ResponseEntity

- ✓

 SBDemo1 [boot] [devtools]
 - src/main/java
 - > # com.rit
 - > # com.rit.p1intro
 - > # com.rit.p2mapping
 - > # com.rit.p3aop
 - > # com.rit.p4crud
 - > 🚜 com.rit.p5crud
 - > # com.rit.p6request
 - > 拇 com.rit.p7crud
 - > # com.rit.p8jpa
 - - > ResponseController.java

ResponseEntity

- ResponseEntity is a class that represents the entire HTTP response.
- It allows you to customize the response, when building RESTful web services with
 - HTTP response status,
 - headers, and
 - body

```
    ■ ResponseController.java ×
19
   @RestController
   @RequestMapping("response")
22 public class ResponseController {
23
24⊖
        @Autowired
25
        private StudentRepository repo;
26
        @PostMapping
27⊝
        public ResponseEntity<Student> addStud(@RequestBody Student stud) {
28
            Student student = repo.save(stud);
29
30
31
            HttpHeaders headers = new HttpHeaders();
            headers.add("login", "success");
32
            headers.add("name", "Vijay");
33
34
35
            return new ResponseEntity<>(student,headers,HttpStatus.CREATED);
36
27
```

```
*ResponseController.java ×
        @GetMapping
38⊜
        public ResponseEntity<List<Student>> viewAll() {
39
            HttpHeaders headers = new HttpHeaders();
40
            headers.add("login", "success");
41
42
43
44
45
            return ResponseEntity
                     .ok()
                     .headers(headers).body(repo.findAll());
46
        @GetMapping("/id/{id}")
47⊖
        public ResponseEntity<Student> viewStud(@PathVariable Long id) {
48
            HttpHeaders headers = new HttpHeaders();
49
            headers.add("login", "success");
50
51
            return ResponseEntity
52
53
                     .status(HttpStatus.OK)
                     .headers(headers).body(repo.findById(id).get());
```

```
*ResponseController.java ×
55
56⊜
       @PutMapping
        public ResponseEntity<?> updateStud(@RequestBody Student stud) {
57
58
59
            if(repo.findById(stud.getRoll()).isEmpty())
                return ResponseEntity
60
                         .status(HttpStatus.NOT_FOUND)
61
62
                         .body("Given Record Doesnt exist to update");
63
            else
64
                return ResponseEntity
                         .ok(repo.save(stud));
65
66
67
```

Exception Handling

Custom Exception

- > # com.rit.p8jpa
- > 🏨 com.rit.p9resp
- - AlreadyExistException.java
 - > CustExcepController.java
 - > NoSuchElementException.java

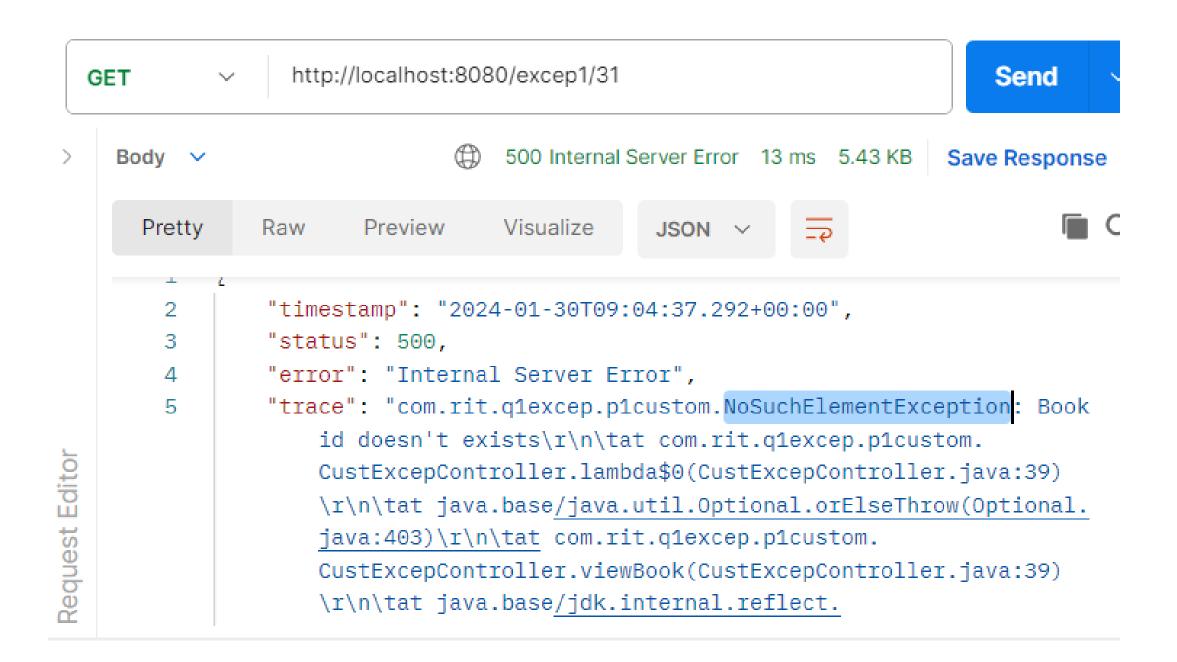
```
AlreadyExistException.java X
 1 package com.rit.q1excep.p1custom;
   public class AlreadyExistException extends RuntimeException {
 5
       private static final long serialVersionUID = 1L;
       public AlreadyExistException() {
 7⊝
           super();
 9
10
       public AlreadyExistException(String message) {
11⊖
12
           super(message);
13
14
15 }
```

