



# Travaux Pratiques : Enoncés

## TP\_ReviewPhoto\_Partie\_01

---

**Objectif :** Consolidation sur le framework Symfony 6  
Création du projet et de la HomePage

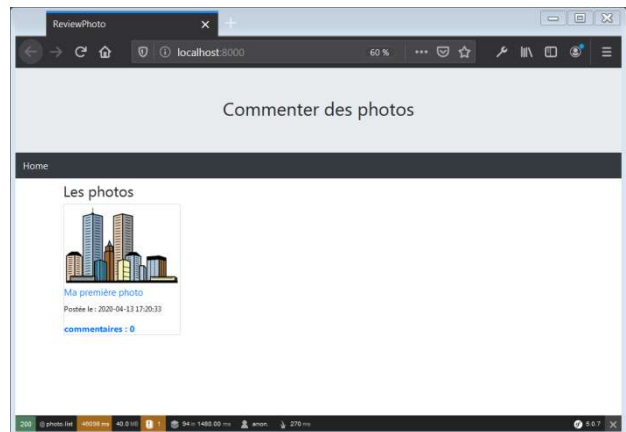
---

On se propose créer une application web de critique de photos.  
L'application est développée par le framework Symfony 6

### CDC :

Sur la page d'accueil, l'internaute aura sous forme de carte les différentes photos qui ont été postées par les inscrits sur le site.

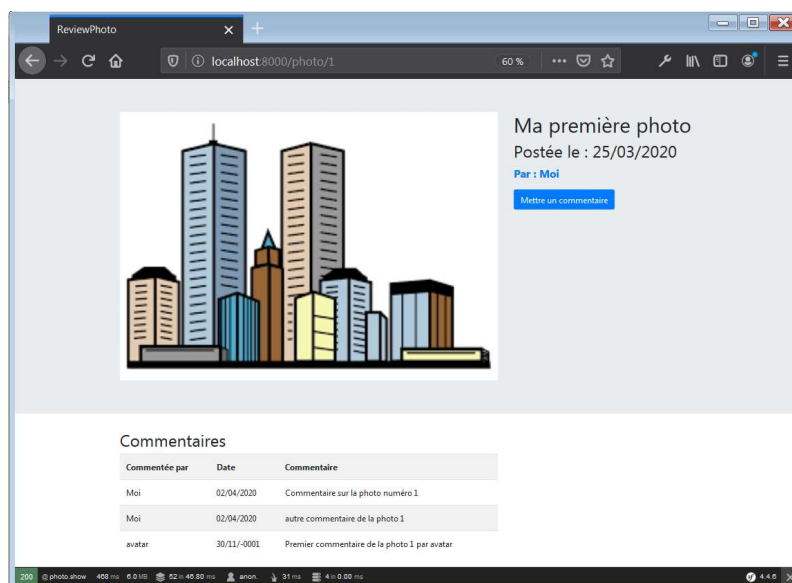
Chaque carte donnera le titre de la photo, son image, le pseudo de celui qui la postée, la date du dépôt et le nombre de commentaires fait.

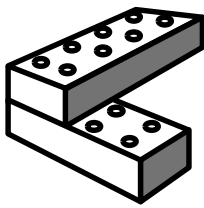


En effet, les internautes inscrits sur le site auront un pseudo et pourront commenter les photos.

En cliquant sur le titre de la photo dans la carte, une page donnera l'ensemble des informations sur cette photo avec l'ensemble des commentaires. Pour chaque commentaire apparaîtra le nom de celui qui l'a fait ainsi que la date.

A partir de cette page, seuls les internautes identifiés après connexion pourront commenter la photo.

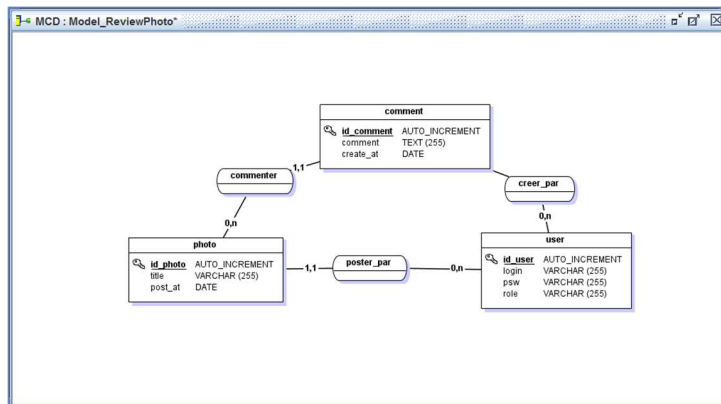




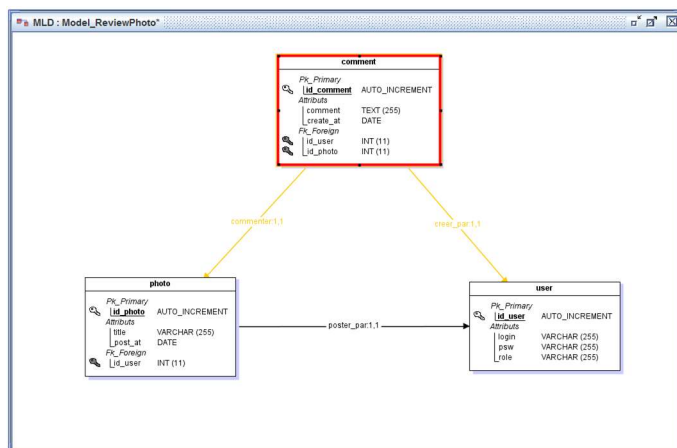
# Travaux Pratiques : Enoncés

## TP\_ReviewPhoto\_Partie\_01

### Le MCD et le MLD :



Num	Nom	Code	type	taille	decimal	Utilisé
1	title	TITLE	Varchar	255		<input type="checkbox"/>
2	post_at	POST_AT	Date			<input type="checkbox"/>
3	comment	COMMENT	Text	255		<input type="checkbox"/>
4	create_at	CREATE_AT	Date			<input type="checkbox"/>
5	login	LOGIN	Varchar	255		<input type="checkbox"/>
6	psw	PSW	Varchar	255		<input type="checkbox"/>
7	role	ROLE	Varchar	255		<input type="checkbox"/>
8	id_photo	ID_PHOTO	Auto_increment			<input type="checkbox"/>
9	id_user	ID_USER	Auto_increment			<input type="checkbox"/>
10	id_comment	ID_COMMENT	Auto_increment			<input type="checkbox"/>



