

Symfony 5

DÉVELOPPEMENT WEB AVANCÉ – UTILISATION D'UN FRAMEWORK

Qui suis-je

2

Lead Dev / DevOps

Technos/outils :

PHP, NodeJS, React
Symfony2/3/4, Prestashop, Laravel, CakePHP
Docker, Admin SYS, Jenkins, Git, Gitflow

Expériences

En poste depuis 2011
Talan Solutions depuis 2017
Gérant de Nowis depuis 2020
Formation Symfony chez SensioLab

En plus

Meetup, Conférences PHP, Hackathon



simon.helene.dvlp@gmail.com
<https://github.com/Nowwis/>

Et vous ?

3

The logo for ESIEE AMIENS is displayed on a white rectangular background. The word "ESIEE" is in a large, bold, blue serif font. Below it, a thin horizontal line separates the word "ESIEE" from the word "AMIENS", which is in a smaller, blue serif font.

ESIEE

AMIENS

Historique

1994 - PHP première version

2005 - Symfony première version

- ▶ Framework MVC libre
- ▶ Développé par SensioLabs
- ▶ Dernière version 5.2 (01/03/2021)
- ▶ Licence MIT

Projets connexes

5



Framework : Norme de codage

PSR-1

6

- ▶ Tag PHP `<?php ?>` (raccourci echo : `<?= ?>` autorisé)
- ▶ Encodage UTF-8
- ▶ Namespace, Class et method en CamelCase

Framework : Norme de codage

PSR-2

- ▶ Doit respecter la norme PSR-1
- ▶ 4 espaces pour indenter (PAS DE TABULATION !)
- ▶ Longueur de ligne MAX de 120 char, recommandé à 80
- ▶ Ligne blanche avant et après le namespace
- ▶ ...

Framework : Norme de codage

PSR-4

8

- ▶ Autoloading de class

- ▶ Le faire de faire un « use Mon\Super\Namespace\Avec\Class

Charge automatiquement le fichier mon/super/namespace/avec/class.php

~~INCLUDE~~

~~REQUIRE~~

~~INCLUDE_ONCE~~

~~REQUIRE_ONCE~~

Développement web AVANCÉ

9

Doc PHP <http://www.php.net/manual/fr/>

Doc PSR <http://www.php-fig.org/>

Développement web AVANCÉ

10

Doc Symfony <https://symfony.com/doc/current/index.html>

Doc Twig <https://twig.symfony.com/>

Doc Doctrine : <https://www.doctrine-project.org/projects/orm.html>

Pré-requis – Méthode classique

11

- ▶ Serveur web
 - ▶ PHP ≥ 7.2
 - ▶ MySQL ≥ 5.7
 - ▶ Apache 2.4 / Nginx ≥ 1.0

Pré-requis – Mode DevOps

12

- ▶ Docker
 - ▶ Docker-compose
 - ▶ Docker-sync (Mac / Windows)

Pré-requis – Mode Symfo

13

- ▶ Symfony possède un bundle qui émule un webserver pour ne rien installer

```
composer require symfony/web-server-bundle --dev
```

Composer

14

« Composer est un logiciel gestionnaire de dépendances libre écrit en PHP. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin. Le développement a débuté en avril 2011 et a donné lieu à une première version sortie le 1er mars 2012. Développé au début par Nils Adermann et Jordi Boggiano (qui continuent encore aujourd'hui à le maintenir), le projet est maintenant disponible sur la plateforme GitHub. Il est ainsi développé par toute une communauté. »

