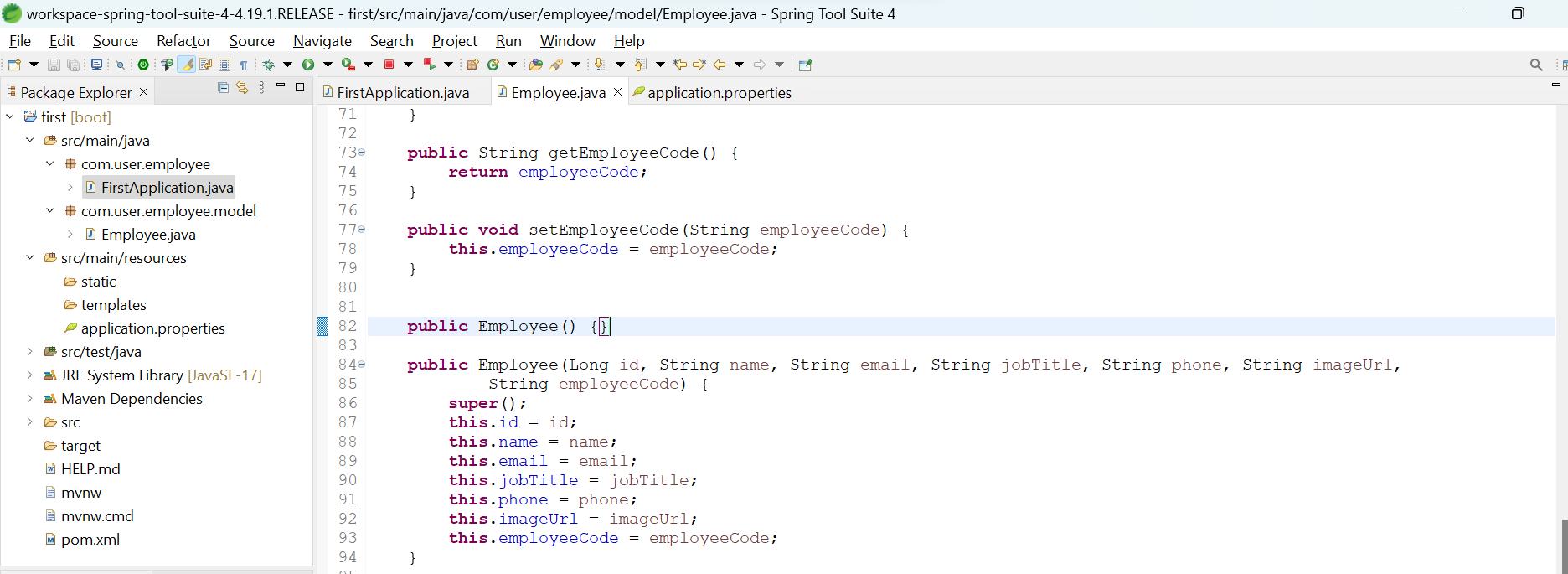
**Choose web,mysql driver, jdbc jpa**

**Controller -> Service -> Repository -> Data**



**FirstApplication.java:**

package com.user.employee;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class FirstApplication {

public static void main(String[] args) {

SpringApplication.run(FirstApplication.class, args);

}

}

**Employee.java**

**package** com.user.employee.model;

**import** jakarta.persistence.Column;

**import** jakarta.persistence.Entity;

**import** jakarta.persistence.GeneratedValue;

**import** jakarta.persistence.GenerationType;

**import** jakarta.persistence.Id;

@Entity

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

@Id

@GeneratedValue(strategy = GenerationType.***AUTO***)

@Column(nullable = **false**, updatable = **false**)

**private** Long id;

**private** String name;

**private** String email;

**private** String jobTitle;

**private** String phone;

**private** String imageUrl;

@Column(nullable = **false**, updatable = **false**)

**private** String employeeCode;

**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 getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getJobTitle() {

**return** jobTitle;

}

**public** **void** setJobTitle(String jobTitle) {

**this**.jobTitle = jobTitle;

}

**public** String getPhone() {

**return** phone;

}

**public** **void** setPhone(String phone) {

**this**.phone = phone;

}

**public** String getImageUrl() {

**return** imageUrl;

}

**public** **void** setImageUrl(String imageUrl) {

**this**.imageUrl = imageUrl;

}

**public** String getEmployeeCode() {

**return** employeeCode;

}

**public** **void** setEmployeeCode(String employeeCode) {

**this**.employeeCode = employeeCode;

}

**public** Employee() {}

**public** Employee(Long id, String name, String email, String jobTitle, String phone, String imageUrl,

String employeeCode) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.email = email;

**this**.jobTitle = jobTitle;

**this**.phone = phone;

**this**.imageUrl = imageUrl;

**this**.employeeCode = employeeCode;

}

@Override

**public** String toString() {

**return** "Employee [id=" + id + ", name=" + name + ", email=" + email + ", jobTitle=" + jobTitle + ", phone="

+ phone + ", imageUrl=" + imageUrl + ", employeeCode=" + employeeCode + "]";

