Spring Boot 기술 문서 기반

Reference & 문제 해결 능력 향상하기

Kim Hye Kyung

topickim@naver.com

https://spring.io/guides/tutorials/rest/

기술 문서의 예제

Controller 애노테이션

@RestController

controller에서 응답하는 데이터가 JSON 형식으로 반환

@ResponseBody를 각 메소드마다 선언했던 것을 한번에 사용할 수 있게 해주는 설정과 흡사

애노테이션 기반의 예외 처리

@ExceptionHandler

@Controller 또는 @RestController가 적용된
Spring Bean내에 발생하는 예외 처리 문법

해당 controller내에서만 사용되는 핸들러

```
@ExceptionHandler(Exception.class)
public String exceptionShow(Exception e) {
   System.out.println("예외 처리 전담 메소드");
   return "redirect:failShow.jsp?msg=" + e.getMessage();
}
```

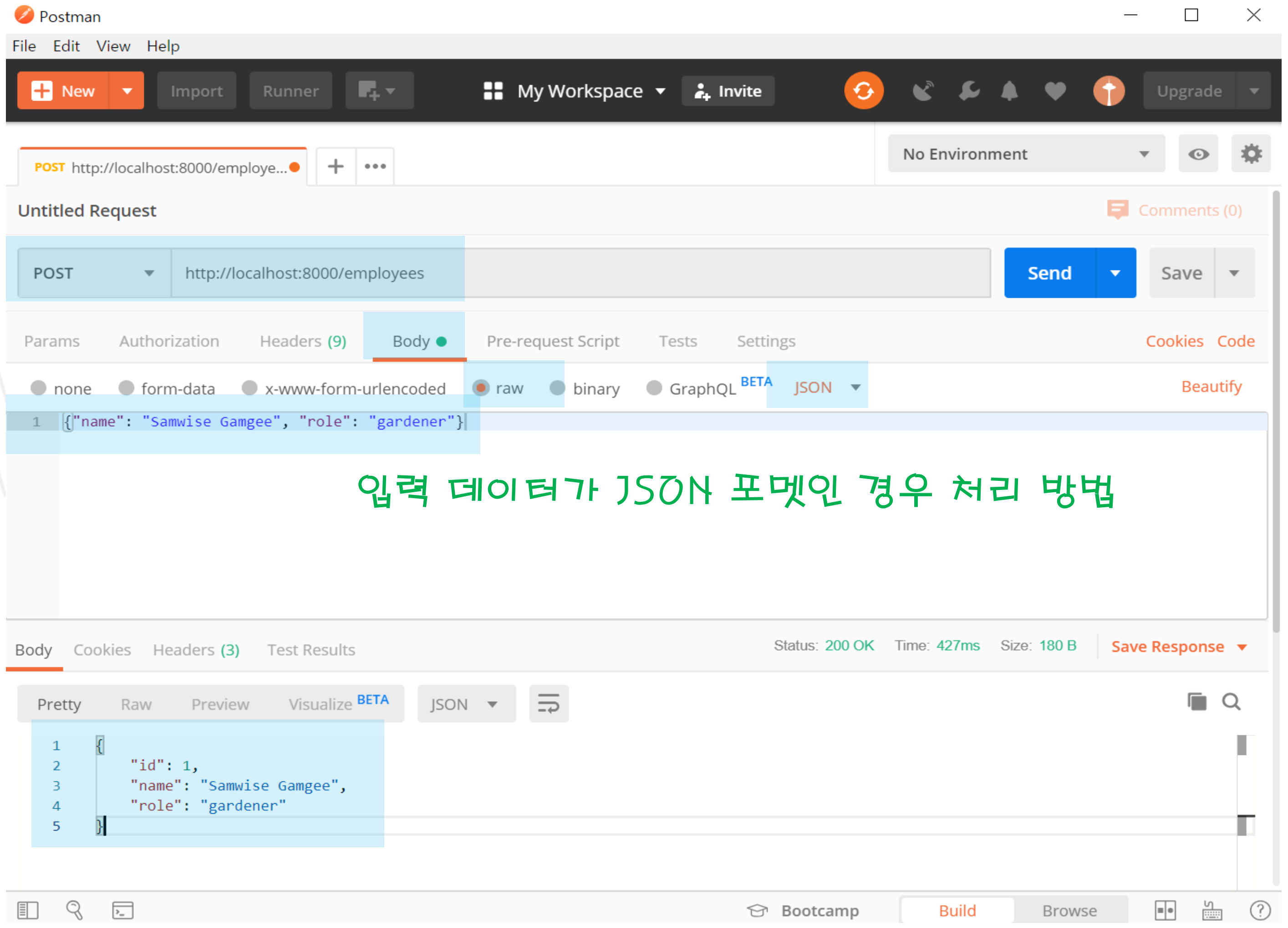
@ControllerAdvice

모든 @Controller 즉 전역에서 발생되는
예외 처리 핸들러 클래스

```
@ControllerAdvice
public class EmployeeNotFoundAdvice {

  @ResponseBody
  @ExceptionHandler(EmployeeNotFoundException.class)
  @ResponseStatus(HttpStatus.NOT_FOUND)
  public String employeeNotFoundHandler(EmployeeNotFoundException ex) {
    System.out.println("global 예외 처리");
    return ex.getMessage();
  }
}
```

