**EmployeeNotFoundException.java**

package com.cognizant.springlearn.service.exception;

import org.springframework.http.HttpStatus;

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

@ResponseStatus(value = HttpStatus.NOT\_FOUND, reason = "Employee Not Found")

public class EmployeeNotFoundException extends Exception {

public EmployeeNotFoundException(String message) {

super(message);

}

}

EmployeeDao.java:

package com.cognizant.springlearn.dao;

import com.cognizant.springlearn.Employee;

import com.cognizant.springlearn.service.exception.EmployeeNotFoundException;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Repository;

import java.util.ArrayList;

import java.util.List;

@Repository

public class EmployeeDao {

public static List<Employee> EMPLOYEE\_LIST;

public EmployeeDao() {

ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");

EMPLOYEE\_LIST = (ArrayList<Employee>) context.getBean("employeeList");

}

public List<Employee> getAllEmployees() {

return EMPLOYEE\_LIST;

}

public void updateEmployee(Employee employee) throws EmployeeNotFoundException {

boolean found = false;

for (int i = 0; i < EMPLOYEE\_LIST.size(); i++) {

if (EMPLOYEE\_LIST.get(i).getId().equals(employee.getId())) {

EMPLOYEE\_LIST.set(i, employee);

found = true;

break;

}

}

if (!found) {

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

}

}

public void deleteEmployee(int id) throws EmployeeNotFoundException {

boolean removed = EMPLOYEE\_LIST.removeIf(emp -> emp.getId() == id);

if (!removed) {

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

}

}

}

EmployeeService.java:

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.Employee;

import com.cognizant.springlearn.dao.EmployeeDao;

import com.cognizant.springlearn.service.exception.EmployeeNotFoundException;

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

import org.springframework.stereotype.Service;

import javax.transaction.Transactional;

import java.util.List;

@Service

public class EmployeeService {

@Autowired

private EmployeeDao employeeDao;

public List<Employee> getAllEmployees() {

return employeeDao.getAllEmployees();

}

@Transactional

public void updateEmployee(Employee employee) throws EmployeeNotFoundException {

employeeDao.updateEmployee(employee);

}

@Transactional

public void deleteEmployee(int id) throws EmployeeNotFoundException {

employeeDao.deleteEmployee(id);

}

}

EmployeeController.java:

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.Employee;

import com.cognizant.springlearn.service.EmployeeService;

import com.cognizant.springlearn.service.exception.EmployeeNotFoundException;

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

import org.springframework.http.ResponseEntity;

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

import javax.validation.Valid;

import java.util.List;

@RestController

@RequestMapping("/employees")

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@GetMapping

public List<Employee> getAllEmployees() {

return employeeService.getAllEmployees();

}

@PutMapping

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

employeeService.updateEmployee(employee);

}

@DeleteMapping("/{id}")

public ResponseEntity<String> deleteEmployee(@PathVariable int id) throws EmployeeNotFoundException {

employeeService.deleteEmployee(id);

return ResponseEntity.ok("Employee with ID " + id + " deleted successfully.");

}

}

employee.xml:

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="employee1" class="com.cognizant.springlearn.Employee">

<property name="id" value="1"/>

<property name="name" value="John Doe"/>

<property name="salary" value="50000"/>

<property name="permanent" value="true"/>

<property name="dateOfBirth" value="15/08/1990"/>

<property name="department">

<bean class="com.cognizant.springlearn.Department">

<property name="id" value="1"/>

<property name="name" value="HR"/>

</bean>

</property>

<property name="skillList">

<list>

<bean class="com.cognizant.springlearn.Skill">

<property name="id" value="1"/>

<property name="name" value="Java"/>

</bean>

<bean class="com.cognizant.springlearn.Skill">

<property name="id" value="2"/>

<property name="name" value="Spring"/>

</bean>

</list>

</property>

</bean>

<bean id="employeeList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="employee1"/>

</list>

</constructor-arg>

</bean>

</beans>