}

}

**Application.properties:**

#MYSQL Configuration

spring.datasource.url=jdbc:mysql://localhost:3306/employeemanager

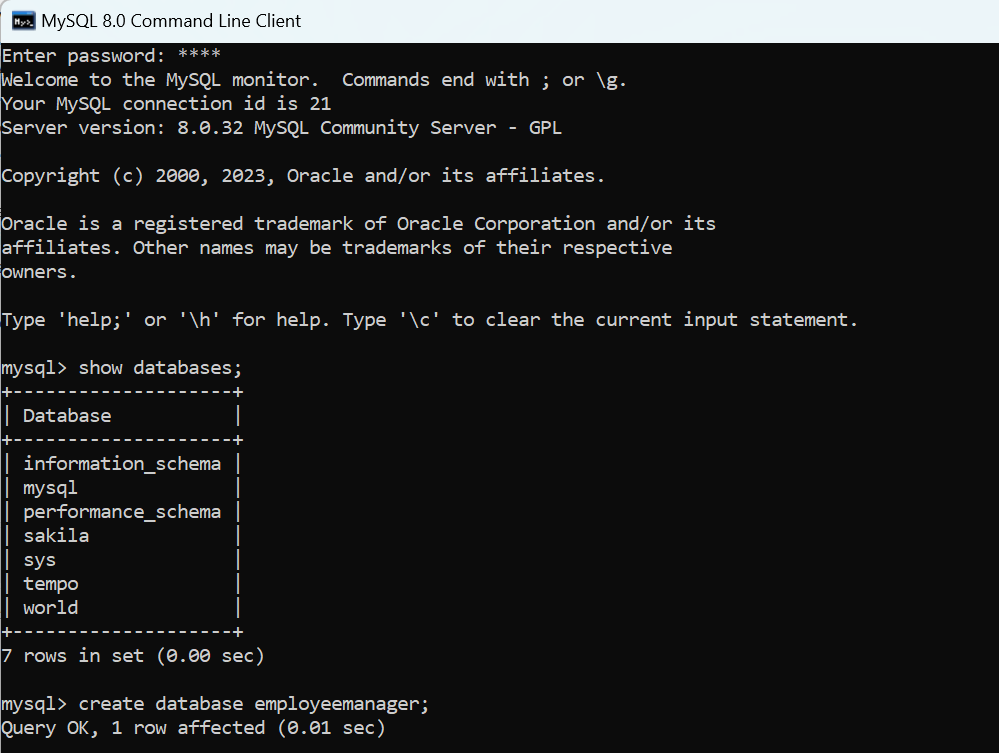
spring.datasource.username=root

spring.datasource.password=1234

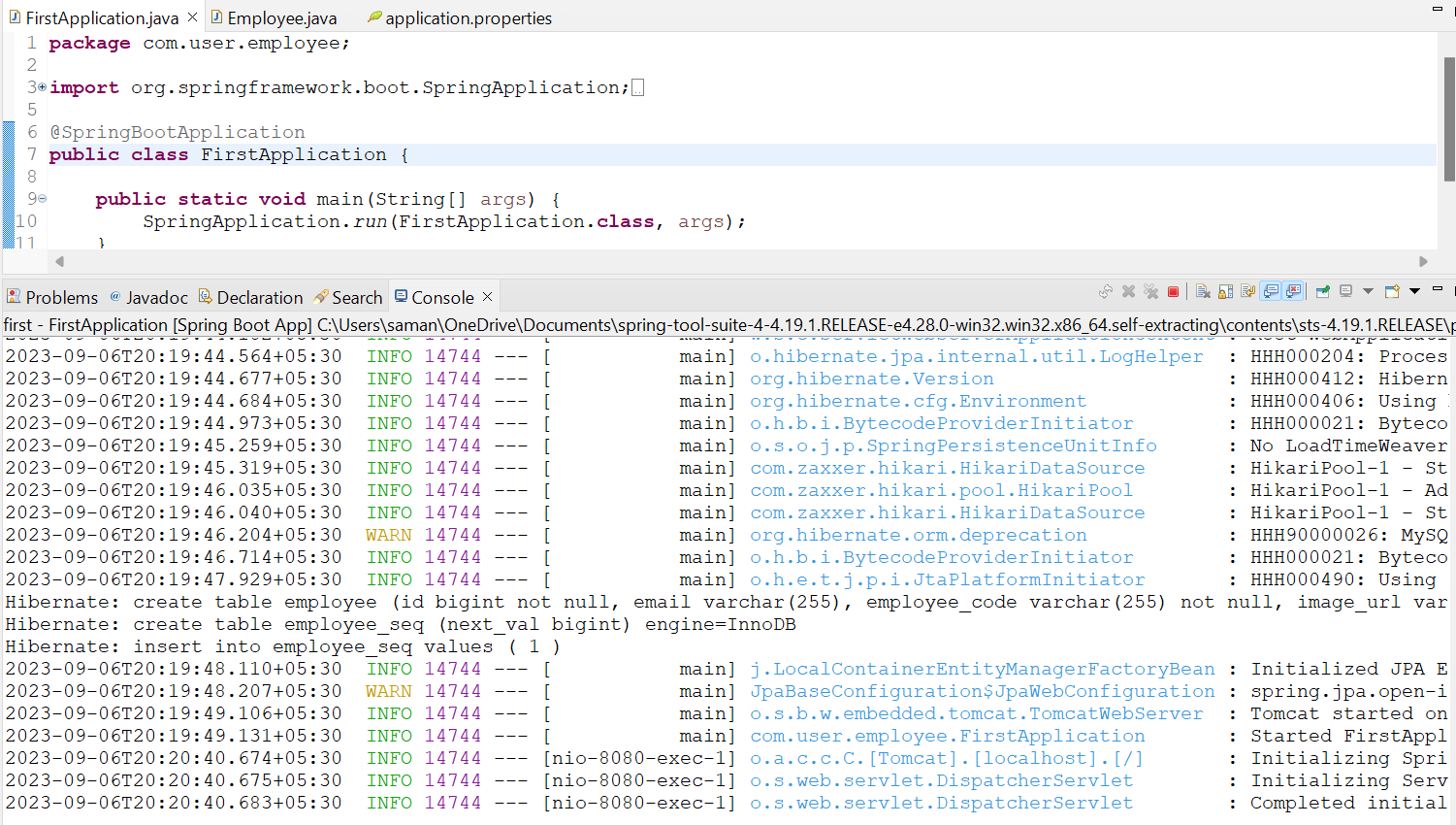
spring.jpa.show-sql=true

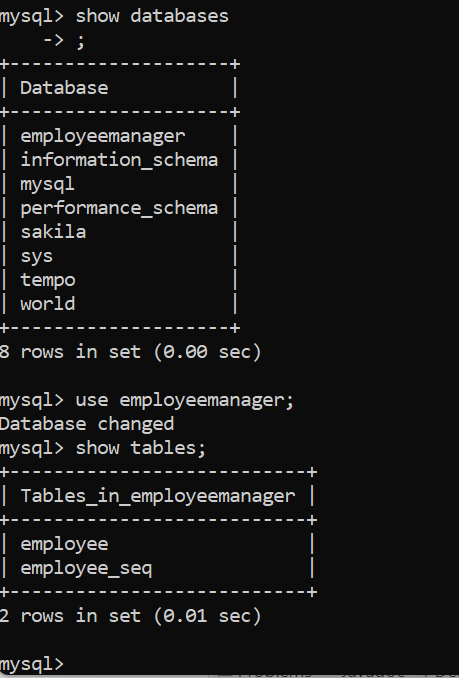
spring.jpa.hibernate.ddl-auto=update

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

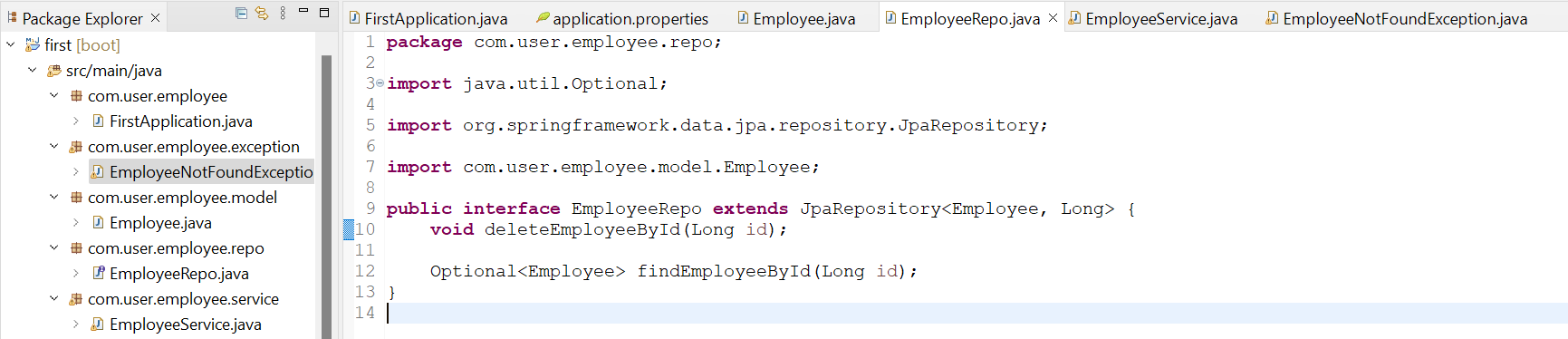


**Run FirstApplication.java**

****



**2ND STAGE:**



**EmployeeRepo.java**

**package** com.user.employee.repo;

**import** java.util.Optional;

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

**import** com.user.employee.model.Employee;

**public** **interface** EmployeeRepo **extends** JpaRepository<Employee, Long> {

**void** deleteEmployeeById(Long id);

Optional<Employee> findEmployeeById(Long id);

}

**EmployeeService.java**

**package** com.user.employee.service;

**import** java.nio.file.attribute.UserPrincipalNotFoundException;

**import** java.util.List;

**import** java.util.UUID;

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

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

**import** com.user.employee.exception.EmployeeNotFoundException;

**import** com.user.employee.model.Employee;

**import** com.user.employee.repo.EmployeeRepo;

@Service

**public** **class** EmployeeService {

**private** **final** EmployeeRepo employeeRepo;

@Autowired

**public** EmployeeService(EmployeeRepo employeeRepo) {

**this**.employeeRepo = employeeRepo;

}

**public** Employee addEmployee(Employee employee) {

employee.setEmployeeCode(UUID.*randomUUID*().toString());

**return** employeeRepo.save(employee);

}

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

**return** employeeRepo.findAll();

}

**public** Employee findEmployeeById(Long id) {

**return** employeeRepo.findEmployeeById(id)

.orElseThrow(() -> **new** EmployeeNotFoundException("Employee with id "+id+" not found."));

}

**public** Employee updateEmployee(Employee employee) {

**return** employeeRepo.save(employee);

}

**public** **void** deleteEmployee(Long id) {

employeeRepo.deleteEmployeeById(id);

