**Hands On 5: Validating country code**   
  
**Country.java**

package com.cognizant.orm\_learn.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.validation.constraints.NotBlank;

import jakarta.validation.constraints.NotNull;

import jakarta.validation.constraints.Size;

@Entity

public class Country {

@Id

@NotNull(message = "Country code must not be null")

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

private String code;

@NotBlank(message = "Country name must not be blank")

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; }

}

**CountryController**

package com.cognizant.orm\_learn.controller;

import com.cognizant.orm\_learn.model.Country;

import com.cognizant.orm\_learn.service.CountryService;

import jakarta.validation.ConstraintViolation;

import jakarta.validation.Validation;

import jakarta.validation.Validator;

import jakarta.validation.ValidatorFactory;

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

import org.springframework.http.HttpStatus;

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

import org.springframework.web.server.ResponseStatusException;

import java.util.ArrayList;

import java.util.List;

import java.util.Set;

@RestController

@RequestMapping("/countries")

public class CountryController {

@Autowired

private CountryService countryService;

@PostMapping

public Country addCountry(@RequestBody Country country) {

System.out.println("Start");

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());

}

System.out.println("Code: " + country.getCode());

System.out.println("Name: " + country.getName());

return countryService.addCountry(country);

}

@GetMapping

public List<Country> getAllCountries() {

return countryService.getAllCountries();

}

@GetMapping("/{code}")

public Country getCountry(@PathVariable String code) {

return countryService.getCountry(code);

}

@PutMapping

public Country updateCountry(@RequestBody Country country) {

return countryService.updateCountry(country);

}

@DeleteMapping("/{code}")

public void deleteCountry(@PathVariable String code) {

countryService.deleteCountry(code);

}

}

**CountryService**

public Country addCountry(Country country) {

return countryRepository.save(country);

}

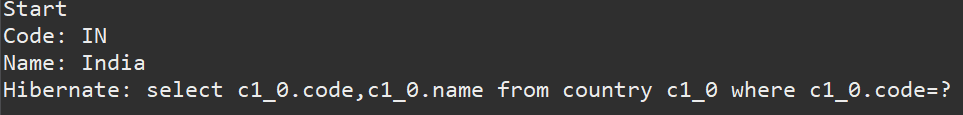
**Output:**

{

"code": "IN",

"name": "India"

}

****

{

"code": "I",

"name": "India"

}

Screenshot 2025-07-13 152733