Les Providers | Symfony 4

Avant d'attaquer directement notre BDD, nous allons dans un premier temps travailler avec des providers.

> Doc de référence :
> https://symfony.com/doc/current/components/yaml.html

Nous allons dans un premier temps mettre nos données *(articles et categories)* dans un fichier YAML.

Création du fichier YAML

Installons tout d'abord le composant symfony qui nous permettra de manipuler via PHP nos données YAML.

```
composer require symfony/yaml
```

Puis dans **src/Service/Article** créons le fichier **articles.yaml**
Nous allons mettre en place nos articles selon le modèle suivant :

```
data:
  article1 :
    id : 1
    titre : 'Tip Aligning Digital Marketing with Business Goals and Objectives'
    contenu    : '<p>CONTENU</p>'
    featuredimage : '3.jpg'
    special : false
    spotlight : true
    datecreation : 2017-02-26 09:37:18
 categorie:
      id : 2
      libelle : 'Business'
    auteur:
      id : 1
      prenom : 'Hugo'
      nom : 'LIEGEARD'
      email : 'wf3@hl-media.fr'
   article2 : ...
   article3 : ...
```

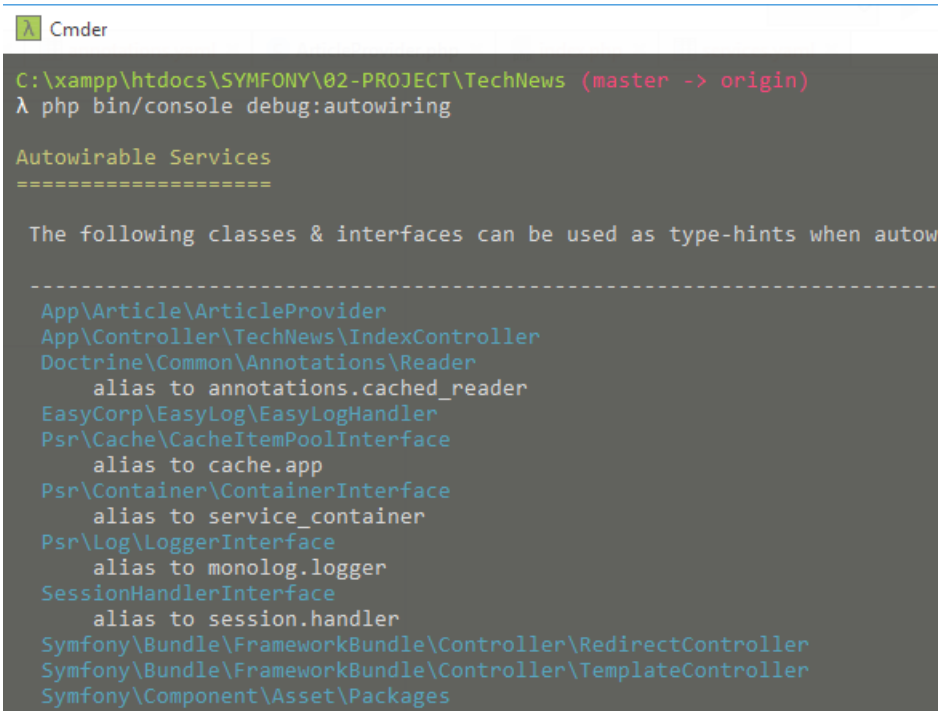
Créez quelques articles pour la démonstration.

Création du Service

Mettons ensuite en place le **service symfony** :

- Création du Service : `YamlProvider` : *Créez la classe YamlProvider dans src/Article/YamlProvider.php*
- Création de la fonction **getArticles()**

Depuis SF4, nos services sont chargé et disponible automatiquement.



Testons dans la console :

```
php bin/console debug:autowiring
```

Nous voyons que notre **YamlProvider** est déjà prêt à être utilisé par symfony !

Utilisation du Service

> Nous allons maintenant récupérer les articles dans notre contrôleur puis afficher les données sur notre page d'accueil.

La récupération du service est simple et rapide, on injecte le service dans le paramètre de notre fonction, symfony fera le reste :

```
public function index(YamlProvider $yamlProvider)
```

Il ne nous reste plus qu'à récupérer les articles et les transmettre à la vue :

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

NOTA BENE : En déclarant notre service dans services.yaml il est possible d'y accéder via le container de symfony.

Affichage dans le template

Nous pouvons ensuite accéder à nos données directement dans twig.

```
{{ dump(articles) }}
```

> Il ne reste plus qu'à faire les boucles et le tour est joué !

Written with ❤ by Hugo LIEGEARD.
Screenshots by Frogg's web tools.