Opérations CRUD dans Symfony 3-3

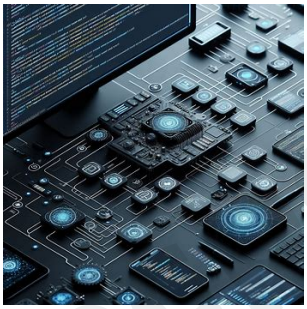
[sommaire]

1. ## Objectifs

- Diviser les résultats de votre requête en pages, permettant ainsi une navigation plus facile pour l'utilisateur, en particulier lorsque vous affichez de grandes quantités de données.

- ## Processus de pagination dans symfony

1. **Configuration :**
    - Cette image représente la configuration initiale du processus de pagination dans Symfony.
    - Elle montre l’installation du bundle KnpPaginatorBundle via Composer.
    - La configuration des paramètres de pagination dans config/packages/knp_paginator.yaml est également illustrée.



2. **Controller :**
    - L’image suivante représente la partie du contrôleur dans le processus de pagination.
    - Elle montre comment récupérer la page actuelle à partir de la requête et créer une requête Doctrine (QueryBuilder) pour récupérer les éléments triés par ID.

apcpedagogie.com



3. **Pagination :**
    - Cette image illustre l’utilisation du service PaginatorInterface pour paginer les résultats.
    - Elle indique comment définir le nombre d’éléments par page et la page actuelle.
    - Le Paginator exécute les requêtes nécessaires pour récupérer les éléments de la page en cours.



4. **Vue Twig :**
    - La dernière image concerne la vue Twig.
    - Elle montre comment afficher les éléments paginés dans la vue.
    - Les variables fournies par le bundle sont utilisées pour afficher les liens vers les pages précédentes, suivantes, etc.



apcpedagogie.com

- ## Comment faire une pagination avec Symfony 7 ?

  - Pour implémenter une pagination dans Symfony 7, vous pouvez utiliser le **bundle KnpPaginatorBundle** qui offre une solution simple et efficace. Voici les étapes de base pour mettre en place la pagination avec ce bundle :
  - Le bundle **KnpPaginatorBundle** offre plus de fonctionnalités avec la possibilité de gérer l'organisation mais aussi la partie template avec la structure HTML d'une pagination. Aussi, si votre objectif est de paginer des résultats dans une page Web je vous conseille d'utiliser ce bundle.

1. **Installer le bundle KnpPaginatorBundle :**

- **Installation du bundle** : Installez le KnpPaginatorBundle via Composer avec la commande suivante :`composer require knplabs/knp-paginator-bundle`

- **Configurer le bundle :**

  - Après l'installation, vous devez configurer le bundle dans votre application Symfony. Assurez-vous d'ajouter la configuration requise dans votre fichier **config/bundles.php** pour activer le bundle.
  - Pour configurer le bundle **KnpPaginatorBundle** dans votre application Symfony, vous n'avez généralement pas besoin d'ajuster de nombreuses configurations, car le bundle est généralement prêt à l'emploi après l'installation. Cependant, vous devez vous assurer que le bundle est activé dans votre application Symfony. Voici comment vous pouvez le faire :

1. **Ouvrez le fichier `config/bundles.php` :**

- C'est ici que vous déclarez tous les bundles que vous souhaitez activer dans votre application Symfony.

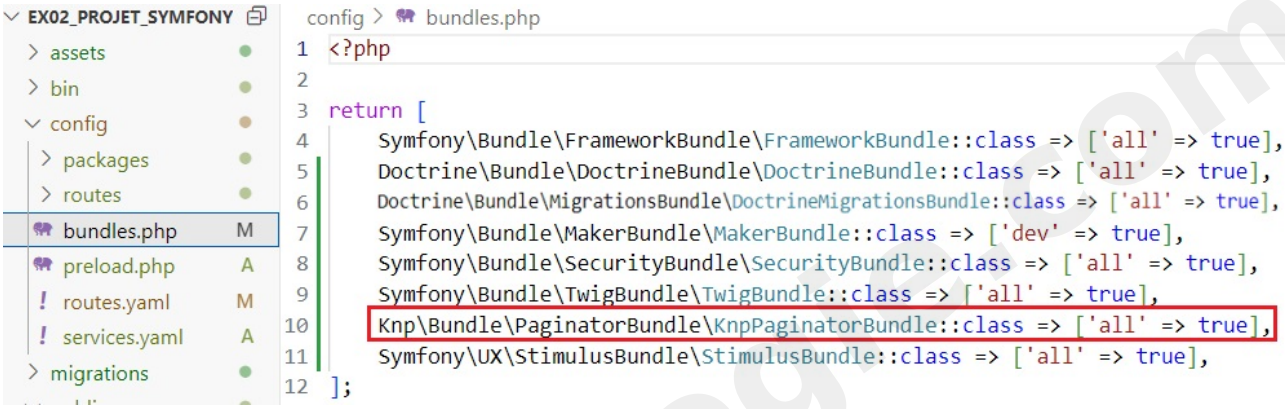
- **Ajoutez la configuration pour le bundle KnpPaginatorBundle:**

  - Dans le fichier **config/bundles.php**, recherchez la ligne où les bundles sont déclarés. Vous devez ajouter une nouvelle ligne pour activer le bundle **KnpPaginatorBundle**. Elle ressemblera à quelque chose comme ceci :

apcpedagogie.com

```
return [
    // ...
    Knp\Bundle\PaginatorBundle\KnpPaginatorBundle::class => ['all' => true],
];
```

- Assurez-vous que la ligne ci-dessus est présente dans votre fichier config/bundles.php. Cela garantira que le bundle est activé et disponible pour être utilisé dans votre application Symfony.



