DemoApplication.java

```
package com.example.demo;

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

@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
       SpringApplication.run(DemoApplication.class, args);
    }
}
```

Employee.java

```
package com.example.demo;

import javax.persistence.*;

@Entity
public class Employee {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;
    private String department;
    private Double salary;

    // Getters and Setters
    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getDepartment() {
        return department;
    }

    public void setDepartment(String department) {
        this.department = department;
    }

    public Double getSalary() {
        return salary;
    }

    public void setSalary(Double salary) {
        this.salary = salary;
    }
}
```

EmployeeController.java

```
package com.example.demo;

import com.example.demo.Employee;
import com.example.demo.EmployeeService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/api/employees")
public class EmployeeController {

    @Autowired
    private EmployeeService employeeService;

    @PostMapping
    public Employee addEmployee(@RequestBody Employee
employee) {
        return employeeService.addEmployee(employee);
    }

    @PutMapping("/{id}")
    public ResponseEntity<Employee>
updateEmployee(@PathVariable Long id, @RequestBody Employee
employeeDetails) {
        return
ResponseEntity.ok(employeeService.updateEmployee(id,
employeeDetails));
    }
    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteEmployee(@PathVariable 
Long id) {
        employeeService.deleteEmployee(id);
        return ResponseEntity.noContent().build();
    }

    @GetMapping
    public List<Employee> getAllEmployees() {
        return employeeService.getAllEmployees();
    }

    @GetMapping("/{id}")
    public ResponseEntity<Employee> 
getEmployeeById(@PathVariable Long id) {
        return employeeService.getEmployeeById(id)
                .map(ResponseEntity::ok)
                .orElse(ResponseEntity.notFound().build());
    }
}
```

EmployeeRepository.java

```
package com.example.demo;

import com.example.demo.Employee;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface EmployeeRepository extends 
JpaRepository<Employee, Long> {
}
```

EmployeeService.java

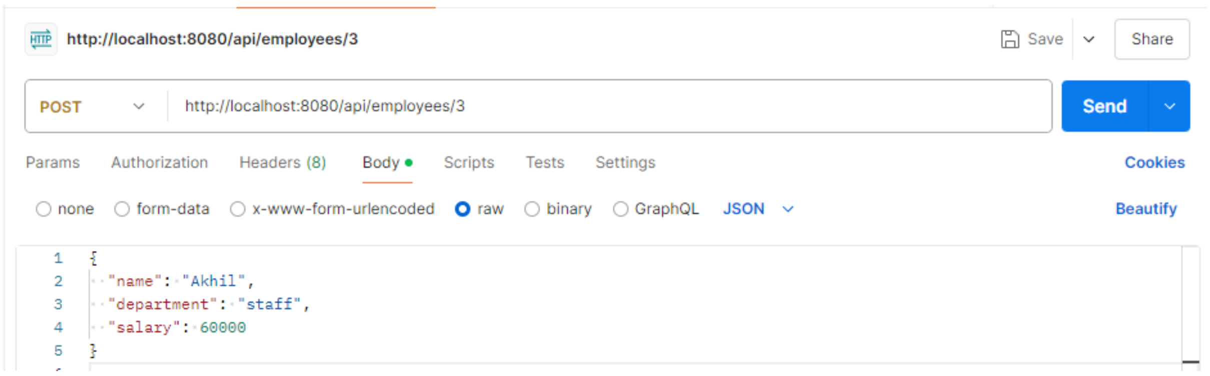
```
package com.example.demo;

import com.example.demo.Employee;
import com.example.demo.EmployeeRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.Optional;

@Service
public class EmployeeService {
    @Autowired
    private EmployeeRepository employeeRepository;

    public Employee addEmployee(Employee employee) {
        return employeeRepository.save(employee);
    }

    public Employee updateEmployee(Long id, Employee 
employeeDetails) {
        Employee employee = 
employeeRepository.findById(id).orElseThrow(() -> new 
RuntimeException("Employee not found"));
        employee.setName(employeeDetails.getName());

employee.setDepartment(employeeDetails.getDepartment());
        employee.setSalary(employeeDetails.getSalary());
        return employeeRepository.save(employee);
    }

    public void deleteEmployee(Long id) {
        Employee employee = 
employeeRepository.findById(id).orElseThrow(() -> new 
RuntimeException("Employee not found"));
        employeeRepository.delete(employee);
    }

    public List<Employee> getAllEmployees() {
        return employeeRepository.findAll();
    }

    public Optional<Employee> getEmployeeById(Long id) {
        return employeeRepository.findById(id);
    }
}
```

