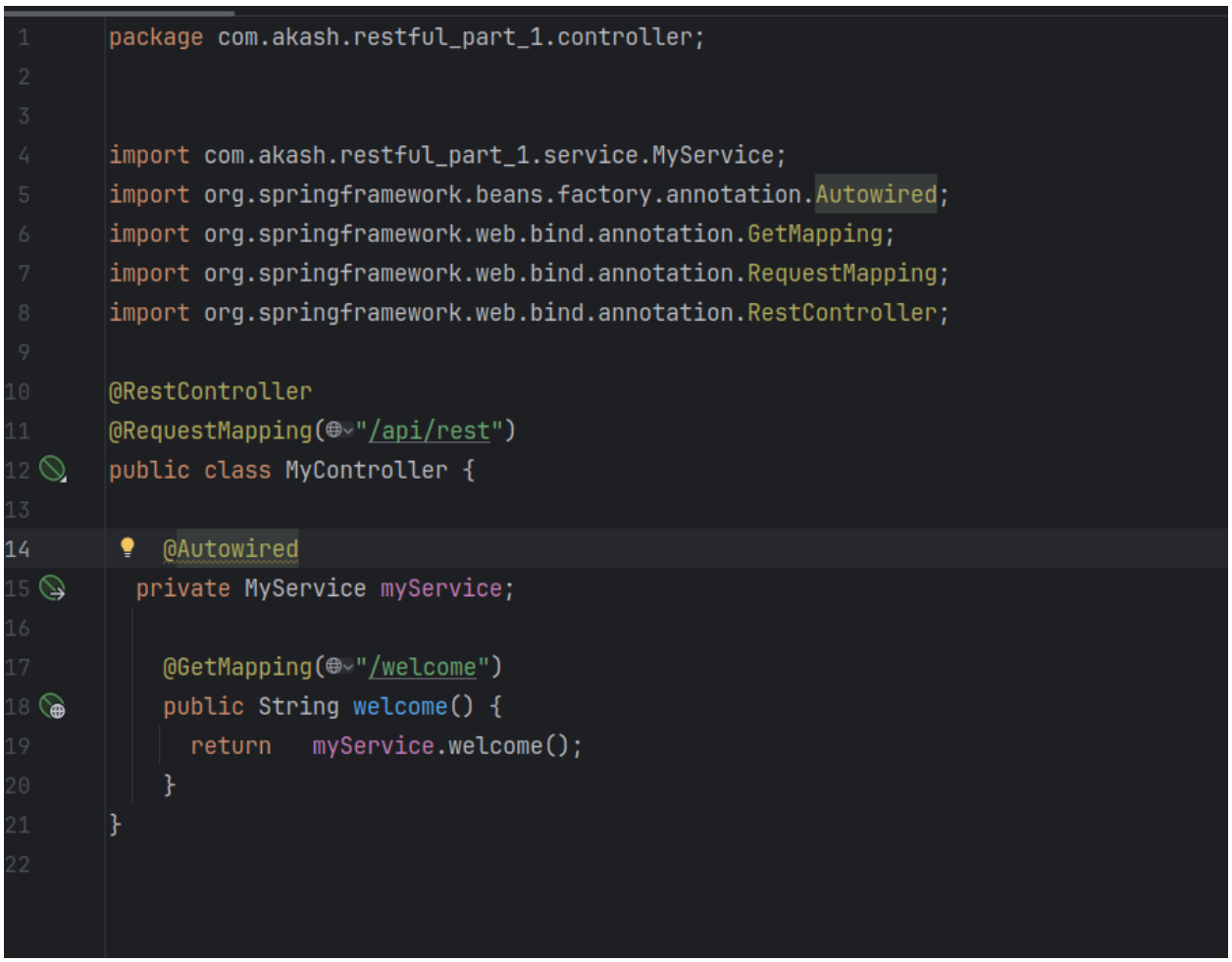
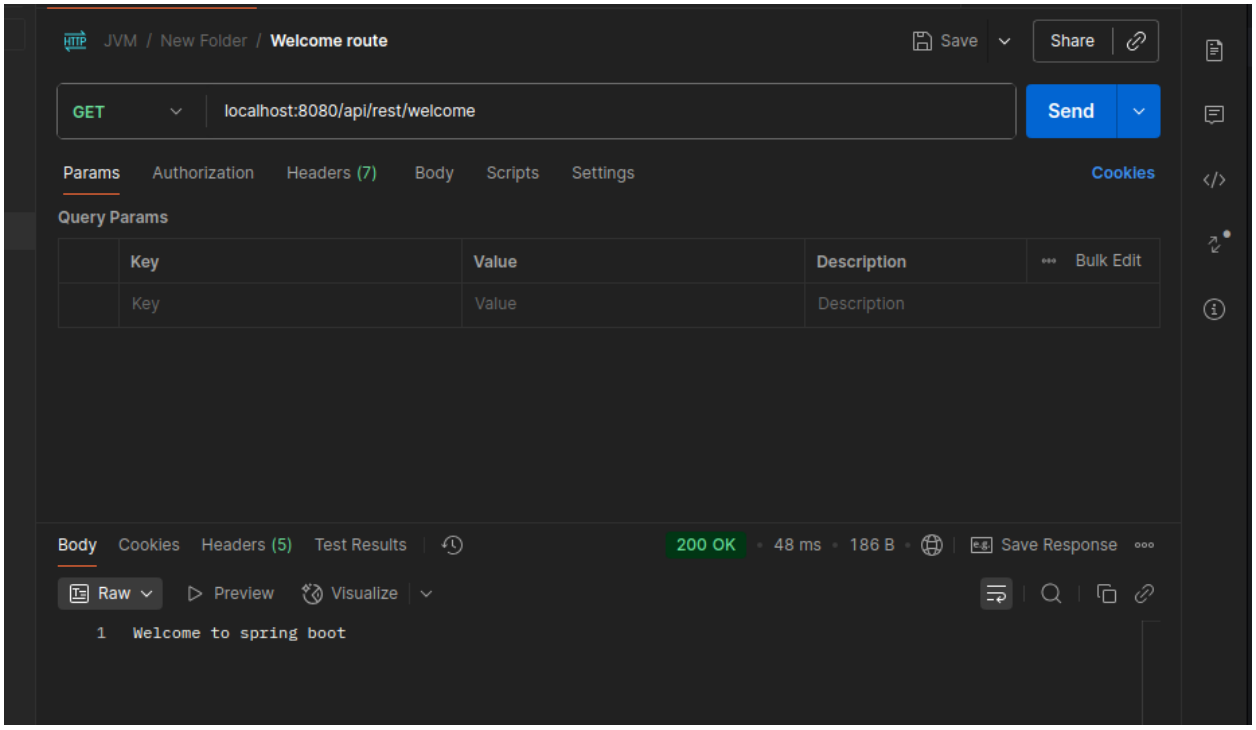
Assignment - RestFul Web Services - Part 1

