



Serviço Nacional de Aprendizagem Industrial

PELO FUTURO DO TRABALHO

Melhorias Controlador

Desenvolvimento de Sistemas

Prof. Me. Reneilson Santos

Abril/2024

Agenda

→ DTO



DTO



DTO

Objeto de Transferência de Dados (do inglês, *Data transfer object*, ou simplesmente DTO), é um padrão de projeto de software usado para **transferir dados entre subsistemas** de um software.

DTOs são frequentemente usados em conjunção com objetos de acesso a dados para **obter dados de um banco de dados**.

Os DTOs também podem ser chamados de "Serializers", pois são utilizados para serializar nossos objetos Java.

No Spring Boot podemos usar DTOs para entrada e saída de dados, fazendo as devidas validações.

DTO

Podemos criar classes de DTO dentro do nosso projeto, no entanto, para simplificar a criação, visto serem objetos imutáveis, torna-se mais cômodo a criação de registros (**Records**) ao invés de classes, definindo apenas os parâmetros e possivelmente algumas validações usando anotações.

DTO (com Record)

```
public record PetInputDTO(  
    String name,  
    Status status  
) { }
```

```
public record PetOutputDTO(  
    Long id,  
    String name,  
    Status status  
) { }
```

DTO (com classe)

```
@Data @AllArgsConstructor  
public class PetInput {  
    private String name;  
    private Status status;  
}
```

```
@Data @AllArgsConstructor  
public class PetOutput {  
    private Long id;  
    private String name;  
    private Status status;  
}
```


DTO

É possível, portanto, criar diversos DTOs para diferentes dados que o usuário faz requisição, podendo tornar o retorno dos dados necessários e desacoplando a camada de modelo da camada de controle, fazendo um intermédio através dos DTOs.

No entanto, quando criamos DTOs, como nossa camada de Dados (modelo e repositório) trabalham diretamente com a classe de modelo, será necessário fazer conversões em ambos os sentidos, de DTO -> Model e Model -> DTO, o que pode ser feito criando construtores específicos.

DTO

```
@Data @AllArgsConstructor
public class PetOutput {
    private Long id;
    private String name;
    private Status status;
    public PetOutput(Pet pet){
        this(pet.getId(), pet.getName(), pet.getStatus());
    }
}
```

Modelo

```
public class Pet {
    ...
    public Pet(PetInput dto){
        this.name = dto.getName();
        this.status = dto.getStatus();
    }
}
```