post 방식으로 데이터 저장

```java
@PostMapping("/employees")
public Employee newEmployee(@RequestBody Employee newEmployee) {
    return repository.save(newEmployee);
}
```


Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
No Environment
Comments (0)
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Cookies Code
none form-data x-www-form-urlencoded raw binary GraphQL BETA
Beautify
Body Cookies Headers (3) Test Results
Save Response
Pretty Raw Preview Visualize BETA
Bootcamp Build Browse

POST http://localhost:8000/employe...

Untitled Request

POST http://localhost:8000/employees Send Save

[✓ raw] JSON

```
{"name": "Samwise Gamgee", "role": "gardener"}
```

입력 데이터가 JSON 포멧인 경우 처리 방법

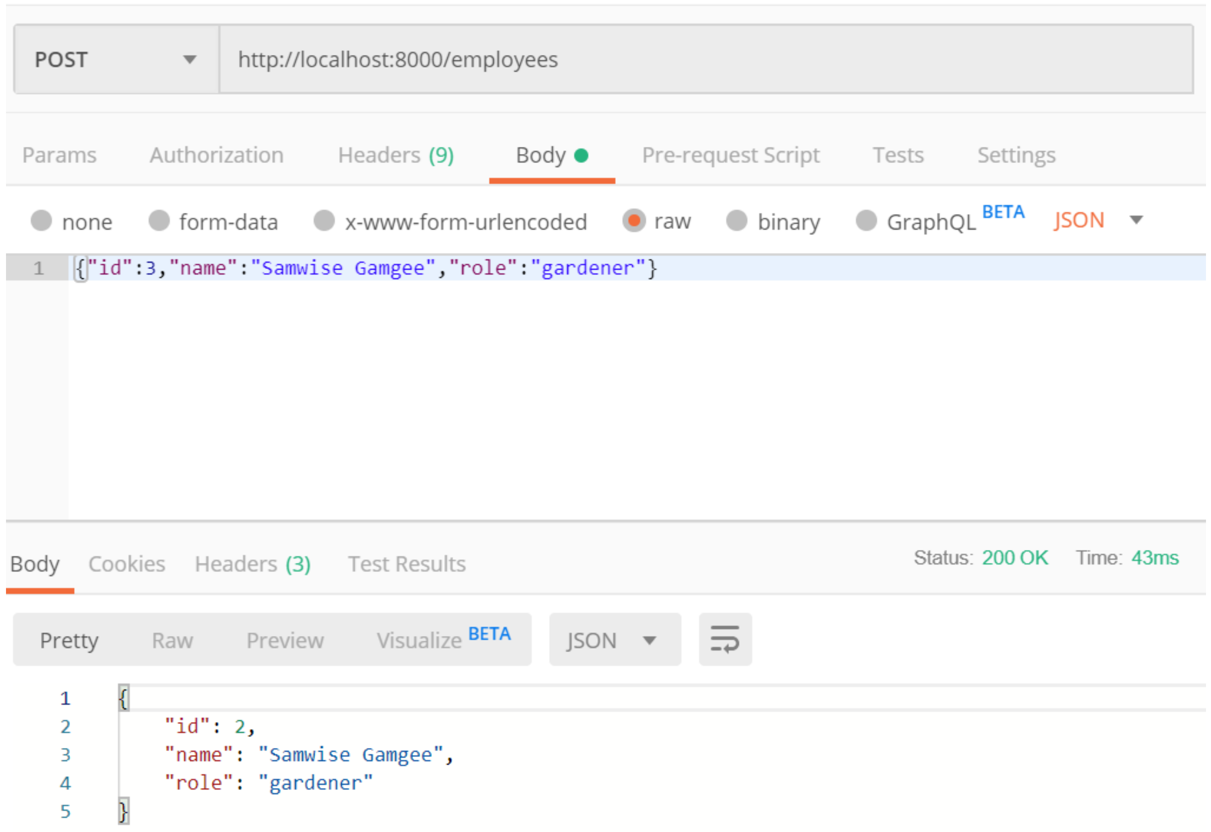
Status: 200 OK Time: 427ms Size: 180 B

JSON

```
{
    "id": 1,
    "name": "Samwise Gamgee",
    "role": "gardener"
}
```

