

Handlers and Response Entity

→ Handlers are Exception handlers.

① Global exception handlers -

→ we use this because spring returns ugly HTML errors

→ with handler: you control JSON response format

It gives:

- cleaner API
- consistent error format
- production ready behaviour.

use `@RestControllerAdvice`

② Response Entity -

It lets us control -

- HTTP status code
- Headers
- Response body.

Eg → `return ResponseEntity.status(201).body(saved data)`

↪ meaning status 201 created
+ body = saved data

Global exception
handlers Maps
Errors along
with messages

```
GlobalExceptionHandler.java
2
3 import org.springframework.http.ResponseEntity;
4 import org.springframework.web.bind.annotation.ExceptionHandler;
5 import org.springframework.web.bind.annotation.RestControllerAdvice;
6
7 import java.util.HashMap;
8 import java.util.Map;
9
10 @RestControllerAdvice no usages
11 public class GlobalExceptionHandler {
12     @ExceptionHandler(StudentNotFoundException.class) no usages
13     public ResponseEntity<Map<String, String>> handleNotFound(StudentNotFoundException ex){
14         Map<String, String> response = new HashMap<>();
```

```
15 response.put("Error", ex.getMessage());  
16 return ResponseEntity.status(404).body(response);  
17 }  
18 }
```

Response
Entity
provides
status along
with body.

StudentNotFoundException.java

```
1 package com.example.Day5.exception;  
2  
3 public class StudentNotFoundException extends RuntimeException{  
4     public StudentNotFoundException(Long id) { 3 usages  
5         super("Student not found with id: " + id);  
6     }  
7 }
```