Wikipédia

Composer – Installation

15

<https://getcomposer.org/>



Installation de Symfony 4.2

16

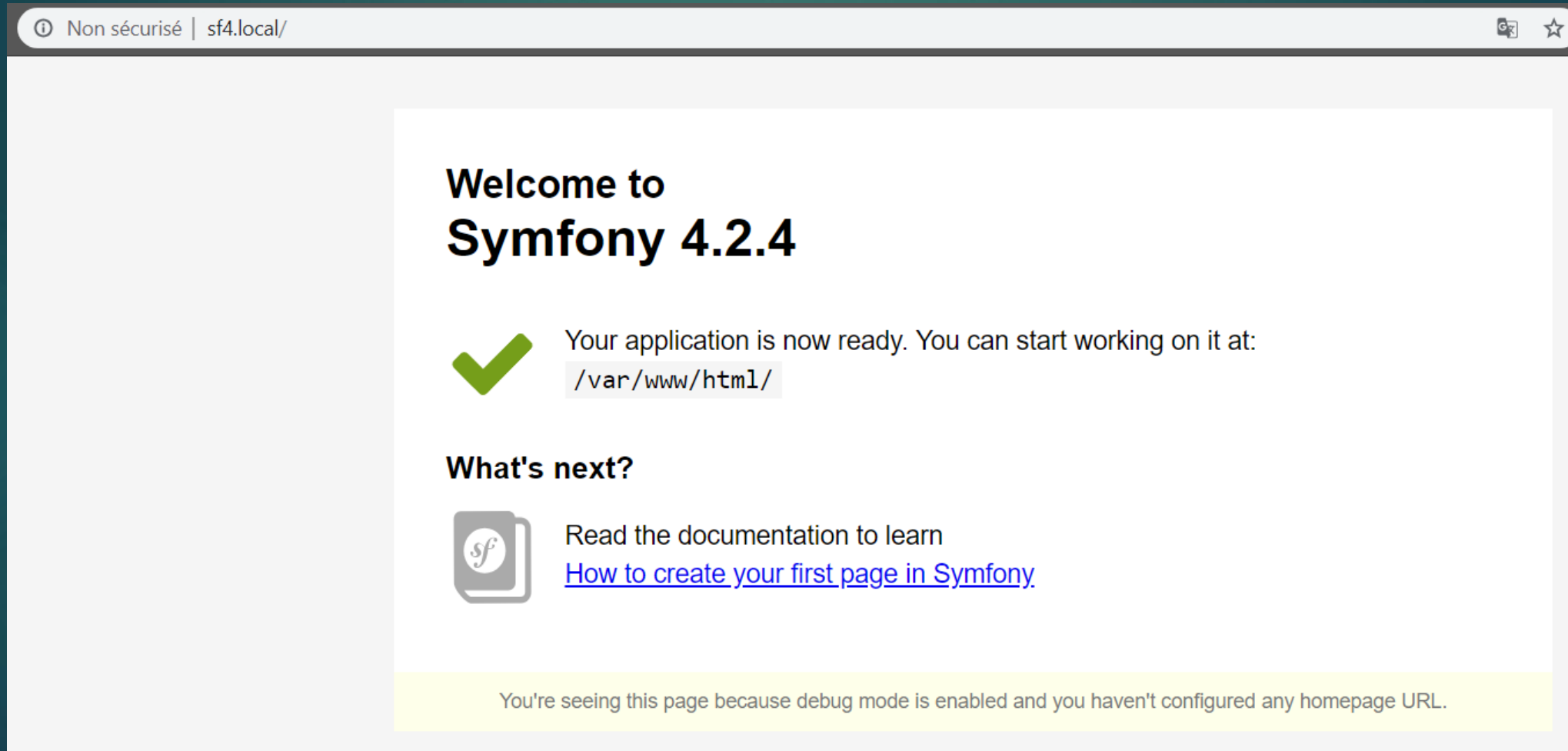
composer create-project symfony/website-skeleton my-project

```
root@c4b87f4dce5d:/var/www/html# composer create-project symfony/website-skeleton ./
Do not run Composer as root/super user! See https://getcomposer.org/root for details
Installing symfony/website-skeleton (v4.2.4.1)
- Installing symfony/website-skeleton (v4.2.4.1): Loading from cache
Created project in ./ - Downloading (100%)
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Package operations: 99 installs, 0 updates, 0 removals
- Installing ocramius/package-versions (1.4.0): Loading from cache
- Installing symfony/flex (v1.2.0): Loading from cache
- Installing symfony/polyfill-mbstring (v1.10.0): Loading from cache
- Installing symfony/contracts (v1.0.2): Loading from cache
- Installing doctrine/lexer (v1.0.1): Loading from cache
- Installing doctrine/annotations (v1.6.0): Loading from cache
- Installing doctrine/reflection (v1.0.0): Loading from cache
- Installing doctrine/event-manager (v1.0.0): Loading from cache
- Installing doctrine/collections (v1.5.0): Loading from cache
```