### Réalisations :

1 : En ligne de commande, créer le projet Symfony en version 6 :

```
symfony new ReviewPhoto --version="6.2.*" --webapp
```

ou

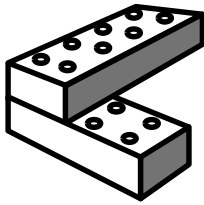
```
composer create-project symfony/skeleton:"6.2.*" ReviewPhoto
cd ReviewPhoto
composer require webapp
```

2 : Lancer VSC (ou NetBeans ou autre IDE)

Ouvrir le dossier « ReviewPhoto »

3 : Création de la « HomePage » :

Il nous faut d'abord créer l'entité « Photo », nous allons utiliser le « maker » de Symfony



# Travaux Pratiques : Enoncés

## TP\_ReviewPhoto\_Partie\_01

- Entrer la commande :

*symfony console make:entity* (ou *php bin/console make:entity*) avec

- nom de l'entité : **Photo**
- Propriété : **title**, de type **string** longueur **255**, **not null**
- Propriété : **post\_at**, de type **datetime\_immutable**, **not null**

Editer le fichier de classe « *src/Entity/Photo.php* » et modifier la méthode « *getPostAt()* » afin qu'elle retourne une chaîne de caractère :

```
public function getPostAt(): string
{
    return date_format($this->post_at, 'Y-m-d H:i:s');
}
```

#### 4 : Créer le contrôleur **PhotoController**

*Symfony console make:controller*

- Nommer la classe **PhotoController**
- Renommer la méthode **index()** en **list()** de type **Response**
- L'annoter avec **#[Route('/', name: 'photo.list')]**
- Renommer la vue « *templates/photo/index.html.twig* » en « *list.html.twig* »
- Dans cette méthode :
  - Créer une nouvelle instance de la classe **Photo** : **\$photo = new Photo()** ;
  - Setter le titre : **\$photo->setTitre('Ma première photo')** ;
  - Setter la date : **\$photo->setPost\_at( new \DateTimeImmutable())** ;
  - Créer un tableau : **\$photos = []** ;
  - Ajouter la photo au tableau : **\$photos[] = \$photo** ;
  - Faire un **Return** de la vue « *photo/list.html.twig* » en lui passant un paramètre **photos** correspondant au tableau **\$photos** :

```
return $this->render('photo/list.html.twig', ['photos' => $photos]) ;
```

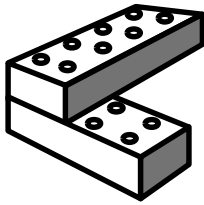
#### 5 : Modifier la vue « *list.html.twig* » du dossier « *templates/photo* »

Les photos seront affichées par une « card » de « bootstrap »

Cette vue reçoit un tableau d'objets **Photo**, on va donc itérer sur ce tableau par une commande Twig : **{% for photo in photos %}.....{% endfor %}**

Placer dans le block twig body le code suivant :

```
<div class="container">
  <h2>Les photos</h2>
```



## Travaux Pratiques : Enoncés

### TP\_ReviewPhoto\_Partie\_01

```
<div class="row flex">
  {% for photo in photos %}
    <div class="col-3">
      <div class="card">
        
        <div class="card-body">
          <h5 class="card-title"><a href="">{{ photo.title }}</a></h5>
          <h5>Postée par : un internaute</h5>
          <p class="card-text">Postée le : {{ photo.postAt }}</p>
          <div class="text-primary" style="font-weight: bold; font-size:
1.2rem;">commentaires : 0</div>
        </div>
      </div>
    </div>
  {% endfor %}
</div>
```

Supprimer le block twig title.

**6 :** Créez un dossier « photos » dans le dossier « public » et copiez la photo « empty.jpg » fournie avec le TP.

**7 :** Dans la vue « base.html.twig » ajouter les liens CSS et JS pour la gestion de Bootstrap comme nous avons déjà fait.  
Les mettre en dehors des blocs Twig « stylesheets » et « javascripts »

Ajouter un « jumbotron » et une « nav-bar » qui aura la route « photo/list » juste avant le bloc « body »

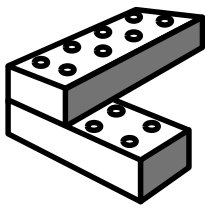
```
<div class="jumbotron text-center mb-0">
  <h1>Commenter des photos</h1>
</div>
<nav class="navbar navbar-dark bg-dark mb-2">
  <a class="navbar-brand" href="{{ path('photo.list') }}">Home</a>
</nav>
{% block body %}{% endblock %}
```

Modifier le block twig title par « Review Photo »

**7 :** Démarrer le serveur web de Php en ligne de commande par :

*symfony server :start (ou php -S localhost :8000 -t public)*

ceci lance le server interne, pour l'arrêter faire ctrl C



## Travaux Pratiques : Enoncés

### TP\_ReviewPhoto\_Partie\_01

**8 : Ouvrir votre navigateur et entrer l'url : <http://localhost:8000>**