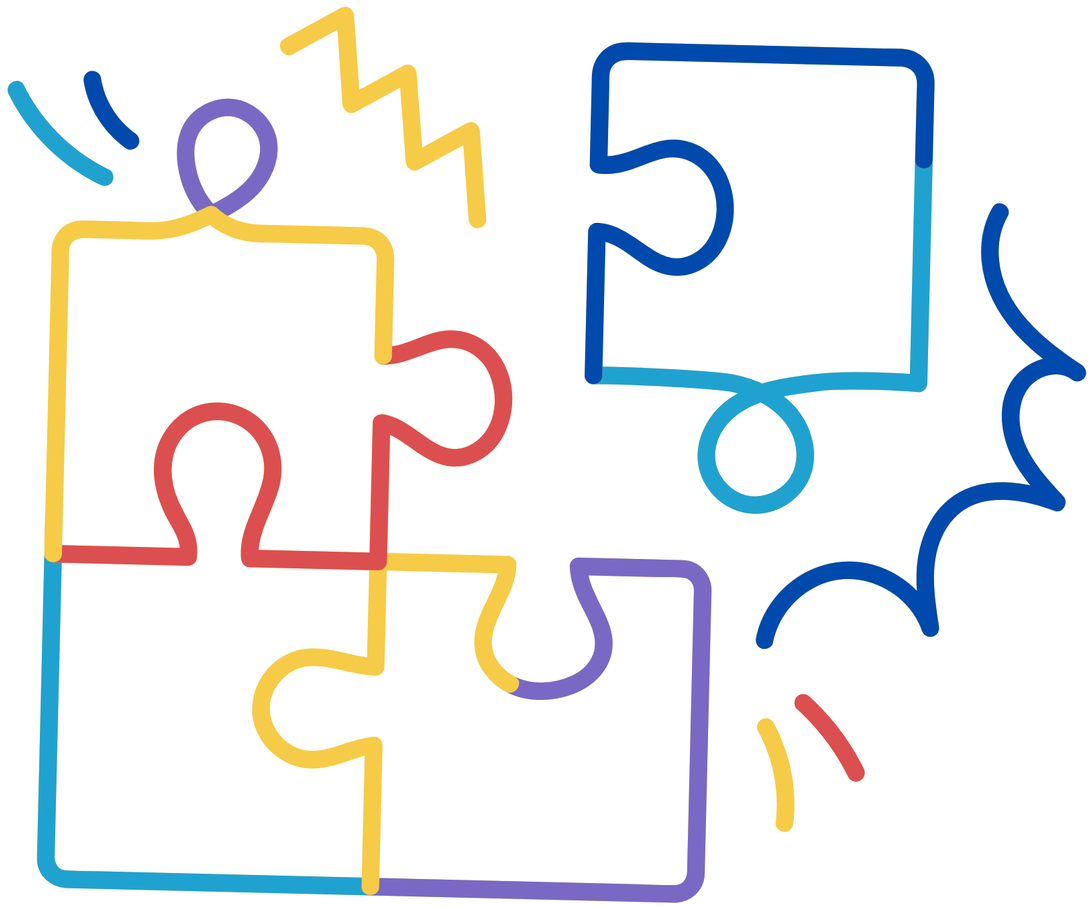
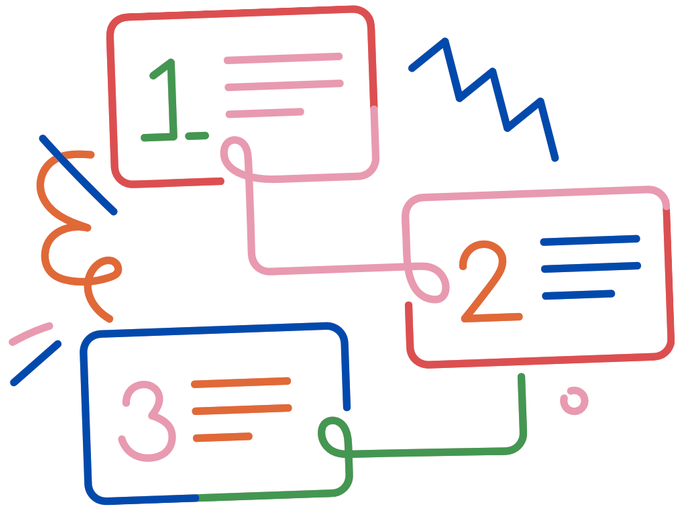
DTO

BAAZIZ WISSAL



Sommaire



DTO

Un DTO (Data Transfer Object) est un objet utilisé pour transférer des données entre différentes couches d'une application, généralement entre la couche de présentation et la couche de service, ou entre la couche de service et la couche de persistance. Le rôle principal d'un DTO est de simplifier le transport de données en encapsulant les informations nécessaires pour une opération spécifique, sans contenir de logique métier ni de comportement.

Pourquoi utiliser un DTO ?

Encapsulation des données

Optimisation des performances

Sécurité

Évolution du code

Types de DTO

DTO de requête (Request DTO) :

- Utilisé pour recevoir des données du client.
- Contient les informations nécessaires pour créer ou modifier une ressource.

DTO de réponse (Response DTO) :

- Utilisé pour envoyer des données au client.
- Contient uniquement les informations pertinentes pour le client, excluant les données sensibles.

Mapping des Objets

Transformation d’un objet source en un objet cible.

Importance : Simplifie le transfert de données entre les couches tout en respectant la structure cible.

Le Mapping Manuel

Utilisation de code Java manuel pour copier les données d'un objet à un autre.

Avantages : Contrôle total sur le processus de transformation.

Inconvénients : Redondance, risque d'erreur, et demande du temps de développement.

MapStruct

Outil de génération de code pour mapper automatiquement les objets Java.

Avantages :

Automatisation des conversions de données
Réduction du code boilerplate
Performance améliorée grâce au code généré à la compilation

Mapping Manuel vs MapStruct

Critères	Mapping Manuel	MapStruct
Temps de Développement	Long	Rapide
Maintenance	Complexe	Simplifiée
Précision	Dépend des développeurs	Standardisé et automatisé

Exemples de Code

Conclusion!!!!