COGNIZANT

Digital Nurture 4.0

Deep Skilling - Java FSE

WEEK-4 HANDS ON

By Kaviya P

4.SPRING-REST -HANDSON

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

**SpringLearnApplication.java**

**package** com.cognizant.spring\_learn;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** SpringLearnApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(SpringLearnApplication.**class**, args);

}

}

**CountryController.java**

**package** com.cognizant.spring\_learn.controller;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

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

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

**import** com.cognizant.spring\_learn.model.Country;

**import** com.cognizant.spring\_learn.service.CountryService;

@RestController

@RequestMapping("/countries")

**public** **class** CountryController {

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

@Autowired

**private** CountryService countryService;

@PostMapping()

**public** **void** addCountry() {

***LOGGER***.info("Start");

// Dummy data just for testing (since no request body is passed)

Country country = **new** Country("IN", "India");

countryService.addCountry(country);

}

}

**Country.java**

**package** com.cognizant.spring\_learn.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.java**

**package** com.cognizant.spring\_learn.service;

**import** org.springframework.stereotype.Service;

**import** com.cognizant.spring\_learn.model.Country;

@Service

**public** **class** CountryService {

**public** **void** addCountry(Country country) {

// Placeholder for logic, e.g., save to database

System.***out***.println("Country added: " + country.getName());

}

}

**index.html**

<!DOCTYPE html>

<html>

<head>

<title>Test POST Country</title>

</head>

<body>

<form action=*"http://localhost:8080/countries"* method=*"post"*>

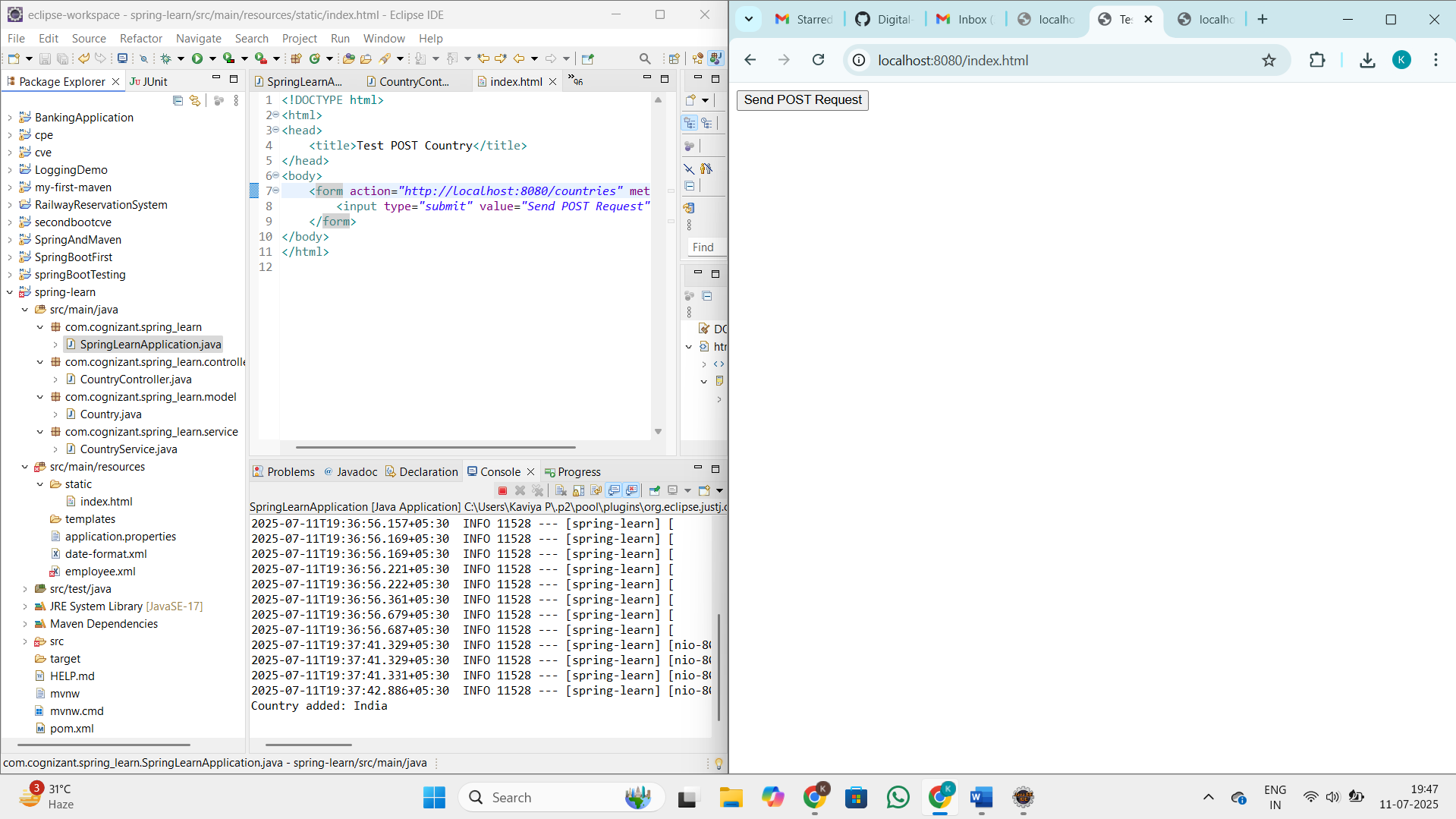
<input type=*"submit"* value=*"Send POST Request"*>

