

Symfony 5 : introduction



Nizar Rouatbi

Technologue en informatique à Iset Sousse

nizar.rouatbi@gmail.com



Plan

- 1 [Introduction](#)
- 2 [Installation](#)
- 3 [Symfony 5 : structure et fonctionnement](#)
- 4 [Symfony 5 : quelques outils](#)

Symfony

Symfony

- framework **PHP** sorti en octobre 2005
- français
- conçu et développé par SensioLabs
- open-source
- basé sur l'architecture **MVC**
- utilisant le protocole **HTTP**

Symfony

Framework ?

- En français : cadre de travail
- Ensemble de composants logiciels et API facilitant le développement d'applications : pour les développeurs maîtrisant certains concepts informatiques (POO, SQL, MVC...)

Ne pas confondre framework et CMS

- IDE (en anglais Integrated Development Environment, Environnement de développement intégré)
- CMS (en anglais Content Management System, Système de gestion de contenu) pour les développeurs novices

Symfony

Pourquoi utiliser un framework ? (Ce n'est pas obligatoire)

- Un code de qualité
- Une meilleure structuration de notre projet
- Conflits entre dépendances gérés par le framework
- Plusieurs composants et API mis à disposition de développeurs

Symfony

Exemple d'utilisation de **Symfony**

- Dailymotion (depuis 2009)
- Composer
- OpenClassroom
- SNCF
- ...

Symfony

Les différentes versions de **Symfony**

- **Symfony 1** : sorti en octobre 2005
- **Symfony 2** : sorti en août 2011
- **Symfony 3** : sorti en novembre 2015
- **Symfony 4** : sorti en novembre 2017
- **Symfony 5** : sorti en novembre 2019

Symfony

Installation sous **Windows** (avec un installer)

Téléchargez <https://get.symfony.com/cli/setup.exe> puis installez

Installation sous **Linux** (avec une commande)

```
wget https://get.symfony.com/cli/installer -O - | bash
```

Installation sous **Mac** (avec une commande)

```
curl -sS https://get.symfony.com/cli/installer | bash
```


Symfony

Pour créer un projet Web Symfony 5

Exécutez `symfony new --full premier_projet_symfony`

Pour créer un projet console ou microservice **Symfony 5**

Exécutez `symfony new premier_projet_symfony`

Pour lancer un projet Symfony 5 créé avec la commande `symfony`, exécutez

- `cd --full premier_projet_symfony`
- `symfony server:start` **ou** `symfony serve`

Symfony

Pour créer un projet Web **Symfony** d'une version antérieure

Exécutez `symfony new --full premier_projet_symfony --version=4.4`

Symfony

Pour configurer un projet Symfony existant (sur GitHub par exemple)

```
# cloner un projet existant
git clone lien_vers_repository_github

# se positionner dans le projet
cd my-project/

# installer les dépendances composer dans vendor
composer install
```

Symfony

Sous Visual Studio Code

- Installer l'extension **PHP Namespace Resolver**
- Elle permet d'importer les namespaces nécessaires pour certaines classes
- Pour l'utiliser, faites clic droit sur la classe concernée
- Allez dans `Import Class` et choisir le namespace correspondant

Symfony

Structure d'un projet **Symfony** 4/5

- `bin/` : contenant deux exécutables, la console de **Symfony** et `phpunit`
- `config/` : contenant les fichiers de configuration (routes, ORM...)
- `public/` : seul dossier accessible de l'extérieur (contenant le contrôleur frontal `index.php`)
- `src/` : contenant les fichiers sources de l'application (contrôleurs, entités, formulaires, DAO...)
- `templates/` : contenant les vues (vue partielle) de l'application
- `tests/` : contenant les fichiers permettant de tester l'application
- `translations/` : contenant les fichiers de l'internationalisation
- `var/` : utilisé par **Symfony** pendant l'exécution, contenant les données de cache, le log et les sessions
- `vendor/` : contenant les fichiers nécessaires pour une application **Symfony** (mentionnés dans `composer.json`)

Symfony

Kernel?