Cliquez pour ajouter des notes

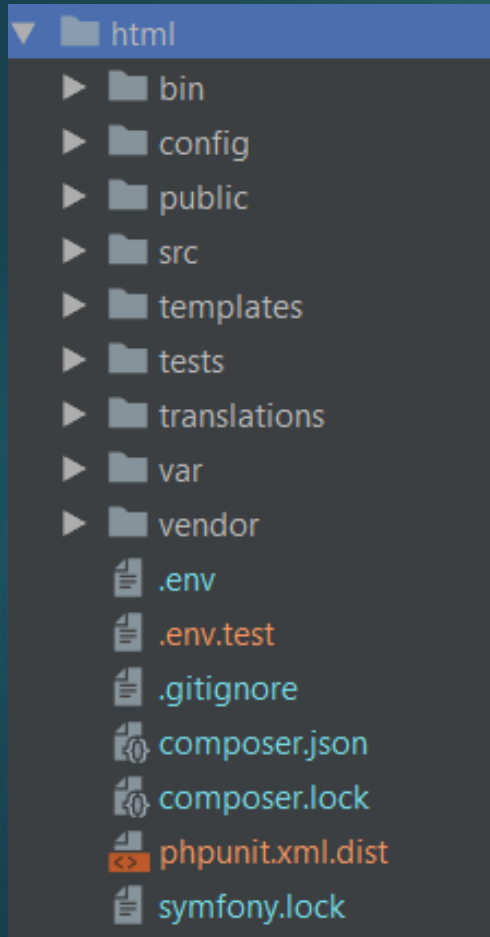
Visite de la homepage

17



Structure applicative

18



Etude d'une requête HTTP avec Symfony

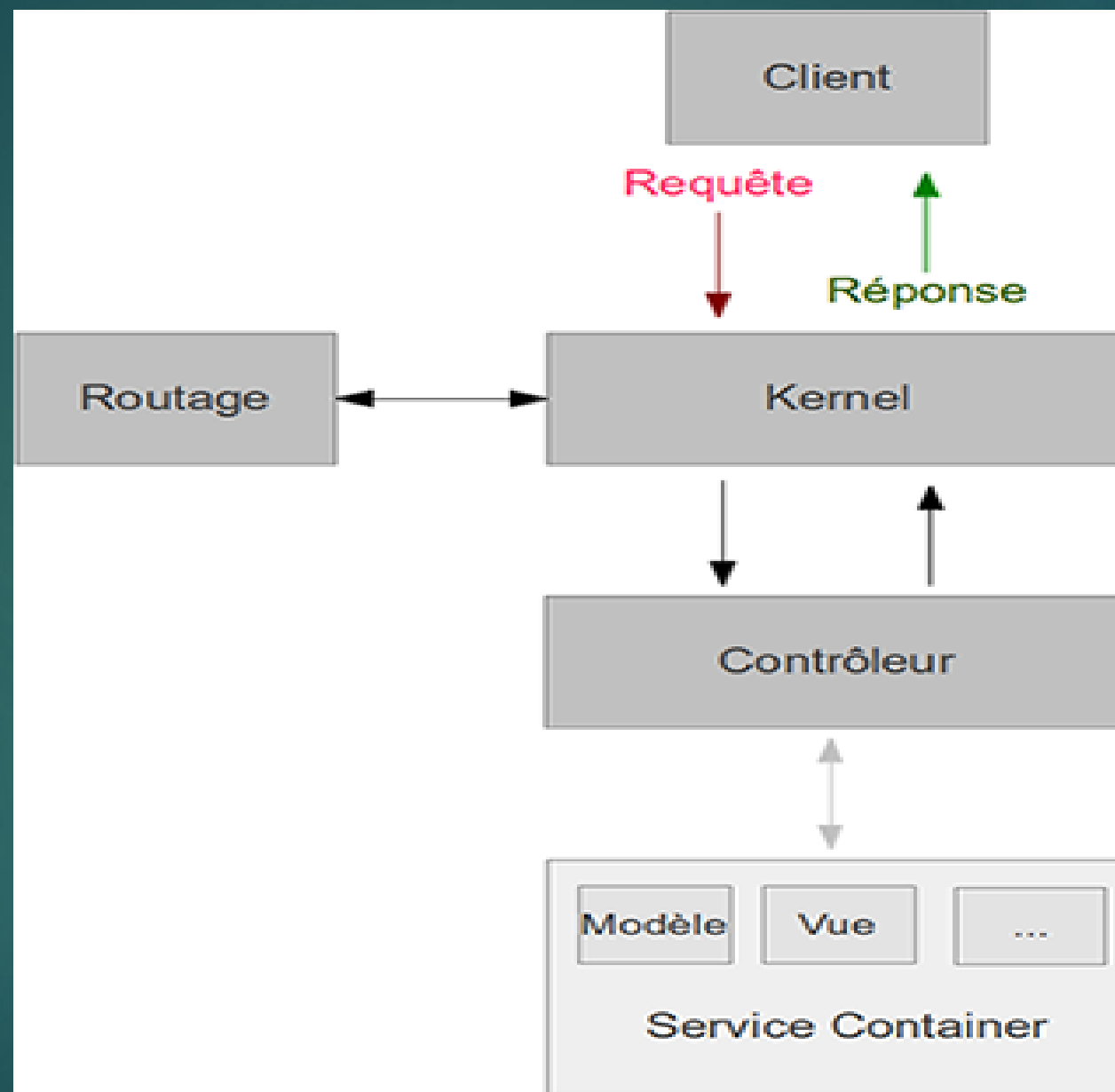
19

- ▶ Exemple : <https://www.mondomaine.com/articles/bienvenue.html>
- ▶ Sans framework : DOCUMENT_ROOT du serveur + dossier « articles » + fichier bienvenue.html
 - ▶ Difficile à maintenir pour un développeur
 - ▶ Non respect du DRY (Don't Repeat Yourself)
- ▶ Avec Symfony :
 - ▶ Contrôleur frontal du DOCUMENT_ROOT (fichier index.php du dossier public »)
 - ▶ Requête GET en interne sur /articles/bienvenue.html

Architecture MVC en revue

20

- ▶ articles/bienvenue.html a été défini dans les routes applicative
 - ▶ Contrôleur : ArticleController.php (CONTROLLER)
 - ▶ Action : voir
 - ▶ Fait appel au Model Article (MODEL)
 - ▶ Return : articles/voir.html.twig (VIEW)



Les bundles

22

- ▶ Composer permet d'ajouter des codes déjà existant
 - ▶ S'occupe des dépendances
 - ▶ Vérifie la bonne combinaison des packages
- ▶ Un bundle est un ensemble de fonctionnalité
 - ▶ API (FosRestBundle, ApiPlatform)
 - ▶ Upload de fichier (VichUploader)
 - ▶ Travail des images (LiipImagine)
 - ▶ ...