</form>

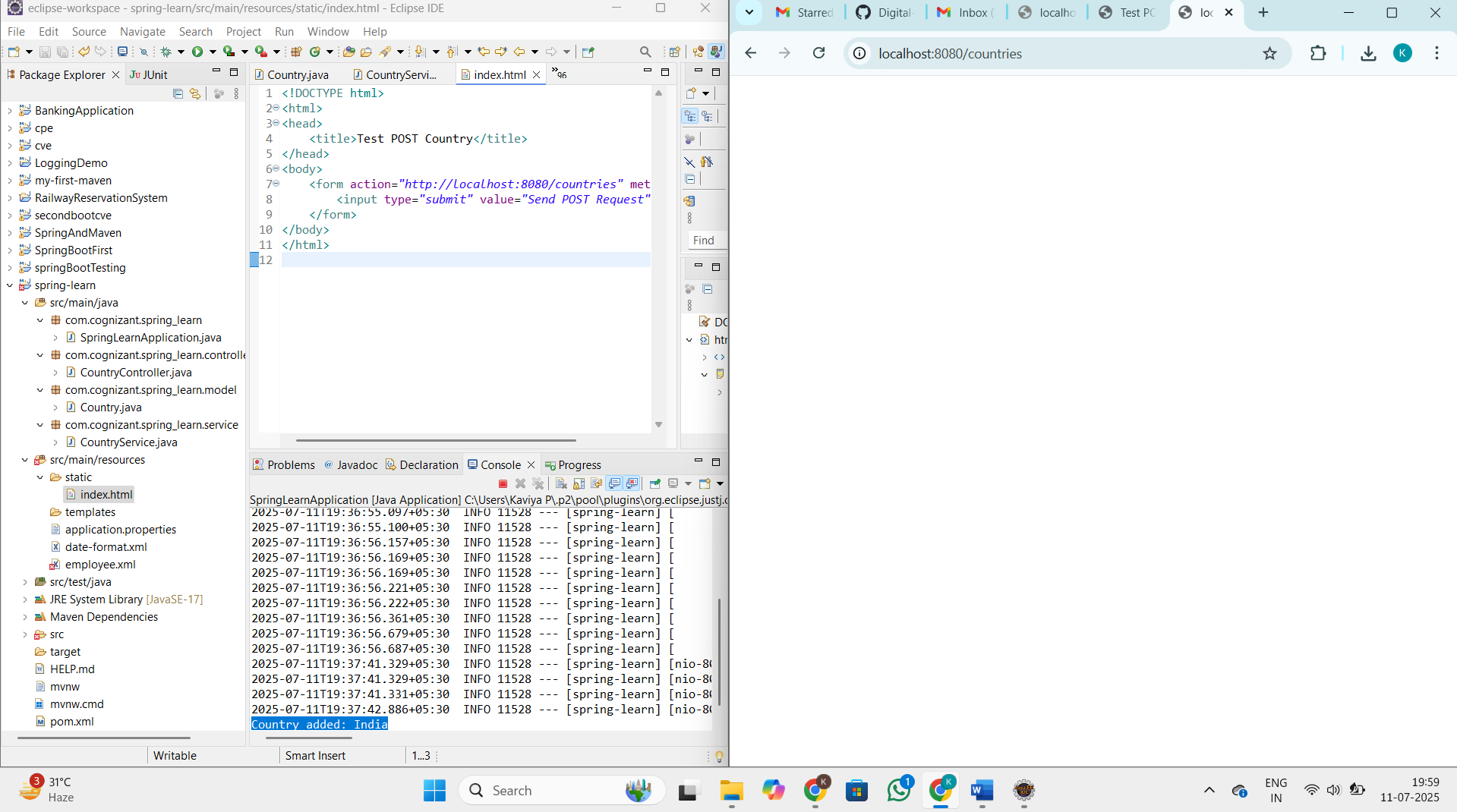
</body>

</html>

**OUTPUT**

****

**When you click the send post request country india will be saved**

****

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

**SpringLearnApplication.java**

**package** com.cognizant.spring\_learn;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** SpringLearnApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(SpringLearnApplication.**class**, args);

}

}

**CountryController.java**

**package** com.cognizant.spring\_learn.controller;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

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

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

**import** com.cognizant.spring\_learn.model.Country;

**import** com.cognizant.spring\_learn.service.CountryService;

@RestController

@RequestMapping("/countries")

