1 - Initialisation et structuration de projet

Séance 1 (4h) GitHub classRoom

API E-commerce - Architecture



IA : avec modération

- Générative : uniquement pour correction/débogage, amélioration, aucune soumission de TAF
- Assistant de codage, type copilot : avec contrôle des productions

Stack technique

- Spring Boot 3+
- Base de données : H2 (fichier)
- JPA (Hibernate)
- Build : Maven
- Tests : JUnit 5
- GitHub classRoom

Spécifications

Mise en place d'une API REST pour la gestion des utilisateurs, des produits et des commandes, utilisable dans le cadre d'une application web e-commerce générique.

End points

Users

```
POST /users - Créer un utilisateur
GET /users/{id} - Récupérer un utilisateur
GET /users/{id}/recommendations - Produits recommandés pour l'utilisateur
```

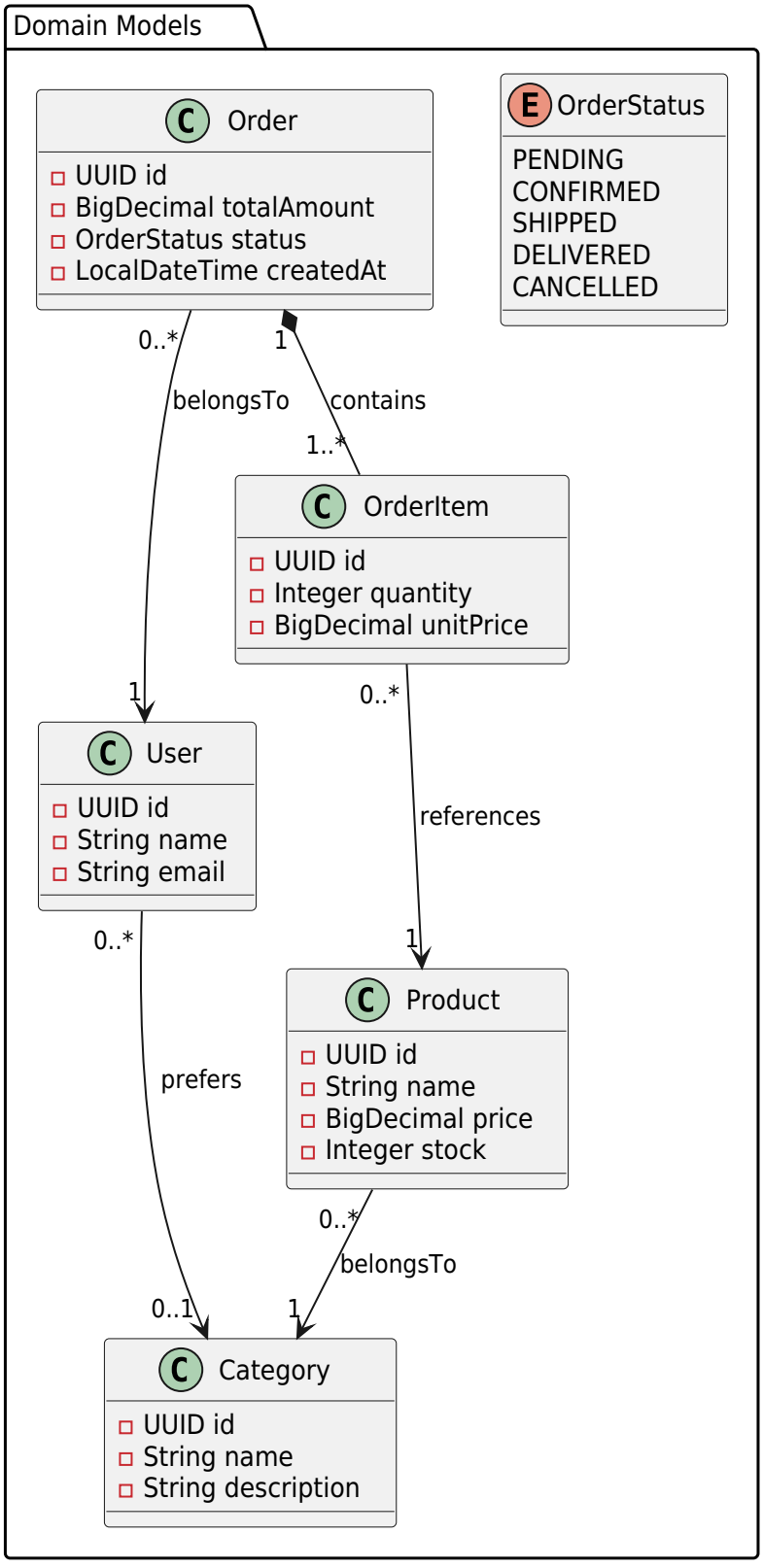
Products

```
POST /products - Créer un produit
GET /products/{id} - Récupérer un produit
PUT /products/{id}/stock - Mettre à jour le stock
GET /products - Lister les produits (avec filtres optionnels)
```

Orders

```
POST /orders - Créer une commande
GET /orders/{id} - Récupérer une commande
GET /users/{userId}/orders - Historique des commandes d'un user
```

Modèle de données



Données exemple

User

```
{
  "id": "550e8400-e29b-41d4-a716-446655440001",
```

```
  "name": "Alice Martin",
  "email": "alice.martin@example.com",
  "preferredCategoryIds": [
    "550e8400-e29b-41d4-a716-446655440010",
    "550e8400-e29b-41d4-a716-446655440011"
  ]
}
```

Product

```
{
  "id": "550e8400-e29b-41d4-a716-446655440020",
  "name": "iPhone 15 Pro",
  "price": 1199.99,
  "stock": 25,
  "categoryId": "550e8400-e29b-41d4-a716-446655440010"
}
```

Order

```
{
  "id": "550e8400-e29b-41d4-a716-446655440030",
  "userId": "550e8400-e29b-41d4-a716-446655440001",
  "items": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440040",
      "productId": "550e8400-e29b-41d4-a716-446655440020",
      "quantity": 2,
      "unitPrice": 1199.99
    }
  ],
  "totalAmount": 2399.98,
  "status": "CONFIRMED",
  "createdAt": "2024-01-15T14:30:00Z"
}
```

Order item

```
{
  "id": "550e8400-e29b-41d4-a716-446655440040",
  "productId": "550e8400-e29b-41d4-a716-446655440020",
  "quantity": 2,
  "unitPrice": 1199.99
}
```

Category

```
{
  "id": "550e8400-e29b-41d4-a716-446655440010",
  "name": "Electronics",
  "description": "Electronic devices and accessories"
}
```

Attendus

Architecture

- Séparation couche métier/applicative :
- Controllers (API REST)
- Services (logique métier)
- Repositories (accès données)
- DTOs pour les échanges API
- Entities pour le domaine métier

Tests

- Tests unitaires sur les services principaux
- Tests d'intégration basiques des controllers

CI/CD

- GitHub Actions : exécution tests sur push/PR
- Pas de déploiement (tests uniquement)



Priorités (4h)

- Entities : User, Product, Category
- CRUD Product + tests unitaires/intégration
- Pipeline CI/CD fonctionnel



- Privilégier la qualité
- Toutes les bonnes initiatives sont les bienvenues

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/eadl/bloc3/dev_av/td1?rev=1758109046

Last update: **2025/09/17 13:37**