

Présentation du framework Symfony



Table des matières

I. Contexte	3
II. Qu'est-ce qu'une bibliothèque et un framework ?	3
III. Exercice : Appliquez la notion	4
IV. Pourquoi utiliser le framework Symfony ?	5
V. Exercice : Appliquez la notion	6
VI. Créer un projet Symfony	7
VII. Exercice : Appliquez la notion	10
VIII. Auto-évaluation	10
A. Exercice final	10
B. Exercice : Défi	12
Solutions des exercices	12

I. Contexte

Durée : 45 minutes

Environnement de travail : Local

Pré-requis : Avoir installé GitBash (Windows), XAMPP et Composer

Contexte

Jusqu'à présent, vous avez probablement commencé à développer de petites applications en PHP, que ce soit en procédural ou grâce à la programmation orientée objet. Elles fonctionnent probablement très bien et possèdent peu de bugs, ce qui ne vous a pas empêché d'être confrontés à plusieurs problématiques : comment organiser votre code, nommer vos dossiers et fichiers, réutiliser un code existant, et surtout ne pas sans arrêt « réinventer la roue ».

Pour répondre à ces questions, des bibliothèques de classes et de fonctions ont vu le jour pour couvrir une ou plusieurs tâches de développement répétitives (accès à la base de données, génération de gabarits HTML, envoi d'e-mails, etc.). Ces bibliothèques de composants ont peu à peu mené à la création de cadres de travail, aussi appelés frameworks : nous allons voir en quoi ils permettent de nous faciliter la tâche en produisant un code plus simple, plus découpé et de meilleure qualité.

II. Qu'est-ce qu'une bibliothèque et un framework ?

Objectifs

- Définir ce qu'est une bibliothèque
- Définir ce qu'est un framework
- Différencier une bibliothèque d'un framework

Mise en situation

Dans votre quotidien de développeur, vous n'aurez probablement pas le temps de réinventer la roue sans arrêt. Pour réduire au maximum la répétition de tâches de développement, vous serez à coup sûr amenés à utiliser des bibliothèques, et peut-être même à en développer.

Imaginez que vous ayez à un moment donné besoin d'envoyer un e-mail. Vous aurez le choix de développer à partir de zéro ou de partir depuis une bibliothèque comprenant déjà toutes ces fonctionnalités. Une telle bibliothèque peut vous paraître simple à développer, mais sachez que les options peuvent se révéler nombreuses, de la logique d'envoi d'e-mails – avec la gestion de plusieurs transports (SMTP, mais aussi des fournisseurs tiers comme Gmail) – jusqu'à la génération de l'e-mail, en y attachant un expéditeur, un destinataire, des fichiers joints, un contenu HTML, un contenu texte, en passant par l'encryptage et la gestion des bugs.

Qu'est-ce qu'une bibliothèque ?

On parle ici, bien sûr, de bibliothèque logicielle. Rien à voir donc avec l'endroit physique où l'on peut trouver des livres, à une exception près. Dans une bibliothèque physique, vous pouvez accéder à des ressources (des livres) partageant une conception autour d'une idée : une bibliothèque logicielle permet d'accéder à une ressource (un code informatique) résultant de la réponse à une problématique donnée.

Exemple

Lorsque vous vous lancez dans la préparation d'une mousse au chocolat, vous savez que vous avez certaines étapes à effectuer, que vous aurez peut-être à faire dans le cadre d'autres préparations, comme monter des œufs en neige. Vous avez donc le choix entre monter les œufs vous-mêmes « à la main », construire votre propre batteur à œufs ou utiliser un batteur créé par quelqu'un d'autre.

En développement, c'est pareil. Si vous avez besoin d'envoyer un e-mail à un utilisateur suite à son inscription, vous pouvez implémenter vous-même la logique d'envoi d'e-mail en utilisant des fonctions de PHP, comme `mail()` par exemple. Vous pouvez choisir d'isoler cette logique dans un fichier ou un ensemble de fichiers à part pour pouvoir le réutiliser à d'autres endroits de votre application, comme pour une newsletter, ou d'utiliser une bibliothèque d'envoi d'e-mails créée par quelqu'un d'autre, telle que Symfony Mailer.

Qu'est-ce qu'un framework ?

Un framework est composé d'une ou plusieurs bibliothèques. Pour faire simple, un framework (ou cadre de travail) permet de définir un ensemble de méthodes de travail, de consignes. Celles-ci peuvent s'appuyer sur des concepts déjà existants ou sur des concepts qui lui sont propres.

Pour reprendre notre exemple de la pâtisserie, on peut considérer une cuisine comme un framework. Au sein de ce cadre de travail, on a des consignes, telles que porter des gants, laver ses ustensiles après chaque utilisation : c'est l'équivalent des bonnes pratiques. On a aussi des outils qui nous permettent de réaliser des tâches, comme un batteur pour battre un œuf, un four pour cuire un gâteau ou un réfrigérateur pour refroidir une crème : ce sont nos librairies. On utilise ces outils pour nous faciliter la tâche et aller plus vite.

