

SYMFONY: 5-Accéder aux données dans Symfony

Nous allons maintenant dans le controller (**BlogController.php**) pour exploiter les données insérées en BDD.

Si nous voulons faire des sélections, nous avons besoin d'un **repository**.

Si nous voulons faire des manipulations, nous avons besoin d'un **manager**.

Pour cet exemple nous allons n'utiliser qu'un **repository**, dans les fixtures, nous avons vu comment manipuler les éléments avec le manager. Nous verrons cela plus en profondeur dans un prochain chapitre.

Rappel : le **repository** permet de sélectionner des données dans une table.

Dans le fichier **BlogController.php**, dans la méthode `index()`, ajouter le code suivant :

```
$repo = $this->getDoctrine()->getRepository(Article::class);
```

Nous avons besoin du repository des articles qui a été créé dans `ArticleRepository.php`

Je veux discuter avec DOCTRINE (ORM), je veux un repository (une sélection), oui mais lequel, nous pourrions avoir plusieurs repository, nous voulons le repository qui gère l'entité Article

Ne pas oublier l'appel de l'entité enfin de la classe :

```
use App\Entity\Article;
```

Ajouter la ligne suivante afin de sélectionner tous les articles :

```
$articles = $repo->findAll();
```

Modifier le return en ajoutant un nouvel indice au tableau ARRAY :

```
return $this->render('blog/index.html.twig', [
    'controller_name' => 'BlogController',
    'articles' => $articles
]);
```

Nous allons maintenant exploiter les données sélectionnées et modifier l'affichage avec le rendu TWIG, rendez-vous dans le fichier **index.html.twig**.

Ne garder qu'un seul article sur la page **index.html.twig**. Ajouter les modifications au fichier, le rendu TWIG, nous allons exploiter les données sélectionnées

```

{% for article in articles %}
    {# la variables article est déclaré au moment du for, c'est une va
riable de reception, qui receptionne un article par tour de boucle, la variabl
es articles correspond à l'indice 'articles' du tableau ARRAY déclarée dans le
controller BlogController.php dans la méthode index() #}

    <article class="col-md-4 mx-auto text-center border border-
dark mb-3 p-3 mt-3 mr-1">
        <h2>{{ article.title }}</h2>
        {# syntaxe TWIG, la variable $title est privée, donc symf
ony essaye d'abord d'y accéder et ensuite fait appel au getteur, c'est accessi
ble, il envoi !! #}
        <div class="metadata">Ecrit le {{ article.createdAt | date
('d/m/Y') }} à {{ article.createdAt | date('H:i:s') }} dans la catégorie Polit
ique</div><hr>
        {# createdAt est un objet DateTime donc il n'est pas possi
ble de l'afficher tel quel, nous devons utiliser un filtre TWIG, qui va conver
tir l'objet DateTime en une chaine de caractères #}
        <div class="content">
            <hr>

            {{ article.content | raw }}
            {# Sans le filtre, les balises <p> s'affichent, nous u
tilisons donc un filtre 'raw' pour l'affichage brut #}

            <a href="{{ path('blog_show') }}" class="btn btn-
primary">Lire la suite</a>
        </div>
    </article>

{% endfor %}

```

Nous allons maintenant faire en sorte d’être redirigé vers le bon article lorsque l’on clique sur ‘Lire la suite’. Nous allons donc modifier la méthode **show()** du fichier **BlogController.php** pour faire en sorte de récupérer le bon ID du bon article. Nous utiliserons le concept de **route paramétrées** qui existe dans tous les Framework.

```

/**
 * @Route("/blog/{id}", name="blog_show")
 */
public function show($id)
{
    $repo = $this->getDoctrine()->getRepository(Article::class);

    $article = $repo->find($id);

    return $this->render('blog/show.html.twig', [
        'article' => $article
    ])
}

```

```
});  
}
```

Retourner ensuite dans le **template show.html.twig** et faire les modifications suivantes :

```
<section class="row articles">  
  
    <article class="col-md-6 mx-auto text-center border border-dark mb-3 p-3 mt-3 mr-1">  
        <h2>{{ article.title }}</h2>  
        <div class="metadata">Ecrit le {{ article.createdAt | date('d/m/Y') }} à {{ article.createdAt | date('H:i:s') }} dans la catégorie Politique</div><hr>  
        <div class="content">  
            <hr>  
            {{ article.content | raw }}  
        </div>  
    </article>  
  
</section>
```

Si on revient sur la page **inde.html.twig**, nous constatons une erreur. La route 'blog_show' doit avoir un paramètre qui est un identifiant

Ajouter au lien <a>, c'est-à-dire à la fonction path() de TWIG un deuxième paramètre, l'ID !!

```
<a href="{{ path('blog_show', {'id' : article.id }) }}" class="btn btn-primary">Lire la suite</a>
```