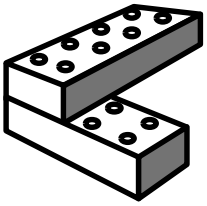

Travaux Pratiques : Enoncés
TP_ReviewPhoto_Partie_07

Objectif : **Consolidation sur le framework Symfony 6**
Gérer les photos d'un utilisateur

Pour un utilisateur identifié, nous allons lui donner les moyens de voir les photos qu'il a déjà postées, d'en supprimer et d'en ajouter une.
Cette dernière possibilité sera vue dans la partie 08 du tp.

Réalisations :

1 : Il faut ajouter au contrôleur des photos deux méthodes :

- **la méthode « manage() » qui va permettre de lister sous forme d'un tableau, les photos de l'utilisateur connecté.**
  La route pour atteindre l'action du contrôleur sera de la forme :
  « photo/manage ».
- **la méthode « delete() » qui va permettre de supprimer la photo à partir du tableau précédent.**
  La route pour atteindre l'action du contrôleur sera de la forme :
  « photo/delete/{id} » avec id qui sera l'id de la photo à supprimer.

  - **création de la méthode « manage() » du « PhotoController » :**

    - **Annoté cette méthode avec la route :**
      #[Route('/photo/manage', name :'photo.manage')

  Dans la méthode :

    - **récupérer le user connecté :**
      $user = $this->getUser() ;
    - **Récupérer ses photos (grâce à la relation entre photo et user) :**
      $photos = $user->getPhotos() ;

    - **Retourner la vue « manage.html.twig » en lui passant cette photo**
      return $this->render('photo/manage.html.twig', ['photos' => $photos]) ;

  - **création de la méthode « delete() » du « PhotoController » :**

    paramètres de la méthode :

      - **le PhotoRepository**
      - **la Request**

    Dans la méthode :

    - **récupérer la photo à partir de l'id passé dans la requête :**
      $photo = $repository->find($request->get('id'));
    - **Récupérer l'EntityManager :**

```
$manager = $this->getDoctrine()->getManager();
```

- **Appeler la méthode « remove() à partir du manager**

```
$manager->remove($photo);
```

- **Valider la suppression en base**

```
$manager->flush();
```

- **Rediriger vers la vue « photo.manage.twig »**

```
return $this->redirectToRoute('photo.manage');
```

2 : Création de la vue « manage.html.twig » dans le dossier « photo » du dossier « templates »

- **Afficher un tableau donnant pour chaque photo :**
  - **son titre**
  - **sa date de post**
  - **le nombre de commentaires**
  - **un bouton permettant de supprimer cette photo**

- **Code :**

```
{% block body %}

  <div class="container">
    <h2>Liste de vos photos</h2>
    <div class="row flex">
      {% for photo in photos %}
        <div class="col-8">
          <table class="table table-striped">
            <tr>
              <th>Titre</th>
              <th>Date</th>
              <th>nb commentaires</th>
              <th></th>
            </tr>
              <tr>
                <td>{{ photo.title }}</td>
                <td>{{ photo.postAt }}</td>
                <td>0</td>
                <td><a href="{{ path("photo.delete", {id: photo.id}) }}"
class="btn btn-secondary"> Supprimer</a></td>
              </tr>
          </table>
        </div>
      {% endfor %}
      <div class="col-4">
```

```
            <a href="" class="btn btn-primary">Ajouter une photo</a>
          </div>
        </div>
      </div>

  {% endblock %}
```

3 : Il faut modifier la vue « base.html.twig » afin de rajouter un item à la navigation

```
    {% if app.user %}
      <li class="nav-item">
        <a class="nav-link" href="{{ path('photo.manage') }}">Gérer mes photos</a>
      </li>
    {% endif %}
```

4 : tester