Évaluation Technique Spring Boot - 1h30

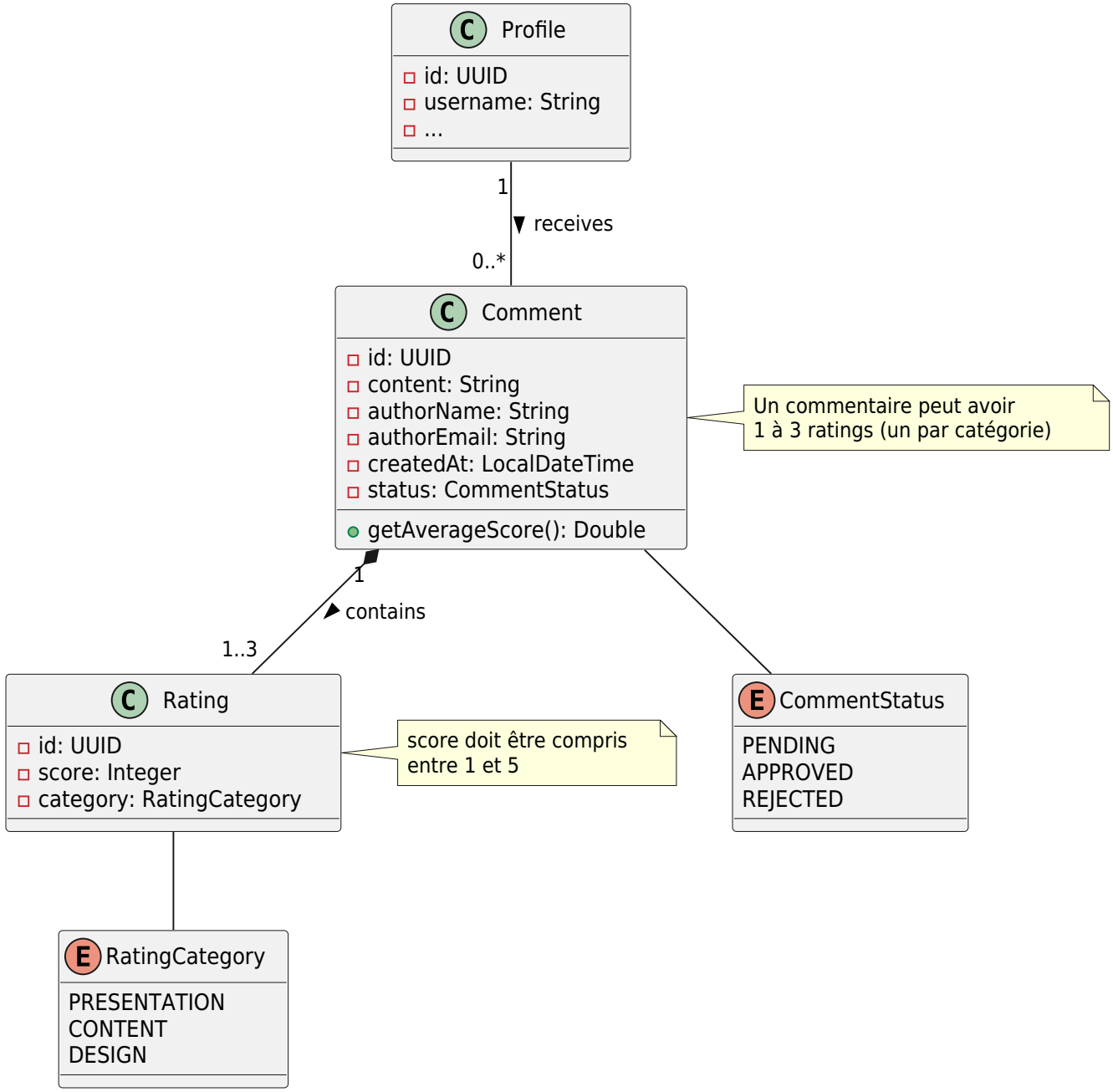
Système de commentaires et notations pour les profils

Contexte

Les utilisateurs du Portfolio-builder souhaitent pouvoir recevoir des retours constructifs sur leurs profils de la part de visiteurs ou recruteurs. Vous allez implémenter un système de commentaires avec notation par catégories (présentation, contenu, design).

Modèle de données

Diagramme PlantUML



Travail demandé

Partie 1 : Modèle de données (30 min)

1. Créer l'enum `CommentStatus`

- Valeurs : `PENDING`, `APPROVED`, `REJECTED`

2. Créer l'enum `RatingCategory`

- Valeurs : `PRESENTATION`, `CONTENT`, `DESIGN`

3. Créer l'entité `Comment`

Attributs :

- `id` : identifiant UUID

- `content` : texte de 500 caractères maximum, obligatoire
- `authorName` : nom du commentateur, obligatoire
- `authorEmail` : email du commentateur, obligatoire
- `createdAt` : date/heure de création
- `status` : statut du commentaire (enum)

Relations :

- un commentaire appartient à un profil
- un commentaire a plusieurs notes
- les ratings ne peuvent pas exister sans commentaire

4. Créer l'entité `Rating`

Attributs :

- `id` : identifiant UUID
- `score` : note de 1 à 5, obligatoire
- `category` : catégorie de notation (enum)

Relations :

- une note appartient à un commentaire

Partie 2 : Repositories (15 min)

5. Créer `CommentRepository`

- Hérite de `JpaRepository<Comment, UUID>`
- Méthode personnalisée : trouver les commentaires par ID de profil ET statut

6. Créer `RatingRepository`

- Hérite de `JpaRepository<Rating, UUID>`
- Méthode personnalisée : trouver toutes les notes d'un commentaire

Partie 3 : DTOs (15 min)

7. Créer `RatingDTO`

Attributs :

- `category`
- `score`

8. Créer `CreateCommentRequest`

Attributs :

- `content`
- `authorName`
- `authorEmail`
- `ratings` (liste de RatingDTO)

9. Créer `CommentDTO`

Attributs :

- `id`
- `content`
- `authorName`
- `createdAt`
- `status`
- `ratings`
- `averageScore`

Critères d'évaluation

Critère	Points
Entités : annotations JPA correctes, relations bidirectionnelles cohérentes	/5
Repositories : déclarations correctes avec méthodes personnalisées	/3
DTOs : structure et validations appropriées	/2

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

Permanent link:
http://slamwiki2.kobject.net/sio/bloc2/2a/td4

Last update: **2025/12/02 20:36**