Installation d'un bundle

23

`composer require vich/uploader-bundle`

Installation d'un bundle

24

```
root@c4b87f4dce5d:/var/www/html# composer require vich/uploader-bundle
Do not run Composer as root/super user! See https://getcomposer.org/root for details
Using version ^1.9 for vich/uploader-bundle
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Restricting packages listed in "symfony/symfony" to "4.2.*"

Prefetching 10 packages ⓘ ⓘ
- Downloading (100%)

Package operations: 10 installs, 0 updates, 0 removals
- Installing symfony/templating (v4.2.4): Loading from cache
- Installing symfony/inflexor (v4.2.4): Loading from cache
- Installing symfony/property-access (v4.2.4): Loading from cache
- Installing symfony/options-resolver (v4.2.4): Loading from cache
- Installing symfony/intl (v4.2.4): Loading from cache
- Installing symfony/polyfill-intl-icu (v1.10.0): Loading from cache
- Installing symfony/form (v4.2.4): Loading from cache
- Installing jms/metadata (2.0.0): Loading from cache
- Installing behat/transliterator (v1.2.0): Loading from cache
- Installing vich/uploader-bundle (1.9.1): Loading from cache
Writing lock file
```


Installation d'un bundle

25

```
Generating autoload files
```

```
Symfony operations: 1 recipe (888ae5e77f79620048123a902dec8e4b)
```

- **WARNING** vich/uploader-bundle (>=1.5): From github.com/symfony/recipes-contrib:master
The recipe for this package comes from the "contrib" repository, which is open to community contributions.
Review the recipe at <https://github.com/symfony/recipes-contrib/tree/master/vich/uploader-bundle/1.5>

```
Do you want to execute this recipe?
```

```
[y] Yes
```

```
[n] No
```

```
[a] Yes for all packages, only for the current installation session
```

```
[p] Yes permanently, never ask again for this project
```

```
(defaults to n): y
```

- Configuring vich/uploader-bundle (>=1.5): From github.com/symfony/recipes-contrib:master

```
Executing script cache:clear [OK]
```

