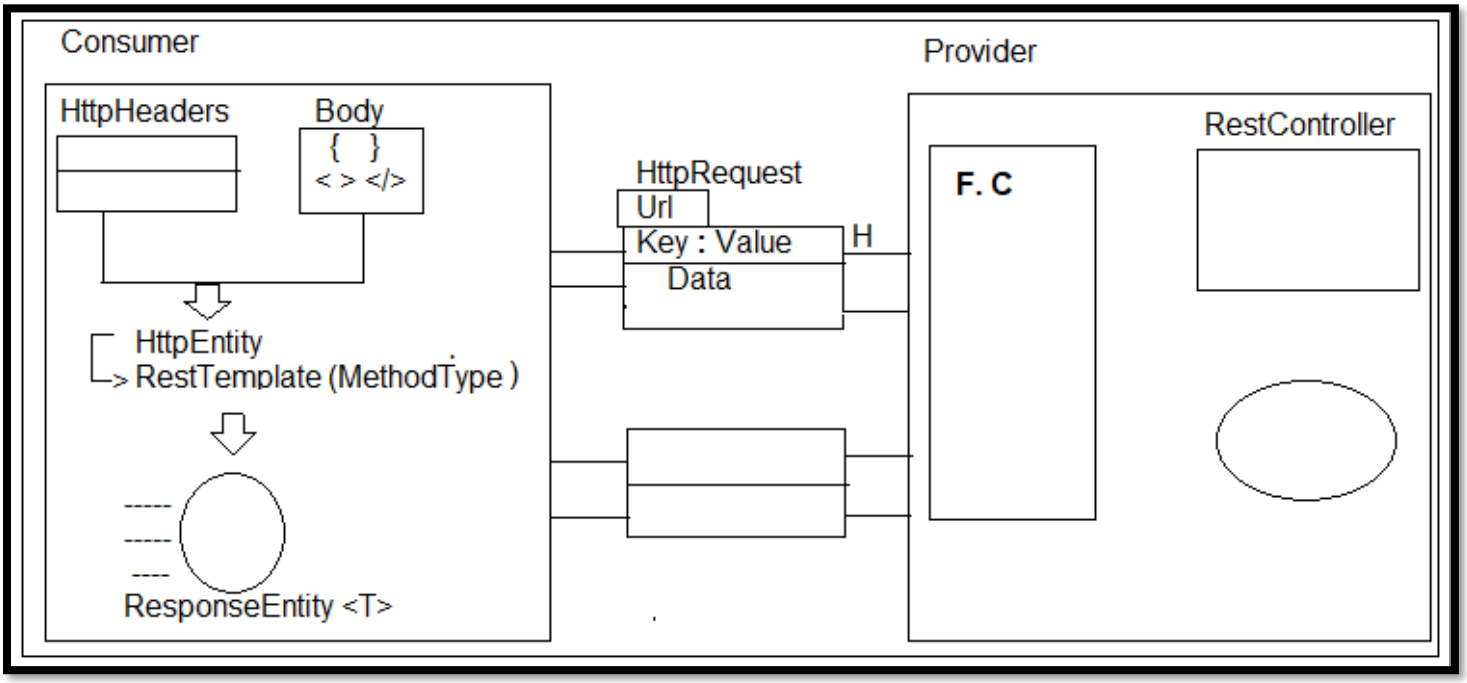
Spring Boot REST API

Introduction:

- To implement RESTFul web service API's using simple annotations and Templates, Spring boot REST F/w has been introduced.
- Even to implement Micro services design Spring boot REST API is used.



- **Consumer** and **Provider** interchanges the data Primitive (String format only), Class Type (Object) and Collections.
- Data will be converted to either **JSON** or **XML** format also known as **Global Formats**
- String data will be converted to any other type directly.
- REST Controller supports 5 types of Http Request Methods for Handling like **GET, POST, PUT, DELETE and PATCH.**

GET: Used to get data.
POST: Used to send data.
DELETE: Used to delete data.
PUT: Used to modify data (complete data) based on Identity (ID-PK).
PATCH: Used to modify partial (few data) based on Identity (ID-PK).

- **@RestController (Stereotype), @RequestMapping** (for URL) annotations are used as Controller class level.

- Controller methods level use annotations.