}

}

**EmployeeNotFoundException:**

**package** com.user.employee.exception;

**public** **class** EmployeeNotFoundException **extends** RuntimeException{

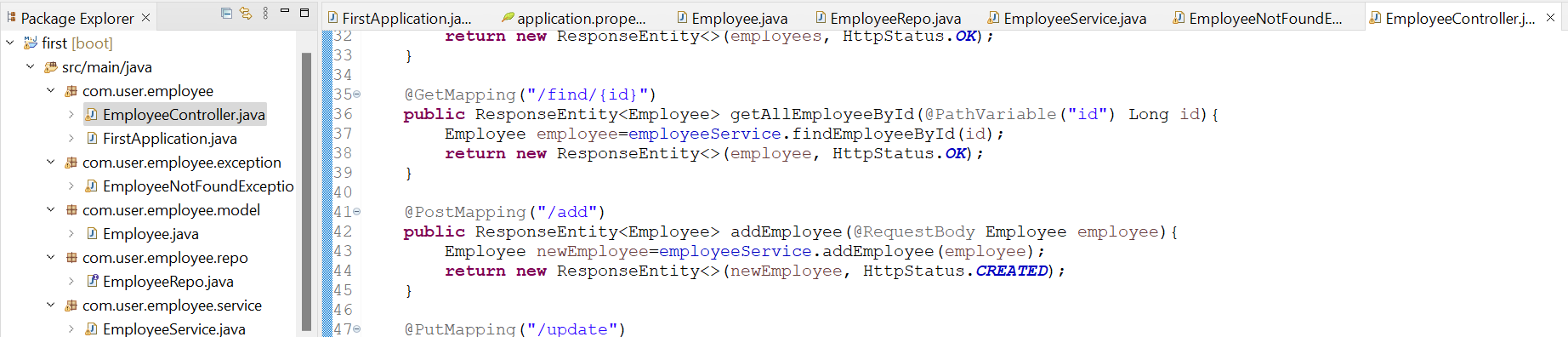
**public** EmployeeNotFoundException(String msg) {

**super**(msg);

}

}

**3rd Stage:**



**EmployeeController.java**

**package** com.user.employee;

**import** java.util.List;

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

**import** org.springframework.http.ResponseEntity;

**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.PostMapping;

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

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

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

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

**import** com.user.employee.model.Employee;

**import** com.user.employee.service.EmployeeService;

@RestController

@RequestMapping("/employee")

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

**private** **final** EmployeeService employeeService;

**public** EmployeeController(EmployeeService employeeService) {

**this**.employeeService = employeeService;

}

@GetMapping("/all")

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

List<Employee> employees=employeeService.findAllEmployees();

**return** **new** ResponseEntity<>(employees, HttpStatus.***OK***);

}

@GetMapping("/find/{id}")

**public** ResponseEntity<Employee> getAllEmployeeById(@PathVariable("id") Long id){

Employee employee=employeeService.findEmployeeById(id);

**return** **new** ResponseEntity<>(employee, HttpStatus.***OK***);

}

@PostMapping("/add")

**public** ResponseEntity<Employee> addEmployee(@RequestBody Employee employee){

Employee newEmployee=employeeService.addEmployee(employee);

**return** **new** ResponseEntity<>(newEmployee, HttpStatus.***CREATED***);

}

@PutMapping("/update")

**public** ResponseEntity<Employee> updateEmployee(@RequestBody Employee employee){

Employee updateEmployee=employeeService.updateEmployee(employee);

**return** **new** ResponseEntity<>(updateEmployee, HttpStatus.***OK***);

}

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

**public** ResponseEntity<?> deleteEmployee(@PathVariable Long id){

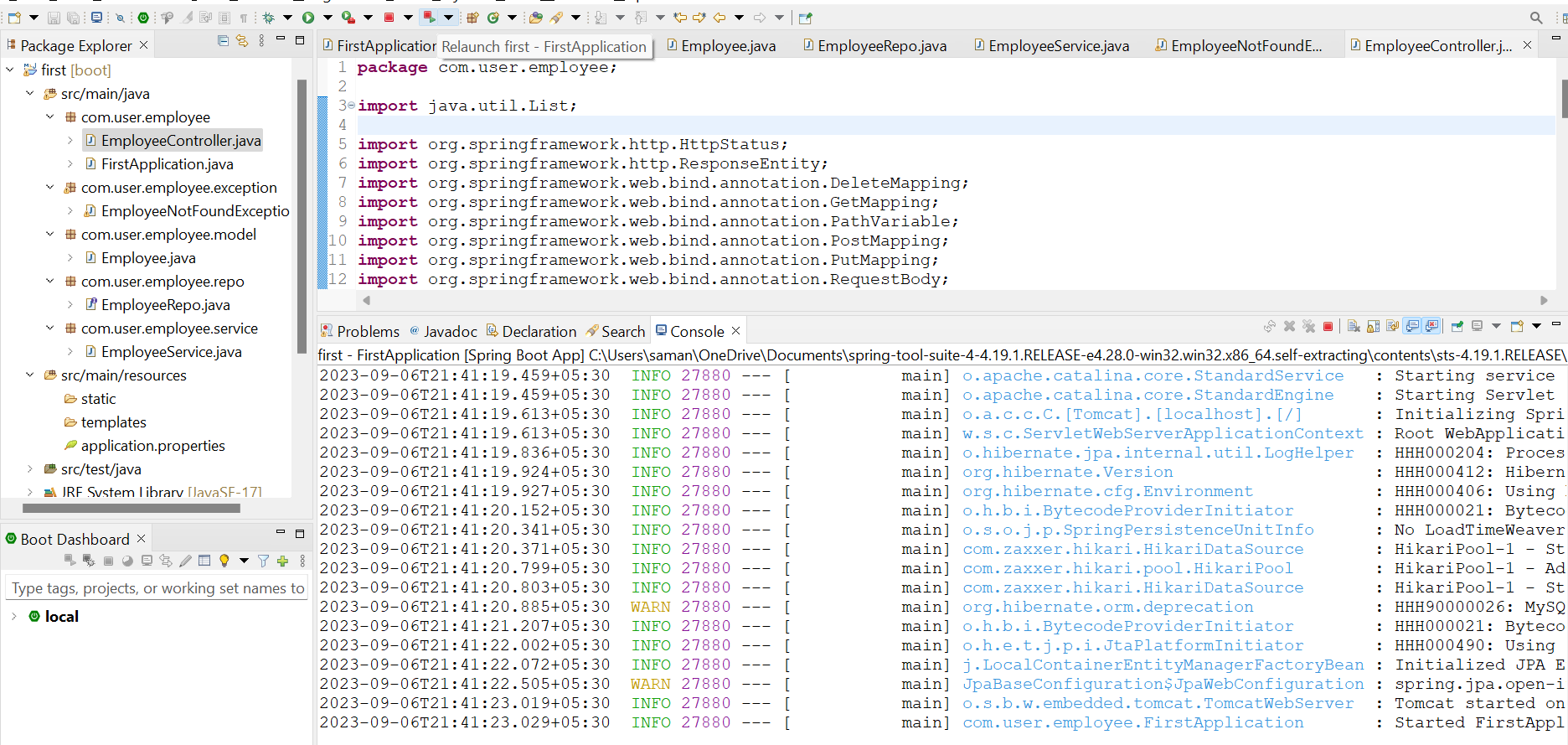
employeeService.deleteEmployee(id);