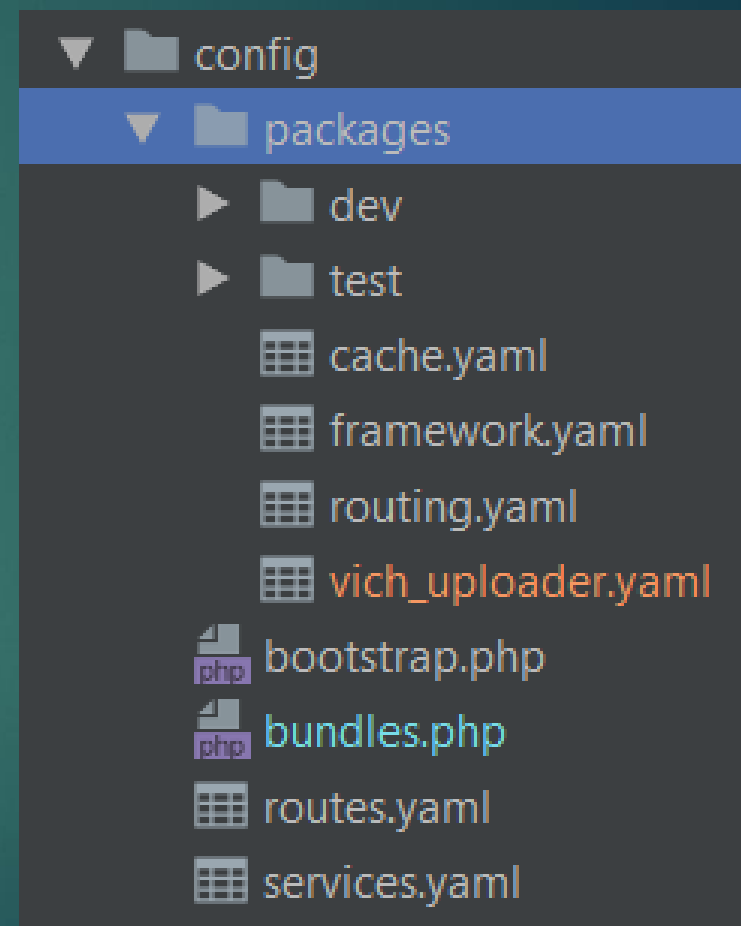
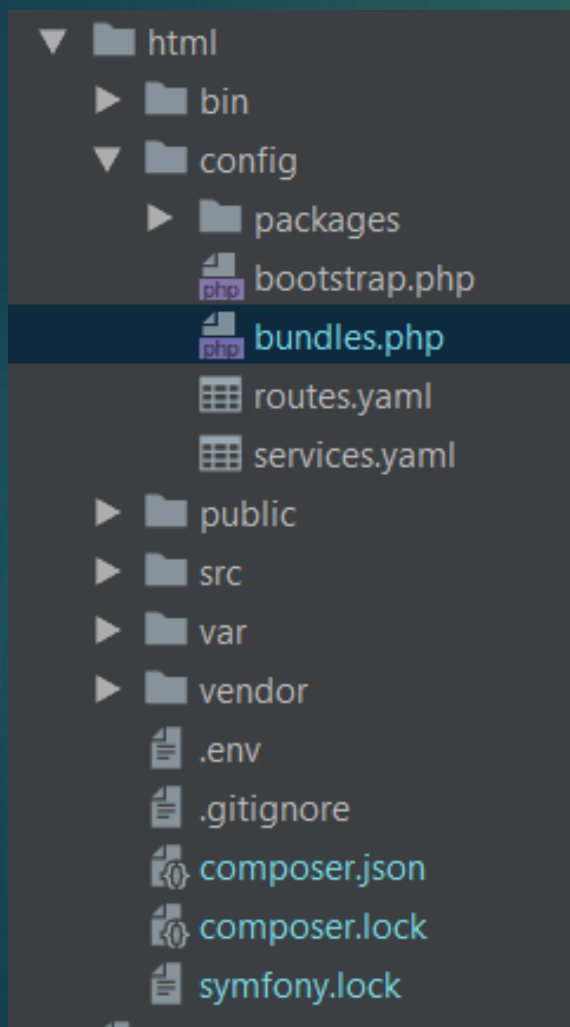
```
Executing script assets:install public [OK]
```

```
Some files may have been created or updated to configure your new packages.
```

```
Please review, edit and commit them: these files are yours.
```

Symfony Flex – La magie

26



Symfony Flex – La magie

27

- ▶ En bref
 - ▶ Lors d'une installation de bundle
 - ▶ Vérification et téléchargement de la version compatible avec le composer.lock
 - ▶ Ajout du paquet dans le composer.json
 - ▶ Inscription des fichier .lock pour les prochaines installation
 - ▶ Ajout d'une mention dans le fichier bundle.php pour prévenir du chargement de ce bundle automatiquement
 - ▶ Création (si besoin) d'un fichier .yaml de configuration