TYPE	Annotations
GET	@GetMapping
POST	@PostMapping
PUT	@PutMapping
DELETE	@DeleteMapping
PATCH	@PatchMapping

- To provide Inputs to Request Methods in RestController, use annotations.

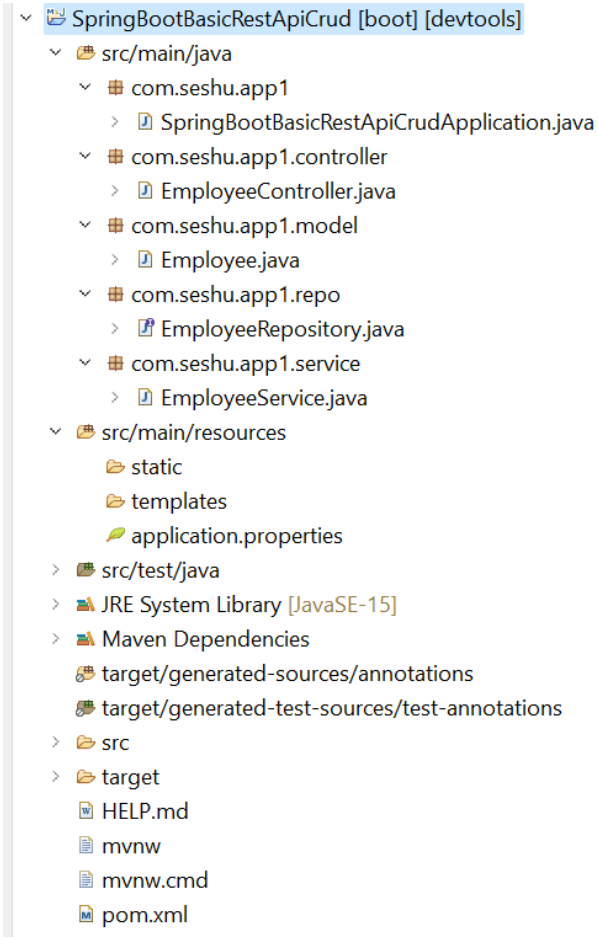
TYPE	Annotation
url?key=val	@RequestParam
/url/val/	@PathVariable
Http Header	@RequestHeader
Http Body	@RequestBody

Note: All above annotations works on **HttpRequest** only, supports reading input data.

Spring Boot Rest API Crud Operation

Create Spring Starter Project as **SpringBootBasicRestApiCrud** with fallowing dependencies;

Spring Web,
Spring Boot DevTools,
Mysql Driver,
Spring Data JPA



application.properties

```
server.port=8181
server.servlet.context-path=/myapp

spring.datasource.url=jdbc:mysql://localhost:3306/adidb
spring.datasource.username=root
spring.datasource.password=seshu

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
```

Employee.java

```
package com.seshu.app1.model;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "emp001")
public class Employee {
	@Id
	@GeneratedValue
	private Integer employeeId;
	private String employeeName;
	private Double employeeSalary;
	private String employeeEmail;
	private Long employeeMobile;

	public Employee() {
		super();
	}

	public Employee(Integer employeeId, String employeeName, Double
employeeSalary, String employeeEmail,
			Long employeeMobile) {
		super();
		this.employeeId = employeeId;
		this.employeeName = employeeName;
		this.employeeSalary = employeeSalary;
		this.employeeEmail = employeeEmail;
		this.employeeMobile = employeeMobile;
	}

	public Integer getEmployeeId() {
		return employeeId;
	}

	public void setEmployeeId(Integer employeeId) {
		this.employeeId = employeeId;
	}
```

```
	public String getEmployeeName() {
		return employeeName;
	}

	public void setEmployeeName(String employeeName) {
		this.employeeName = employeeName;
	}

	public Double getEmployeeSalary() {
		return employeeSalary;
	}

	public void setEmployeeSalary(Double employeeSalary) {
		this.employeeSalary = employeeSalary;
	}

	public String getEmployeeEmail() {
		return employeeEmail;
	}

	public void setEmployeeEmail(String employeeEmail) {
		this.employeeEmail = employeeEmail;
	}

	public Long getEmployeeMobile() {
		return employeeMobile;
	}

	public void setEmployeeMobile(Long employeeMobile) {
		this.employeeMobile = employeeMobile;
	}

	@Override
	public String toString() {
		return "Employee [employeeId=" + employeeId + ", employeeName=" +
employeeName + ", employeeSalary="
				+ employeeSalary + ", employeeEmail=" + employeeEmail + ",
employeeMobile=" + employeeMobile + "]";
	}
}
```

