country> violation : violations) {

errors.add(violation.getMessage());

}

if (!errors.isEmpty()) {

throw new ResponseStatusException(HttpStatus.BAD\_REQUEST, errors.toString());

}

return country;

}

}



**5.Include global exception handler for validation errors** 

**GlobalException**

package com.example.countryrest;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.http.HttpHeaders;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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,

HttpStatus status,

WebRequest request) {

LOGGER.info("Start");

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);

LOGGER.info("End");

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

}

}