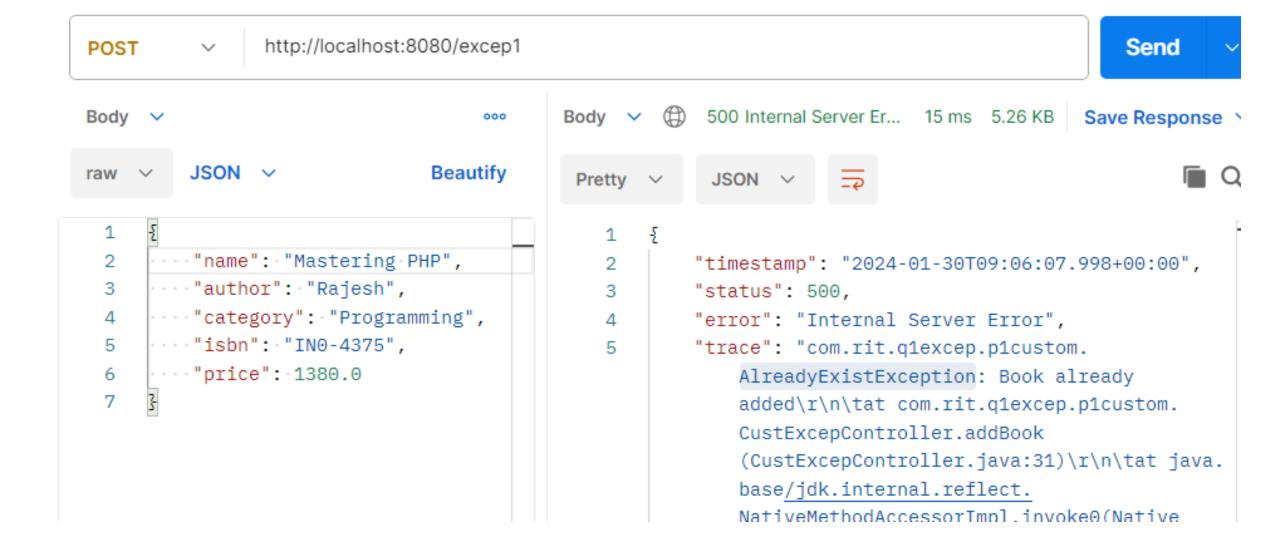
```
AlreadyExistException.java

    NoSuchElementException.java 
    X

 1 package com.rit.q1excep.p1custom;
   public class NoSuchElementException extends RuntimeException {
        private static final long serialVersionUID = 1L;
 5
 6
        public NoSuchElementException() {
 7⊝
             super();
 8
10
        public NoSuchElementException(String message) {
11⊖
             super(message);
12
13
14
15 }
```

```
AlreadyExistException.java
                 *CustExcepController.java ×
   @RequestMapping("excep1")
   public class CustExcepController {
20
21<sup>©</sup>
22
<u>23</u>
24<sup>©</sup>
        @Autowired
        private BookRepository repo;
        @PostMapping
25
        public Book addBook(@RequestBody Book book) {
26
            if(repo.findByName(book.getName()).isPresent())
                 throw new AlreadyExistException("Book already added");
27
28
29
            return repo.save(book);
30
31
32⊝
        @GetMapping("/{id}")
33
        public Book viewBook(@PathVariable long id) {
34
35
36
            return repo.findById(id)
            .orElseThrow(()-> new NoSuchElementException("Book id doesn't exists"));
```





