**Objectives**

* Demonstrate implementation of RESTful Web Service using POST/PUT/DELETE method with input validation
  + HTTP method types (GET, POST, PUT, DELETE), REST service URL naming guidelines, @RequestMapping, @GetMapping, @PostMapping, @PutMapping, @DeleteMapping, setting POST request payload and invoking the REST service in Postman and curl, JSON to bean mapping, @RequestBody, validating input request using javax.validation and hibernate validators, @Size, @NotNull, @NotBlank, @Min, @Max, @JsonFormat, @Valid, global exception handling, handle number formatting errors
    - HTTP Request Methods - https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods
    - RESTful API naming guide - https://restfulapi.net/resource-naming/
    - Request Mapping - https://docs.spring.io/spring/docs/5.2.0.RELEASE/spring-framework-reference/web.html#mvc-ann-requestmapping
    - Validation - https://www.mkyong.com/spring-boot/spring-rest-validation-example/

**Significance of HTTP Method Types in RESTful Web Services**   
  
SME to explain the importance of HTTP Method Types for RESTful Web Services. 

|  |  |
| --- | --- |
| **HTTP Method** | **Usage Scenario** |
| GET | Used to get data about a resource |
| POST | Used to create a resource |
| PUT | Used to update a resource |
| DELETE | Used to delete a resource |

The method type is just a classification and does not actually have the persistence implemented. The respective application is expected to take responsibility in implementing the persistence.

**RESTful Web Service resource naming guidelines**   
  
**Country.java**

package com.cognizant.orm\_learn.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

@Entity

public class Country {

@Id

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;

}

}

**CountryRepository**

package com.cognizant.orm\_learn.repository;

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

import org.springframework.data.jpa.repository.JpaRepository;

public interface CountryRepository extends JpaRepository<Country, String> {

}

**CountryController**

package com.cognizant.orm\_learn.controller;

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

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

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

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

import java.util.List;

@RestController

@RequestMapping("/countries") // Class level: plural resource name

public class CountryController {

@Autowired

private CountryService countryService;

@GetMapping

public List<Country> getAllCountries() {

return countryService.getAllCountries();

}

@GetMapping("/{code}")

public Country getCountry(@PathVariable String code) {

return countryService.getCountry(code);

}

@PostMapping

public void addCountry(@RequestBody Country country) {

countryService.addCountry(country);

}

@PutMapping

public void updateCountry(@RequestBody Country country) {

countryService.updateCountry(country);

}

@DeleteMapping("/{code}")

public void deleteCountry(@PathVariable String code) {

countryService.deleteCountry(code);

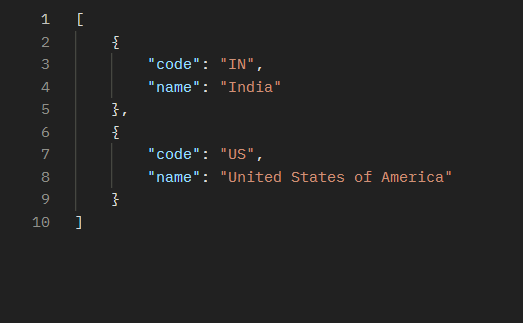
}

}

**Output**

**Get all countries**

**Url:** GET <http://localhost:8080/countries>



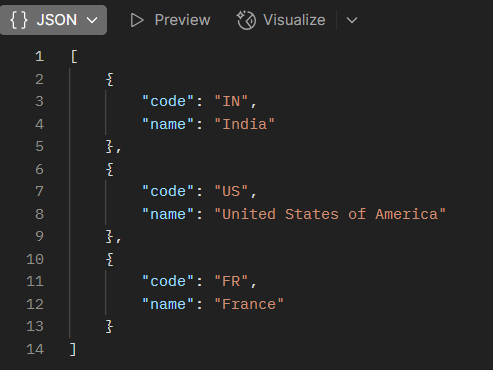
**Get specific countries**

**Url: GET** <http://localhost:8080/countries/IN>



**Create new country**

**Url:** POST <http://localhost:8080/countries>



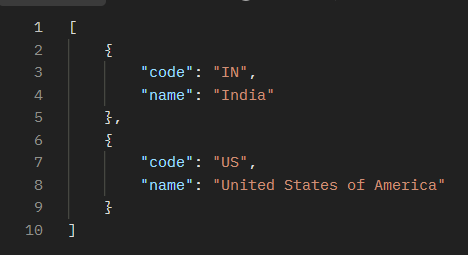
**Update a country**

**Url:** PUT <http://localhost:8080/countries>



**Delete a country**

**Url:** DELETE <http://localhost:8080/countries/FR>



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

**CountryController**

package com.cognizant.orm\_learn.controller;