post방식으로 데이터 저장 2

- id:3으로 저장해도 id값은 2로 자동 반영



POST http://localhost:8000/employees

Params Authorization Headers (9) Body Pre-request Script Tests Settings

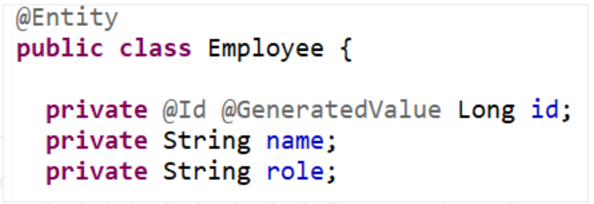
none form-data x-www-form-urlencoded [✓ raw] binary GraphQL BETA JSON

```
{"id":3,"name":"Samwise Gamgee","role":"gardener"}
```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 43ms

Pretty Raw Preview Visualize BETA JSON

```
{
    "id": 2,
    "name": "Samwise Gamgee",
    "role": "gardener"
}
```

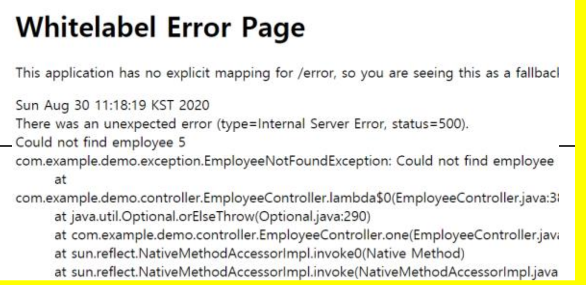


```
@Entity
public class Employee {

  private @Id @GeneratedValue Long id;
  private String name;
  private String role;
```

get 방식으로 데이터 검색

- 존재하는 데이터 검색 & 미존재하는 데이터 검색

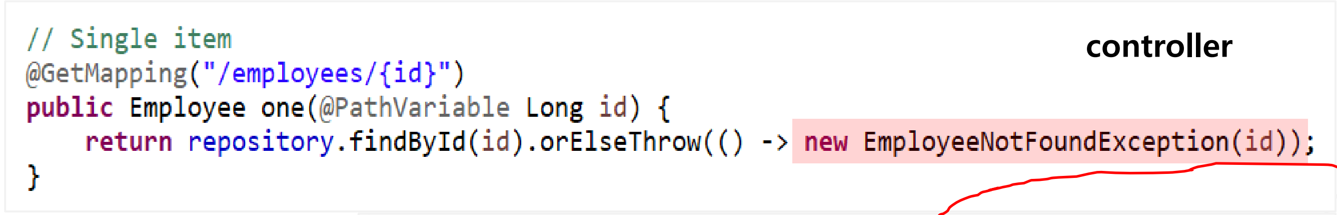
생략시 Whitelabel Error Page 실행



Whitelabel Error Page

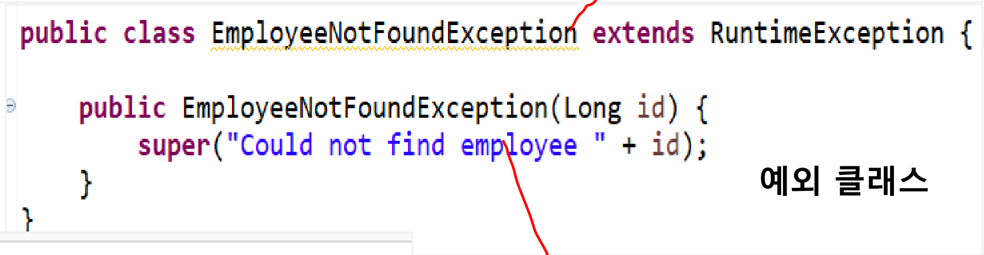
```
This application has no explicit mapping for /error, so you are seeing this as a fallbac

Sun Aug 30 11:18:19 KST 2020
There was an unexpected error (type=Internal Server Error, status=500).
Could not find employee 5
com.example.demo.exception.EmployeeNotFoundException: Could not find employee
        at
com.example.demo.controller.EmployeeController.lambda$0(EmployeeController.java:3
        at java.util.Optional.orElseThrow(Optional.java:290)
        at com.example.demo.controller.EmployeeController.one(EmployeeController.java
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java
```