GlobalExceptionHandler.java

```
package com.example.demo;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import 
org.springframework.web.bind.annotation.ControllerAdvice;
import 
org.springframework.web.bind.annotation.ExceptionHandler;

@ControllerAdvice
public class GlobalExceptionHandler {

    @ExceptionHandler(RuntimeException.class)
    public ResponseEntity<String>
handleRuntimeException(RuntimeException ex) {
        return new ResponseEntity<>(ex.getMessage(),
HttpStatus.NOT_FOUND);
    }
}
```

OUTPUT

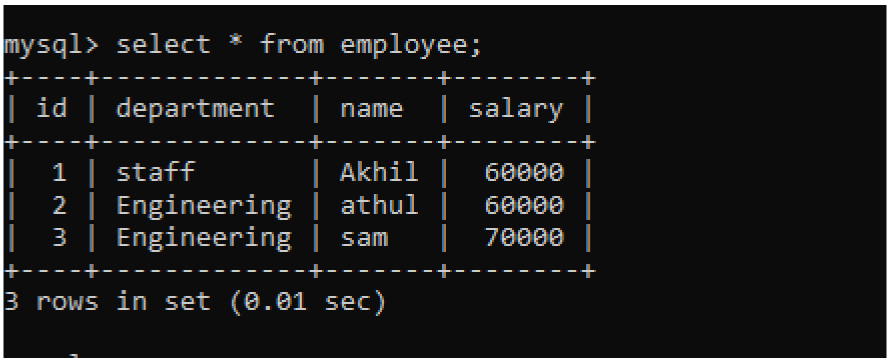
CREATE (POST)



POST http://localhost:8080/api/employees/3

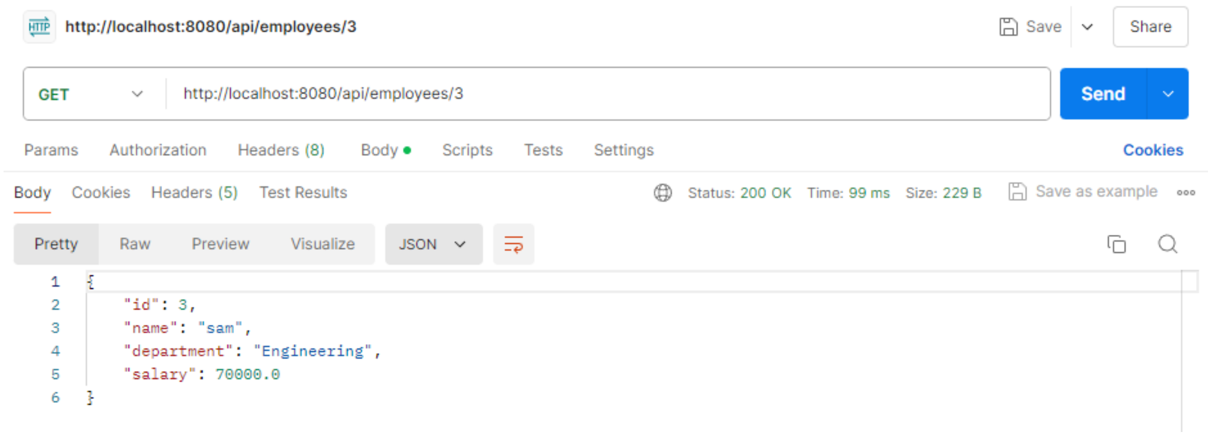
Body: [✓ raw] JSON

```
{
  "name": "Akhil",
  "department": "staff",
  "salary": 60000
}
```



```
mysql> select * from employee;
+----+-------------+-------+--------+
| id | department  | name  | salary |
+----+-------------+-------+--------+
|  1 | staff       | Akhil |  60000 |
|  2 | Engineering | athul |  60000 |
|  3 | Engineering | sam   |  70000 |
+----+-------------+-------+--------+
3 rows in set (0.01 sec)
```