**public** **class** CountryController {

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

@PostMapping

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

***LOGGER***.info("Start");

***LOGGER***.info("Country Code: {}", country.getCode());

***LOGGER***.info("Country Name: {}", country.getName());

**return** country;

}

}

**Country.java**

**package** com.cognizant.spring\_learn.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.java**

**package** com.cognizant.spring\_learn.service;

**import** org.springframework.stereotype.Service;

**import** com.cognizant.spring\_learn.model.Country;

@Service

**public** **class** CountryService {

**public** **void** addCountry(Country country) {

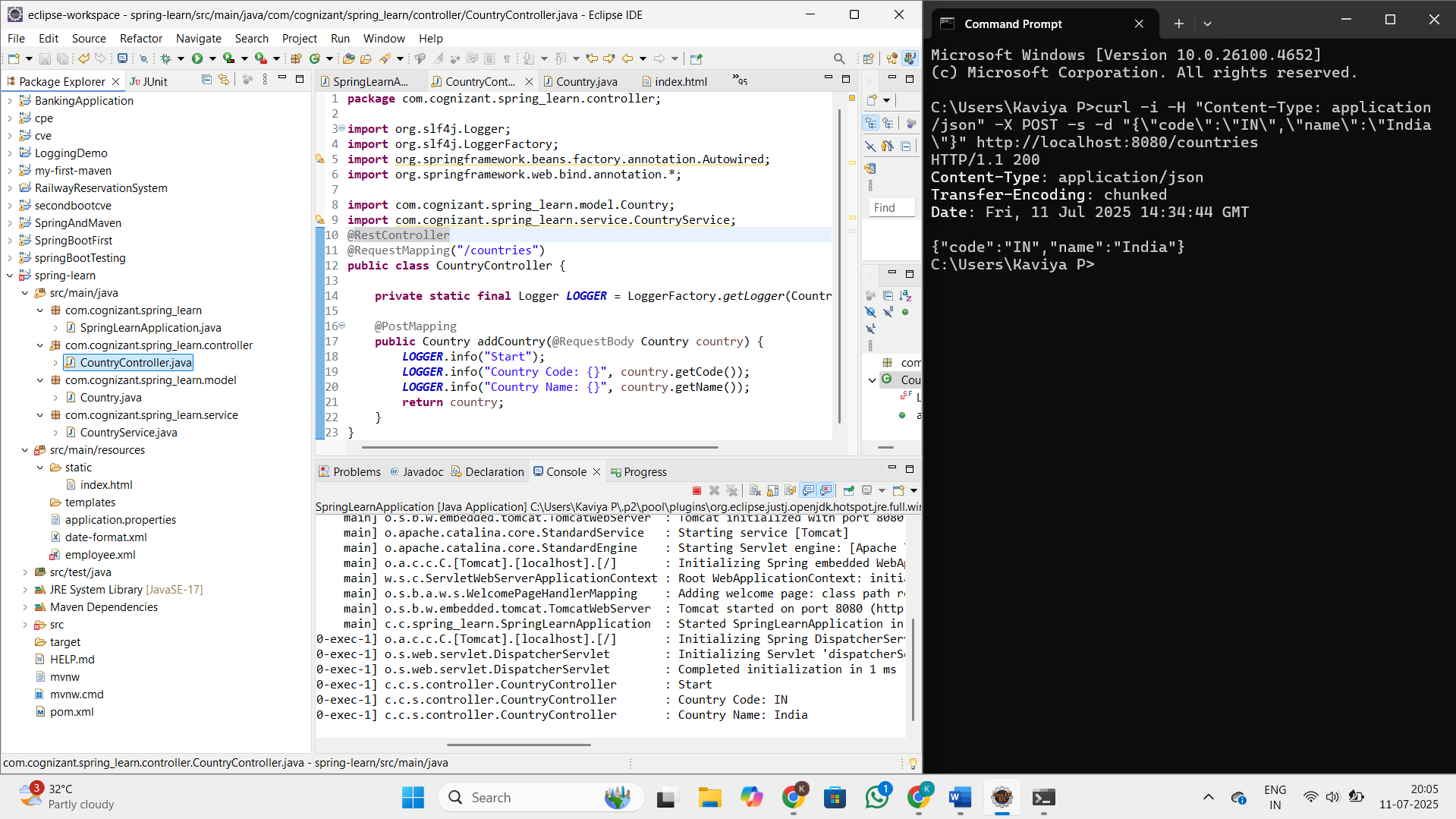
// Placeholder for logic, e.g., save to database

System.***out***.println("Country added: " + country.getName());

}

}

**OUTPUT**

****

**Validating country code**

**SpringLearnApplication.java**

**package** com.cognizant.spring\_learn;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** SpringLearnApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(SpringLearnApplication.**class**, args);

}

}

**CountryController.java**

**package** com.cognizant.spring\_learn.controller;

**import** com.cognizant.spring\_learn.model.Country;

**import** jakarta.validation.ConstraintViolation;

**import** jakarta.validation.Validation;

**import** jakarta.validation.Validator;

**import** jakarta.validation.ValidatorFactory;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

**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 {

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

@PostMapping

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

***LOGGER***.info("Start");

// Set up validation

ValidatorFactory factory = Validation.*buildDefaultValidatorFactory*();

Validator validator = factory.getValidator();