- **Assurez-vous que les dépendances sont chargées :**
  - Assurez-vous également que les dépendances requises par KnpPaginatorBundle sont correctement chargées dans votre fichier **config/packages/knp_paginator.yaml**.
  - Ce fichier de configuration est généralement créé automatiquement lors de l'installation du bundle, mais vous devez vous assurer qu'il est correctement configuré.

- **Vérification :**
  - Une fois que vous avez ajouté la ligne pour activer le bundle dans **config/bundles.php**, vérifiez que le bundle est correctement chargé en exécutant la commande Symfony suivante dans votre terminal :`php bin/console debug:container | grep paginator`
  - Vous devriez voir des services relatifs à la pagination du bundle KnpPaginatorBundle.

- **Configuration du contrôleur :**
  - Utilisez le service **PaginatorInterface** pour paginer les données dans votre contrôleur. Par exemple :

```
use App\Repository\ModuleRepository;
use Knp\Component\Pager\PaginatorInterface;

public function index(Request $request, ModuleRepository $moduleRepository,
PaginatorInterface $paginator)
{
    // Utilisez le repository pour récupérer les modules
    $queryBuilder = $moduleRepository->createQueryBuilder('m')
        ->orderBy('m.id', 'ASC');

    // Paginez les résultats
    $pagination = $paginator->paginate(
        $queryBuilder, // QueryBuilder à paginer
        $request->query->getInt('page', 1), // Numéro de la page à afficher
        10 // Nombre d'éléments par page
    );

    // Transférez la pagination à votre vue Twig
    return $this->render('module/index.html.twig', [
        'pagination' => $pagination,
    ]);
}
```

- Dans cet exemple, nous utilisons le **ModuleRepository** pour obtenir un **QueryBuilder**. Ensuite, nous utilisons ce **QueryBuilder** pour paginer les résultats avec le Paginator.

- **Affichage dans la vue :**
  - Dans votre fichier **Twig**, utilisez la variable pagination pour afficher les résultats et les liens de pagination. Par exemple :

```
{% for item in pagination %}
    {# Affichez vos éléments ici #}
{% endfor %}

{{ knp_pagination_render(pagination) }}
```

  - Ces étapes devraient vous aider à démarrer avec la pagination dans Symfony 7. Pour plus de détails, vous pouvez consulter la documentation officielle ou des tutoriels en ligne comme celui proposé par Grafikart1.

apcpedagogie.com

- **Mettre en forme la pagination avec Bootstrap dans votre application Symfony**

  - Pour mettre en forme la pagination avec Bootstrap dans votre application Symfony, vous pouvez utiliser les classes CSS fournies par Bootstrap pour styliser les liens de pagination générés par le **bundle KnpPaginatorBundle**.
  - Voici comment vous pouvez faire cela :
  - **Inclure Bootstrap dans votre application Symfony** : Si vous n'avez pas déjà inclus Bootstrap dans votre application Symfony, vous devez le faire. Vous pouvez le faire en ajoutant les fichiers CSS de Bootstrap dans votre template de base ou en utilisant Webpack Encore si vous utilisez Symfony Flex avec Webpack Encore.
  - **Modifier la vue Twig pour afficher la pagination** : Dans votre vue Twig où vous affichez la pagination, vous pouvez utiliser les classes CSS de Bootstrap pour styliser les liens de pagination. Voici un exemple :

```
{#
/**
 * @file
 * Twitter Bootstrap v4 Sliding pagination control implementation.
 *
 * View that can be used with the pagination module
 * from the Twitter Bootstrap CSS Toolkit
 * https://getbootstrap.com/docs/4.5/components/pagination/
 *
 */
#}
{% if pageCount > 1 %}


        {% set classAlign = (align is not defined) ? '' : align=='center' ? ' justify-
content-center' : (align=='right' ? ' justify-content-end' : '') %}
        {% set classSize = (size is not defined) ? '' : size=='large' ? ' pagination-lg' :
(size=='small' ? ' pagination-sm' : '') %}



                {% if previous is defined %}

    ◦
                        « {{ 'label_previous'|trans({}, 'KnpPaginatorBundle') }}
```

```
            {% else %}
```

- « {{ 'label_previous'|trans({}, 'KnpPaginatorBundle') }}

```
            {% endif %}
            {% if startPage > 1 %}
```

- 1

```
                {% if startPage == 3 %}
```

- 2

```
                {% elseif startPage != 2 %}
```

- …

```
                {% endif %}
            {% endif %}
            {% for page in pagesInRange %}
                {% if page != current %}
```

- {{ page }}

```
                {% else %}
```

- {{ page }}

apcpedagogie.com

```
                {% endif %}

            {% endfor %}

            {% if pageCount > endPage %}
                {% if pageCount > (endPage + 1) %}
                    {% if pageCount > (endPage + 2) %}

                        …

                    {% else %}

                        {{ pageCount -1 }}

                    {% endif %}
                {% endif %}

                {{ pageCount }}

            {% endif %}

            {% if next is defined %}

                {{ 'label_next'|trans({}, 'KnpPaginatorBundle') }} »

            {% else %}

                {{ 'label_next'|trans({}, 'KnpPaginatorBundle') }} »

            {% endif %}
```

apcpedagogie.com

```
{% endif %}
```

Cours Symfony

Source : https://apcpedagogie.com/operations-crud-dans-symfony-3-3 · téléchargé le 03/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com