**return** **new** ResponseEntity<>(HttpStatus.***OK***);

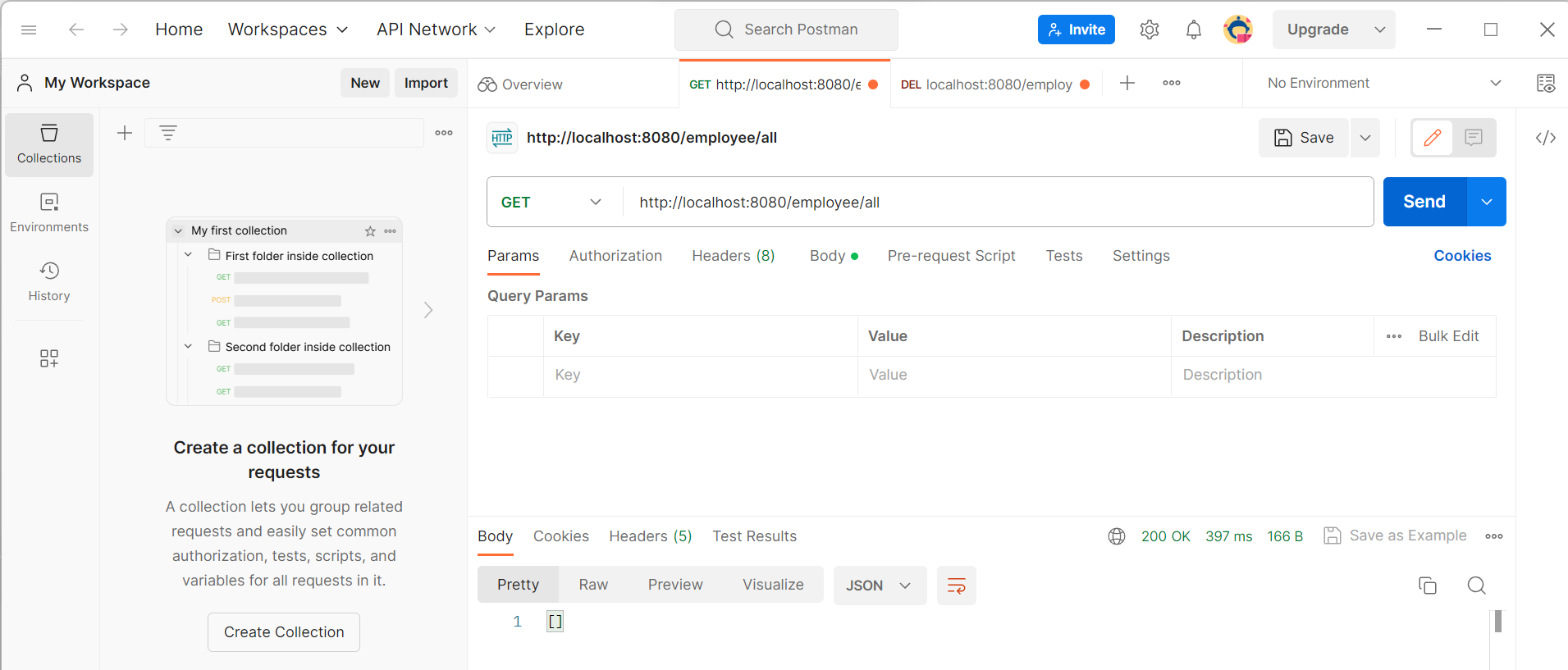
}

}

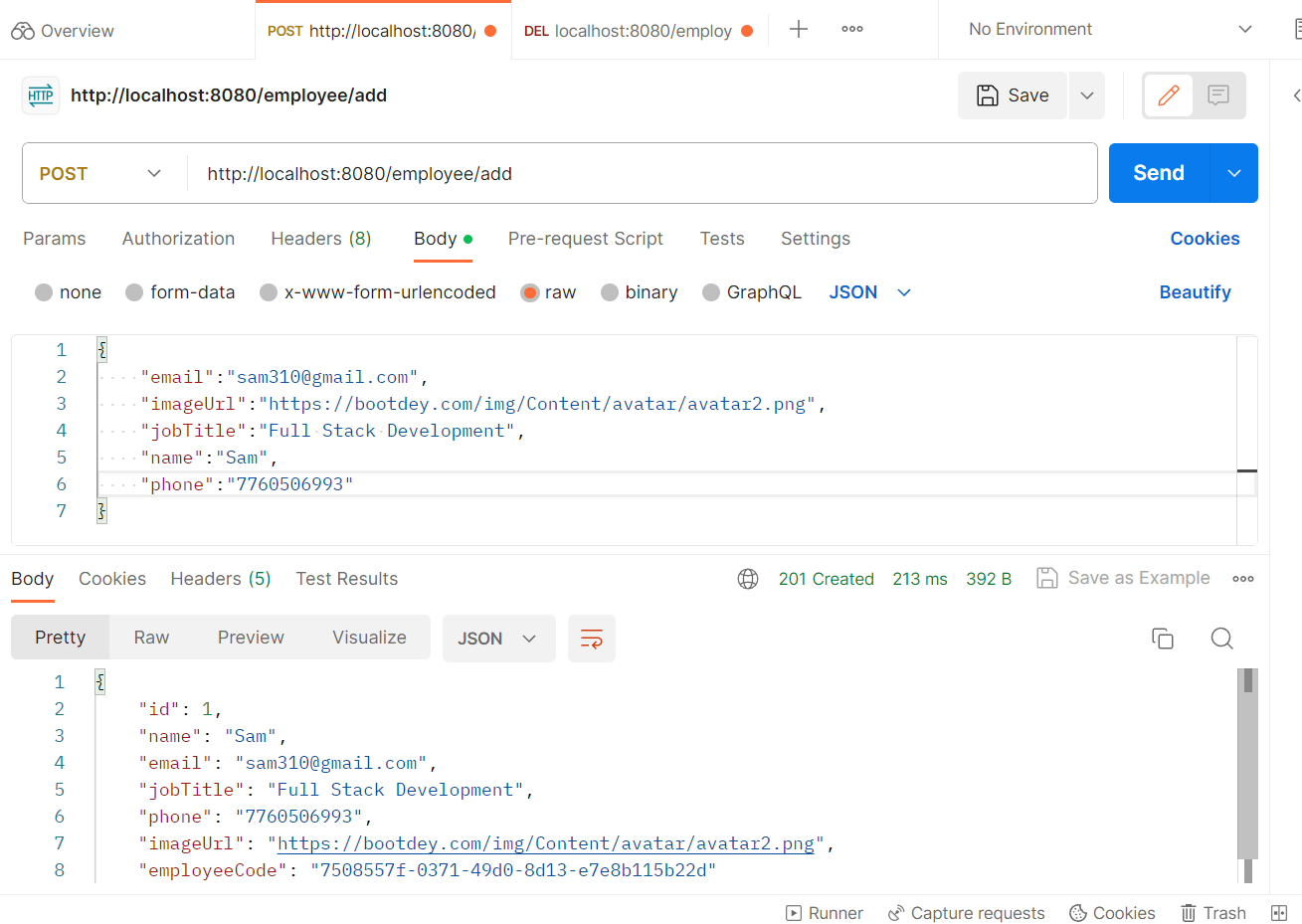
**Click on relaunch as showed in image:**