controller

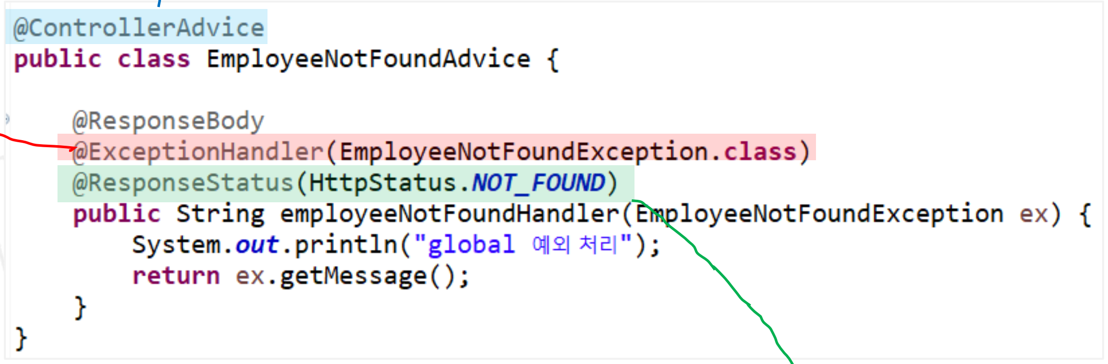


```java
// Single item
@GetMapping("/employees/{id}")
public Employee one(@PathVariable Long id) {
    return repository.findById(id).orElseThrow(() -> new EmployeeNotFoundException(id));
}
```

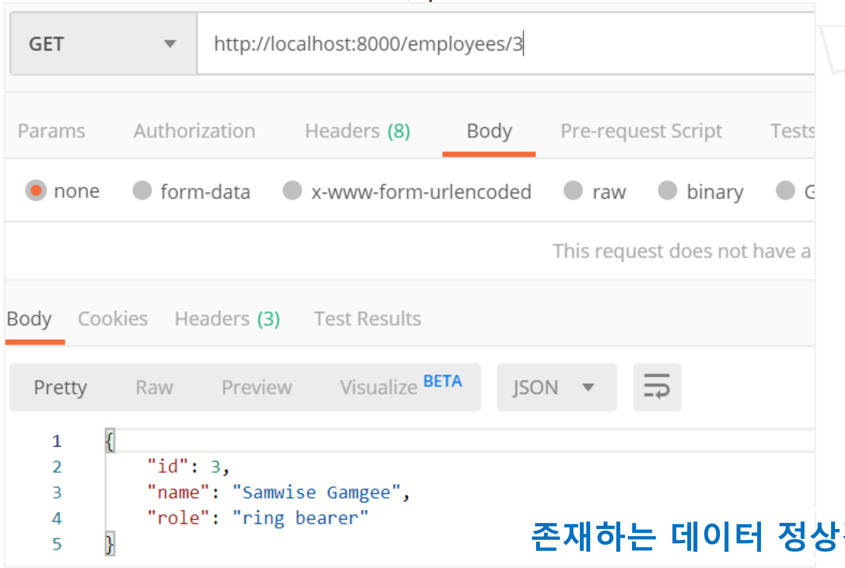
예외 클래스



```java
public class EmployeeNotFoundException extends RuntimeException {

    public EmployeeNotFoundException(Long id) {
        super("Could not find employee " + id);
    }
}
```



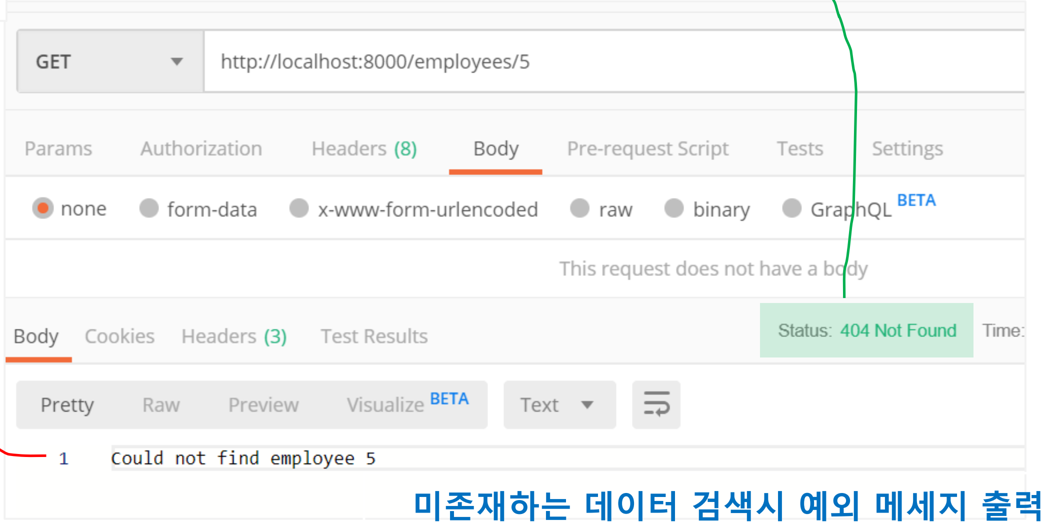
```java
@ControllerAdvice
public class EmployeeNotFoundAdvice {

    @ResponseBody
    @ExceptionHandler(EmployeeNotFoundException.class)
    @ResponseStatus(HttpStatus.NOT_FOUND)
    public String employeeNotFoundHandler(EmployeeNotFoundException ex) {
        System.out.println("global 예외 처리");
        return ex.getMessage();
    }
}
```