// Validate country bean

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 the country if valid

**return** country;

}

}

**Country.java**

**package** com.cognizant.spring\_learn.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;

**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;

}

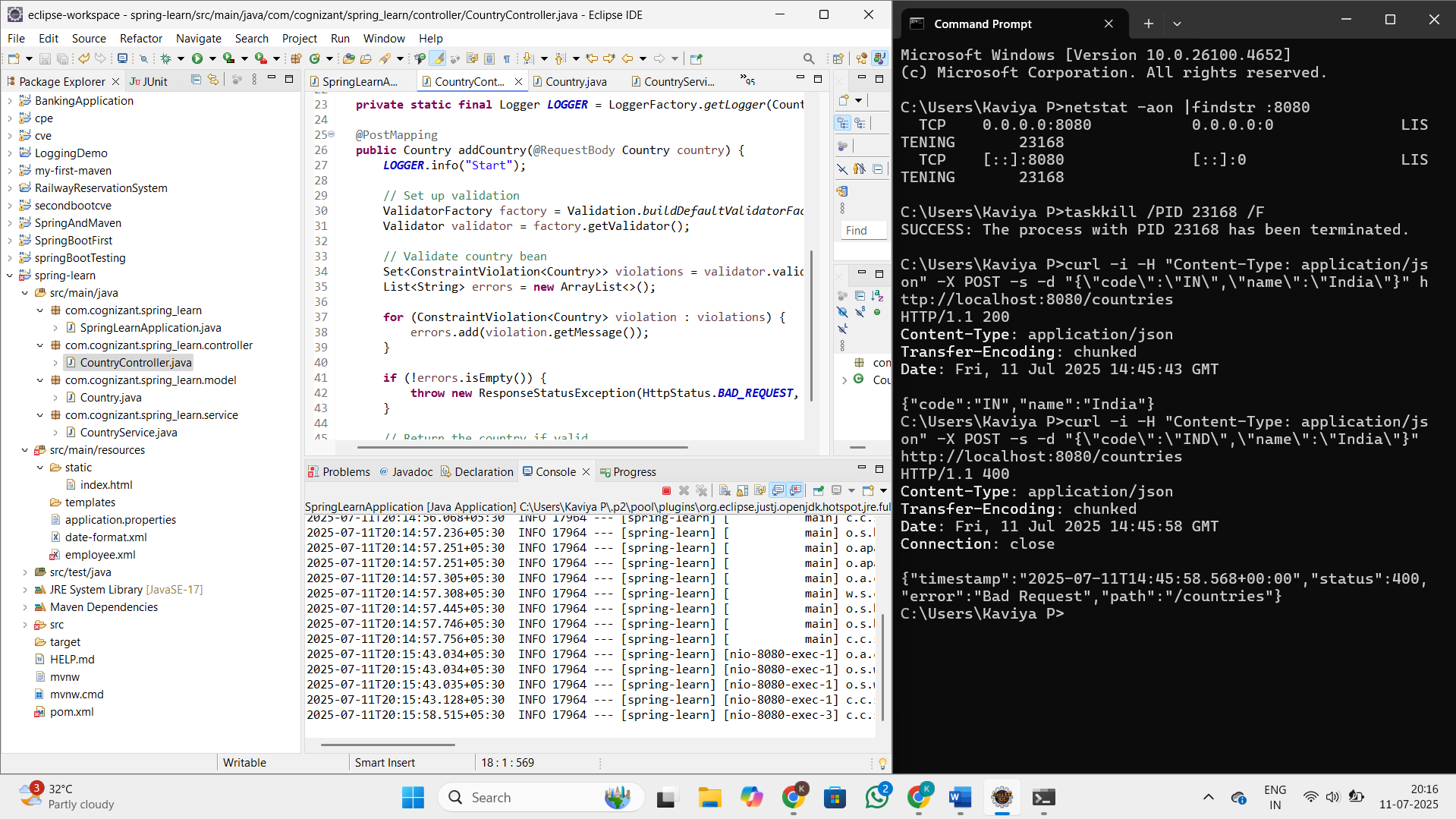
**public** **void** setName(String name) {

**this**.name = name;

}

}

**OUTPUT**

****

**Include global exception handler for validation errors**

**package** com.cognizant.spring\_learn;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** SpringLearnApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(SpringLearnApplication.**class**, args);

}

}

**CountryController.java**

**package** com.cognizant.spring\_learn.controller;

**import** com.cognizant.spring\_learn.model.Country;

**import** jakarta.validation.Valid;

**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 @Valid Country country) {

***LOGGER***.info("Start - Inside CountryController");

**return** country;

}

}

**Country.java**

**package** com.cognizant.spring\_learn.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;

**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;

}

}

**GlobalExceptionHandler.java**

**package** com.cognizant.spring\_learn;

**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.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 - Global Exception Handler");

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);

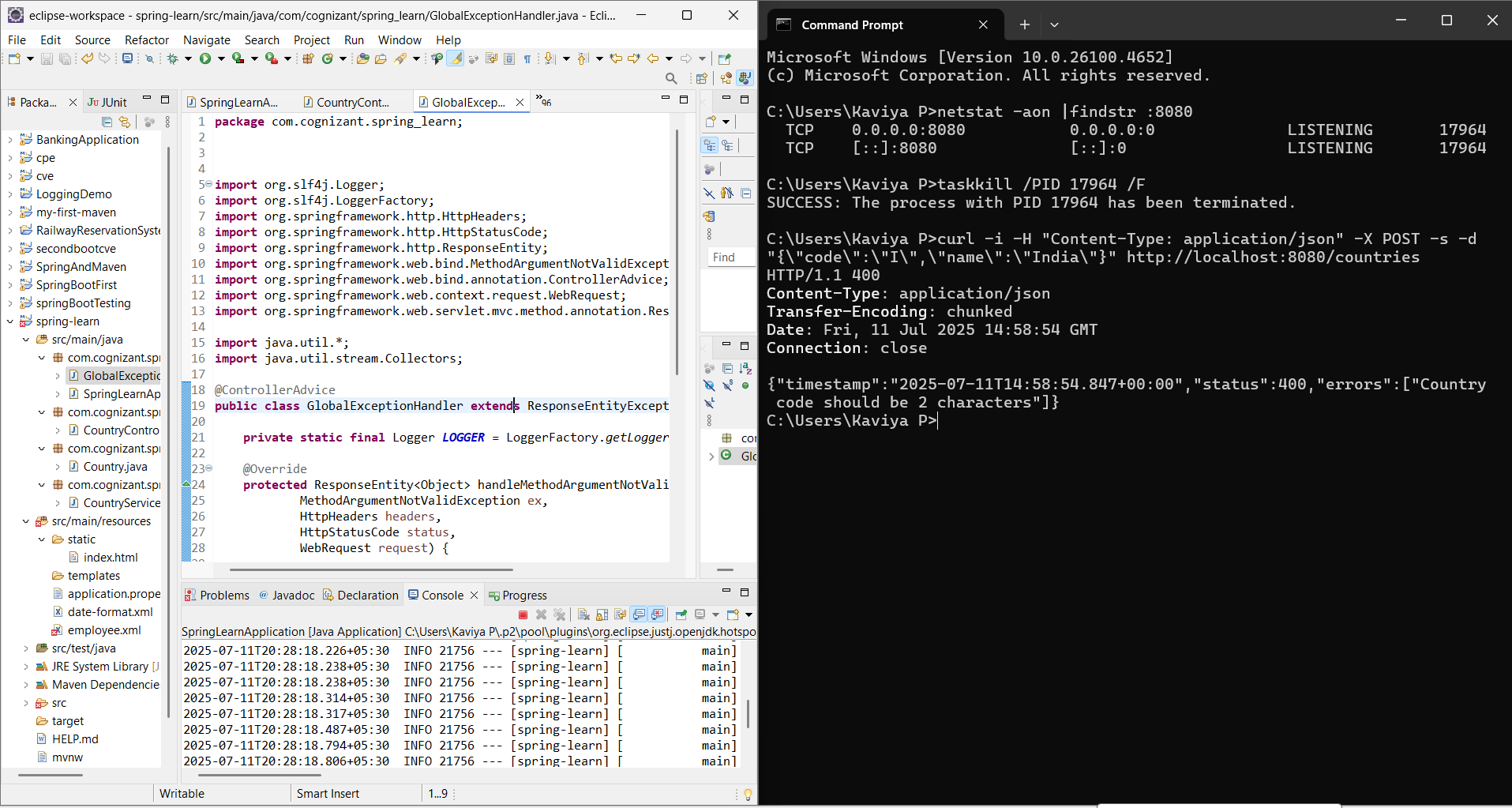
***LOGGER***.info("End - Global Exception Handler");

