## Folder Structure

A screenshot of a computer program

AI-generated content may be incorrect.

## Employee.java

package com.example.CRUD\_application.model;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Entity //specifies that the class is an entity and is mapped to a database table

@AllArgsConstructor //generates a constructor with 1 parameter for each field in your class

@NoArgsConstructor //generates a no-argument constructor

@Data //generates getters and setters for all fields

@Table(name = "employees")

public class Employee {

    @Id //primary key

    @GeneratedValue(strategy = GenerationType.IDENTITY) //auto-increment

    private Long id;

    @Column(name = "first\_name", nullable = false) //column name in the database

    private String firstName;

    @Column(name = "last\_name", nullable = false) //column name in the database

    private String lastName;

    @Column(name = "email", nullable = false, unique = true) //column name in the database

    private String email;

}

## EmployeeRepository.java

package com.example.CRUD\_application.repository;

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

import com.example.CRUD\_application.model.Employee;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

    // crud methods are provided by JpaRepository

}

## EmployeeService.java

package com.example.CRUD\_application.service;

import java.util.List;

import com.example.CRUD\_application.model.Employee;

public interface EmployeeService {

    Employee saveEmployee (Employee employee); // create

    List<Employee> getAllEmployees(); // read all

    Employee getEmployeeById(Long id); // read by id

    Employee updateEmployee(Employee employee, Long id); // update

    void deleteEmployee(Long id); // delete

}

## EmployeeServiceImpl.java

package com.example.CRUD\_application.service.impl;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.example.CRUD\_application.model.Employee;

import com.example.CRUD\_application.repository.EmployeeRepository;

import com.example.CRUD\_application.service.EmployeeService;

@Service

public class EmployeeServiceImpl implements EmployeeService {

   @Autowired

   private EmployeeRepository employeeRepository;

    @Override

    public Employee saveEmployee(Employee employee) {

        return employeeRepository.save(employee);

    }

    @Override

    public List<Employee> getAllEmployees() {

        return employeeRepository.findAll();

    }

    @Override

    public Employee getEmployeeById(Long id) {

        Optional<Employee> employee = employeeRepository.findById(id);

        if(employee.isPresent()) {

            return employee.get();

        } else {

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

        }

    }

    @Override

    public Employee updateEmployee(Employee employee, Long id) {

        Employee exitingEmployee = employeeRepository.findById(id).orElseThrow(

            () -> new RuntimeException("Employee not found with id: " + id)

        );

        exitingEmployee.setFirstName(employee.getFirstName());

        exitingEmployee.setLastName(employee.getLastName());

        exitingEmployee.setEmail(employee.getEmail());

        employeeRepository.save(exitingEmployee);

        return exitingEmployee;

    }

    @Override

    public void deleteEmployee(Long id) {

        employeeRepository.findById(id).orElseThrow(

            () -> new RuntimeException("Employee not found with id: " + id)

        );

        employeeRepository.deleteById(id);

    }

}

## EmployeeController.java

package com.example.CRUD\_application.controller;

import java.util.List;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

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

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

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

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

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

import com.example.CRUD\_application.model.Employee;

import com.example.CRUD\_application.service.EmployeeService;

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

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

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

@RestController

@RequestMapping("/api/employees")

@CrossOrigin(origins = "\*")

public class EmployeeController {

    @Autowired

    private EmployeeService employeeService;

    @PostMapping

    public ResponseEntity<Employee> saveEmployee(@RequestBody Employee employee) {

        return new ResponseEntity<Employee>(employeeService.saveEmployee(employee), HttpStatus.CREATED);

    }

    @GetMapping

    public List<Employee> getAllEmployees() {

        return employeeService.getAllEmployees();

    }

    @GetMapping("{id}")

    public ResponseEntity<Employee> getEmployeeById(@PathVariable("id") Long employeeID) {

    return new ResponseEntity<Employee>(employeeService.getEmployeeById(employeeID), HttpStatus.OK);

    }

    @PutMapping("{id}")

    public ResponseEntity<Employee> updateEmployee(@PathVariable("id") Long id, @RequestBody Employee employee) {

        return new ResponseEntity<Employee>(employeeService.updateEmployee(employee, id), HttpStatus.OK);

    }

    @DeleteMapping("{id}")

    public ResponseEntity<String> deleteEmployee(@PathVariable("id") Long id) {

        employeeService.deleteEmployee(id);

        return new ResponseEntity<String>("Employee deleted successfully!.", HttpStatus.OK);

    }

}

## Postman

## A screenshot of a computer AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## A screenshot of a computer AI-generated content may be incorrect.

## A screenshot of a computer AI-generated content may be incorrect.

## A screenshot of a computer AI-generated content may be incorrect.

## 

## Frontend

A screenshot of a computer

AI-generated content may be incorrect.