**Employee.java**

@Entity

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String firstName;

private String lastName;

private Department department;

// getters and setters

}

**Department.java**

@Entity

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String firstName;

private String lastName;

private Department department;

// getters and setters

}

**Employeerepository.java**

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

List<Employee> findByDepartment(Department department);

List<Employee> findByFirstName(String firstName);

List<Employee> findByLastName(String lastName);

}

**DepartmentRepository.java:**

public interface DepartmentRepository extends JpaRepository<Department, Long> {

List<Department> findByName(String name);

List<Department> findByLocation(String location);

}

**EmployeeService.java:**

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

public List<Employee> getAllEmployees() {

return employeeRepository.findAll();

}

public Employee getEmployee(Long id) {

return employeeRepository.findById(id).orElseThrow();

}

public Employee createEmployee(Employee employee) {

return employeeRepository.save(employee);

}

public Employee updateEmployee(Employee employee) {

return employeeRepository.save(employee);

}

public void deleteEmployee(Long id) {

employeeRepository.deleteById(id);

}

}

**DepartmentService.java:**

@Service

public class DepartmentService {

@Autowired

private DepartmentRepository departmentRepository;

public List<Department> getAllDepartments() {

return departmentRepository.findAll();

}

public Department getDepartment(Long id) {

return departmentRepository.findById(id).orElseThrow();

}

public Department createDepartment(Department department) {

return departmentRepository.save(department);

}

public Department updateDepartment(Department department) {

return departmentRepository.save(department);

}

public void deleteDepartment(Long id) {

departmentRepository.deleteById(id);

}

}

**EmployeeController.java:**

@RestController

@RequestMapping("/api/employees")

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@GetMapping

public List<Employee> getAllEmployees() {

return employeeService.getAllEmployees();

}

@GetMapping("/{id}")

public Employee getEmployee(@PathVariable Long id) {

return employeeService.getEmployee(id);

}

@PostMapping

public Employee createEmployee(@RequestBody Employee employee) {

return employeeService.createEmployee(employee);

}

@PutMapping("/{id}")

public Employee updateEmployee(@PathVariable Long id, @RequestBody Employee employee) {

return employeeService.updateEmployee(employee);

}

@DeleteMapping("/{id}")

public void deleteEmployee(@PathVariable Long id) {

employeeService.deleteEmployee(id);

}

}

**DepartmentController.java:**

@RestController

@RequestMapping("/api/departments")

public class DepartmentController {

@Autowired

private DepartmentService departmentService;

@GetMapping

public List<Department> getAllDepartments() {

return departmentService.getAllDepartments();

}

@GetMapping("/{id}")

public Department getDepartment(@PathVariable Long id) {

return departmentService.getDepartment(id);

}

@PostMapping

public Department createDepartment(@RequestBody Department department) {

return departmentService.createDepartment(department);

}

@PutMapping("/{id}")

public Department updateDepartment(@PathVariable Long id, @RequestBody Department department) {

return departmentService.updateDepartment(department);

}

@DeleteMapping("/{id}")

public void deleteDepartment(@PathVariable Long id) {

departmentService.deleteDepartment(id);

}

}