- noyau de **Symfony**
- défini dans `vendor/symfony/http-kernel`
- utilisé par le contrôleur frontal pour désigner le contrôleur adéquat pour répondre à la requête **HTTP** reçue

Contrôleur frontal

- point d'entrée d'une application **Symfony**
- défini dans `public/index.php`

Symfony

Deux environnements de travail

- **prod** (destiné aux utilisateurs finaux de l'application) montrant
 - l'application telle qu'elle sera visible par les visiteurs
 - ■ rapide à exécuter
 - n'affichant pas les messages d'erreur.
- **dev** (destinés aux développeurs)
 - Plus lent que la version de production
 - Environnement de débogage complet
 - Possibilité d'ajouter des nouvelles fonctionnalités

Symfony

Remarque

- Par défaut, une application est configurée à l'environnement **dev**
- Pour changer d'environnement, allez dans `.env` et mettez la valeur de **APP_ENV** à `prod`

Symfony

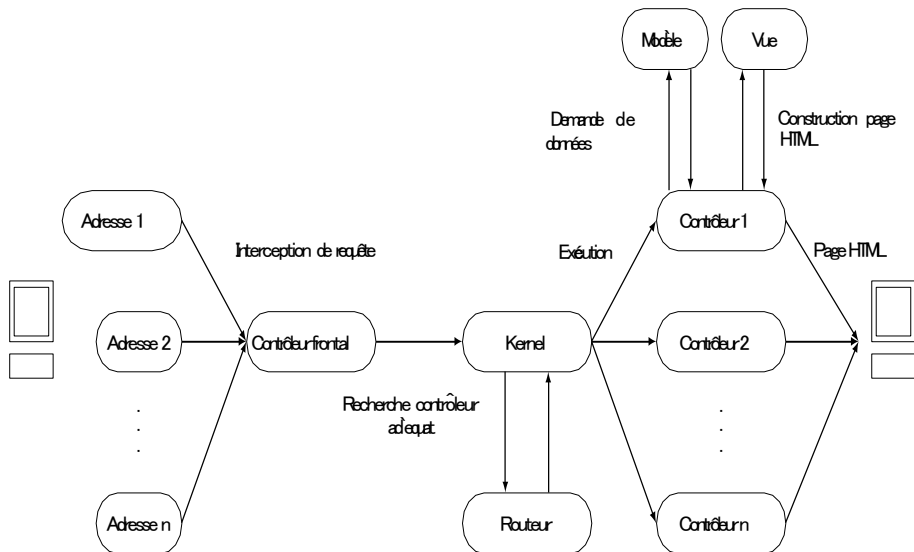
Extrait d'`index.php`

```
$kernel = new Kernel($_SERVER['APP_ENV'], (bool) $_SERVER['APP_DEBUG']);  
$request = Request::createFromGlobals();  
$response = $kernel->handle($request);  
$response->send();  
$kernel->terminate($request, $response);
```

Explication

- Le contrôleur frontal charge le `kernel` selon l'environnement précisé dans `.env`
- Il récupère la requête
- Il utilise le `kernel` pour retourner une réponse

Symfony



Symfony

Déroulement

- L'utilisateur saisit l'adresse d'une page de notre site
- Le contrôleur frontal intercepte la requête et il la transmet au Kernel
- Le Kernel demande au Routeur le contrôleur adéquat à la page demandée
- à la réception d'une réponse, le Kernel exécute le contrôleur
- Le contrôleur communique avec le modèle pour récupérer ou stocker certaines données
- Ensuite il renvoie ces données à la vue pour qu'elle construise la page HTML et la lui retourne.
- Enfin le contrôleur envoie à l'utilisateur la réponse (page HTML).

Symfony

Quelques outils **Symfony 5**

- Console
- Flex

Symfony

Console

- Outil (composant) pour développeur
- Disponible depuis le lancement de **Symfony**
- Permettant d'interagir avec l'application **Symfony** en ligne de commandes (**CLI** ou Command Line Interface).
- Défini dans le répertoire `bin` par le fichier `console`

Pour consulter la liste des commandes disponibles

```
php bin/console
```

Symfony

La console, pourquoi ?

Pour :

- faciliter la création/génération de contrôleurs, entités, formulaires, bundles...
- éviter les erreurs
- accélérer le développement

Symfony

Pour consulter la liste des éléments que l'on puisse générer avec la console

```
php bin/console list make
```

Symfony

Flex

- Outil pour développeur
- Disponible depuis la version 4 de **Symfony**
- Exécuté lorsque **Composer** est utilisé pour installer une nouvelle dépendance (recette ou **recipes** en anglais) afin de la configurer

recipe = bibliothèque standard (définie dans `packagist` : dépôt de **Composer**) + configuration

Pour consulter la liste des recettes

<https://flex.symfony.com/>