Q1)Create a simple REST ful service in Spring Boot which returns the Response "Welcome to spring boot".

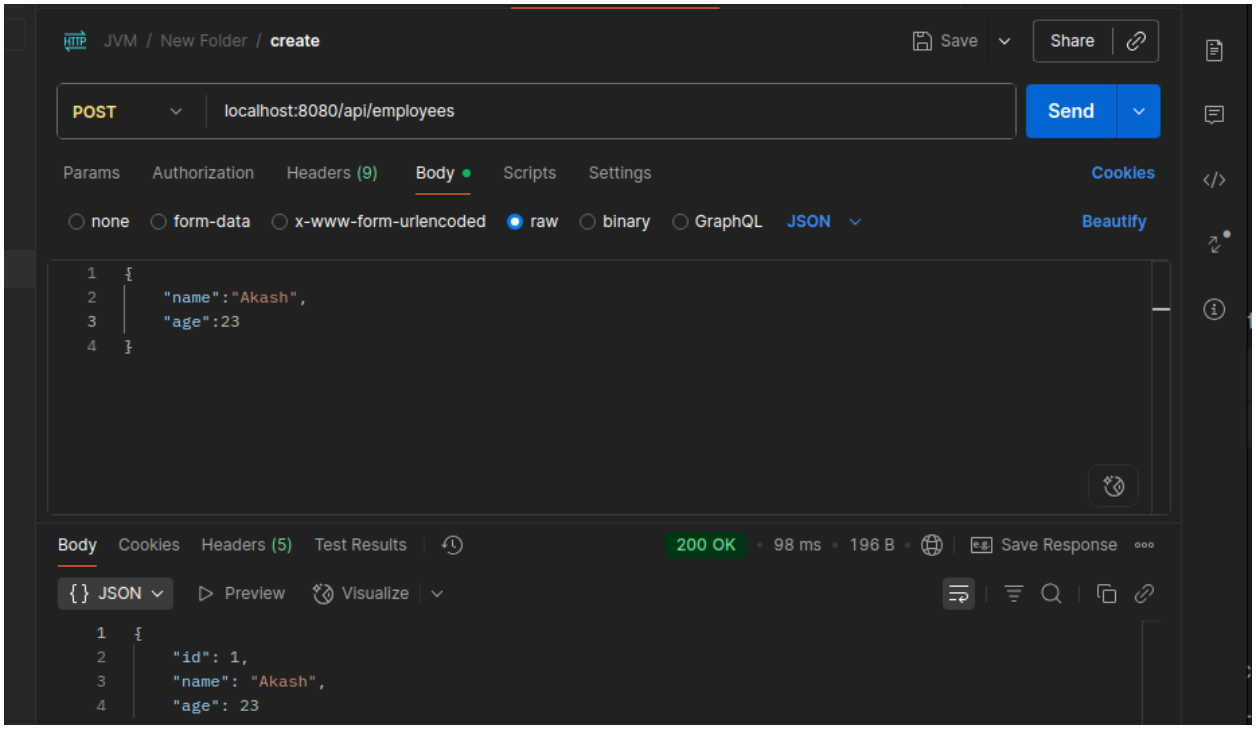


```java
package com.akash.restful_part_1.controller;


import com.akash.restful_part_1.service.MyService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/api/rest")
public class MyController {

    @Autowired
    private MyService myService;

    @GetMapping("/welcome")
    public String welcome() {
        return   myService.welcome();
    }
}
```

```java
package com.akash.restful_part_1.service;

import org.springframework.stereotype.Service;

Rename usages
@Service
public class MyService {
    public String welcome(){  no usages
        return "Welcome to spring boot";
    }
}

```



Q2)Create an Employee Bean(id, name, age) and service to perform different operations related to employee.



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

    @Autowired
    private EmployeeService service;

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

    @GetMapping("/{id}")
    public Employee getEmployeeById(@PathVariable Long id) {
        return service.getEmployeeById(id);
    }