Exception Handling

Global Exception with custom response

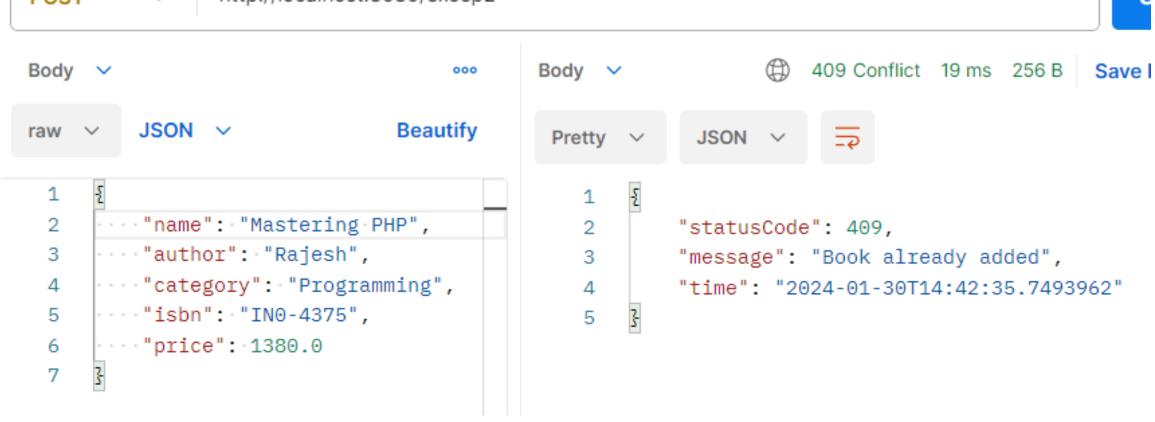
- pa commuporcop
- # com.rit.q1excep.p1custom
- - AdviceExcepController.java
 - AlreadyExistException.java
 - › D CustomHandler.java
 - ErrorResponse.java
 - NoSuchElementException.java

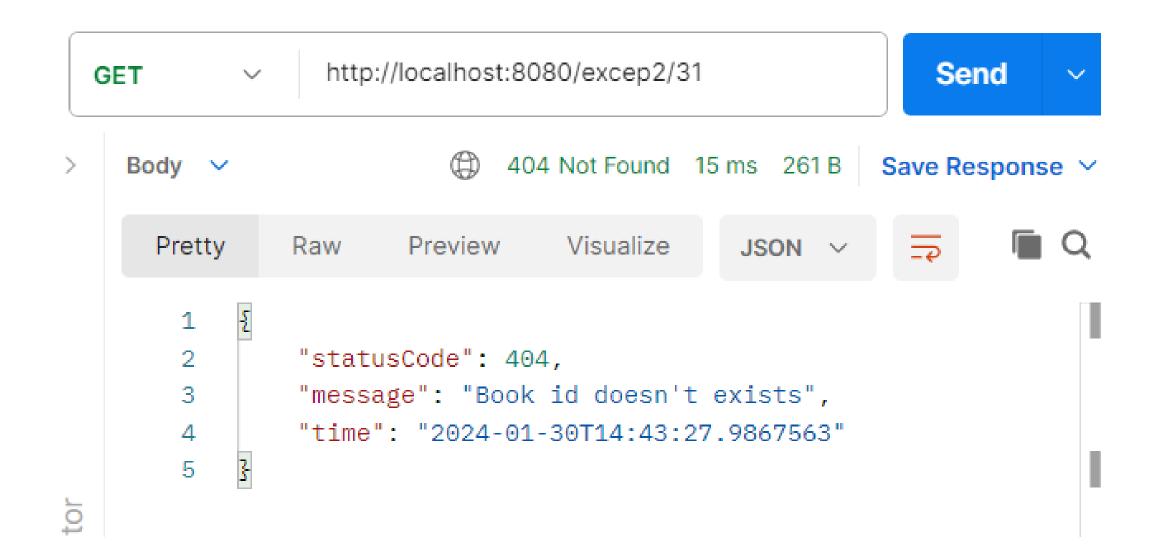
```
*ErrorResponse.java ×
   public class ErrorResponse {
        private int statusCode;
       private String message;
       private LocalDateTime time;
        public ErrorResponse() {
10⊝
11
            super();
12
13
        public ErrorResponse(String message, int status) {
14⊝
            this.statusCode = status;
15
            this.message = message;
16
            this.time = LocalDateTime.now();
17
18
19
```

```
Custom Handler.java ×
   @RestControllerAdvice
   public class CustomHandler {
11
       @ExceptionHandler(NoSuchElementException.class)
12⊝
       @ResponseStatus(HttpStatus.NOT_FOUND)
13
       public ErrorResponse handler1(NoSuchElementException ex) {
14
            return new ErrorResponse(ex.getMessage(),
15
                    HttpStatus.NOT_FOUND.value());
16
17
18
       @ExceptionHandler(AlreadyExistException.class)
19⊜
       @ResponseStatus(HttpStatus.CONFLICT)
20
21
       public ErrorResponse handler2(AlreadyExistException ex) {
            return new ErrorResponse(ex.getMessage(),
22
                    HttpStatus.CONFLICT.value());
23
24
25
26
```

```
*ErrorResponse.java
             *AdviceExcepController.java ×
16 @RestController
   @RequestMapping("excep2")
18 public class AdviceExcepController {
19
       @Autowired
20⊝
       private BookRepository repo;
21
22
23⊝
       @PostMapping
       public Book addBook(@RequestBody Book book) {
24
            if(repo.findByName(book.getName()).isPresent())
25
                throw new AlreadyExistException("Book already added");
26
27
28
            return repo.save(book);
29
30
31⊜
       @GetMapping("/{id}")
       public Book viewBook(@PathVariable long id) {
32
            return repo.findById(id)
33
                     .orElseThrow(()-> new NoSuchElementException("Book id does
34
35
```

```
http://localhost:8080/excep2
POST
```





Thank you