

# Spring Boot

Part6

---

*ResponseEntity<T>를 사용한 응답*

React에서 Spring 컨트롤러로 요청이 전달되면, 컨트롤러에서는 필요한 데이터를 return을 통해 React로 전달하였다. 이것을 응답이라 한다.

우리는 컨트롤러에서 필요한 코드 실행 후 React로 가져갈 데이터가 있으면 return을 사용하였고, React로 전달할 데이터가 없다면 return을 사용하지 않았다. 하지만 이러한 코드 작성은 http 응답을 이루는 상태코드(status), 헤더(header), 본문(body) 중 본문만 사용한 것으로, 응답에 대한 세밀한 컨트롤을 할 수 없다.

## HTTP 응답 구조

상태코드(status code) : 응답의 상태를 알수 있는 정보로 200~500번대의 숫자로 이루어져 있다.

HTTP 헤더(header) : http 응답에 대한 모든 부가 정보가 들어가 있음. 데이터는 key-value형태로 이루어져 있음.

HTTP 본문(body) : 실제 전송하는 데이터(html 문서, 이미지, json데이터 등)

## ResponseEntity<T>

Spring Boot에서 HTTP 응답을 보다 세밀하게 제어할 수 있도록 해주는 클래스로 클라이언트에게 상태 코드, 헤더, 본문(body)까지 원하는 대로 설정해서 응답결과를 전달할 수 있게 해준다.

## 상태코드(status code)

범주	설명	예시 코드
2xx	성공 (성공적으로 처리됨)	200 OK : 요청 성공 201 Created: 리소스 생성 성공 (POST 후) 204 No Content: 성공했지만 반환할 본문 없음
3xx	리다이렉션 (다른 위치로 이동 필요)	
4xx	클라이언트 오류 (요청 잘못된)	400 Bad Request: 요청이 잘못된 (형식 오류 등) 401 Unauthorized: 인증 필요 403 Forbidden: 권한 없음 404 Not Found: 리소스 없음
5xx	서버 오류 (서버 처리 중 문제 발생)	500 Internal Server Error: 서버에서 에러 발생

## ResponseEntity<T> 사용 예제

```
@GetMapping("")
public ResponseEntity<String> testResponseEntity1(){
    return ResponseEntity.status(HttpStatus.OK).body("hello");
}
```

```
@GetMapping("")
public ResponseEntity<String> testResponseEntity2(){
    return ResponseEntity.ok("hello");
}
```

```
@GetMapping("")
public ResponseEntity<BookDTO> testResponseEntity3(BookDTO bookDTO){
    return ResponseEntity.status(HttpStatus.OK).body(bookDTO);
}
```

```
@GetMapping("")
public ResponseEntity<BookDTO> testResponseEntity4(BookDTO bookDTO){
    return ResponseEntity.ok(bookDTO);
}
```

## ResponseEntity<T> 사용 예제 - 2

*// 404 상태 및 본문에 데이터 추가*

```
@GetMapping("/notfound")
public ResponseEntity<String> notFound() {
    return ResponseEntity.status(HttpStatus.NOT_FOUND).body("요청하신 리소스를 찾을 수 없습니다.");
}
```

*// 200 상태, 헤더 추가, 본문 데이터 추가*

```
@GetMapping("/custom-header")
public ResponseEntity<String> withHeader() {
    HttpHeaders headers = new HttpHeaders();
    headers.add("myName", "hong");
    headers.add("myAge", "30");

    return ResponseEntity.status(HttpStatus.OK).headers(headers).body("성공");
}
```

*// insert 성공 + 본문없이. 예시에서는 GetMapping을 사용했지만, postMapping 사용해야 함.*

```
@GetMapping("/users")
public ResponseEntity<BookDTO> createUser(@RequestBody BookDTO user) {
    return ResponseEntity.status(HttpStatus.CREATED).build(); //201
}
```