    @PostMapping
    public Employee createEmployee(@Valid @RequestBody Employee employee) {
        return service.createEmployee(employee);
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<String> deleteEmployee(@PathVariable Long id) {
        service.deleteEmployee(id);
```

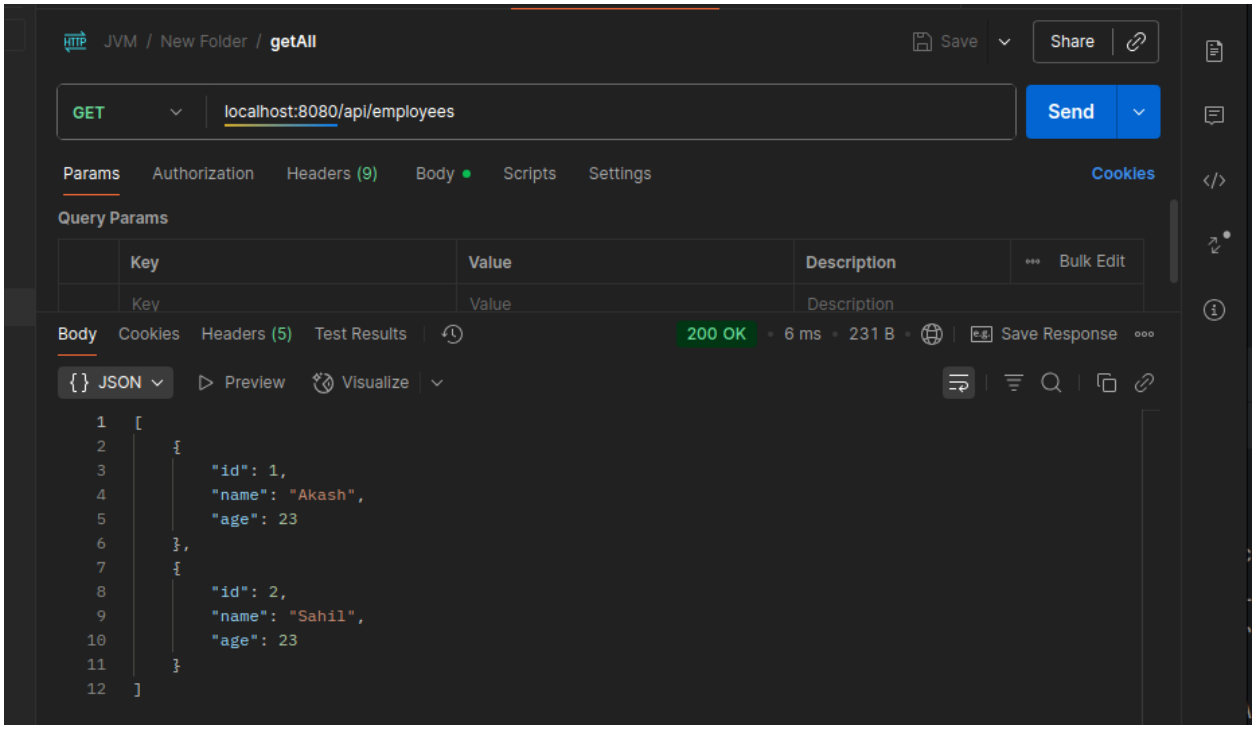
```java
import java.util.Map;

@Service  no usages
public class EmployeeService {

    private Map<Long, Employee> employees = new HashMap<>();  7 usages
    private long customId = 1L;  1 usage

    private synchronized Long generateCustomId() {  1 usage
        return customId++;
    }

    public List<Employee> getAllEmployees() {  no usages
        return new ArrayList<>(employees.values());
    }

    public Employee getEmployeeById(Long id) {  no usages
        Employee emp = employees.get(id);
        if (emp == null) throw new ResourceNotFoundException("Employee not found
        return emp;
    }

    public Employee createEmployee(Employee employee) {  no usages
        employee.setId(generateCustomId());
        employees.put(employee.getId(), employee);
        return employee;
```

```java
package com.akash.restful_part_1.model;

import org.springframework.stereotype.Component;

@Component  16 usages
public class Employee {
    private Long id;  2 usages
    private String name;  2 usages
    private int age;  2 usages


    public Long getId() {  1 usage
        return id;
    }

    public void setId(Long id) {  2 usages
        this.id = id;
    }

    public String getName() {  no usages
```

Q3)Implement GET http request for Employee to get list of employees.