Fondamental À retenir

En conclusion, une bibliothèque est un ensemble de classes et de méthodes permettant de réaliser une tâche bien précise, telle que l'envoi d'e-mails, l'accès à la base de données... Alors qu'un framework est un cadre de travail mettant en place des bonnes pratiques (normes de codage, nommage des fichiers, etc.) et pouvant être constitué d'un ensemble de bibliothèques. Parmi ceux-ci, VueJS, ou encore Symfony, dont nous allons parler dans la prochaine partie.

Complément

Bibliothèque logicielle¹

Framework²

When should I use a framework?³

Exercice : Appliquez la notion

Exercice

Qu'est-ce qui permet d'envoyer des e-mails ?

- ☐ Une bibliothèque
- ☐ Un framework

Exercice

Qu'est-ce qui permet de mettre en place des bonnes pratiques de développement ?

¹ https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle

² <https://fr.wikipedia.org/wiki/Framework>

³ <https://symfony.com/when-use-a-framework> [en]

- ☐ Une bibliothèque
- ☐ Un framework

Exercice

Qu'est-ce qui permet de ne pas réinventer la roue ?

- ☐ Une bibliothèque
- ☐ Un framework

III. Pourquoi utiliser le framework Symfony ?

Objectifs

- Détailler l'intérêt d'un framework
- Décrire les avantages liés au framework Symfony

Mise en situation

L'intérêt d'utiliser des frameworks est simple : gagner du temps. Nous allons voir les différents avantages liés à leur utilisation, mais également leurs quelques inconvénients.

Contexte

L'utilisation d'un framework offre plusieurs bénéfices par rapport à l'utilisation simple de bibliothèques ou au fait de réécrire le même code encore et encore, qui peut se révéler répétitif. Cependant, nous allons aussi aborder le fait qu'il n'y a pas que des avantages à utiliser un framework, même si les inconvénients se révèlent vite compensés.

De plus, il existe plusieurs frameworks dans le monde du développement PHP. Nous allons donc voir plus en détails quels sont les avantages à utiliser Symfony, l'un des frameworks les plus populaires. Beaucoup des concepts que nous verrons dans le cadre de Symfony peuvent être transposés sur d'autres frameworks, pour peu qu'ils utilisent aussi une approche MVC.

Pourquoi utiliser un framework ?

Tout d'abord, il faut savoir qu'il n'est pas obligatoire d'utiliser un framework lorsque vous faites du développement. Néanmoins, il peut vous faire gagner énormément de temps et vous faciliter la tâche, que ce soit pour un gros projet ou pour des projets beaucoup plus petits. Vous verrez cela plus tard lorsque nous commencerons à coder.

Un des nombreux avantages liés à l'utilisation d'un framework est de bénéficier d'un cadre de travail plus ou moins imposé (arborescence de dossiers, nommage de fichier, etc.), aussi appelé « architecture logicielle ».

Comme nous l'avons vu précédemment, un autre avantage majeur est la mise à disposition d'une ou plusieurs bibliothèques de composants permettant d'éviter de redévelopper sans cesse les mêmes composants basiques d'applications (gestion des sessions, génération de gabarit HTML, etc.).

Parmi les inconvénients liés à l'utilisation d'un framework, on peut noter que la courbe d'apprentissage peut être plus pentue que dans le cadre de l'apprentissage de PHP, même si le gain sur le long terme contrebalance largement cette difficulté. Un cadre de travail plus ou moins imposé peut, par exemple, vous empêcher de faire les choses à votre manière. Il est donc important de choisir un framework qui soit simple à appréhender et facilement personnalisable.

Pourquoi utiliser Symfony ?

Il existe une importante panoplie de frameworks dans le monde du développement PHP, tels que Symfony, Laravel, Zend Framework, CakePHP ou encore Phalcon, pour ne citer que les plus populaires. Ceux-ci font partie de la famille des frameworks *open source*.

Nous nous concentrerons dans ce module sur le framework Symfony, car il répond à plusieurs problématiques :

- Création en 2005, 2000 contributeurs, 48 000 *commits* et 23 000 étoiles GitHub,
- Largement plébiscité par la communauté PHP et nombreuses ressources disponibles en lignes (cours, bundles, tutoriel, forums),
- Projet *open source*, donc une grande majorité de contributions proviennent de la communauté, ce qui lui assure un bon niveau de mise à jour et une pérennité,
- Utilisé par de grands noms tels que Yahoo, Dailymotion, et dans certains produits, tels que Drupal, Joomla! et phpBB,
- Facile à mettre à jour et assure une grande maintenabilité par la mise en place de bonnes pratiques,
- Compatible avec de nombreuses versions de PHP et avec tous les systèmes d'exploitation.

Syntaxe À retenir

Il n'est pas obligatoire d'utiliser un framework, mais c'est très fortement conseillé pour les avantages qu'il peut apporter, tels que la qualité et la maintenabilité du code, la mise à jour des applications, ou encore la comptabilité et le support d'une large communauté.