GET http://localhost:8000/employees/3

```
{
    "id": 3,
    "name": "Samwise Gamgee",
    "role": "ring bearer"
}
```

존재하는 데이터 정상검색



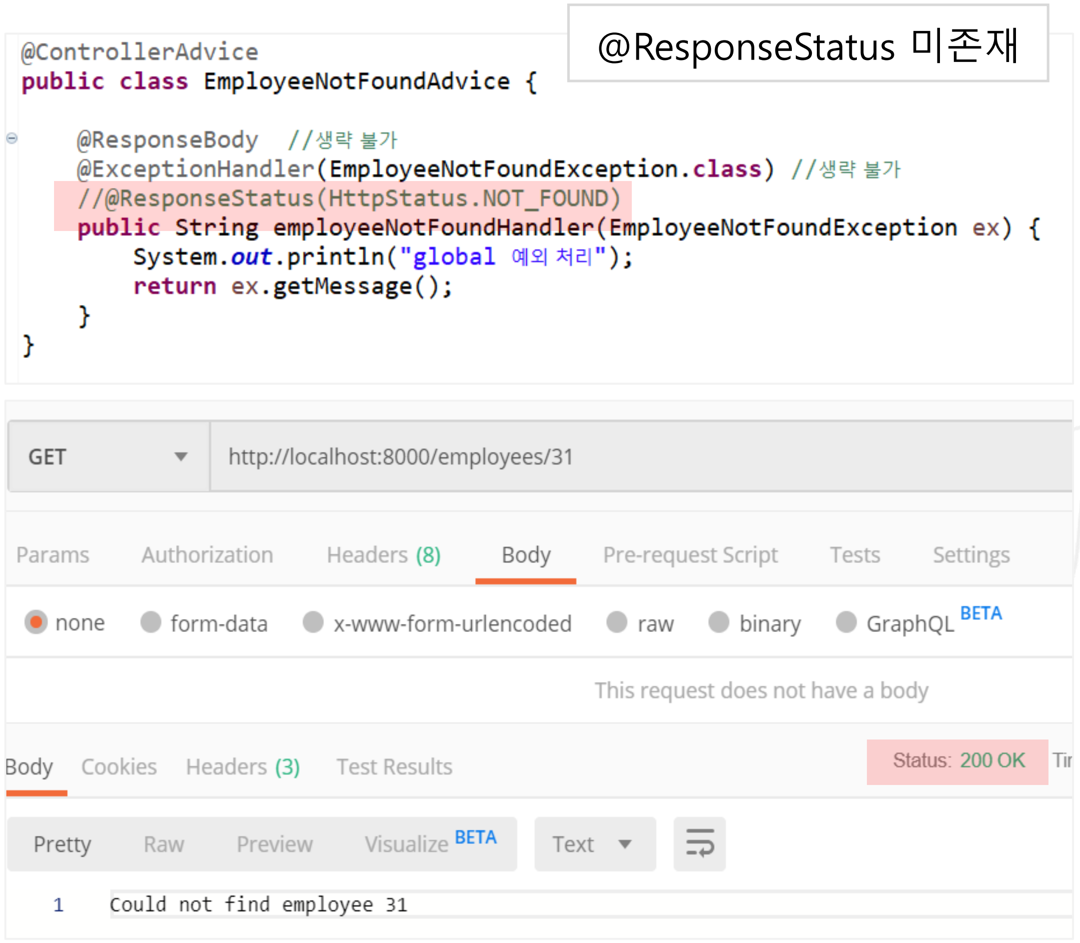
GET http://localhost:8000/employees/5

Status: 404 Not Found

```
Could not find employee 5
```

미존재하는 데이터 검색시 예외 메세지 출력

@ResponseStatus(HttpStatus.NOT_FOUND) 여부에 따른 실행 결과



@ResponseStatus 미존재

```java
@ControllerAdvice
public class EmployeeNotFoundAdvice {

    @ResponseBody  //생략 불가
    @ExceptionHandler(EmployeeNotFoundException.class) //생략 불가
    //@ResponseStatus(HttpStatus.NOT_FOUND)
    public String employeeNotFoundHandler(EmployeeNotFoundException ex) {
        System.out.println("global 예외 처리");
        return ex.getMessage();
    }
}
```