```java
public List<Employee> getAllEmployees() {  no usages
    return new ArrayList<>(employees.values());
}
```

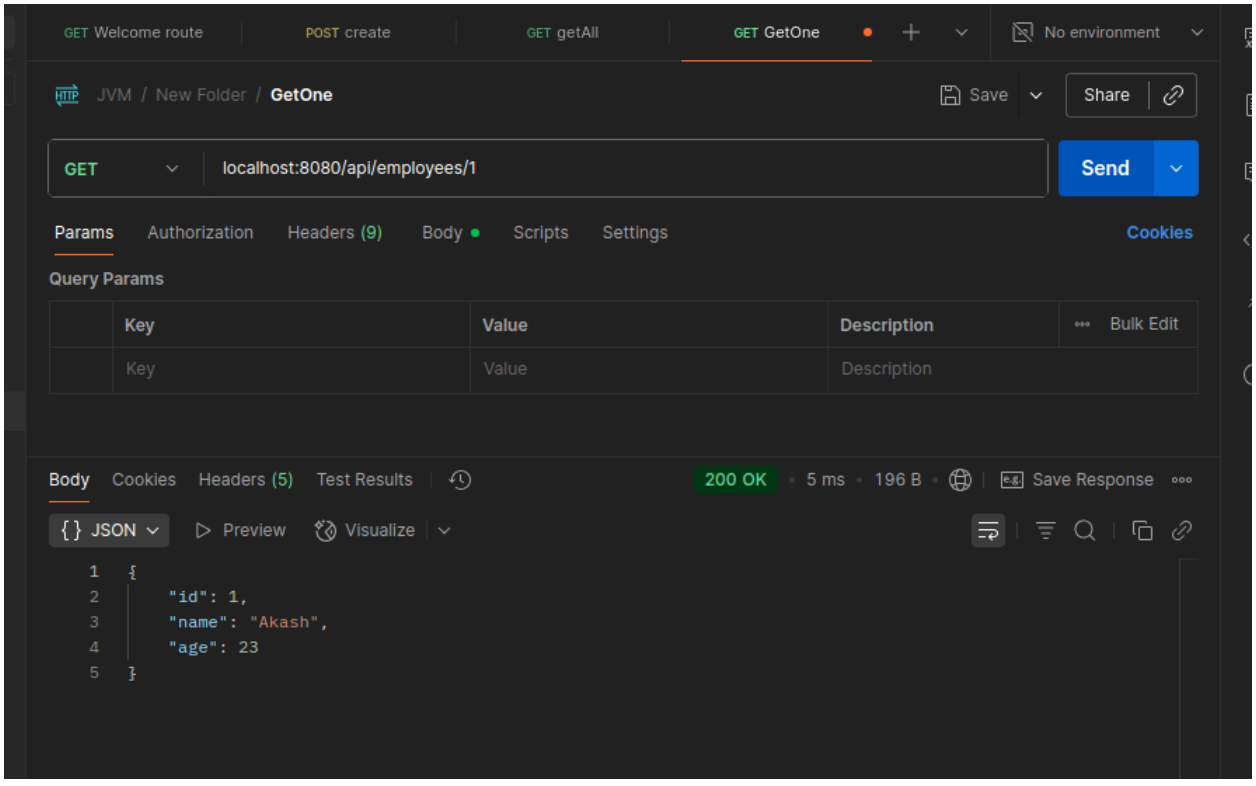
```java
@GetMapping
public List<Employee> getAllEmployees() {
    return service.getAllEmployees();
}
```



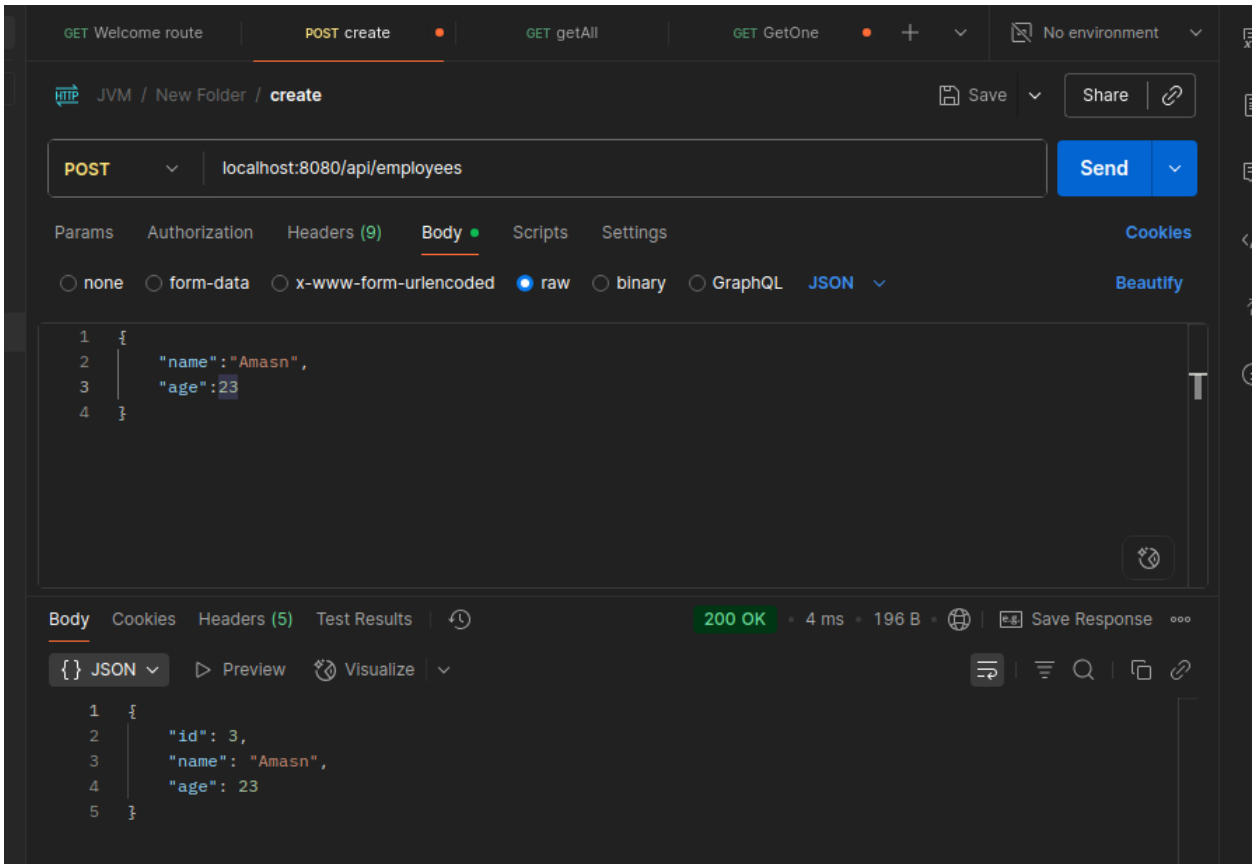
Q4)Implement GET http request using path variable top get one employee

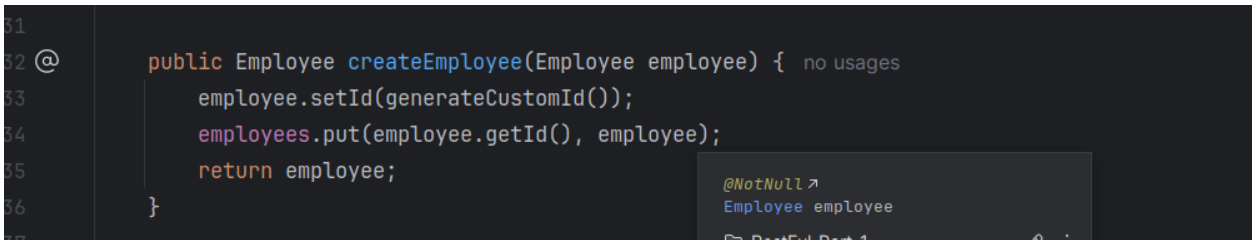
```java
@GetMapping("/{id}")
public Employee getEmployeeById(@PathVariable Long id) {
    return service.getEmployeeById(id);
}
```

