# Question 1 - Create RESTful Web Service to handle POST request of Country

Code 1: CountryController.java

package com.cognizant.springlearn.controller;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RestController;  
import com.cognizant.springlearn.Country;  
  
@RestController  
public class CountryController {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(CountryController.class);  
  
 @PostMapping("/countries")  
 public Country addCountry(@RequestBody Country country) {  
 LOGGER.info("Start");  
 LOGGER.info("Country Received: {}", country);  
 return country;  
 }  
}

Code 2: Country.java

package com.cognizant.springlearn;  
  
public class Country {  
 private String code;  
 private String 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 + "]";  
 }  
}

Output (Sample - curl command):  
curl -i -H 'Content-Type: application/json' -X POST -s -d '{"code":"IN","name":"India"}' http://localhost:8090/countries  
  
HTTP/1.1 200  
Content-Type: application/json;charset=UTF-8  
{"code":"IN","name":"India"}

# Question 2 - Validating Country Code

Code 1: Country.java (with validation)

package com.cognizant.springlearn;  
  
import javax.validation.constraints.NotNull;  
import javax.validation.constraints.Size;  
  
public class Country {  
  
 @NotNull  
 @Size(min=2, max=2, message="Country code should be 2 characters")  
 private String code;  
  
 private String 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 + "]";  
 }  
}

Code 2: CountryController.java (with @Valid)

@PostMapping("/countries")  
public Country addCountry(@RequestBody @Valid Country country) {  
 LOGGER.info("Start");  
 return country;  
}

Output (Invalid Code - curl command):  
curl -i -H 'Content-Type: application/json' -X POST -s -d '{"code":"I","name":"India"}' http://localhost:8090/countries  
  
HTTP/1.1 400  
Content-Type: application/json;charset=UTF-8  
{"timestamp":"2019-10-02T10:28:56.506+0000","status":400,"error":"Bad Request","message":"[Country code should be 2 characters]","path":"/countries"}

# Question 3 - Global Exception Handler for Validation Errors

Code 1: GlobalExceptionHandler.java

package com.cognizant.springlearn;  
  
import java.util.Date;  
import java.util.LinkedHashMap;  
import java.util.List;  
import java.util.Map;  
import java.util.stream.Collectors;  
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;  
  
@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);  
 }  
}

Output (Global Validation - curl command):  
curl -i -H 'Content-Type: application/json' -X POST -s -d '{"code":"I","name":"India"}' http://localhost:8090/countries  
  
HTTP/1.1 400  
Content-Type: application/json;charset=UTF-8  
{"timestamp":"2019-10-03T04:10:17.277+0000","status":400,"errors":["Country code should be 2 characters"]}