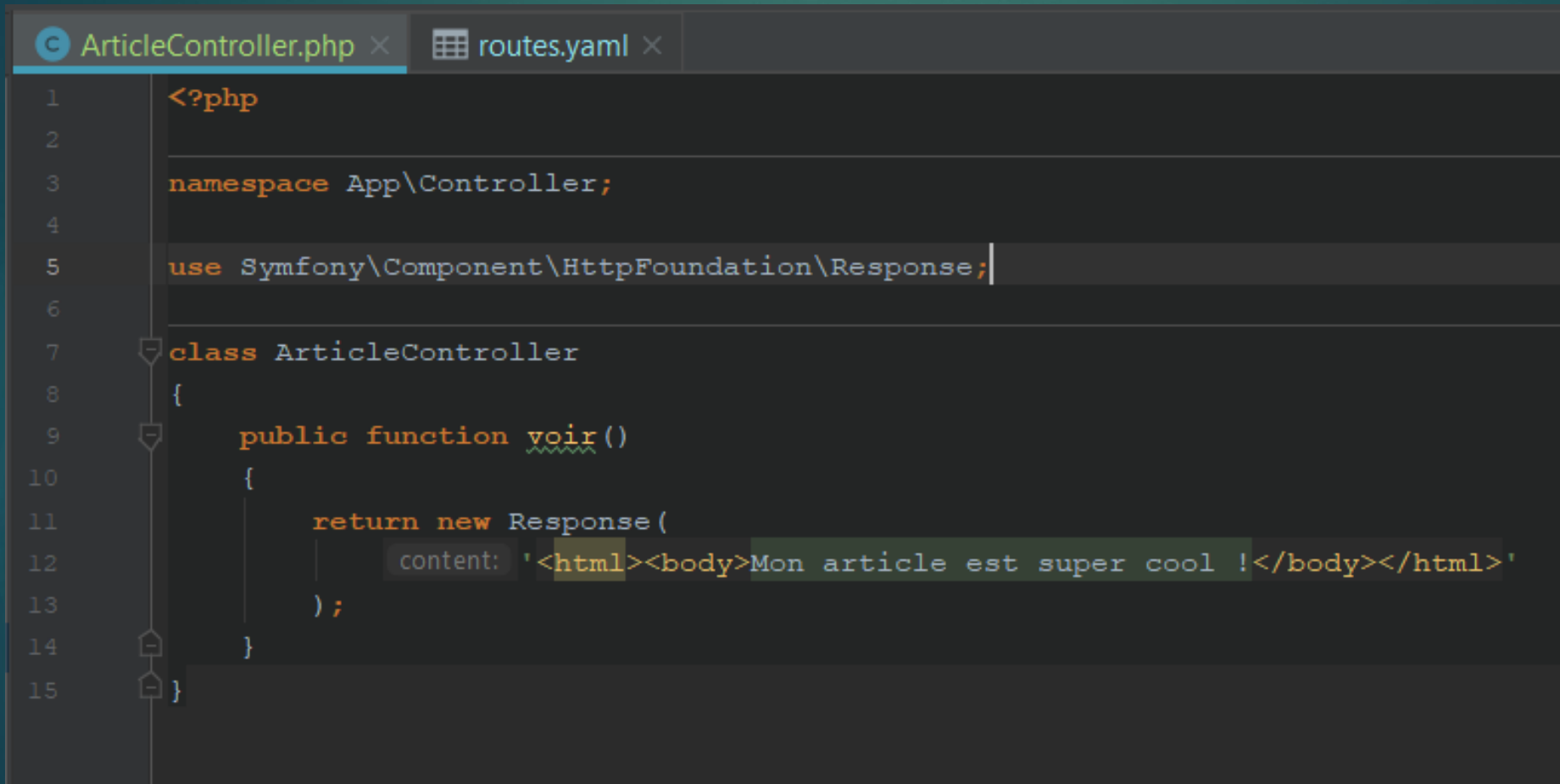
Controller

28

- ▶ Un controller permet d'orchestrer les services de notre application (les modèles, les vues, l'envoi de mail, la manipulation de fichier...)
- ▶ Listés dans le dossier `/src/Controller/`
- ▶ Norme de nommage `MonSuperController.php`