```java
public Employee getEmployeeById(Long id) {  no usages
    Employee emp = employees.get(id);
    if (emp == null) throw new ResourceNotFoundException("Employee not found with id: " + id);
    return emp;
}
```



Q5)Implement POST http request for Employee to create a new employee.





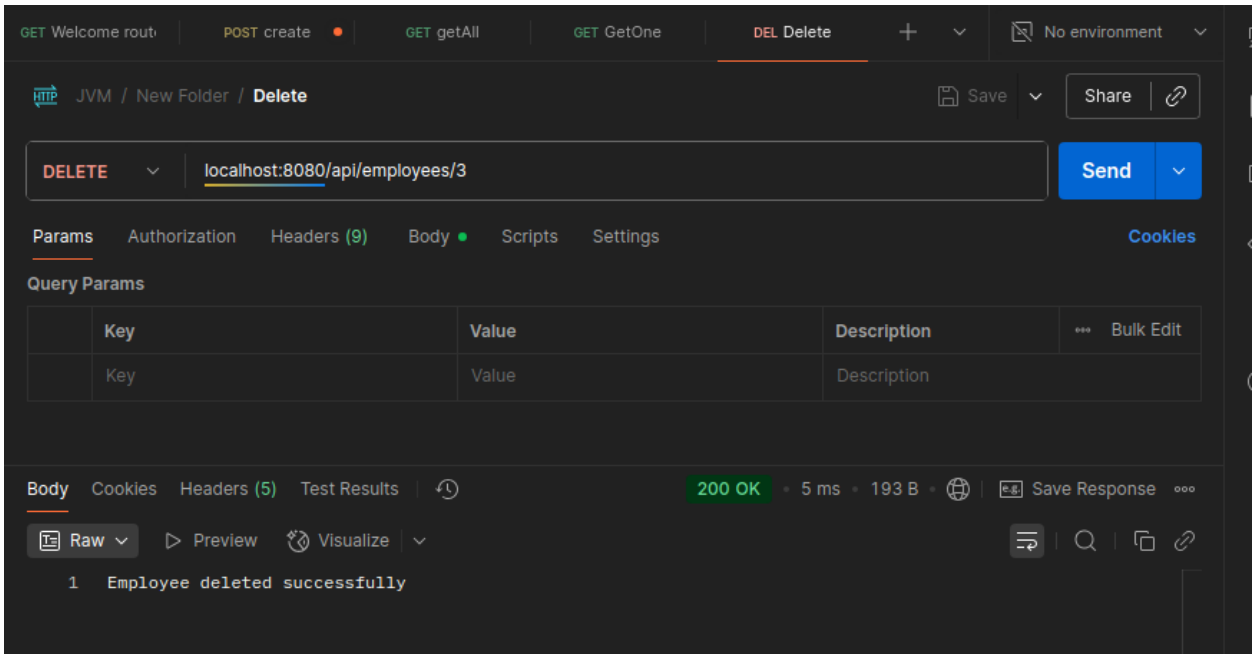
Q6)Implement Exception Handling for resource not found

```java
package com.akash.restful_part_1.exception;

public class ResourceNotFoundException extends RuntimeException {  4 usages
    public ResourceNotFoundException(String message) {  3 usages
        super(message);
    }
}
```

Q7)Implement DELETE http request for Employee to delete employee

```java
@DeleteMapping("/{id}")
public ResponseEntity<String> deleteEmployee(@PathVariable Long id) {
    service.deleteEmployee(id);
    return ResponseEntity.ok( body: "Employee deleted successfully");
}
```

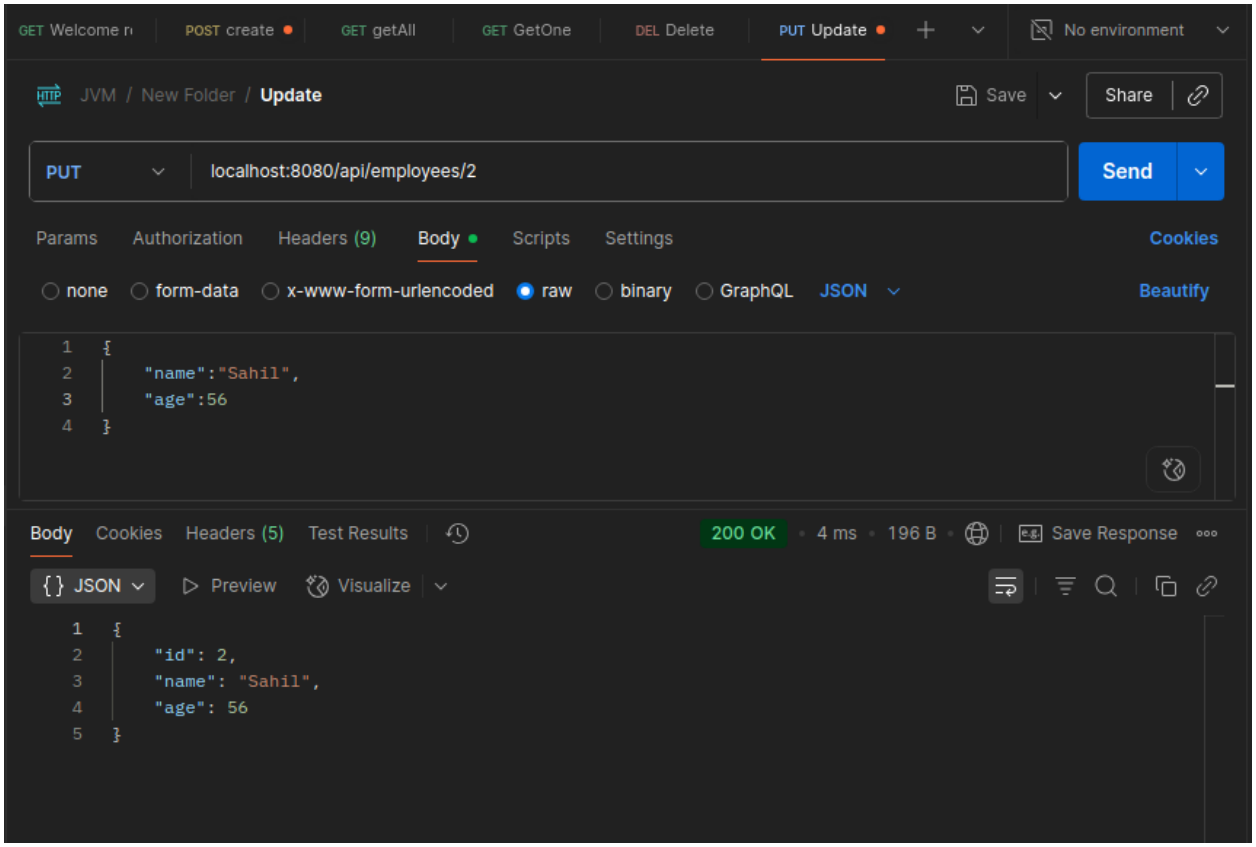
```java
public void deleteEmployee(Long id) {  no usages
    if (!employees.containsKey(id)) throw new ResourceNotFoundException("Employee not found with id: " + id);
    employees.remove(id);
}
```



Q8)Implement PUT http request for Employee to update employee