READ (GET)

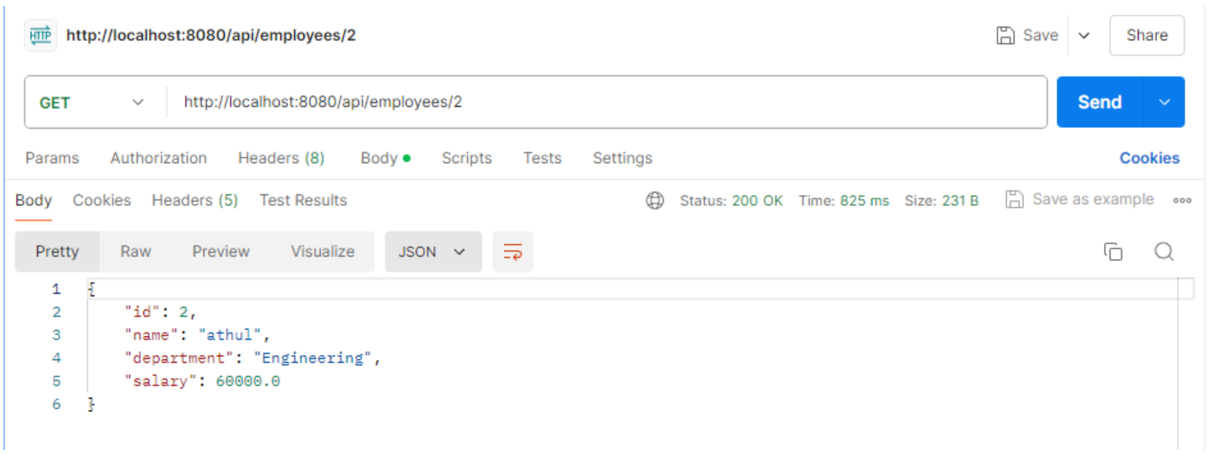


http://localhost:8080/api/employees/3

GET http://localhost:8080/api/employees/3

Status: 200 OK Time: 99 ms Size: 229 B

```
{
    "id": 3,
    "name": "sam",
    "department": "Engineering",
    "salary": 70000.0
}
```



