

Workshop :

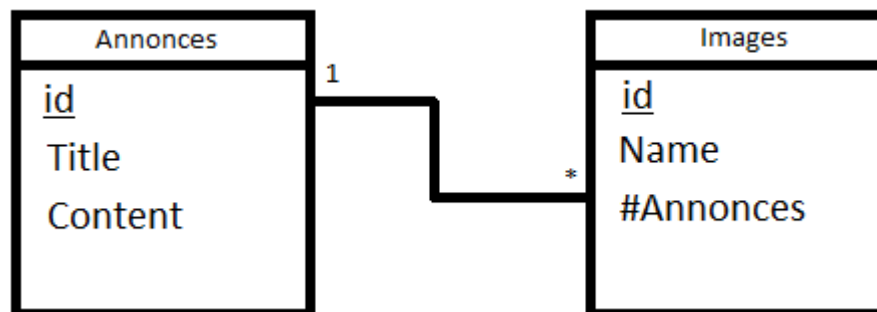
Upload images multiple

Objectif :

- Créer une application proposant des annonces.
- Chaque annonce pourra avoir une galerie d'images.

Etude de cas :

1. Configurer et créer une nouvelle base de donnée.
2. Créer les 2 entités : Annonces et Images :



Entité Annonces :

```
Php bin/console make:entity annonces
```

Entité Images:

```
Php bin/console make:entity images
```

3. Faire la migration :

```
php bin/console make:migration  
php bin/console doctrine:migration:migrate
```

4. Installer la recette qui permet de créer les formulaires et en créer un :

```
php bin/console make:form AnnoncesType
```

Changer le contenu de la fonction buildForm par le code suivant :

```
public function buildForm(FormBuilderInterface $builder, array $options)  
{  
    $builder  
        ->add('titre')  
        //On Ajoute un champ image: il n'est pas lié à la bdd (pas mapped)  
        ->add('images', FileType::class, [  
            'label' => false,  
            'multiple' => true,  
            'mapped' => false,  
            'required' => false  
        ])  
    ;  
}
```

Assurez-vous que multiple est sur true, pour indiquer qu'on pourra upload plusieurs images.

Sans oublier d'ajouter le use pour le filetype au debut :

```
use Symfony\Component\Form\Extension\Core\Type\FileType;
```

5. Paramétrer l'emplacement de stockage des images uploadées dans config/services.yaml sous paramètres.

On va les mettre sous le dossier public.

On donne “%kernel.project_dir%” comme repertoire parent pour indiquer le dossier de notre projet.

```
parameters:  
    upload_directory: '%kernel.project_dir%/public/uploads'
```

6. Créer un contrôleur annoncesController :

```
php bin/console make:controller annoncesController
```

On ajoute ensuite le code suivant dans notre contrôleur, sans oublier de mettre les uses nécessaires avec :

```

/**
 * @Route("/", name="annonces_index", methods={"GET"})
 */
public function index(AnnoncesRepository $annoncesRepository): Response
{
    return $this->render('annonces/index.html.twig', [
        'annonces' => $annoncesRepository->findAll(),
    ]);
}

/**
 * @Route("/new", name="annonces_new", methods={"GET", "POST"})
 */
public function new(Request $request): Response
{
    $annonce = new Annonces();
    $form = $this->createForm(AnnoncesType::class, $annonce);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        //recuperation des images transmises
        $images = $form->get('images')->getData();
        // Boucle sur les images
        foreach ($images as $image) {
            // on genere un nouveau nom de fichier avec md5. guessExtension
            recupere l'extension du fichier
            $fichier = md5(uniqid()) . '.' . $image->guessExtension();
            //On copie le fichier dans le dossier upload
            $image->move(
                $this->getParameter('upload_directory'),
                $fichier
            );

            // on stocke l'image dans la bdd (son nom)
            $img = new Images();
            $img->setName($fichier);
            $annonce->addImage($img);
        }
        $entityManager = $this->getDoctrine()->getManager();
        $entityManager->persist($annonce);
        $entityManager->flush();

        return $this->redirectToRoute('annonces_index');
    }

    return $this->render('annonces/new.html.twig', [
        'annonce' => $annonce,
        'form' => $form->createView(),
    ]);
}

/**
 * @Route("/{id}", name="annonces_show", methods={"GET"})
 */
public function show(Annonces $annonce): Response
{
    return $this->render('annonces/show.html.twig', [
        'annonce' => $annonce,
    ]);
}

```

7. Créer les vues nécessaires :

a) Pour ajouter une nouvelle annonce :

```
{% extends 'base.html.twig' %}

{% block title %}New Annonces{% endblock %}

{% block body %}
    <h1>Create new Annonces</h1>

    {{ form_start(form) }}
    {{ form_widget(form) }}
    <button class="btn">{{ button_label|default('Save') }}</button>
    {{ form_end(form) }}

    <a href="{{ path('annonces_index') }}">back to list</a>
{% endblock %}
```

b) Pour afficher la liste des annonces

Index.html.twig

```
{% extends 'base.html.twig' %}

{% block title %}Annonces index{% endblock %}

{% block body %}
    <h1>Annonces index</h1>

    <table class="table">
        <thead>
            <tr>
                <th>Id</th>
                <th>Titre</th>
                <th>actions</th>
            </tr>
        </thead>
        <tbody>
            {% for annonce in annonces %}
                <tr>
                    <td>{{ annonce.id }}</td>
                    <td>{{ annonce.title }}</td>
                    <td>
                        <a href="{{ path('annonces_show', {'id':
annonce.id}) }}">show</a>
                    </td>
                </tr>
            {% else %}
                <tr>
                    <td colspan="3">no records found</td>
                </tr>
            {% endfor %}
        </tbody>
    </table>

    <a href="{{ path('annonces_new') }}">Create new</a>
{% endblock %}
```

c) Pour afficher une seule annonce avec sa galerie d'images:

Show.html.twig

```
{% extends 'base.html.twig' %}

{% block title %}Annonces{% endblock %}

{% block body %}
    <h1>Annonces</h1>

    <table class="table">
        <tbody>
            <tr>
                <th>Id</th>
                <td>{{ annonce.id }}</td>
            </tr>
            <tr>
                <th>Titre</th>
                <td>{{ annonce.title }}</td>
            </tr>
            <tr>
                <th>Images</th>
                <td>{% for image in annonce.images %}
                    
                    {% endfor %}
                </td>
            </tr>
        </tbody>
    </table>

    <a href="{{ path('annonces_index') }}">back to list</a>
{% endblock %}
```