EmployeeRepository.java

```
package com.seshu.app1.repo;

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

import com.seshu.app1.model.Employee;

public interface EmployeeRepository extends JpaRepository<Employee, Integer>{

}
```

EmployeeService.java

```
package com.seshu.app1.service;

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

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

import com.seshu.app1.model.Employee;
import com.seshu.app1.repo.EmployeeRepository;

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

	public Employee saveEmployee(Employee e) {
		return repo.save(e);
	}

	public List<Employee> getEmployees() {
		return repo.findAll();
	}

	public Optional<Employee> getEmployee(Integer id) {
		return repo.findById(id);
	}
```

```
	public void updateEmployee(Employee e) {
		repo.save(e);

	}

	public void deleteEmployee(Integer id) {
		repo.deleteById(id);
	}
}
```

EmployeeController.java

```
package com.seshu.app1.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.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.seshu.app1.model.Employee;
import com.seshu.app1.service.EmployeeService;

@RestController
@RequestMapping("/api")
public class EmployeeController {
	@Autowired
	private EmployeeService service;

	@PostMapping("/employee")
	public ResponseEntity<Employee> save(@RequestBody Employee employee) {
		Employee newEmployee = service.saveEmployee(employee);
		return new ResponseEntity<>(newEmployee, HttpStatus.CREATED);
	}
```

```
	@GetMapping("/employee")
	public ResponseEntity<List<Employee>> getAllEmployees() {
		List<Employee> employeeList = service.getEmployees();
		return new ResponseEntity<>(employeeList, HttpStatus.OK);
	}

	@GetMapping("/employee/{id}")
	public ResponseEntity<?> getEmployeeById(@PathVariable Integer id) {
		Employee employee = service.getEmployee(id).get();
		return new ResponseEntity<Employee>(employee, HttpStatus.OK);
	}

	@PutMapping("/employee")
	public ResponseEntity<?> update(@RequestBody Employee employee){
		service.updateEmployee(employee);
		return new ResponseEntity<>(HttpStatus.OK);
	}

	@DeleteMapping("/employee/{id}")
	public ResponseEntity<?> delete(@PathVariable Integer id){
		service.deleteEmployee(id);
		return new ResponseEntity<>(HttpStatus.OK);
	}
}
```