****

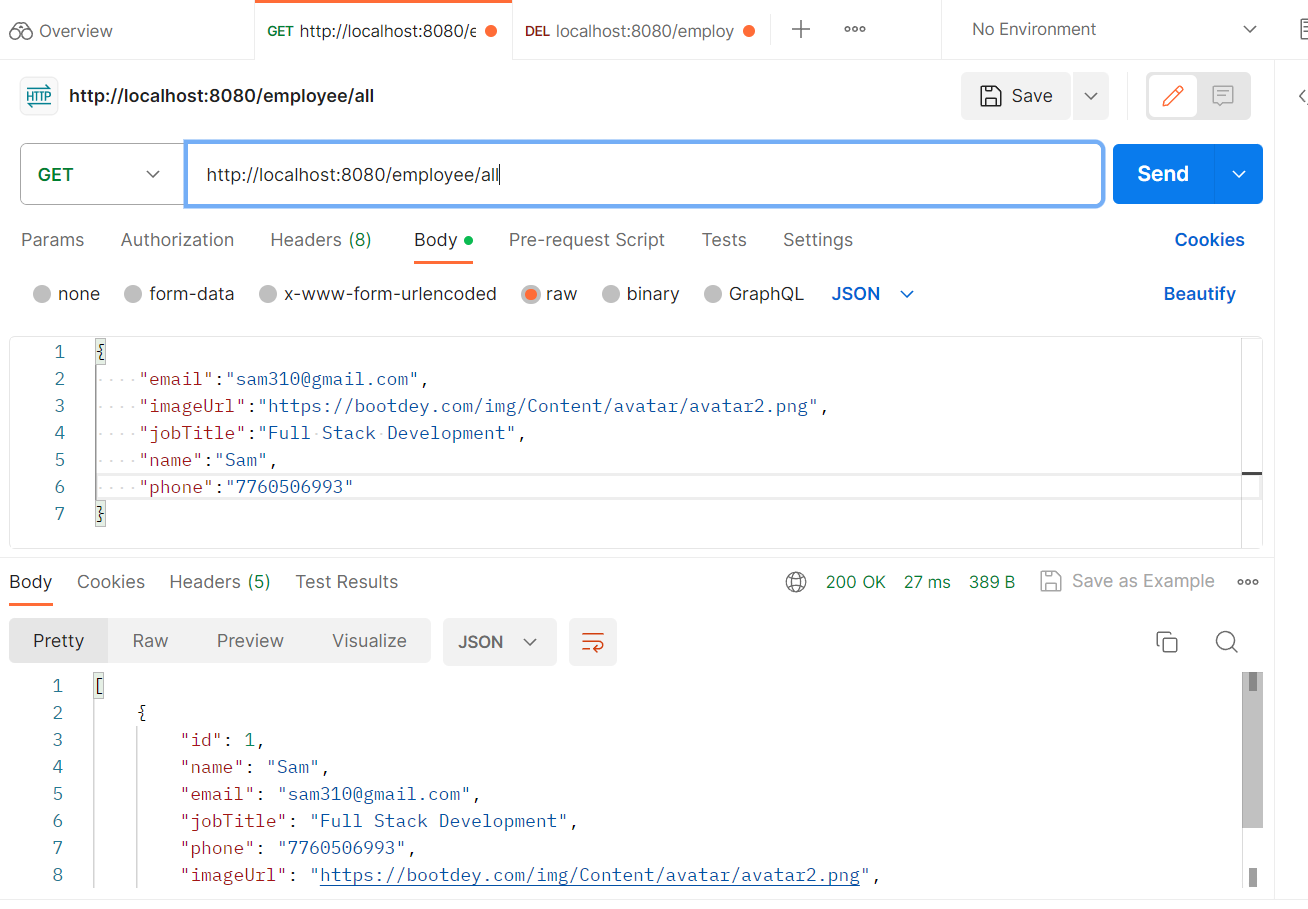
**TESTING Through POSTMAN:**

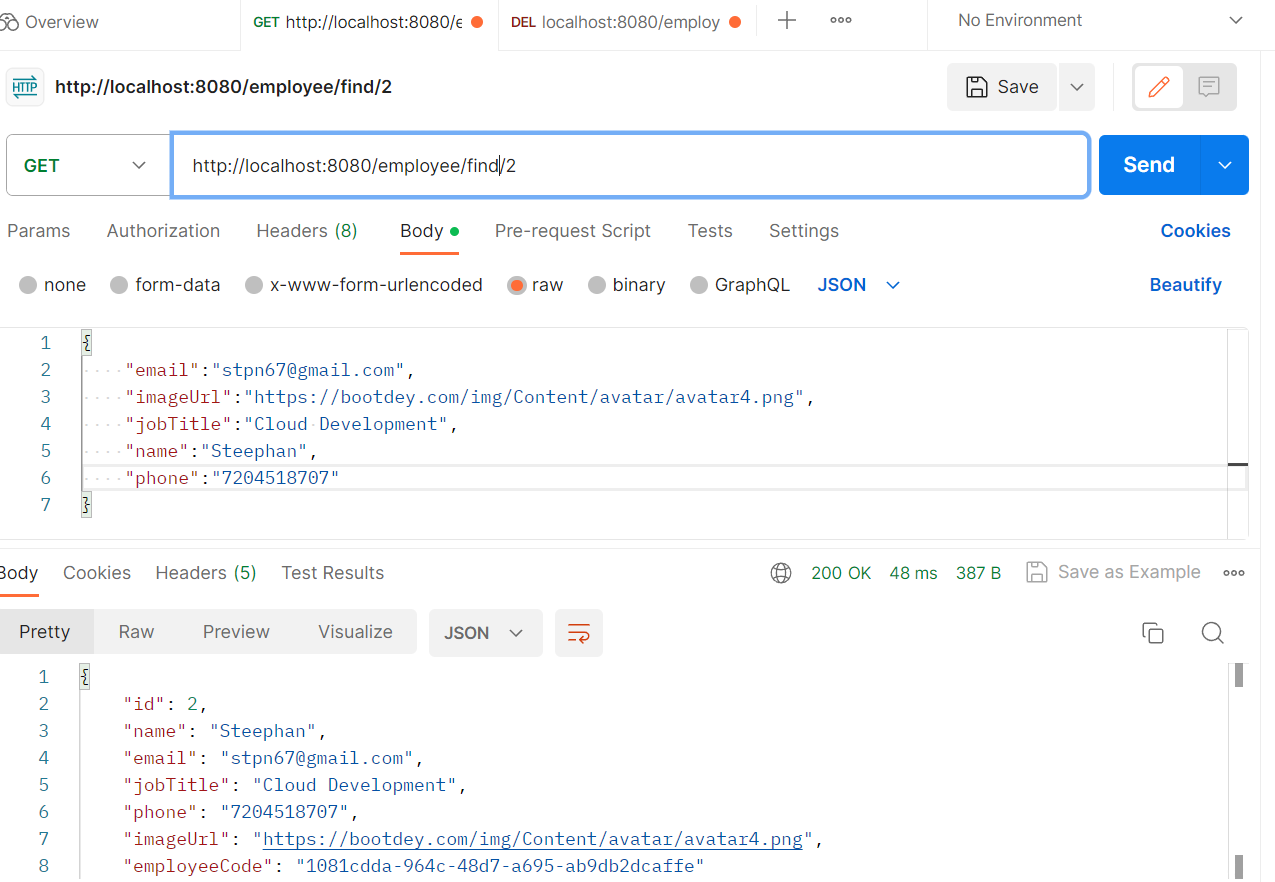


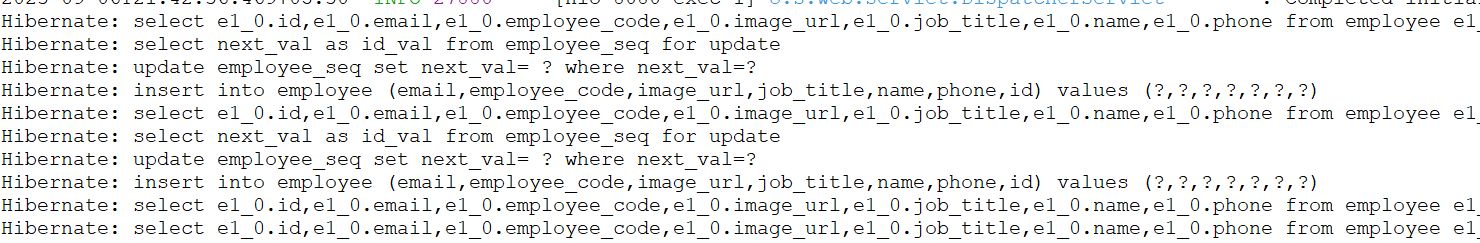
Returns empty list since there are no data in database.