```java
    public Employee updateEmployee(Long id, Employee updated) {  no usages
        if (!employees.containsKey(id)) throw new ResourceNotFoundException("Employee not found with id: " + id);
        updated.setId(id);
        employees.put(id, updated);
        return updated;
    }
}
```

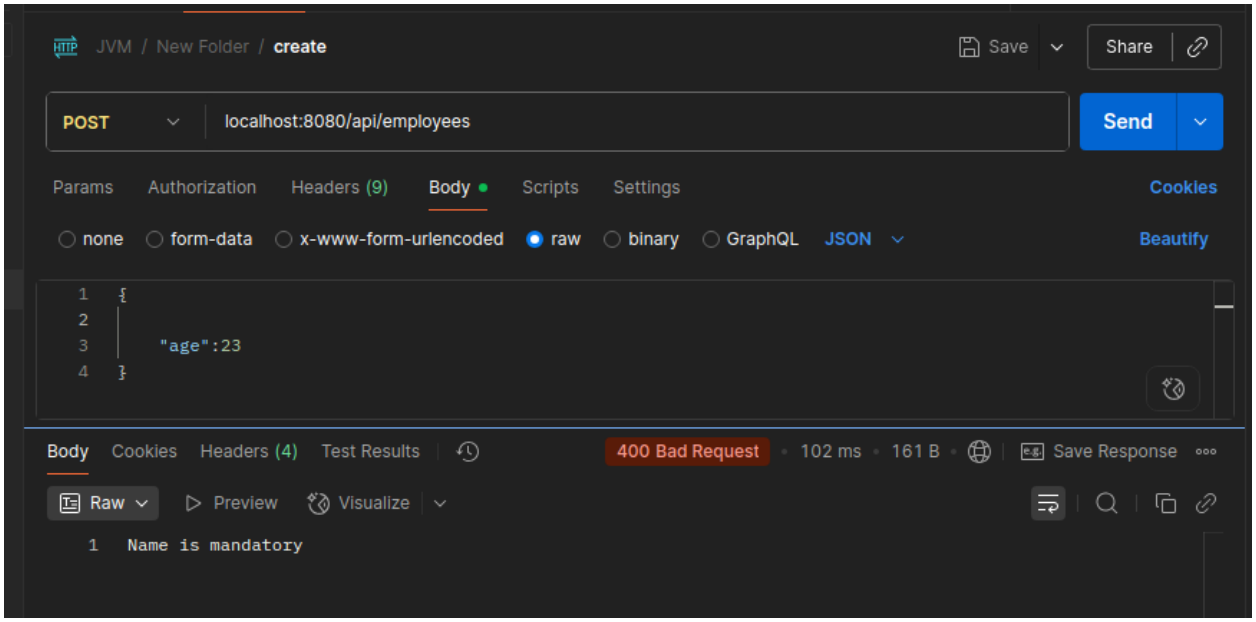
```java
    }