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

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

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

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

import java.util.List;

@RestController

@RequestMapping("/countries")

public class CountryController {

@Autowired

private CountryService countryService;

@GetMapping

public List<?> getAllCountries() {

return countryService.getAllCountries();

}

@GetMapping("/{code}")

public Object getCountry(@PathVariable String code) {

return countryService.getCountry(code);

}

@PostMapping

public void addCountry() {

System.out.println("Start");

}

@PutMapping

public void updateCountry(@RequestBody Country country) {

countryService.updateCountry(country);

}

@DeleteMapping("/{code}")

public void deleteCountry(@PathVariable String code) {

countryService.deleteCountry(code);

}

}

**Output**

**Url:** POST <http://localhost:8080/countries>



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

**CountryController**

package com.cognizant.orm\_learn.controller;

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

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

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

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

import java.util.List;

@RestController

@RequestMapping("/countries") // ✔️ Plural resource name at class level

public class CountryController {

@Autowired

private CountryService countryService;

@PostMapping

public Country addCountry(@RequestBody Country country) {

System.out.println("Start");

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

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

return country; }

@GetMapping

public List<Country> getAllCountries() {

return countryService.getAllCountries();

}

@GetMapping("/{code}")

public Country getCountry(@PathVariable String code) {

return countryService.getCountry(code);

}

@PutMapping

public void updateCountry(@RequestBody Country country) {

countryService.updateCountry(country);

}

@DeleteMapping("/{code}")

public void deleteCountry(@PathVariable String code) {

countryService.deleteCountry(code);

}

}

**Output**

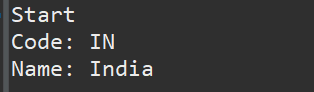
**Url:**POST <http://localhost:8080/countries>

{

  "code": "IN",

  "name": "India"

}



**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);

}

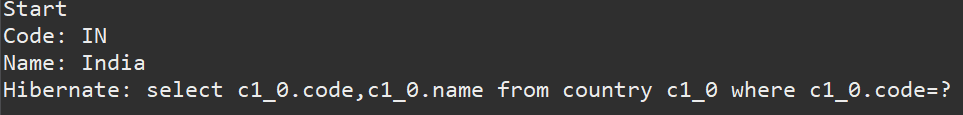
**Output:**

{

"code": "IN",

"name": "India"

}

****

{

"code": "I",

"name": "India"

}

Screenshot 2025-07-13 152733

**Include global exception handler for validation errors**   
  
**CountryController**

@PostMapping

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

LOGGER.info("CountryController Start");

return countryService.addCountry(country);

}

**GlobalExceptionHandler**

package com.cognizant.orm\_learn;

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);

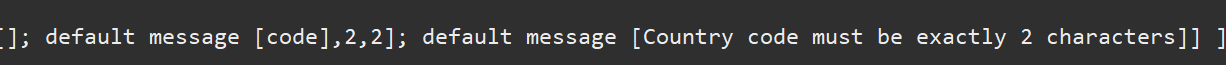
LOGGER.info("End");

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

}

}

**Output**

**Screenshot 2025-07-13 184134**

**Implement REST service for updating an employee**

**EmployeeDao**

package com.cognizant.orm\_learn.dao;

import com.cognizant.orm\_learn.exception.EmployeeNotFoundException;

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

import org.springframework.stereotype.Repository;

import java.util.ArrayList;

import java.util.List;

@Repository

public class EmployeeDao {

private static List<Employee> employeeList = new ArrayList<>();

static {

employeeList.add(new Employee(1, "John Doe", 50000.0, true, "01/01/1990"));

employeeList.add(new Employee(2, "Jane Smith", 60000.0, false, "02/02/1985"));

}

public void updateEmployee(Employee updatedEmployee) throws EmployeeNotFoundException {

boolean found = false;

for (Employee emp : employeeList) {

if (emp.getId() == updatedEmployee.getId()) {

emp.setName(updatedEmployee.getName());

emp.setSalary(updatedEmployee.getSalary());

emp.setPermanent(updatedEmployee.getPermanent());

emp.setDateOfBirth(updatedEmployee.getDateOfBirth());

emp.setDepartment(updatedEmployee.getDepartment());

emp.setSkillList(updatedEmployee.getSkillList());

found = true;

break;