Controller – Classique

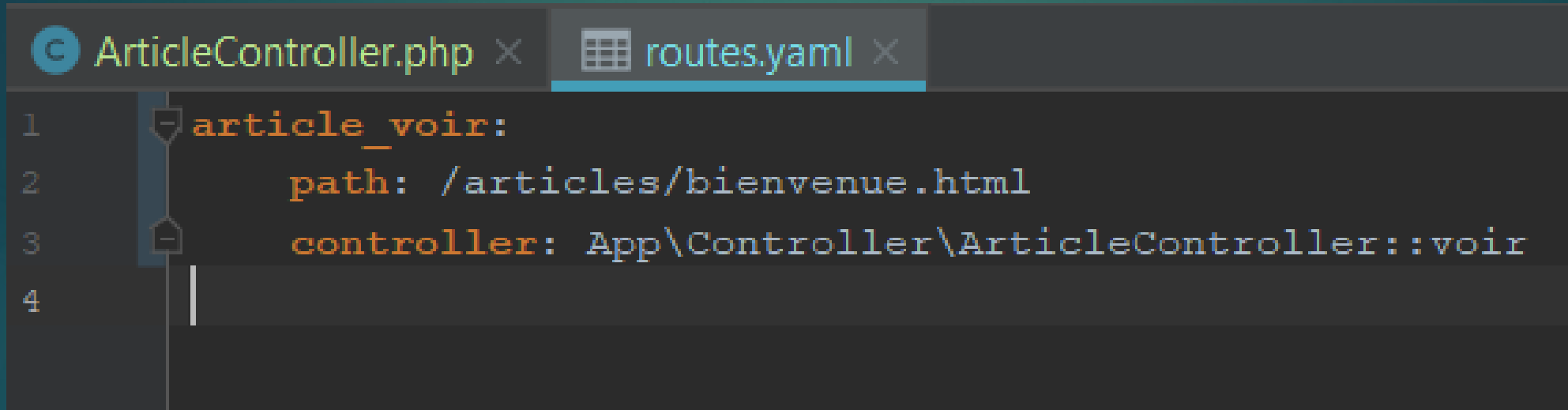
29



```
1 <?php
2
3 namespace App\Controller;
4
5 use Symfony\Component\HttpFoundation\Response;
6
7 class ArticleController
8 {
9     public function voir()
10     {
11         return new Response(
12             content: '<html><body>Mon article est super cool !</body></html>'
13         );
14     }
15 }
```

Controller – Classique

30



The image shows a code editor with two tabs: 'ArticleController.php' and 'routes.yaml'. The 'routes.yaml' tab is active and displays the following configuration:

```
1 article_voir:
2     path: /articles/bienvenue.html
3     controller: App\Controller\ArticleController::voir
4
```

Controller – Annotations

31

- ▶ En chargeant un bundle Symfony, on bénéficie de la puissance des commentaires PHP pour annoter des méthodes

composer require annotations

Controller – Annotations

32

```
<?php

namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class ArticleController
{
    /**
     * @Route("/articles/bienvenue.html", name="article_voir")
     */
    public function voir()
    {
        return new Response(
            content: '<html><body>Mon article est super cool !</body></html>'
        );
    }
}
```


Controller – Annotations²

33

```
<?php

namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

/**
 * @Route("/articles")
 */
class ArticleController
{
    /**
     * @Route("/bienvenue.html", name="article_voir")
     */
    public function voir()
    {
        return new Response(
            content: '<html><body>Mon article est super cool !</body></html>'
        );
    }
}
```

- ▶ S'occupe de la partie design de l'application
- ▶ Aucun code HTML/CSS présent dans les Controllers
- ▶ Moteur de template TWIG majoritairement utilisé
 - ▶ Peut être simplement du PHP
 - ▶ Peut être un autre moteur de template tel que Smarty

View – Twig

35

```
<?php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\Routing\Annotation\Route;

/**
 * @Route("/articles")
 */
class ArticleController extends AbstractController
{
    /**
     * @Route("/bienvenue.html")
     */
    public function voir()
    {
        return $this->render('view: articles/bienvenue.html.twig');
    }
}
```

▼ templates
▼ articles
 ▼ bienvenue.html.twig

View – Passage de paramètres

36

```
public function voir()  
{  
    $prenom = "ESIEE Amiens";  
  
    return $this->render( view: 'articles/bienvenue.html.twig', [  
        'name' => $prenom  
    ] );  
}
```

```
<!DOCTYPE html>  
<html>  
    <head>  
        <meta charset="utf-8">  
    </head>  
    <body>  
        <h1>Bienvenue {{ name }}</h1>  
    </body>  
</html>
```