

Presentation

AMT

10 - Presentation

Résumé du document

Cette note explique le rôle des DTO (Data Transfer Objects), qui sont des structures de données servant à transporter des informations entre les différentes couches d'une application, sans logique métier.

Table des matières

1. DTO	2
2. Mapper, Assembler, Adapter	3
2.1. Exemple	3

1. DTO

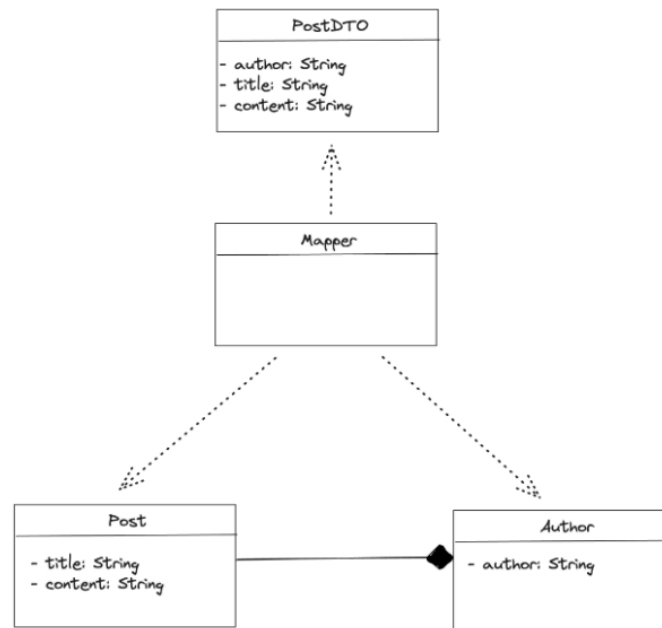
Un DTO (Data Transfer Object) est un objet qui transporte des données entre les couches de l'application. Il est utilisé pour encapsuler des données et les transférer entre les différentes couches de l'application. Un DTO ne contient pas de logique métier, il est simplement une structure de données.

- Ils sont souvent utilisés dans la couche de présentation pour transférer des données entre le client et le serveur.
- Ils sont étroitement liés à la sérialisation et à la désérialisation.
- Ils ne doivent pas contenir de logique métier.
- Ils ne doivent pas refléter le modèle du domaine et fuir la logique du domaine.

Dans certains cas les DTO peuvent mapper les entités.

2. Mapper, Assembler, Adapter

Les objets de transfert de données sont souvent mis en correspondance avec les objets du domaine à l'aide de mappeurs, d'assembleurs, d'adaptateurs, etc.



2.1. Exemple

```

@Entity
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    ...
}

@Entity
public class Address {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private Long userId;
    private String city;
    ...
}

public class UserWithAddressDTO {
    private Long userId;
    private String userName;
    private String city;

    public UserWithAddressDTO(Long userId, String userName, String city) {
        this.userId = userId;
        this.userName = userName;
        this.city = city;
    }
    ...
}

@ApplicationScoped

```

```
public class UserWithAddressMapper {  
    public UserWithAddressDTO toDTO(User user, Address address) {  
        if (user == null || address == null) {  
            return null;  
        }  
        return new UserWithAddressDTO(  
            user.getId(),  
            user.getName(),  
            address.getCity()  
        );  
    }  
}
```