

Exception Handling

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- Controller Advice
- Exception Handler
- Exceptions customizadas
- Programas

Vamos falar de erros, no Java temos tratamentos de exception através de trys e catches, mas para utilizarmos esse tratamento de erros nos endpoints de um programa Spring Boot precisamos adicionar uma Classe com a anotação `@ControllerAdvice` e cada tipo de exception será tratada em um método com a Anotação `@ExceptionHandler`.

Vamos ver um pouco mais dessa classe e como podemos juntar os erros que temos no programa com os status que temos disponíveis nos métodos HTTP.

Controller Advice

```
@ControllerAdvice
public class GlobalHandler {
    @ExceptionHandler(Exception.class) // exception handled
    public ResponseEntity<ErrorResponse> handleExceptions (
        Exception e
    ) {
        // ... logica

        return ResponseEntity.internalServerError().body(new
        ErrorResponse());
    }
}
```

Essa classe acima é uma resposta que é acionada sempre que temos uma exception no programa, cada método é uma resposta para uma determinada exception.

Exception Handler

A resposta dessa classe será dada no endpoint em que ocorreu o erro. Então é como se estivéssemos “procurando pelos erros”. Podemos ter tanto exceptions que já existem no Java, como podemos ter exceptions criadas no programa nesse tipo de tratativa.

Podemos criar exceptions para alguns cenários no nosso programa e tratá-los através dos `ExceptionHandler`.

Exception Handler

A classe abaixo é um erro que estende da exception `RuntimeException` nós temos que utilizar esse erro em blocos try/catch ou em validações para que ele seja encontrado pelo handler.

```
public class NotFoundException extends RuntimeException {  
    public NotFoundException () {  
        super ();  
    }  
}
```

```
    public NotFoundException (String message) {  
        super (message);  
    }  
}
```

INTERVALO DE AULA

DEV!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

Nos vemos em 20 minutos.

Início: 20:20

Retorno: 20:40



Exceptions customizadas

No Java podemos criar exceptions novas estendendo exceptions já existentes:

```
public class ServerErrorException extends RuntimeException {  
    public ServerErrorException () {  
        super ();  
    }  
  
    public ServerErrorException (String message) {  
        super (message);  
    }  
}
```

Essa exception criada tem 2 construtores, um que não recebe nenhum parâmetro e outro que recebe uma mensagem e adiciona a mensagem a Exception.

Exceptions customizadas

Para retornarmos essa exception Customizada temos que adicionar um Exception Handler para ela:

```
@ExceptionHandler(ServerErrorException.class)
public ResponseEntity<ErrorResponse> handleServerErrorExceptions (
    Exception e
) {
    return new ResponseEntity<ErrorResponse>(new
ErrorResponse (LocalDateTime.now(), HttpStatus.INTERNAL_SERVER_ERROR.value()
, e), HttpStatus.INTERNAL_SERVER_ERROR);
}
```

Exceptions customizadas

Podemos utilizar essa exception no nosso programa seguindo o seguinte exemplo:

```
public List<NotaResponse> encontraTodasNotas () {  
    try {  
        // lógica  
    } catch (Exception e) {  
        throw new ServerErrorException ("Erro ao buscar notas");  
    }  
}
```

Vamos adicionar ao nosso programa tratamentos de erro:

1. Adicione 3 exceptions para representar os status 404, 400 e 500
2. Adicione um Hadler Global
3. Adicione um Exception Handler para cada exception criada

AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>