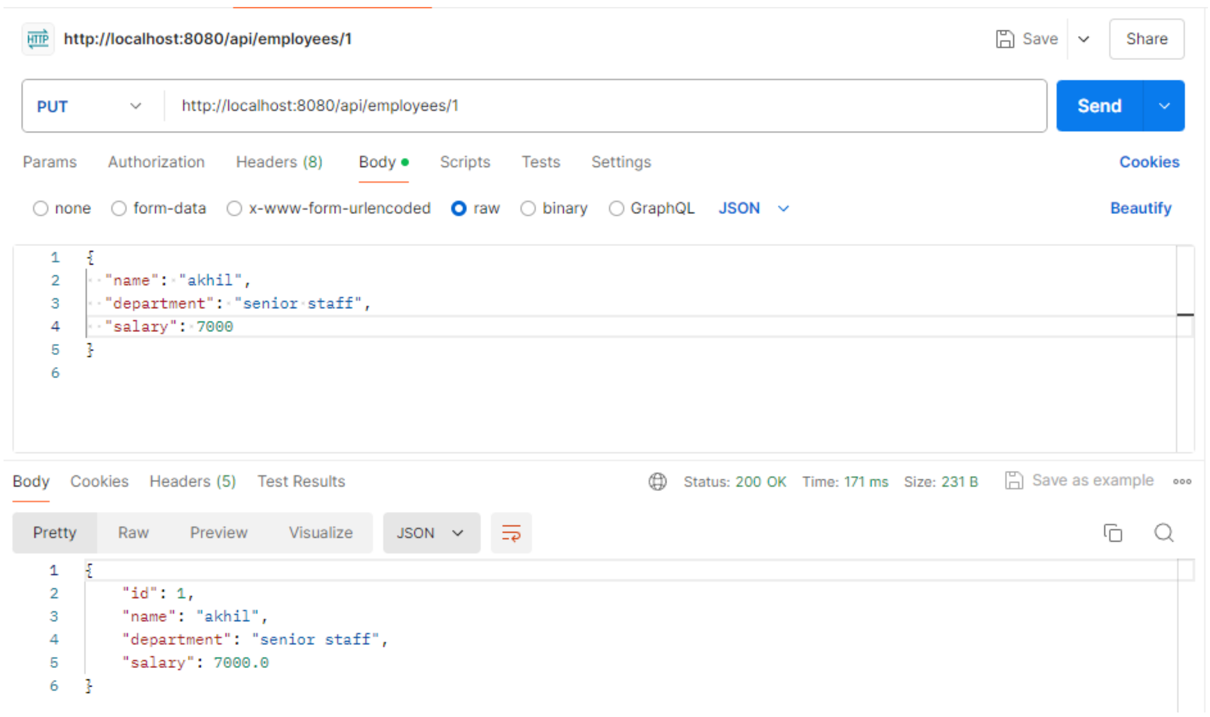
http://localhost:8080/api/employees/2

GET http://localhost:8080/api/employees/2

Status: 200 OK Time: 825 ms Size: 231 B

```
{
    "id": 2,
    "name": "athul",
    "department": "Engineering",
    "salary": 60000.0
}
```

UPDATE (PUT)



http://localhost:8080/api/employees/1

Save | Share

PUT | http://localhost:8080/api/employees/1 | Send

Params | Authorization | Headers (8) | Body | Scripts | Tests | Settings | Cookies

none | form-data | x-www-form-urlencoded | [✓ raw] | binary | GraphQL | JSON | Beautify

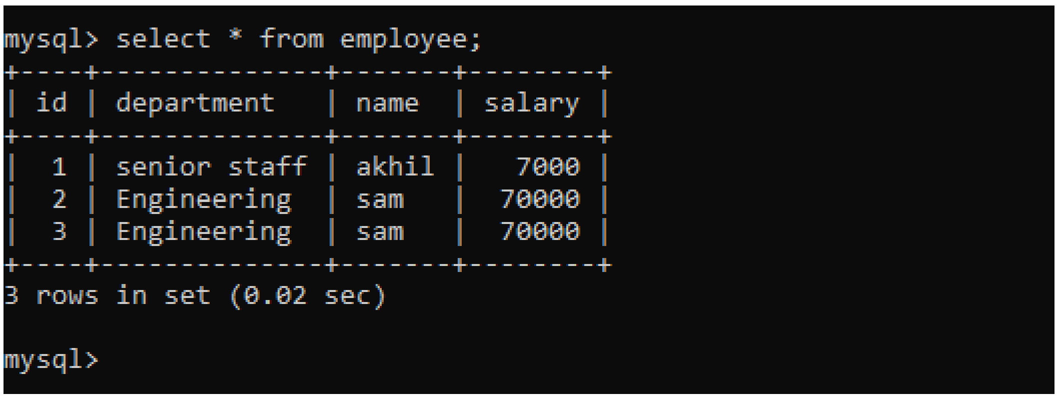
```
{
  "name": "akhil",
  "department": "senior staff",
  "salary": 7000
}
```