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

}

}

**OUTPUT**

****

**Implement REST service for updating an employee**

**EmployeeNotFoundException.java**

**package** com.cognizant.spring\_learn;

**import** org.springframework.http.HttpStatus;

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

@ResponseStatus(HttpStatus.***NOT\_FOUND***)

**public** **class** EmployeeNotFoundException **extends** Exception {

**public** EmployeeNotFoundException(String message) {

**super**(message);

}

}

**GlobalExceptionHandler.java**

**package** com.cognizant.spring\_learn;

**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) {

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

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

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

List<String> errors = ex.getBindingResult()

.getFieldErrors()

.stream()

.map(err -> err.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) {

**for** (InvalidFormatException.Reference ref : ((InvalidFormatException) ex.getCause()).getPath()) {

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

}

}

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

}

}

**SpringLearnApplication.java**

**package** com.cognizant.spring\_learn;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** SpringLearnApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(SpringLearnApplication.**class**, args);

}

}

**EmployeeController.java**

**package** com.cognizant.spring\_learn.controller;

**import** com.cognizant.spring\_learn.EmployeeNotFoundException;

**import** com.cognizant.spring\_learn.model.Employee;

**import** com.cognizant.spring\_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

**public** **class** EmployeeController {

@Autowired

**private** EmployeeService employeeService;

@PutMapping("/employees")

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

employeeService.updateEmployee(employee);

}

@GetMapping("/employees")

**public** List<Employee> getAllEmployees() {

**return** employeeService.getAllEmployees();

}

}

**EmployeeDao.java**

**package** com.cognizant.spring\_learn.dao;

**import** com.cognizant.spring\_learn.EmployeeNotFoundException;

**import** com.cognizant.spring\_learn.model.Employee;

**import** com.cognizant.spring\_learn.model.Department;

**import** com.cognizant.spring\_learn.model.Skill;

**import** jakarta.annotation.PostConstruct;

**import** org.springframework.stereotype.Repository;

**import** java.util.\*;

@Repository

**public** **class** EmployeeDao {

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

@PostConstruct

**public** **void** init() {

// Create sample employee

Employee emp = **new** Employee();

emp.setId(1);

emp.setName("Default Emp");

emp.setSalary((**float**)(3000.0f));

emp.setPermanent(**true**);

emp.setDateOfBirth(**null**);

Department dept = **new** Department();

dept.setId(101);

dept.setName("HR");

emp.setDepartment(dept);

Skill skill1 = **new** Skill();

skill1.setId(1);

skill1.setName("Communication");

Skill skill2 = **new** Skill();

skill2.setId(2);

skill2.setName("Teamwork");

emp.setSkillList(Arrays.*asList*(skill1, skill2));

*employeeList*.add(emp);

}

**public** **void** updateEmployee(Employee updatedEmployee) **throws** EmployeeNotFoundException {

**for** (**int** i = 0; i < *employeeList*.size(); i++) {

**if** (*employeeList*.get(i).getId().equals(updatedEmployee.getId())) {

*employeeList*.set(i, updatedEmployee);

**return**;

}

}

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

}

**public** List<Employee> getAllEmployees() {

**return** *employeeList*;

}

}

**Department.java**

**package** com.cognizant.spring\_learn.model;

**import** jakarta.validation.constraints.\*;

**public** **class** Department {

@NotNull

**private** Integer id;

@NotBlank

@Size(min = 1, max = 30)

**private** String name;

**public** Integer getId() {

**return** id;

}

**public** **void** setId(Integer id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

// Getters and Setters

}

**Employee.java**

**package** com.cognizant.spring\_learn.model;

**import** com.fasterxml.jackson.annotation.JsonFormat;

**import** jakarta.validation.constraints.\*;

**import** java.util.Date;

**import** java.util.List;

**public** **class** Employee {

@NotNull

**private** Integer id;

@NotBlank

@Size(min = 1, max = 30)

**private** String name;

@NotNull

@PositiveOrZero

**private** Float salary;

@NotNull

**private** Boolean permanent;

@NotNull

@JsonFormat(shape = JsonFormat.Shape.***STRING***, pattern = "dd/MM/yyyy")