}

}

if (!found) {

throw new EmployeeNotFoundException("Employee not found with ID: " + updatedEmployee.getId());

}

}

public List<Employee> getAllEmployees() {

return employeeList;

}

}

**EmployeeNotFoundException**

package com.cognizant.orm\_learn.exception;

import org.springframework.http.HttpStatus;

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

@ResponseStatus(HttpStatus.NOT\_FOUND)

public class EmployeeNotFoundException extends Exception {

private static final long serialVersionUID = 1L;

public EmployeeNotFoundException(String message) {

super(message);

}

}

**EmployeeController**

package com.cognizant.orm\_learn.controller;

import com.cognizant.orm\_learn.exception.EmployeeNotFoundException;

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

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

import jakarta.validation.Valid;

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

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

import java.util.List;

@RestController

@RequestMapping("/employees")

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@PutMapping

public void updateEmployee(@RequestBody @Valid Employee employee) throws EmployeeNotFoundException {

employeeService.updateEmployee(employee);

}

@GetMapping

public List<Employee> getAllEmployees() {

return employeeService.getAllEmployees();

}

}

**EmployeeService**

package com.cognizant.orm\_learn.service;

import com.cognizant.orm\_learn.dao.EmployeeDao;

import com.cognizant.orm\_learn.exception.EmployeeNotFoundException;

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

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class EmployeeService {

@Autowired

private EmployeeDao employeeDao;

public void updateEmployee(Employee employee) throws EmployeeNotFoundException {

employeeDao.updateEmployee(employee);

}

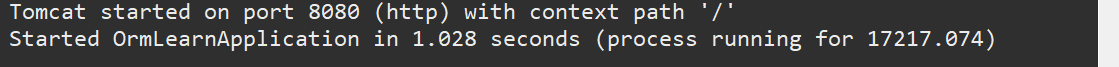
public List<Employee> getAllEmployees() {

return employeeDao.getAllEmployees();

}

}

**Output**

****

****

**Implement REST DELETE Service**   
  
**EmployeeDao**

public void deleteEmployee(int id) throws EmployeeNotFoundException {

boolean removed = employeeList.removeIf(emp -> emp.getId() == id);

if (!removed) {

throw new EmployeeNotFoundException("Employee not found with id: " + id);

}

}

**EmployeeService**

public void deleteEmployee(int id) throws EmployeeNotFoundException {

employeeDao.deleteEmployee(id);

}

**EmployeeController**

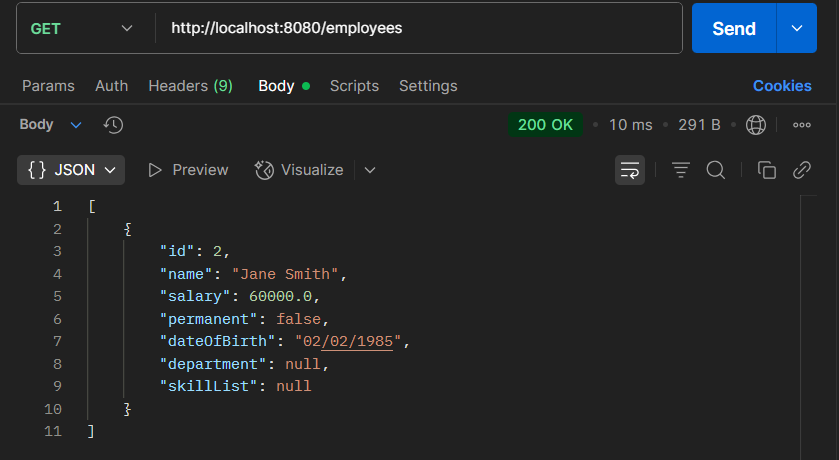
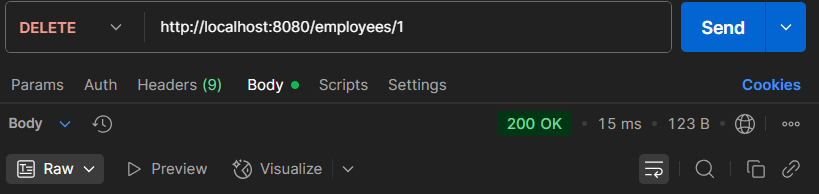
@DeleteMapping("/employees/{id}")

public void deleteEmployee(@PathVariable int id) throws EmployeeNotFoundException {

employeeService.deleteEmployee(id);

}

**Output**

****