    @PutMapping("/{id}")
    public Employee updateEmployee(@PathVariable Long id, @RequestBody Employee updatedEmployee) {
        return service.updateEmployee(id, updatedEmployee);
    }
}
```



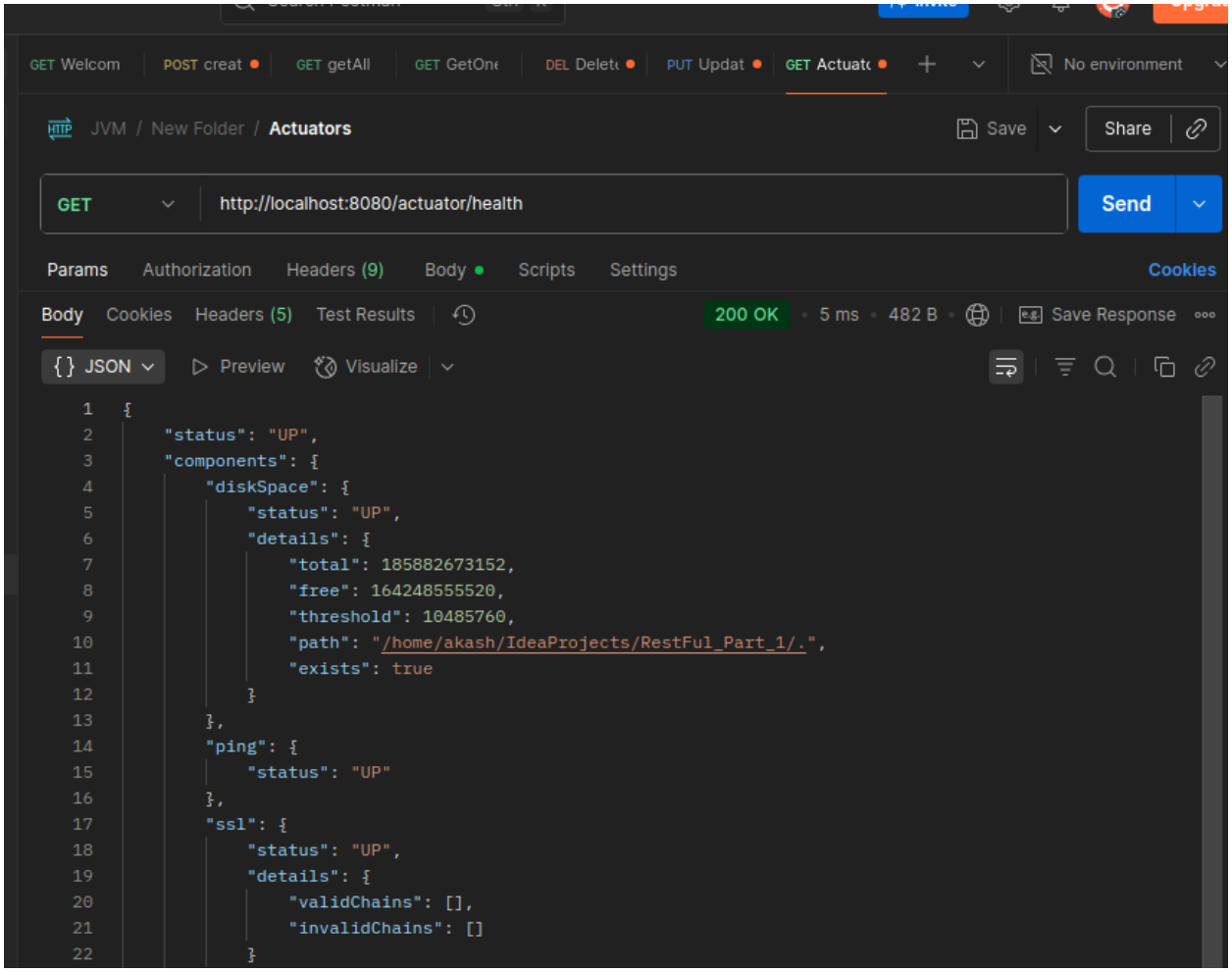
Q9)Apply validation while create a new employee using POST http Request.

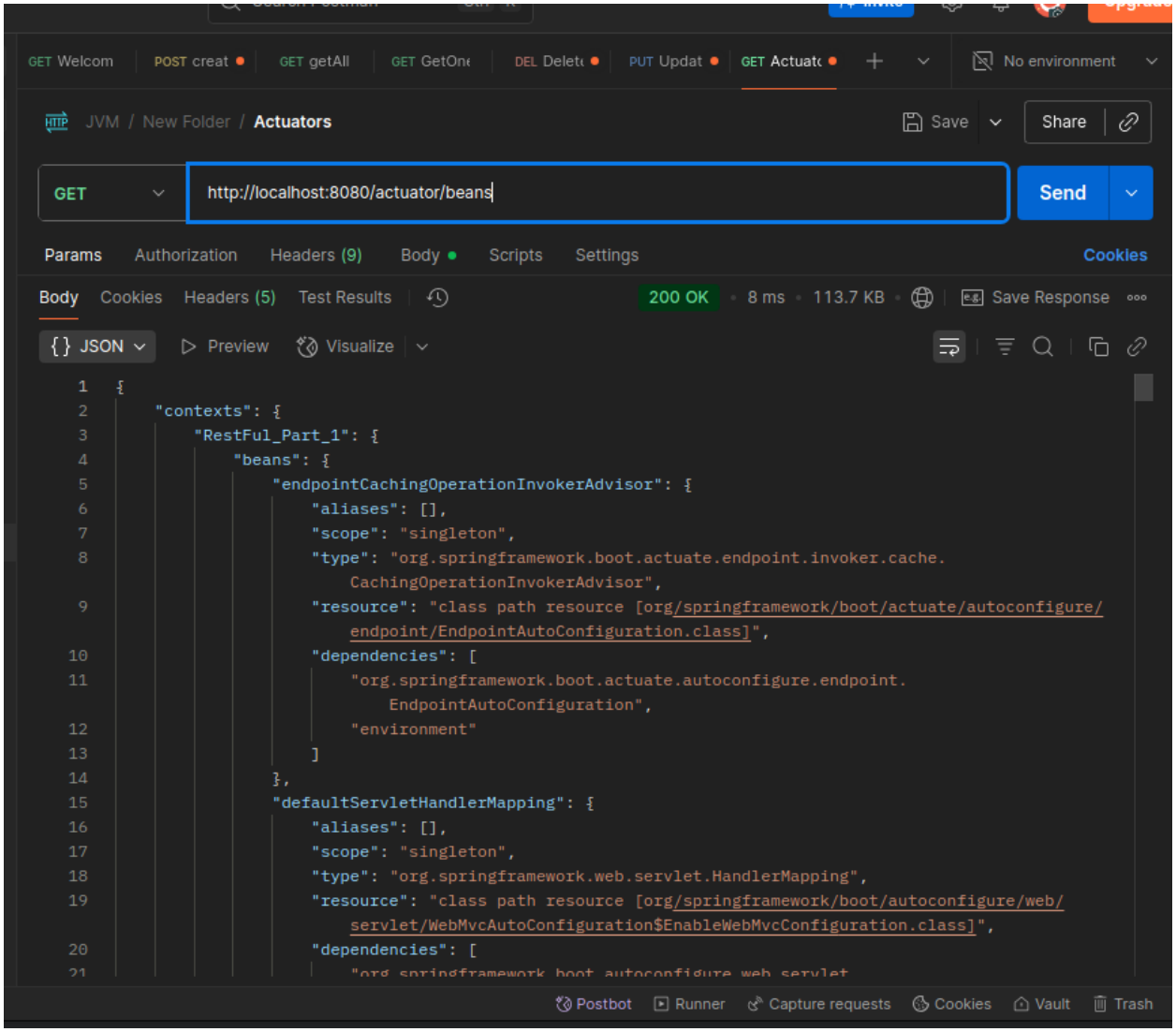
```java
    @PostMapping
    public Employee createEmployee(@Valid @RequestBody Employee employee) {
        return service.createEmployee(employee);
    }
```

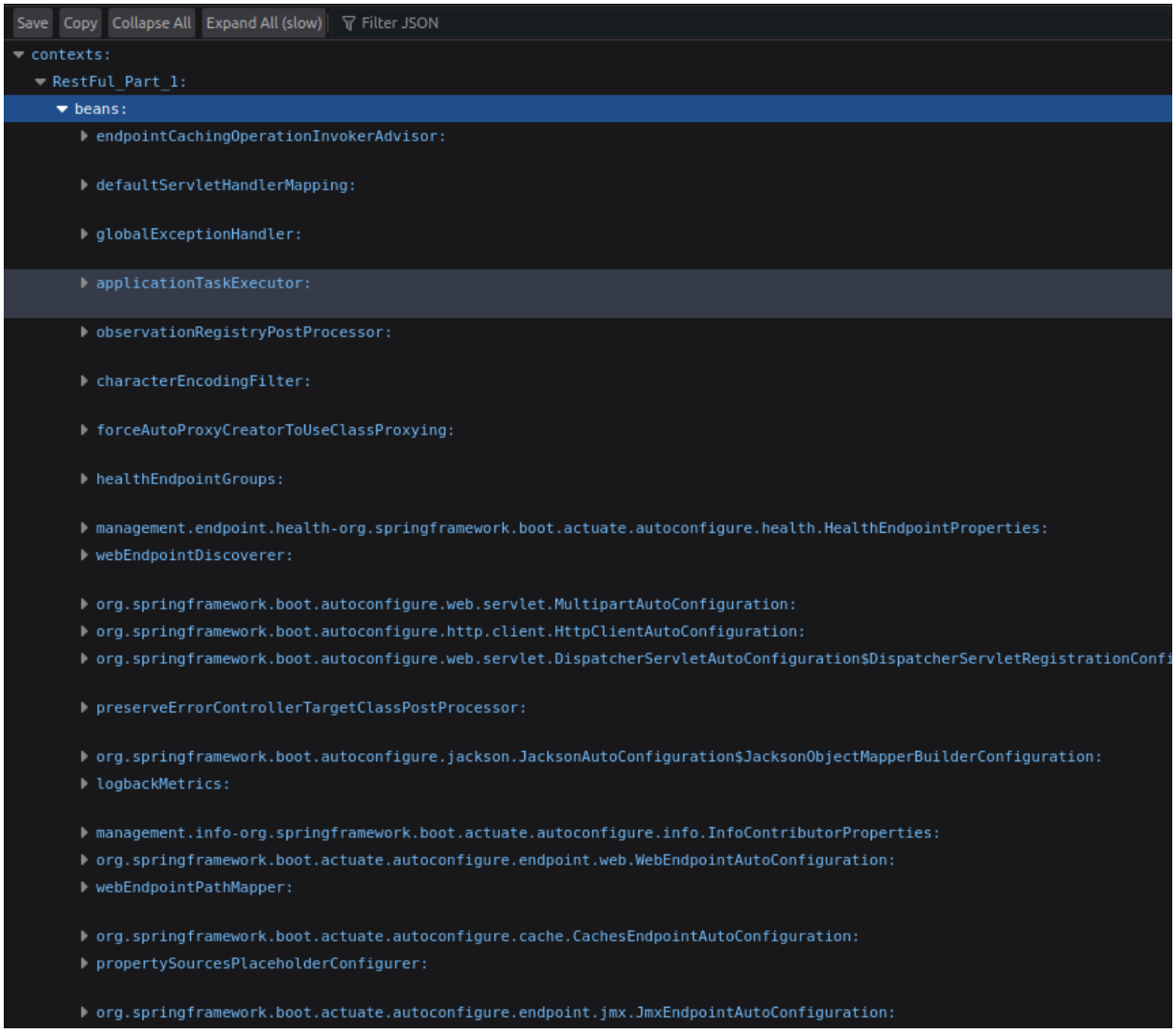
```java
@Component  16 usages
public class Employee {
    private Long id;  3 usages