**private** Date dateOfBirth;

@NotNull

**private** Department department;

@NotNull

**private** List<Skill> skillList;

**public** Employee()

{

}

**public** Integer getId() {

**return** id;

}

**public** **void** setId(Integer id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** Float getSalary() {

**return** salary;

}

**public** **void** setSalary(Float salary) {

**this**.salary = salary;

}

**public** Boolean getPermanent() {

**return** permanent;

}

**public** **void** setPermanent(Boolean permanent) {

**this**.permanent = permanent;

}

**public** Date getDateOfBirth() {

**return** dateOfBirth;

}

**public** **void** setDateOfBirth(Date dateOfBirth) {

**this**.dateOfBirth = dateOfBirth;

}

**public** Department getDepartment() {

**return** department;

}

**public** **void** setDepartment(Department department) {

**this**.department = department;

}

**public** List<Skill> getSkillList() {

**return** skillList;

}

**public** **void** setSkillList(List<Skill> skillList) {

**this**.skillList = skillList;

}

**public** Employee(@NotNull Integer id, @NotBlank @Size(min = 1, max = 30) String name,

@NotNull @PositiveOrZero Float salary, @NotNull Boolean permanent, @NotNull Date dateOfBirth,

@NotNull Department department, @NotNull List<Skill> skillList) {

**this**.id = id;

**this**.name = name;

**this**.salary = salary;

**this**.permanent = permanent;

**this**.dateOfBirth = dateOfBirth;

**this**.department = department;

**this**.skillList = skillList;

}

// Getters and Setters

}

**Skill.java**

**package** com.cognizant.spring\_learn.model;

**import** jakarta.validation.constraints.\*;

**public** **class** Skill {

@NotNull

**private** Integer id;

@NotBlank

@Size(min = 1, max = 30)

**private** String name;

**public** Integer getId() {

**return** id;

}

**public** **void** setId(Integer id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

// Getters and Setters

}

**EmployeeService.java**

**package** com.cognizant.spring\_learn.service;

**import** com.cognizant.spring\_learn.EmployeeNotFoundException;

**import** com.cognizant.spring\_learn.dao.EmployeeDao;

**import** com.cognizant.spring\_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);

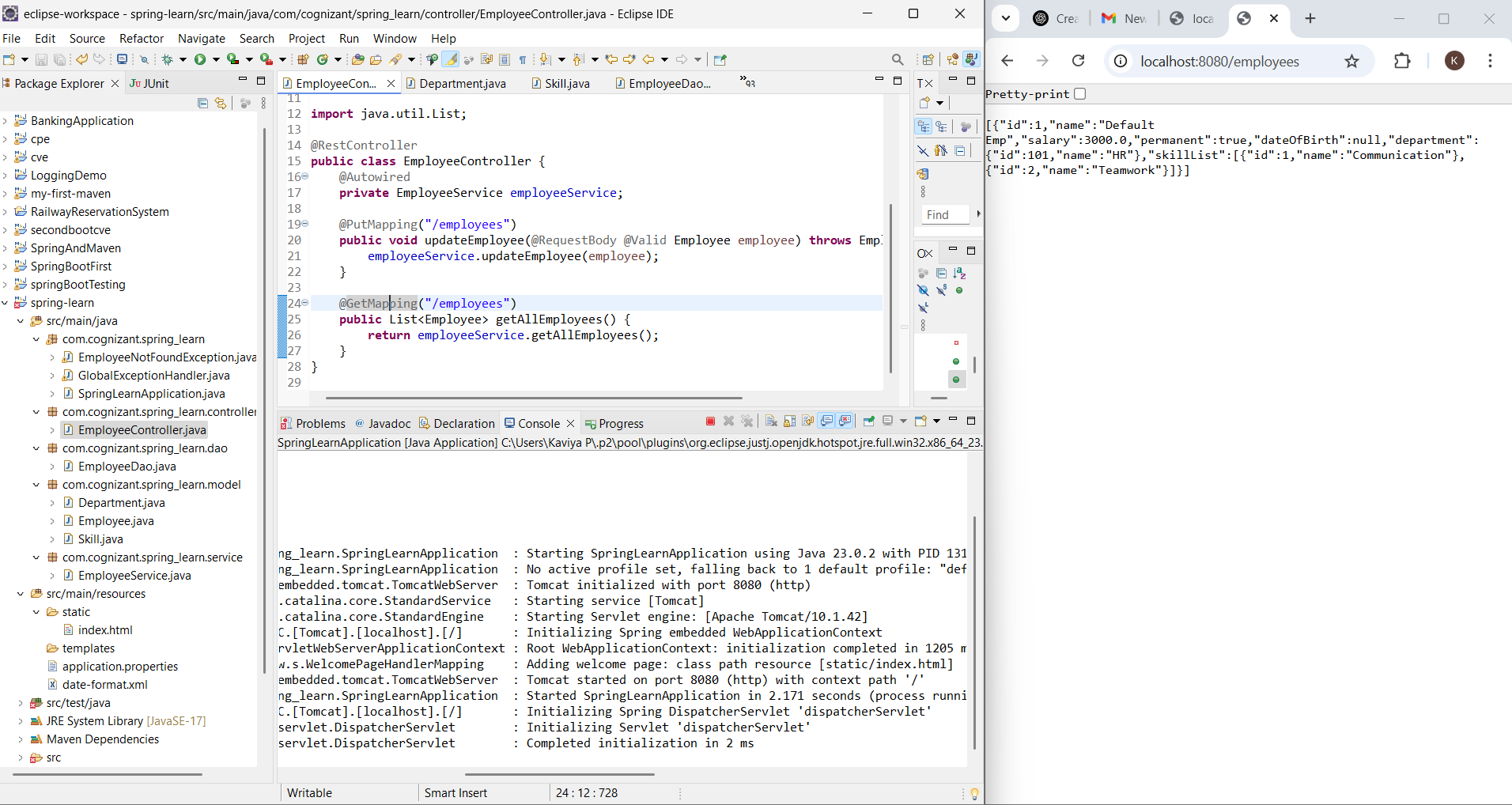
}

**public** List<Employee> getAllEmployees() {

**return** employeeDao.getAllEmployees();

}

}

****

**Implement REST DELETE Service**   
  
Implement steps below to complete this:

* Implement a new delete service for employee by incorporating relevant code in EmployeeController, EmployeeService and EmployeeDao.
* The EmployeeDao should have the code to remove the respective item from the list or throw EmployeeNotFoundException if the id not found.
* Test the service and check if deletion happens correctly

**EmployeeNotFoundException.java**

**package** com.cognizant.spring\_learn;

**import** org.springframework.http.HttpStatus;

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

@ResponseStatus(HttpStatus.***NOT\_FOUND***)

**public** **class** EmployeeNotFoundException **extends** Exception {

**public** EmployeeNotFoundException(String message) {

**super**(message);

}

}

**GlobalExceptionHandler.java**

**package** com.cognizant.spring\_learn;

**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) {

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

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

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

List<String> errors = ex.getBindingResult()

.getFieldErrors()

.stream()

.map(err -> err.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) {

**for** (InvalidFormatException.Reference ref : ((InvalidFormatException) ex.getCause()).getPath()) {

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

}

}

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

}

}

**SpringLearnApplication.java**

**package** com.cognizant.spring\_learn;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** SpringLearnApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(SpringLearnApplication.**class**, args);

}

}

**EmployeeController.java**

package com.cognizant.spring\_learn.controller;

import com.cognizant.spring\_learn.EmployeeNotFoundException;

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

import com.cognizant.spring\_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

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@PutMapping("/employees")

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

employeeService.updateEmployee(employee);

}

@GetMapping("/employees")

public List<Employee> getAllEmployees() {

return employeeService.getAllEmployees();

}

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

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

employeeService.deleteEmployee(id);

}

}

**EmployeeDao.java**

**package** com.cognizant.spring\_learn.dao;

**import** com.cognizant.spring\_learn.EmployeeNotFoundException;

**import** com.cognizant.spring\_learn.model.Employee;

**import** com.cognizant.spring\_learn.model.Department;

**import** com.cognizant.spring\_learn.model.Skill;

**import** jakarta.annotation.PostConstruct;

**import** org.springframework.stereotype.Repository;

**import** java.util.\*;

@Repository

**public** **class** EmployeeDao {

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

@PostConstruct

**public** **void** init() {

// Create sample employee

Employee emp = **new** Employee();

emp.setId(1);

emp.setName("Default Emp");

emp.setSalary((**float**)(3000.0f));

emp.setPermanent(**true**);

emp.setDateOfBirth(**null**);

Department dept = **new** Department();

dept.setId(101);

dept.setName("HR");

emp.setDepartment(dept);

Skill skill1 = **new** Skill();

skill1.setId(1);

skill1.setName("Communication");

Skill skill2 = **new** Skill();

skill2.setId(2);

skill2.setName("Teamwork");

emp.setSkillList(Arrays.*asList*(skill1, skill2));

*employeeList*.add(emp);

}

**public** **void** deleteEmployee(**int** id) **throws** EmployeeNotFoundException {

Iterator<Employee> iterator = *employeeList*.iterator();

**while** (iterator.hasNext()) {

Employee emp = iterator.next();

**if** (emp.getId() == id) {

iterator.remove();

**return**;

}

}

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

}

**public** **void** updateEmployee(Employee updatedEmployee) **throws** EmployeeNotFoundException {

**for** (**int** i = 0; i < *employeeList*.size(); i++) {

**if** (*employeeList*.get(i).getId().equals(updatedEmployee.getId())) {

*employeeList*.set(i, updatedEmployee);

**return**;

}

}

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

}

**public** List<Employee> getAllEmployees() {

**return** *employeeList*;

}

}

**Department.java**

**package** com.cognizant.spring\_learn.model;

**import** jakarta.validation.constraints.\*;

**public** **class** Department {

@NotNull

**private** Integer id;

@NotBlank

@Size(min = 1, max = 30)

**private** String name;

**public** Integer getId() {

**return** id;

}

**public** **void** setId(Integer id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

// Getters and Setters

}

**Employee.java**

**package** com.cognizant.spring\_learn.model;

**import** com.fasterxml.jackson.annotation.JsonFormat;

**import** jakarta.validation.constraints.\*;

**import** java.util.Date;

**import** java.util.List;

**public** **class** Employee {

@NotNull

**private** Integer id;

@NotBlank

@Size(min = 1, max = 30)

**private** String name;

@NotNull

@PositiveOrZero

**private** Float salary;

@NotNull

**private** Boolean permanent;

@NotNull

@JsonFormat(shape = JsonFormat.Shape.***STRING***, pattern = "dd/MM/yyyy")

**private** Date dateOfBirth;

@NotNull

**private** Department department;

@NotNull

**private** List<Skill> skillList;

**public** Employee()

{

}

**public** Integer getId() {

**return** id;

}

**public** **void** setId(Integer id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** Float getSalary() {

**return** salary;

}

**public** **void** setSalary(Float salary) {

**this**.salary = salary;

}

**public** Boolean getPermanent() {

**return** permanent;

}

**public** **void** setPermanent(Boolean permanent) {

**this**.permanent = permanent;

}

**public** Date getDateOfBirth() {

**return** dateOfBirth;

}

**public** **void** setDateOfBirth(Date dateOfBirth) {

**this**.dateOfBirth = dateOfBirth;

}

**public** Department getDepartment() {

**return** department;

}

**public** **void** setDepartment(Department department) {

**this**.department = department;

}

**public** List<Skill> getSkillList() {

**return** skillList;

}

**public** **void** setSkillList(List<Skill> skillList) {

**this**.skillList = skillList;

}

**public** Employee(@NotNull Integer id, @NotBlank @Size(min = 1, max = 30) String name,

@NotNull @PositiveOrZero Float salary, @NotNull Boolean permanent, @NotNull Date dateOfBirth,