SpringBootBasicRestApiCrudApplication.java

```
package com.seshu.app1;

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

@SpringBootApplication
public class SpringBootBasicRestApiCrudApplication {

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

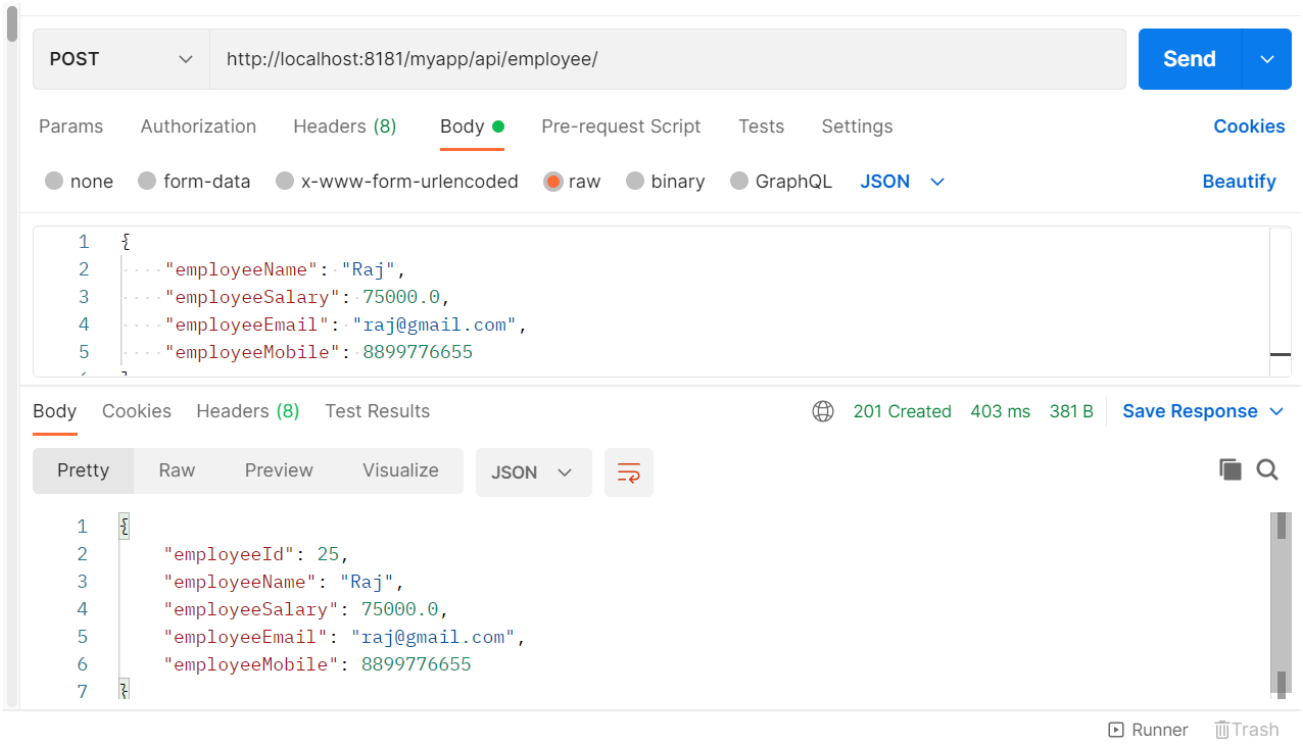
Right click on SpringBootBasicRestApiCrudApplication.java -> Run As -> Spring Boot App

Test the Employee API service using Postman.

POST Request;

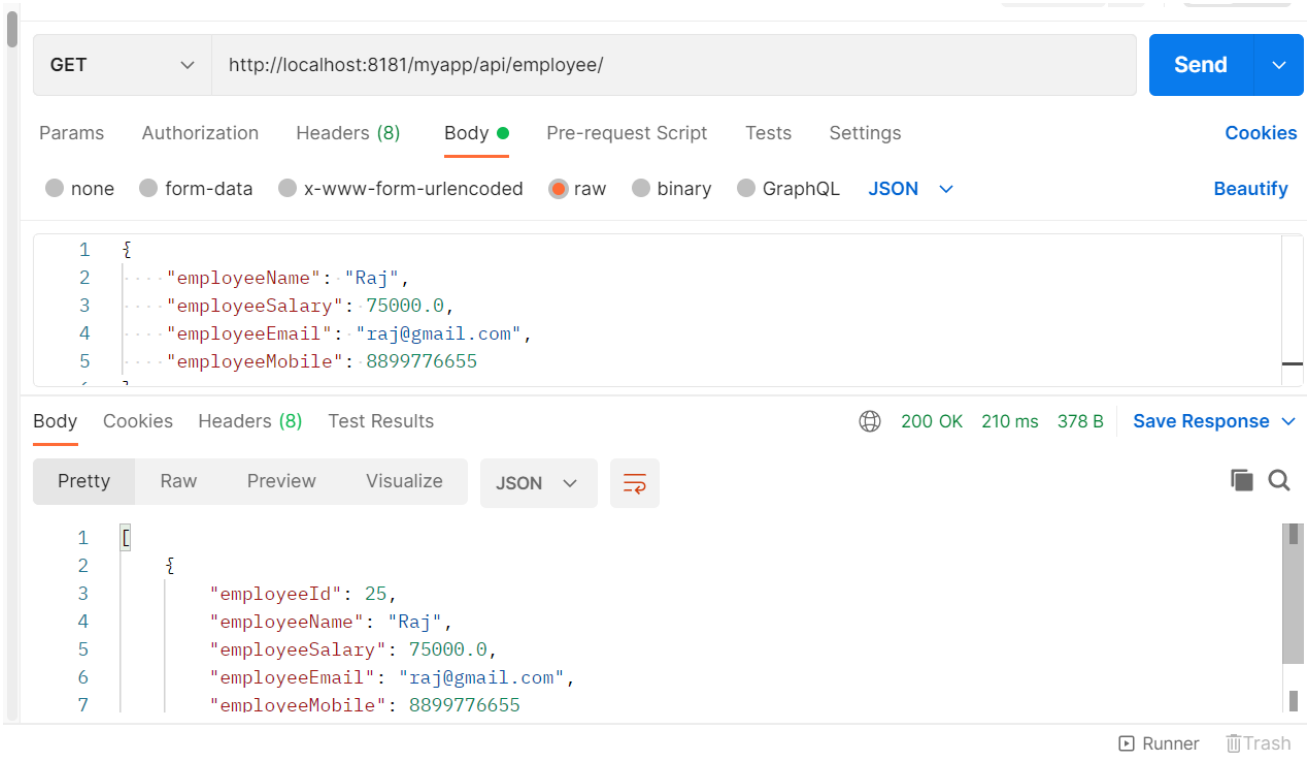
http://localhost:8181/myapp/api/employee/

