

NestJS Interceptors

Os Interceptors nos entregam um conjunto de recursos capazes de:

- Adicionar lógicas antes / depois da execução de nossos métodos



- Transformar o resultado de uma função
- Transformar uma exception lançada por uma função
- Ampliar o comportamento de uma função
- Dependendo de condições específicas, sobrescrever completamente uma função (exemplo: fins de cache)

Características de implementação:

- Um interceptor deve implementar a interface `NestInterceptor` e deverá sobrescrever o método `intercept()`, onde teremos acesso dois argumentos:
 - **Instância de ExecutionContext** que herda de `ArgumentsHost`. Nós já interagimos com este objeto, quando implementamos nosso Tratamento global de exceção, onde exploramos os recursos de "Execution Context" que o Nest nos entrega.
 - Um `CallHandler`. Esta é uma interface que implementa o método `handle()`, que podemos utilizar para acionar um route handle method em algum ponto do nosso interceptor. Se não invocarmos o método `handle()`, em nossa implementação do método `intercept()`, nosso route handle method não será executado. É importante ressaltar que este método `handle()` retorna um observable, onde podemos utilizar o poder dos operadores RxJs para manipular o response, interagindo antes ou depois da execução do nosso route handle method.