@NotNull Department department, @NotNull List<Skill> skillList) {

**this**.id = id;

**this**.name = name;

**this**.salary = salary;

**this**.permanent = permanent;

**this**.dateOfBirth = dateOfBirth;

**this**.department = department;

**this**.skillList = skillList;

}

// Getters and Setters

}

**Skill.java**

**package** com.cognizant.spring\_learn.model;

**import** jakarta.validation.constraints.\*;

**public** **class** Skill {

@NotNull

**private** Integer id;

@NotBlank

@Size(min = 1, max = 30)

**private** String name;

**public** Integer getId() {

**return** id;

}

**public** **void** setId(Integer id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

// Getters and Setters

}

**EmployeeService.java**

**package** com.cognizant.spring\_learn.service;

**import** com.cognizant.spring\_learn.EmployeeNotFoundException;

**import** com.cognizant.spring\_learn.dao.EmployeeDao;

**import** com.cognizant.spring\_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();

}

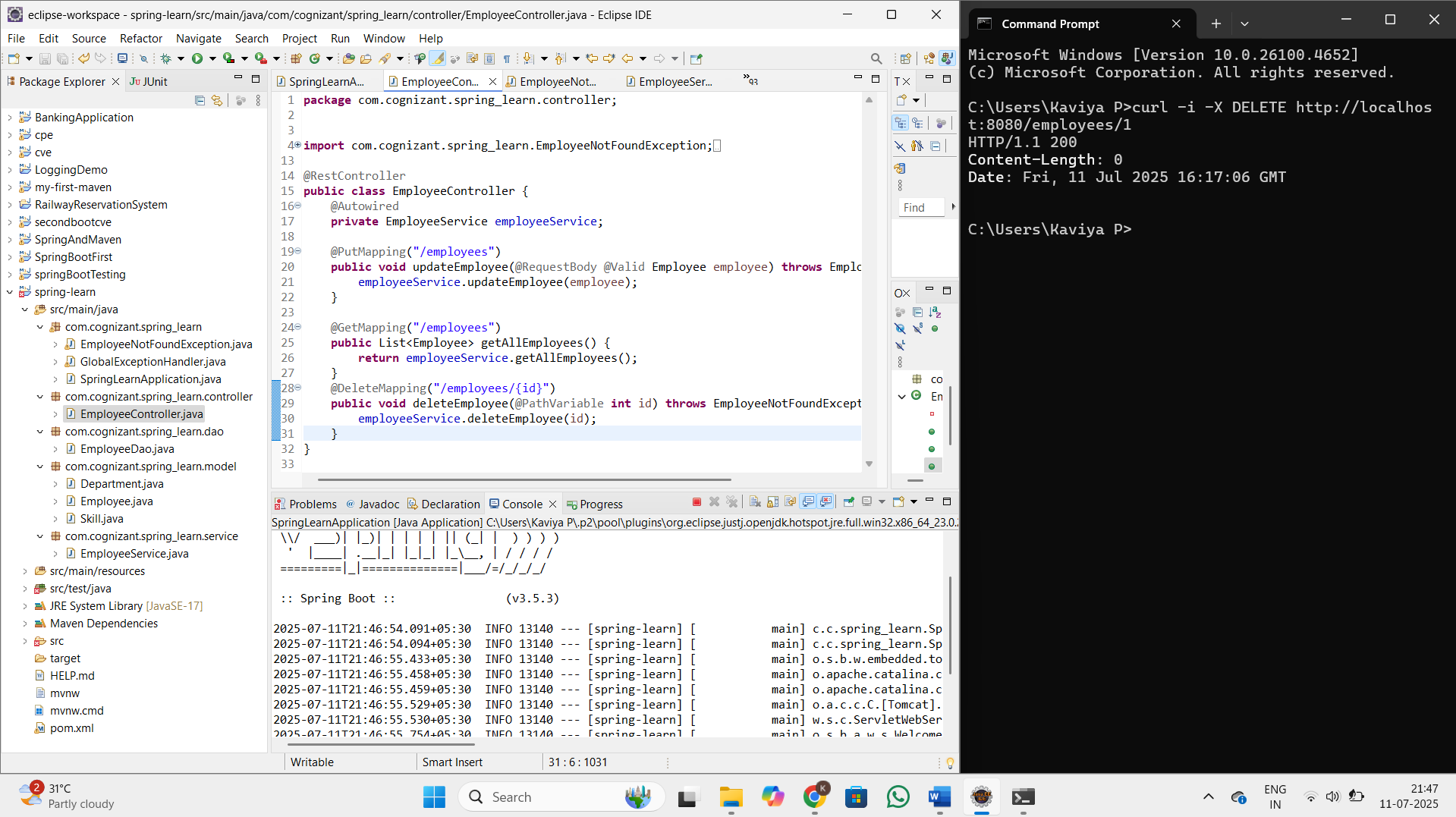
**public** **void** deleteEmployee(**int** id) **throws** EmployeeNotFoundException {

employeeDao.deleteEmployee(id);

}

}

**OUTPUT**

****