Look for raw and JSON is selected for POST request.







Also we can verify using Httpie

Where we have to provide

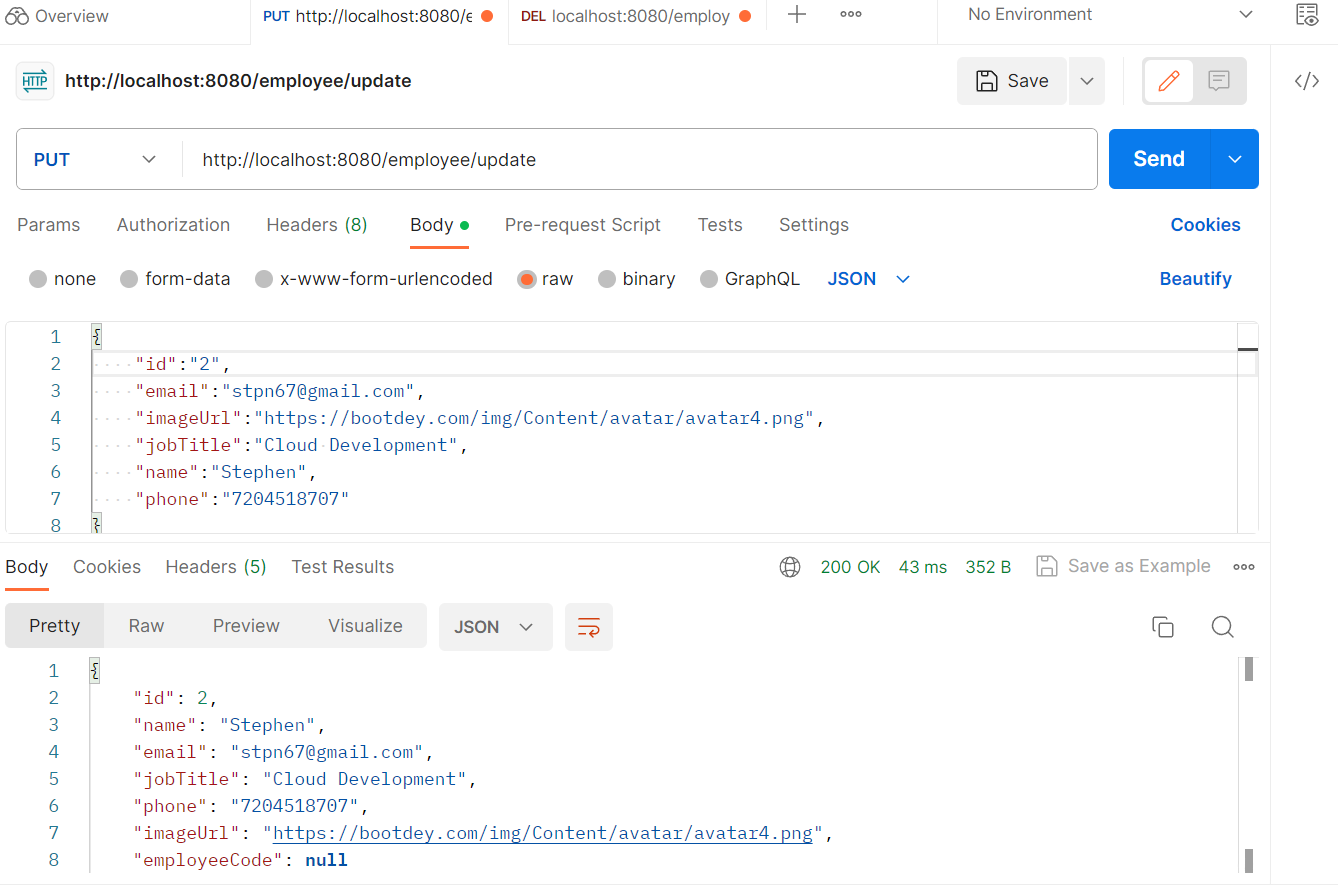
http GET localhost:8080/employee/all

OR

http :8080/employee/all we can skip GET since by default GET, and localhost can be skipped since it treats as localhost before port number.

For passing JSON in body use cmd

* touch data.json
* nano data.json
* paste the code
* http POST :8080/employee/add < data.json



Remember to add employeeCode in JSON