GET http://localhost:8000/employees/31

Params | Authorization | Headers (8) | Body | Pre-request Script | Tests | Settings

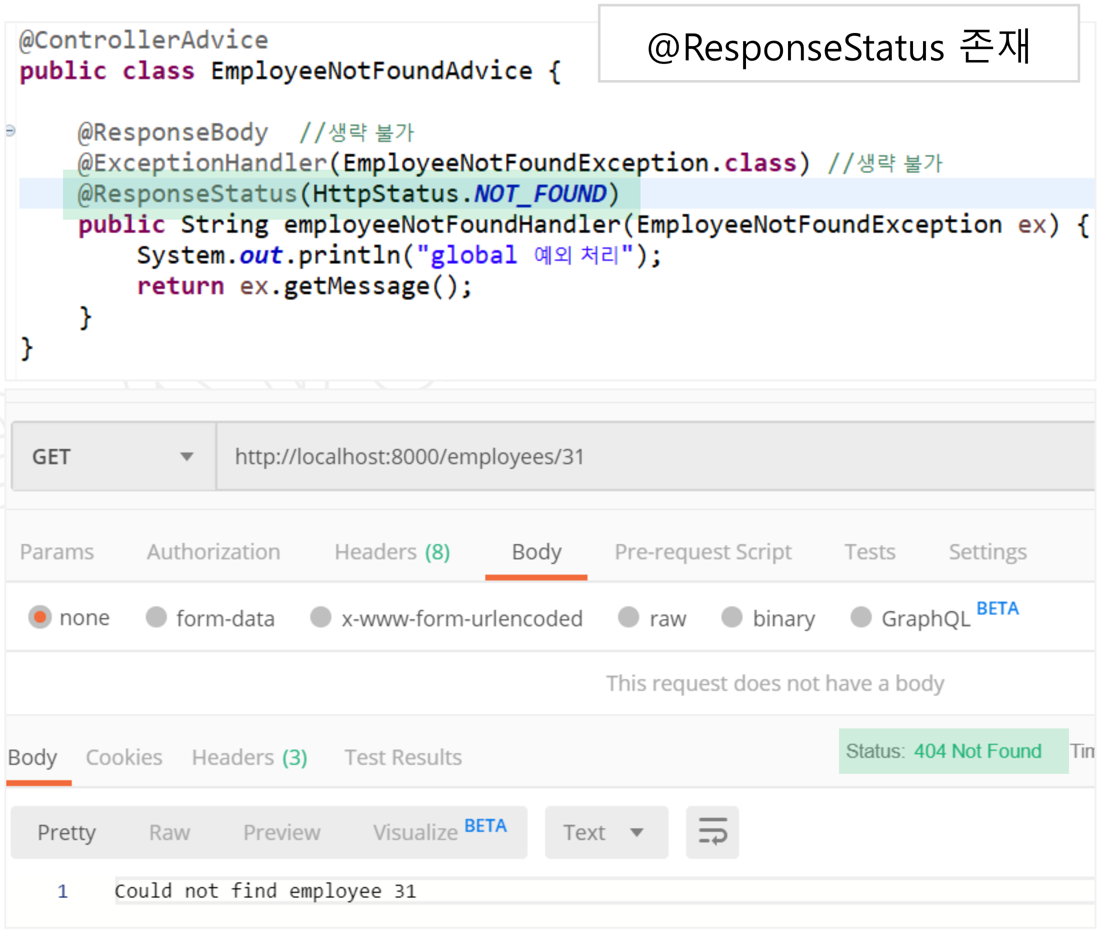
none | form-data | x-www-form-urlencoded | raw | binary | GraphQL BETA

This request does not have a body

Body | Cookies | Headers (3) | Test Results — Status: 200 OK

Pretty | Raw | Preview | Visualize BETA | Text

```
Could not find employee 31
```



@ResponseStatus 존재

```java
@ControllerAdvice
public class EmployeeNotFoundAdvice {

    @ResponseBody  //생략 불가
    @ExceptionHandler(EmployeeNotFoundException.class) //생략 불가
    @ResponseStatus(HttpStatus.NOT_FOUND)
    public String employeeNotFoundHandler(EmployeeNotFoundException ex) {
        System.out.println("global 예외 처리");
        return ex.getMessage();
    }
}
```

