

Fialoux mateo

Symfony 8 upload fichiers et anonymisation

1- Intégration des données de test

1.1 Lancer la commande de chargement des fixtures (données de test). Quelle commande avez-vous exécutée?

```
bin/console doctrine:fixtures:load
```

```
yes
```

1.2 Relancer la commande et observer l'évolution des id de la table Book et Author. Que se passe-t-il?

Il n'y a aucun changement car dans la class AppFixtures on clear d'abord la tables pour y intégrer les informations.

1.3 Dans la classe AppFixtures, trouver et mettre en commentaire la ligne suivante: Relancer le chargement des fixtures. Que se passe-t-il?

Ici ca n'a pas clear les id alors cela continue d'augmenter dans la bdd.

		<i>idx</i>			
1	11	11	1984	NULL	Dys
2	12	12 <i>firstname: 'Ha...</i>	To Kill a Mocking...	NULL	Sou
3	13	13 <i>firstname: 'F...</i>	The Great Gatsby	NULL	Tra
4	14	14 <i>firstname: 'Ja...</i>	Pride and Prejudi...	NULL	Cla
5	15	15 <i>firstname: 'Ma...</i>	Adventures of Huc...	NULL	Pic
6	16	16 <i>firstname: 'J...</i>	The Hobbit	NULL	Fan
7	17	17 <i>firstname: 'Le...</i>	War and Peace	NULL	His

1.4 Ajouter l'option "--purge-with-truncate" à la commande. Cette option purge les données en utilisant la méthode TRUNCATE, ce qui est plus rapide que la suppression standard et réinitialise également tous les auto-incréments. Est-ce que cela fonctionne? Faire une recherche et expliquer pourquoi.

Cela ne fonctionne pas car sqlite3 ne supporte pas la commande truncate

1.5 Sur la base du TP "Symfony - authentification et administration", créer un dashboard d'administration. Quelles commandes doit-on lancer? Note: pour les besoins de simplification de ce TP, nous ne gérons pas les utilisateurs. L'accès à l'admin se fera sans identification

Elle installe le bundle EasyAdmin dans ton projet Symfony : composer require admin

Elle génère un "Dashboard" (tableau de bord) pour ton interface d'admin : bin/console make:admin:dashboard

clear le cache symfony : php bin/console cache:clear

1.6 Sur la base du TP "Symfony - CRUD: Create, Read, Update, Delete" Créer les CRUD pour gérer la liste des livres et des auteurs. Vous ferez en sorte de gérer le lien livre/auteur sous forme d'une liste (AssociationField). Détailler les actions menées.

php bin/console make:admin:crud pour ajouter les controller book et author dans dashboardcontroller on a ajouter les book et author

```
use App\Entity\Book;  
  
use App\Entity\Author;
```

```
yield MenuItem::linkToCrud('book', 'fas fa-list', book::class);

yield MenuItem::linkToCrud('author', 'fas fa-list', author::class);
```

2- Upload d'une page de couverture.

2.1 Après avoir contrôlé le bon fonctionnement de l'ajout/suppression de cover, faire une recherche sur le namer:

Vich\UploaderBundle\Naming\UniqidNamer Expliquer son principe et détailler les autres namer disponibles.

UniqidNamer génère un nom de fichier unique basé sur la fonction PHP uniqid().

Cela signifie que chaque fichier uploadé aura un nom unique, composé d'un identifiant généré de manière quasi aléatoire, pour éviter les collisions de noms.

Exemple : 5f4dcc3b5aa765d61d8327deb882cf99.jpg ou
5f4dcc3b5aa765d61d8327deb882cf99.png.

Ce namer garantit que plusieurs fichiers uploadés avec le même nom original ne vont pas écraser les anciens.

Autres namers disponibles dans VichUploaderBundle

- SmartUniqueNamer (celui que tu utilises) : Combine un nom généré unique avec l'extension originale. Par exemple, a1b2c3d4e5f6.jpg.
- OriginalNamer : Utilise le nom original du fichier. Risque de collisions si deux fichiers ont le même nom.

- PropertyNamer : Utilise la valeur d'une propriété de l'entité pour nommer le fichier (ex: coverName ou title).
- SlugNamer : Crée un nom basé sur un slug (texte « friendly »), idéal pour SEO.
- Or tu peux créer ton propre Namer en implémentant l'interface `Vich\UploaderBundle\Naming\NamerInterface`.

2.2 En quoi l'usage du renommage de fichier peut-être très utile pour l'anonymisation de document?

Renommer les fichiers uploadés évite d'utiliser les noms d'origine, qui peuvent contenir des informations personnelles (ex : prénom.nom.jpg).

Cela prévient les fuites d'informations sensibles, surtout dans des contextes où des documents confidentiels sont uploadés.

Permet aussi de cacher les métadonnées potentiellement présentes dans le nom (date, lieu, etc.).

En anonymisant le nom, on réduit le risque que des tiers accèdent à des données personnelles juste en regardant les noms de fichiers.

2.3 Éditer les twig de visualisation de l'accueil et de la fiche détail pour afficher la Cover. Quelles sont les modifications effectuées?

```
{# Affiche la couverture seulement si elle existe #}  
  
{% if book.coverName %}  
  

{% endif %}
```

2.4 Nous souhaitons maintenant pouvoir télécharger un extrait pdf des ouvrages qui soit consultable par les visiteurs sur la fiche détail. Réaliser les évolutions nécessaires pour leur téléchargement dans l'administration et leur accès sur le site. Indiquer les modifications apportées au projet.

1. Étendre l'entité Book

```
#[Assert\File(
    maxSize: "10M",
    mimeTypes: ["application/pdf"],
    mimeTypesMessage: "Veuillez télécharger un fichier PDF valide (max
5 Mo)."]
)]

#[Vich\UploadableField(mapping: 'book_excerpt', fileNameProperty:
'excerptName')]

private ?File $excerptFile = null;
```

ajout également des Getters/Setters

2. Ajouter la configuration VichUploader pour le PDF

```
book_excerpt:

    uri_prefix: /uploads/book_excerpts

    upload_destination:
'%kernel.project_dir%/public/uploads/book_excerpts'

    namer: Vich\UploaderBundle\Naming\SmartUniqueNamer

    delete_on_update: true

    delete_on_remove: true
```

3. Modifier ton contrôleur EasyAdmin (BookCrudController)

```
// Champ d'upload pour le PDF

TextField::new('excerptFile')

    ->setFormType(VichFileType::class)

    ->setLabel('Extrait PDF')

    ->setRequired(false)

    ->onlyOnForms(),

TextField::new('excerptName')

    ->setLabel('Fichier PDF')

    ->onlyOnIndex(),
```

et l'inclusion

```
use Vich\UploaderBundle\Form\Type\VichFileType;
```

4. Migration de base de données

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


```
php bin/console doctrine:migrations:migrate
```

5. Modifier la vue publique (show.html.twig)

```
{% if book.excerptName %}

<p>

  <a href="{{ asset('/uploads/book_excerpts/' ~ book.excerptName) }}"
  class="btn btn-outline-primary" target="_blank">

     Télécharger l'extrait PDF

  </a>

</p>

{% endif %}
```

3- Aller plus loin