

Workshop : Utilisation de StofDoctrineExtensionsBundle dans Symfony 6.4

Introduction :

StofDoctrineExtensionsBundle est un bundle Symfony qui facilite l'intégration des extensions Doctrine de Gedmo. Il permet d'ajouter facilement des fonctionnalités avancées comme :

- **Timestampable** : Ajoute automatiquement les dates de création et de mise à jour.
- **Sluggable** : Génère un slug à partir d'un champ (exemple : titre d'un article).
- **Softdeleteable** : Permet une suppression logique des entités sans les supprimer de la base de données.
- **Translatable** : Gère des traductions de champs.
- **Tree** : Gère des structures arborescentes.

Objectifs :

- Mettre à jour les champs de date lors de la création et de la modification
- Eviter les id dans les paths en utilisant les slugs

Prérequis :

- Avoir un projet Symfony6 configuré.
- Une entité Product

Les étapes :

▪ Etape1 : Installation du Bundle

Exécute la commande suivante pour ajouter le bundle à ton projet Symfony :

```
composer require stof/doctrine-extensions-bundle
```

Workshop : Utilisation de StofDoctrineExtensionsBundle dans Symfony 6.4

Vérifie ensuite que le bundle est bien enregistré dans [config/bundles.php](#)

```
return [  
    // ...  
    Stof\DoctrineExtensionsBundle\StofDoctrineExtensionsBundle::class => ['all' => true],  
];
```

▪ Etape2 : Configuration du Bundle :

Crée le fichier de configuration `config/packages/stof_doctrine_extensions.yaml` si ce n'est pas déjà fait :

```
stof_doctrine_extensions:  
    default_locale: fr  
    orm:  
        default:  
            timestampable: true  
            sluggable: true  
            tree: true  
            softdeleteable: true
```

Puis ajoute la configuration des mappings Gedmo dans le fichier de configuration `config/packages/doctrine.yaml`

```
doctrine:  
    orm:  
        mappings:  
            gedmo:  
                type: attribute
```

Workshop : Utilisation de StofDoctrineExtensionsBundle dans Symfony 6.4

```
prefix: Gedmo  
dir: "%kernel.project_dir%/vendor/gedmo/doctrine-extensions/src"  
alias: Gedmo  
is_bundle: false
```

- Etape3 : Utilisation des extensions Doctrine dans les entités

Ajouter les attributs : \$createdAt, \$updatedAt et \$slug dans l'entité Product

```
use Doctrine\ORM\Mapping as ORM;  
use Gedmo\Mapping\Annotation as Gedmo;  
  
#[ORM\Entity]  
class Article  
{  
    #[ORM\Id, ORM\GeneratedValue, ORM\Column(type: 'integer')]  
    private ?int $id = null;  
  
    #[ORM\Column(type: 'string', length: 255)]  
    private string $title;  
  
    #[Gedmo\Timestampable(on: "create")]  
    #[ORM\Column(type: "datetime")]  
    private \DateTime $createdAt;  
  
    #[Gedmo\Timestampable(on: "update")]  
    #[ORM\Column(type: "datetime")]  
    private \DateTime $updatedAt;  
}
```

Workshop :

Utilisation de StofDoctrineExtensionsBundle dans Symfony 6.4

- 1) Ajouter le champ « **Slug** » : Slug unique pour l'étudiant, généré à partir du nom ainsi que les getters et les setters.

```
#[ORM\Column(length: 255, unique: true)]
#[Assert\NotNull()]
#[Assert\NotBlank()]
private ?string $slug = null;
public function getSlug(): ?string
{
    return $this->slug;
}

public function setSlug(string $slug): static
{
    $this->slug = $slug;
    return $this;
}
#[Gedmo\Slug(fields: ['title'])] #[ORM\Column(type: 'string', length: 255, unique: true)] private string $slug;
```

L'ajout d'un champ slug dans votre entité **Product** présente plusieurs avantages, notamment en termes de référencement, de convivialité des URL et de gestion des données. Voici quelques raisons pour lesquelles vous pourriez vouloir ajouter un slug :

- **SEO (Search Engine Optimization)** : Les slugs sont souvent utilisés pour améliorer le référencement des pages web. Un slug bien structuré et descriptif peut aider les moteurs de recherche à comprendre le contenu de la page, ce qui peut améliorer le classement de votre site dans les résultats de recherche.
- **Convivialité des URL** : Les slugs rendent les URL plus lisibles et plus conviviales pour les utilisateurs. Par exemple, une URL comme `example.com/student/john-doe` est plus facile à comprendre et à mémoriser qu'une URL avec un identifiant numérique comme `example.com/student/123`.

Workshop : Utilisation de StofDoctrineExtensionsBundle dans Symfony 6.4

- **Stabilité des URL** : Les slugs permettent de créer des URL stables qui ne changent pas même si d'autres attributs de l'entité changent. Cela est particulièrement utile pour les liens permanents (permalinks) dans les blogs et les sites de contenu.
- **Gestion des Données** : Les slugs peuvent être utilisés comme identifiants uniques pour les entités, ce qui facilite la recherche et la gestion des données. Par exemple, vous pouvez rechercher un étudiant par son slug plutôt que par son identifiant numérique.
- **Prévention des Conflits** : En utilisant des slugs uniques, vous évitez les conflits de noms dans votre base de données. Cela est particulièrement important lorsque vous avez des entités avec des noms similaires ou identiques.

Etape 4 : Mettre à jour la base des données

Exécution des migrations pour appliquer les modifications en lançant ces 2 commandes :

- **Symfony console make : migration**
- **Symfony console doctrine : migrations : migrate**

Etape 5 : Tester le bon fonctionnement de l'ajout d'un produit

Références :

<https://symfony.com/bundles/StofDoctrineExtensionsBundle/current/index.html>