GET http://localhost:8000/employees/31

Params | Authorization | Headers (8) | Body | Pre-request Script | Tests | Settings

none | form-data | x-www-form-urlencoded | raw | binary | GraphQL BETA

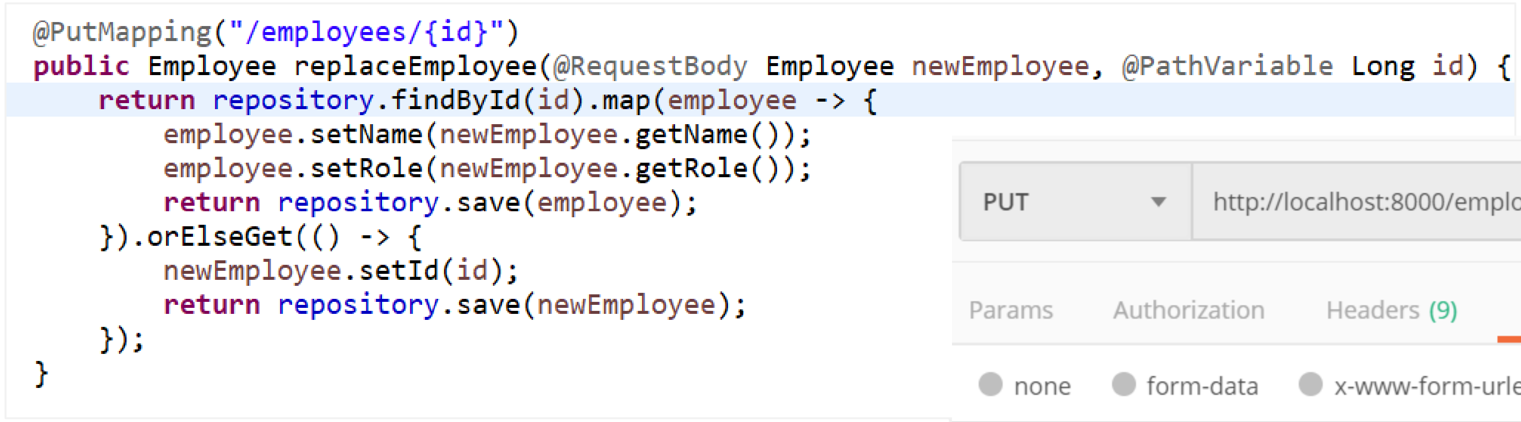
This request does not have a body

Body | Cookies | Headers (3) | Test Results — Status: 404 Not Found

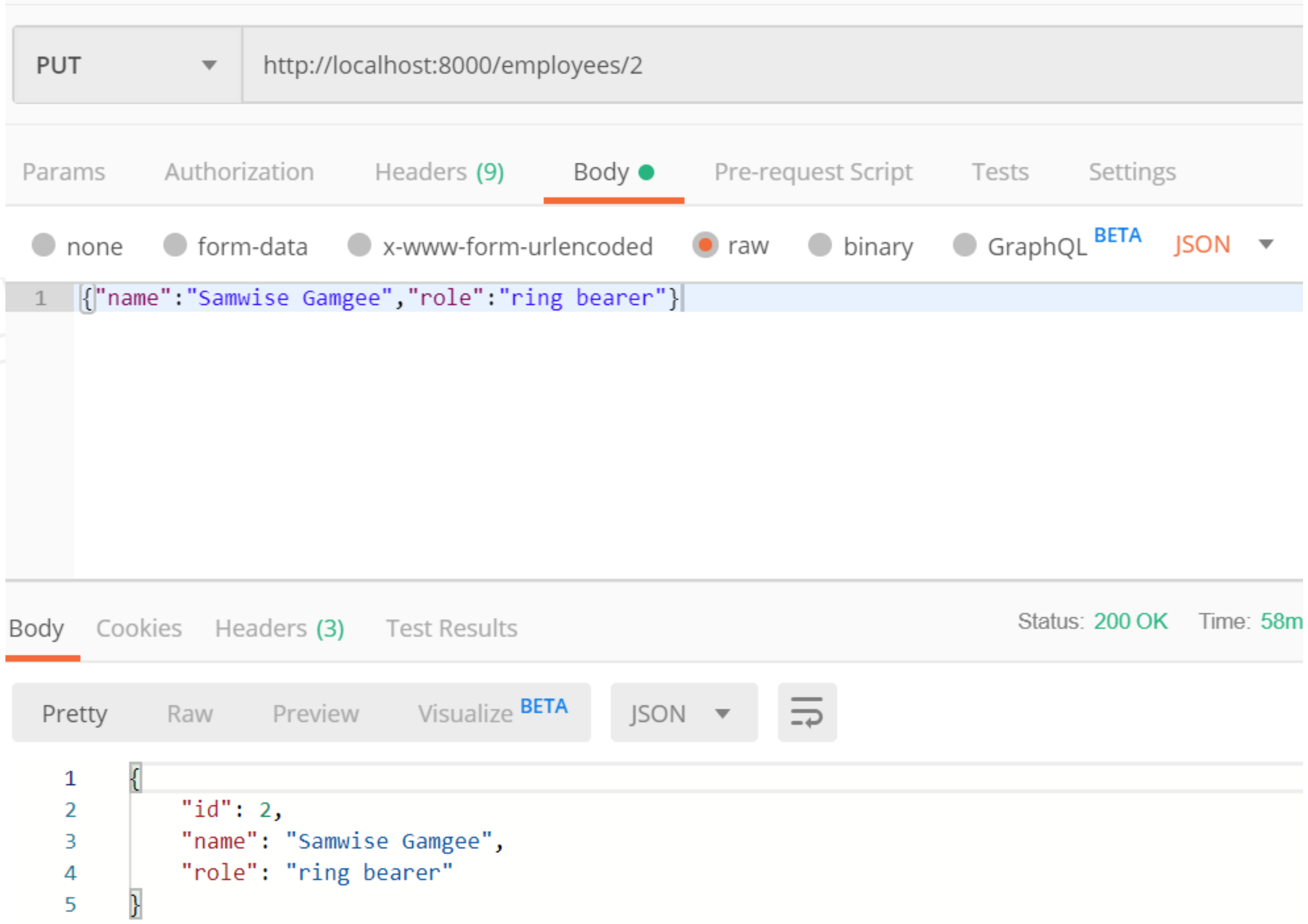
Pretty | Raw | Preview | Visualize BETA | Text

```
Could not find employee 31
```