Body | Cookies | Headers (5) | Test Results

Status: 200 OK Time: 171 ms Size: 231 B | Save as example

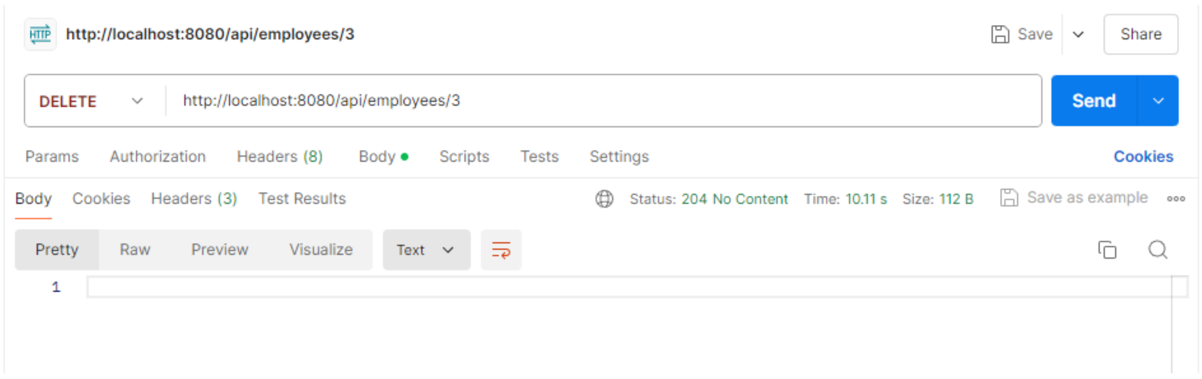
Pretty | Raw | Preview | Visualize | JSON

```
{
    "id": 1,
    "name": "akhil",
    "department": "senior staff",
    "salary": 7000.0
}
```



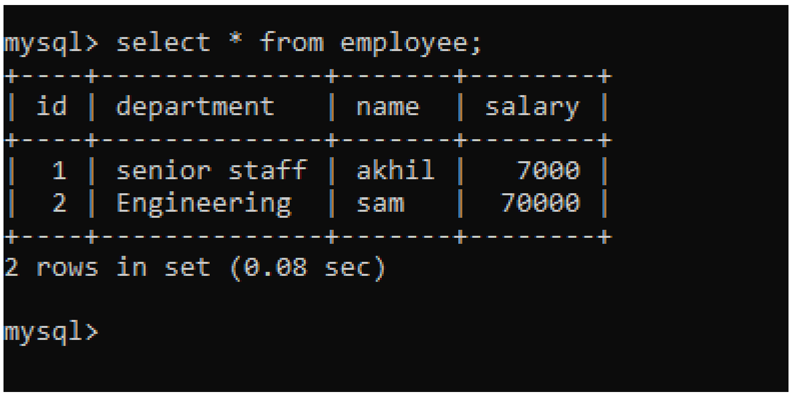
```
mysql> select * from employee;
+----+--------------+-------+--------+
| id | department   | name  | salary |
+----+--------------+-------+--------+
|  1 | senior staff | akhil |   7000 |
|  2 | Engineering  | sam   |  70000 |
|  3 | Engineering  | sam   |  70000 |
+----+--------------+-------+--------+
3 rows in set (0.02 sec)

mysql>
```

DELETE (DELETE)



DELETE http://localhost:8080/api/employees/3

Status: 204 No Content Time: 10.11 s Size: 112 B



```
mysql> select * from employee;
+----+--------------+-------+--------+
| id | department   | name  | salary |
+----+--------------+-------+--------+
|  1 | senior staff | akhil |   7000 |
|  2 | Engineering  | sam   |  70000 |
+----+--------------+-------+--------+
2 rows in set (0.08 sec)

mysql>
```