```
{
   "employeeName": "Raj",
   "employeeSalary": 75000.0,
   "employeeEmail": "raj@gmail.com",
   "employeeMobile": 8899776655
}
```



GET REQUEST;

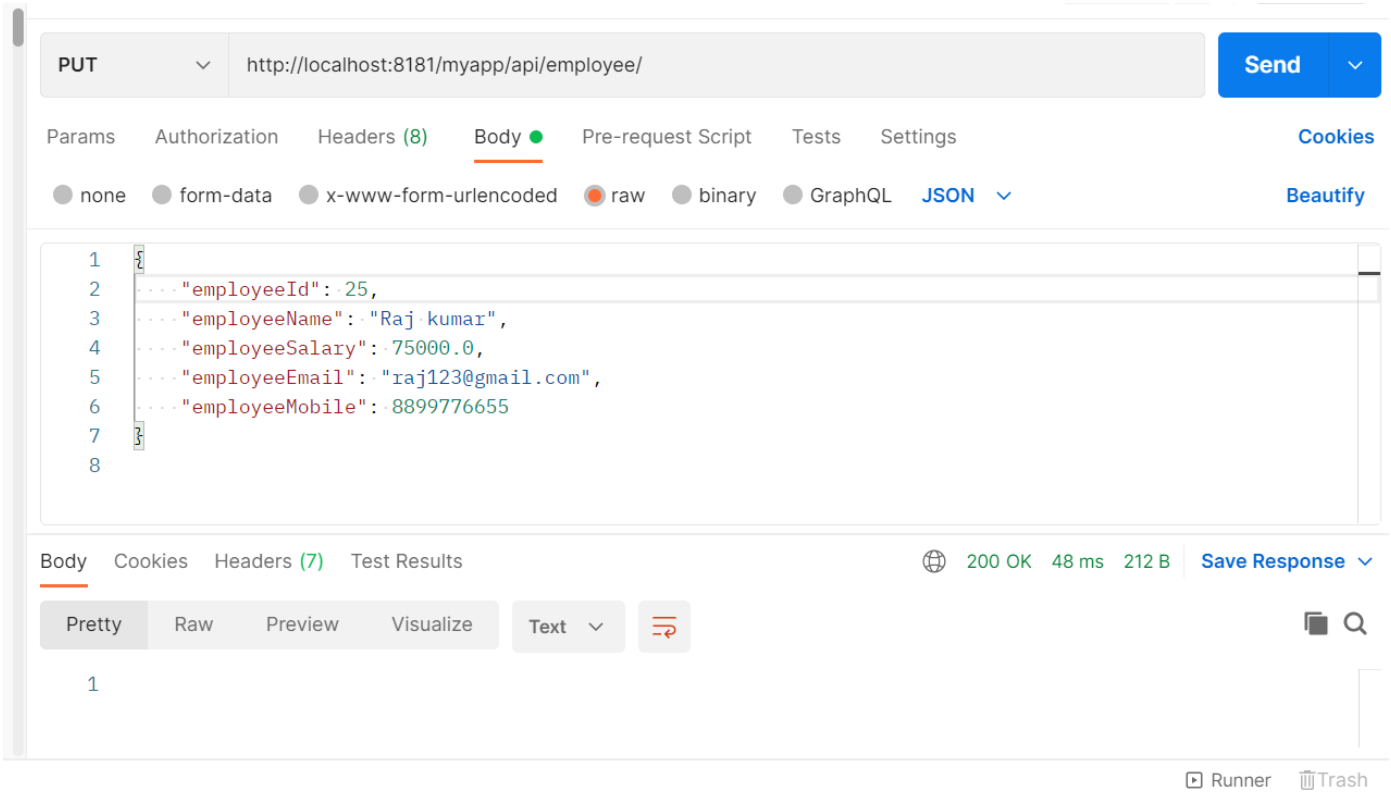
http://localhost:8181/myapp/api/employee/



PUT REQUEST;

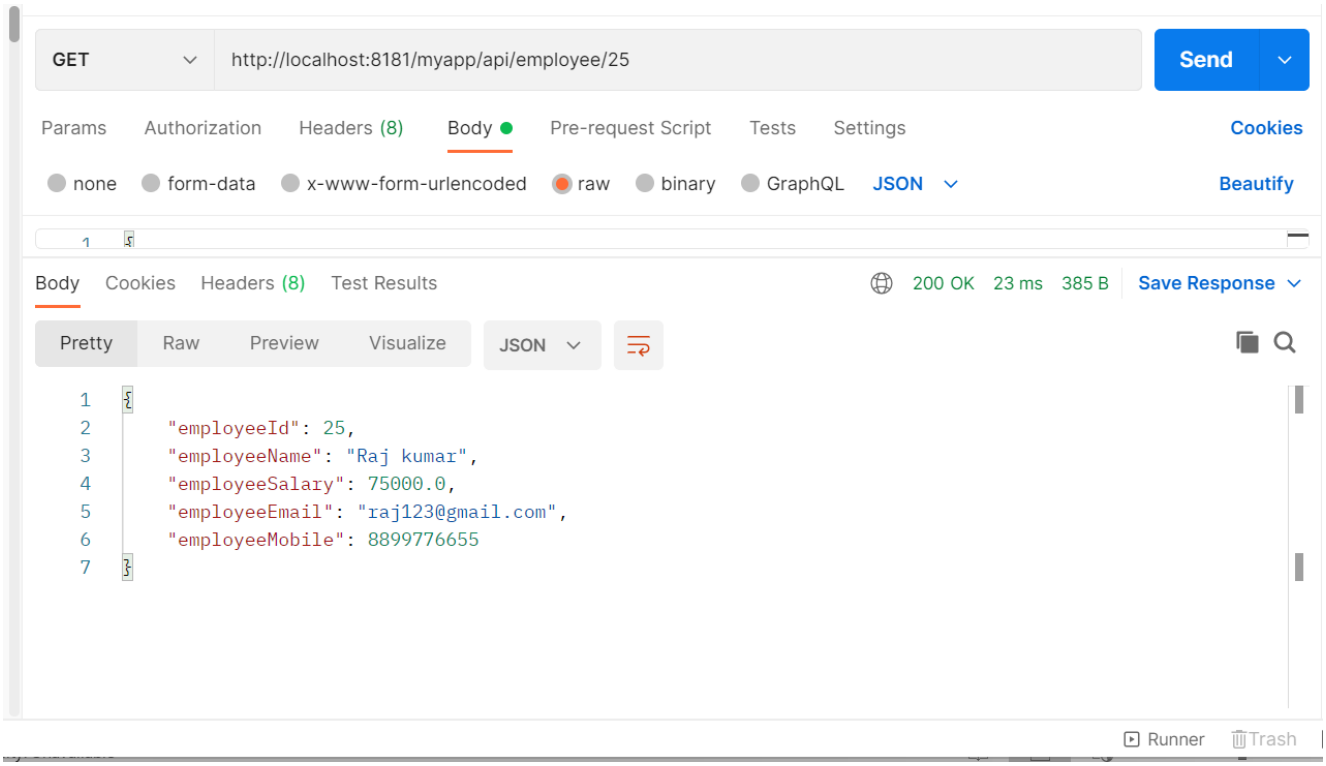
http://localhost:8181/myapp/api/employee/

```
{
   "employeeId": 25,
   "employeeName": "Raj kumar",
   "employeeSalary": 75000.0,
   "employeeEmail": "raj123@gmail.com",
   "employeeMobile": 8899776655
}
```



GET REQUEST;

http://localhost:8181/myapp/api/employee/25



DELETE REQUEST;

http://localhost:8181/myapp/api/employee/25