put 방식으로 이미 존재하는 데이터 수정



```
@PutMapping("/employees/{id}")
public Employee replaceEmployee(@RequestBody Employee newEmployee, @PathVariable Long id) {
    return repository.findById(id).map(employee -> {
        employee.setName(newEmployee.getName());
        employee.setRole(newEmployee.getRole());
        return repository.save(employee);
    }).orElseGet(() -> {
        newEmployee.setId(id);
        return repository.save(newEmployee);
    });
}
```



http://localhost:8000/employees/2

Body: [✓ raw] JSON

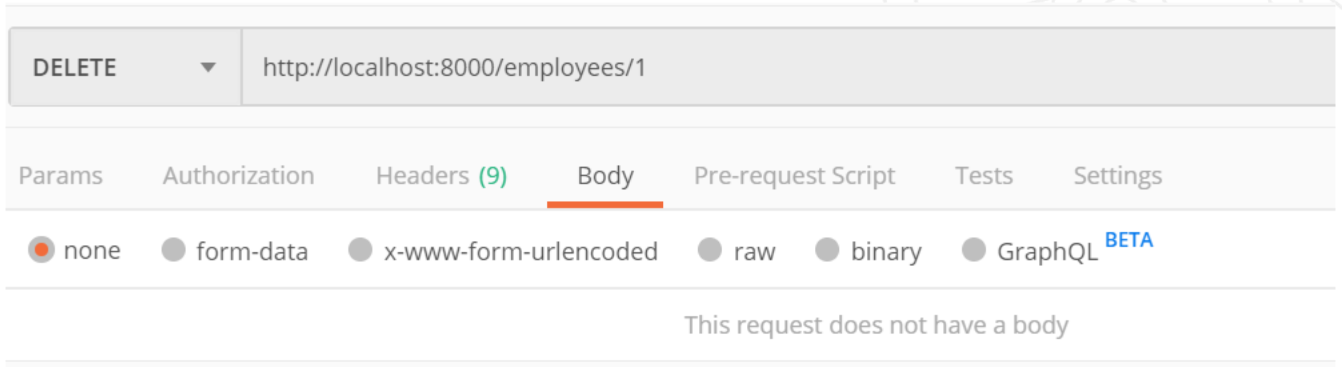
```
{"name":"Samwise Gamgee","role":"ring bearer"}
```

Status: 200 OK Time: 58m

```
{
    "id": 2,
    "name": "Samwise Gamgee",
    "role": "ring bearer"
}
```

delete 방식으로 id값을 활용하여 직원 삭제

```java
@DeleteMapping("/employees/{id}")
public void deleteEmployee(@PathVariable Long id) {
    repository.deleteById(id);
}
```



DELETE http://localhost:8000/employees/1

Params Authorization Headers (9) Body Pre-request Script Tests Settings

[✓ none] form-data x-www-form-urlencoded raw binary GraphQL BETA

This request does not have a body