    @NotBlank(message = "Name is mandatory")  3 usages
    private String name;

    @Min(value = 18, message = "Age must be >= 18")  3 usages
    private int age;

    public Employee() {}

    public Employee(Long id, String name, int age) {
        this.id = id;
        this.name = name;
        this.age = age;
    }
```


JVM / New Folder / create
Save
Share
POST
localhost:8080/api/employees
Send
Params
Authorization
Headers (9)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"age":23
}
Body
Cookies
Headers (4)
Test Results
400 Bad Request
102 ms
161 B
Save Response
Raw
Preview
Visualize
Name is mandatory

Q10)Configure actuator in your project to check the health of application and get the information about various beans configured in your application




GET Welcom
POST creat
GET getAll
GET GetOne
DEL Delete
PUT Updat
GET Actuato
No environment
JVM / New Folder / Actuators
Save
Share
GET
http://localhost:8080/actuator/beans
Send
Params
Authorization
Headers (9)
Body
Scripts
Settings
Cookies
Body
Cookies
Headers (5)
Test Results
200 OK
8 ms
113.7 KB
Save Response
JSON
Preview
Visualize
{
"contexts": {
"RestFul_Part_1": {
"beans": {
"endpointCachingOperationInvokerAdvisor": {
"aliases": [],
"scope": "singleton",
"type": "org.springframework.boot.actuate.endpoint.invoker.cache.
CachingOperationInvokerAdvisor",
"resource": "class path resource [org/springframework/boot/actuate/autoconfigure/
endpoint/EndpointAutoConfiguration.class]",
"dependencies": [
"org.springframework.boot.actuate.autoconfigure.endpoint.
EndpointAutoConfiguration",
"environment"
]
},
"defaultServletHandlerMapping": {
"aliases": [],
"scope": "singleton",
"type": "org.springframework.web.servlet.HandlerMapping",
"resource": "class path resource [org/springframework/boot/autoconfigure/web/
servlet/WebMvcAutoConfiguration$EnableWebMvcConfiguration.class]",
"dependencies": [
Postbot
Runner
Capture requests
Cookies
Vault
Trash




```xml
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```

EmployeeService.java | EmployeeController.java | Employee.java | pom.xml (RestFul_Part_1) | application.properties

```
spring.application.name=RestFul_Part_1
management.endpoints.web.exposure.include=*
management.endpoint.health.show-details=always
```