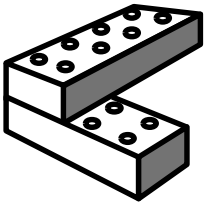

Travaux Pratiques : Enoncés
TP_ReviewPhoto_Partie_09

Objectif : **Consolidation sur le framework Symfony 6**
Création de la table des commentaires

Lorsque l'on clique sur le titre d'une photo, la vue « photo/show.html.twig » affiche cette photo en plus grand avec ses caractéristiques et en dessous la liste des commentaires sous forme d'un tableau qui pour l'instant est vide.
Pour cela on va créer la table des commentaires conforme au MCD.

Rappel sur le MCD :

- **Chaque « COMMENTAIRE » créé par « UN UTILISATEUR »**
- **Chaque « UTILISATEUR » peut créer « PLUSIEURS COMMENTAIRES»**

- **Chaque « COMMENTAIRE » commente « UNE PHOTO »**
- **Chaque « PHOTO » est commentée par « PLUSIEURS COMMENTAIRES»**

Réalisations :

1 : Création de l'entité Comment

- **Entrer la commande : symfony console make:entity (ou php bin/console make:entity)**
- **Répondre :**
  - **Nom : Comment**
  - **Propriété : content string 255 not null**
  - **Propriété : create_at datetime not null**
  - **Propriété : user de type relation**
    - **vers la classe User**
    - **ManyToOne**
    - **null no**
    - **ajout une propriété à User yes**
    - **nom de cette propriété : comments**
    - **supprimer les orphelins : yes**
  - **Propriété photo de type relation**
    - **vers la classe Photo**
    - **ManyToOne, null no, le reste valeur par défaut**

2 : Créer la table dans la base

- **Entrer la commande :**
  - **php bin/console make:migration**
- **Puis**
  - **php bin/console doctrine:migrations:migrate**

Vous pouvez vérifier avant « doctrine:migrations:migrate » quels sont les ordres SQL qui vont être appliqués (voir dernier fichier de migration, méthode up()) :

1. **CREATE TABLE comment …**
2. **ALTER TABLE user …**
3. **ALTER TABLE photo …**

- **créer par phpMyAdmin plusieurs commentaires :**

Respectez bien les ids des utilisateurs et photos

3 : Il faut maintenant modifier la vue « photo/show.html.twig » afin de remplir le tableau des commentaires par une boucle « for ».

Comment faire pour récupérer les commentaires de la photo : tout est déjà prévu

Explication :

Dans l'objet « Photo » récupéré de la base de données, Symfony via Doctrine a valorisé toutes les propriétés à l'exception des propriétés de type relation :

la propriété « user » contient un objet de type « User » incomplet (il n-y a que son id) et la propriété « comments » est un tableau d'objets « Comment » incomplets (il n(y a que leur id).

Ce n'est que si on demande ces propriétés (par les getters) que Symfony demandera à Doctrine les valeurs en faisant des requêtes SQL dans la base. Ceci est transparent pour nous. On appelle cela du « lazy loading » (on va chercher les données que si on en a besoin).

Ce sera le cas dans la vue que l'on va créer ci-après, elle va récupérer l'objet $photo et appeler les getters via twig :

{{ photo.user.pseudo }} fait $photo->getUser()->getPseudo()

Il sera donc possible de faire une boucle pour afficher tous les commentaires :

{% for comment in photo.comments %}

- **Modification de la vue « show.html.twig » dans le dossier « templates/photo »**
  - **Code de la partie commentaire :**

```
{% block body %}
    ……
  <div class="container mt-4">
    <div class="row">
      <div class="col-md-8">
        <h2>Commentaires</h2>
        <table class="table table-striped">
          <tr>
            <th>Commentée par</th>
            <th>Date</th>
```

```
                <th>Commentaire</th>
              </tr>
              {% for comment in photo.comments %}   // on récupère un tableau des commentaires
              <tr>
                <td>{{ comment.user.pseudo }}</td>
                <td>{{ comment.create_at|date("d/m/Y") }}</td>
                <td>{{ comment.content }}</td>
              </tr>
              {% endfor %}
            </table>
          </div>
        </div>
    {% endblock %}
```

4 : tester