



Framework Symfony

Axel Wagnier - [axessweb](#)

Framework Symfony (introduction)

Symfony est un framework MVC PHP très populaire. Il a été développé par la société Sensiolabs. Une grande communauté de développeurs s'articule autour de ce projet afin de le faire évoluer.

- **C'est quoi un framework PHP ?**

Un framework PHP est un ensemble de composants qui cohabitent parfaitement entre eux. Cet ensemble est articulé, autour d'un squelette, permettant de former une base de développement solide.

- **L'avantage d'un framework ?**

Bien que la prise en main d'un framework type Symfony semble compliquée, le framework va apporter des avantages. - Ne pas réinventer la roue ! En effet, à l'aide de ces nombreux composants, certaines fonctionnalités vont pouvoir être mises en oeuvre rapidement.

- Avoir un code propre et organisé : Symfony, comme tout autre MVC, oblige à produire un code normé. Par conséquent, celui-ci sera bien plus maintenable dans la durée et évolutif.

- Sécurité : Symfony intègre des mesures de sécurités préventives afin de lutter contre les différentes failles possibles.

Framework Symfony (installation)



Le binaire de Symfony est un petit programme permettant l'installation facile de symfony et de ces composants.

Il est téléchargeable à l'adresse suivante : <https://symfony.com/download>

Une fois téléchargé, il est bien évidemment, nécessaire de l'installer.

NB: L'installateur risque de vous proposer d'installer `git`, faites le !

Framework Symfony (création d'un projet)

La création d'un projet Symfony est l'installation des dépendances nécessaires au bon fonctionnement. Ce projet vous permettra d'effectuer vos développements.

Pour créer un projet Symfony, dans un terminal (powershell sous Windows) entrez la ligne de commande suivante dans le dossier de votre choix : `symfony new --full nom_de_mon_projet`

Bien entendu, "nom_de_mon_projet" est à personnaliser en fonction de vos besoins.

Toujours dans le terminal, saisissez ensuite la commande `cd nom_de_mon_projet` puis `symfony server:start`

Vous pouvez également lancer le projet en tâche de fond avec la commande ci-dessous. Cela évite d'avoir un terminal ouvert constamment.

`symfony server:start -d`

Cela exécutera le serveur afin de profiter pleinement de Symfony. Rendez vous ensuite à l'adresse <http://localhost:8000/> pour accéder à votre projet.

Framework Symfony (les routes)

Un framework MVC se base sur des routes, elles sont particulièrement importantes. Elles nous permettront d'aiguiller les requêtes des visiteurs dans notre site web.

Dans Symfony, les routes peuvent se gérer via les annotations, ou à l'aide du fichier [config/routes.yaml](#), tel que ci-dessous.

```
default_index: # nom de la route doit être unique, sans espace, sans caractères spéciaux
    path: / #chemin d'accès
    controller: App\Controller\DefaultController::index #chemin du controller :: nom_de_la_fonction
```

Ces routes sont l'url d'accès à une page. Comme l'exemple, le démontre, une route fait référence à un Controller spécifique et une fonction spécifique, dont le nom apparaît après les :: (deux doubles points).

Framework Symfony (les contrôleurs)

Un contrôleur est une class PHP qui contient la logique de code. C'est dans celui-ci qu'on effectuera les validations d'un formulaire quelconque, la logique d'inscription d'un utilisateur, ou bien même l'envoi d'un email par exemple.

Afin de faciliter la création d'un contrôleur, à la racine de votre projet, il suffit d'utiliser la commande suivante :

```
php bin/console make:controller ArticlesController
```

Le contrôleur `ArticlesController` a été créé. Dans la console, vous pouvez également constater la création de deux fichiers, à savoir :

- `src/Controller/ArticlesController.php` - Le contrôleur
- `templates/articles/index.html.twig` - Le fichier de vue correspondant

On remarquera que le fichier vue a été créé dans un dossier nommé `/articles/`. En effet, en MVC, l'organisation est primordiale. Chaque Controller disposera d'un dossier de vues, du même nom, en minuscule.

Chaque méthode du contrôleur `ArticlesController`, aura ainsi son fichier de vue dans le dossier `templates/articles/`. Bien que déroutant au début, cela permet d'avoir un projet bien organisé.

Framework Symfony (les vues)

Dans Symfony, les fichiers de vues sont gérés à l'aide d'un moteur de templates nommé [twig](#). Les fichiers de vues se placeront dans le dossier `/templates/` de notre application. Ce moteur de template viendra accélérer l'affichage du site (à l'aide de la mise en cache) et sécuriser les données.

Rappel : la création d'un contrôleur en ligne de commande permet de créer le fichier de vue associé automatiquement.

Voici le code permettant, depuis un contrôleur, de transmettre une variable à une vue :

```
// Depuis un contrôleur

// le chemin d'accès est relatif au dossier /templates/
return $this->render('default/index.html.twig', [
    'my_number' => $number,
]);
```

On utilisera alors, la clé du tableau ci-dessus dans le fichier de vue pour récupérer sa valeur, comme ci-dessous.

```
<p>Votre numéro est le {{ my_number }} </p>
```

L'ensemble des fonctionnalités de twig dans Symfony sont disponibles twig sur la [documentation officielle](#).

Framework Symfony (les modèles - doctrine ORM)

Les modèles permettent de communiquer avec la base de données. Sous Symfony, on utilise un ORM (object-relational mapping) nommé Doctrine. Pour créer un modèle, une ligne de commande est disponible :

```
php bin/console make:entity Articles
```

Attention, la base (et non la table) de données doit être créée au préalable et configurée dans le fichier `.env` à la ligne `DATABASE_URL`.

Exemple: `DATABASE_URL="mysql://root@127.0.0.1:3306/nom_bdd"`

Cette commande vous demandera ensuite de créer vos différents champs ainsi que leurs types. Le champs `id` est créé automatiquement en auto-increment, par conséquent, il est inutile de le renseigner.

Une fois les différentes informations saisies, vous devez lancer une migration. Ces migrations, une fois maîtrisées, ont pour objectif de faciliter la création et la maintenances de la base de données de votre projet.

Framework Symfony (les modèles - doctrine ORM)

Les migrations permettent de palier à un problème d'évolution de projet. En effet, faire évoluer une base de données nécessite de savoir quel champs a été modifié, son type et dans quel table il se trouve. Ces opérations peuvent devenir rapidement fastidieuses, même sur de petits projets. Les fichiers de migrations permettent donc d'effectuer une mise à jour de la base de données de manière sécurisée.

Pour fabriquer une migration, entrez la commande suivante dans votre terminal :

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

Cette commande permettra de surveiller les différents changements qui ont été effectuées dans vos tables et de les enregistrer dans un fichier. Une fois exécutée, il suffira de reporter les modifications des tables à l'aide de la commande suivante :

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

Votre base de données est désormais à jour !

La documentation officielle de doctrine dans Symfony est accessible [à cette adresse](#).