Complément

Why should I use a framework?¹

Open source²

Symfony³

Laravel⁴

Zend Framework⁵

CakePHP⁶

Phalcon⁷

Six good reasons to use Symfony⁸

Exercice : Appliquez la notion

Exercice

Il est obligatoire d'utiliser un framework pour bien développer.

¹ <https://symfony.com/why-use-a-framework> [en]

² https://fr.wikipedia.org/wiki/Open_source

³ <https://symfony.com/> [en]

⁴ <https://laravel.com/> [en]

⁵ <https://framework.zend.com/> [en]

⁶ <https://cakephp.org/> [en]

⁷ <https://phalcon.io/fr-fr> [en]

⁸ <https://symfony.com/six-good-reasons> [en]

- ☐ Vrai
- ☐ Faux

Exercice

Le framework Symfony est connu pour :

- ☐ Sa grande communauté
- ☐ Sa difficulté de mise à jour
- ☐ Sa compatibilité sur les différents systèmes d'exploitation et versions de PHP

IV. Créer un projet Symfony

Objectifs

- Créer un projet Symfony
- Configurer l'environnement de travail
- Afficher un « *hello world* »

Créer un projet Symfony

Tout d'abord, vous devez avoir installé PHP 7.4 et Composer sur votre machine, et les avoir disponibles depuis votre ligne de commande.

Complément	Installer Composer
-------------------	---------------------------

Si vous n'avez pas encore installé Composer, vous pouvez le faire en suivant la documentation d'installation disponible sur le site officiel¹.

Pour installer un projet Symfony sur Windows, il suffit de se placer dans le répertoire où vous avez installé XAMPP (C:\xampp), de créer le répertoire "**apps**" qui contiendra l'application, puis d'exécuter la commande de création du projet grâce à Composer.

```
1 cd C:\xampp\  
2 mkdir apps  
3 cd apps  
4 composer create-project symfony/skeleton symfony
```

Pour installer un projet Symfony sur Mac, il suffit de se placer dans le répertoire "htdocs" de XAMPP qui contiendra l'application et enfin d'exécuter la commande de création du projet grâce à Composer.

```
1 cd /Applications/XAMPP/xamppfiles/htdocs  
2 composer create-project symfony/skeleton symfony
```

Cette commande crée un nouveau répertoire "**symfony**" contenant une architecture Symfony.

```
1 symfony/  
2 | .env  
3 | bin/console  
4 | composer.json  
5 | composer.lock  
6 | config/  
7 | public/index.php
```

¹ <https://getcomposer.org/download/>

```
8 | src/
9 | symfony.lock
10 | var/
11 | vendor/
```

Remarque

Il s'agit ici d'une architecture correspondant à **symfony/skeleton**. Ce squelette d'application comporte le strict minimum pour faire fonctionner un projet Symfony. Il en existe un autre qui vous permet d'installer un projet entier pour une application web, appelé **symfony/website-skeleton**. Bien qu'il puisse paraître plus adapté dans certain cas, Symfony est conçu pour être facilement composable, c'est-à-dire que vous pouvez facilement ajouter des composants grâce à Composer. Je vous conseille donc d'installer la version de base **symfony/skeleton**, et nous verrons ensuite comment ajouter ce dont nous avons besoin.

Servir son application avec XAMPP

Nous allons ensuite configurer Apache pour accéder à notre site web. Pour cela, nous allons ouvrir le fichier "`C:\xampp\apache\conf\extra\httpd-vhosts.conf`" avec un éditeur de texte et ajouter les lignes suivantes :

```
1 <VirtualHost *:80>
2     ServerName symfony.localhost
3
4     DocumentRoot "C:/xampp/apps/symfony/public"
5     DirectoryIndex index.php
6
7     <Directory "C:/xampp/apps/symfony/public">
8         Require all granted
9
10        FallbackResource /index.php
11    </Directory>
12</VirtualHost>
```

Servir son application avec MAMP

Nous allons ensuite configurer Apache pour accéder à notre site web. Pour cela, nous allons ouvrir le fichier "`/Applications/XAMPP/xamppfiles/etc/httpd.conf`" avec un éditeur de texte et supprimer le commentaire (le "#") devant la ligne incluant le fichier "`httpd-vhosts.conf`" :

```
1 # Virtual Hosts
2 Include etc/extra/httpd-vhosts.conf
```

Ouvrez ensuite le fichier "`/Applications/XAMPP/etc/extra/httpd-vhosts.conf`". En bas du fichier, se trouvent deux exemples de définitions de virtual hosts. Modifiez les afin qu'ils ressemblent à cela :

```
1 <VirtualHost *:80>
2     ServerName symfony.localhost
3     DocumentRoot "/Applications/XAMPP/xamppfiles/htdocs/symfony/public"
4     DirectoryIndex index.php
5
6     <Directory "/Applications/XAMPP/xamppfiles/htdocs/symfony/public">
7         Require all granted
8         FallbackResource /index.php
9     </Directory>
10</VirtualHost>
```

Redémarrez ensuite le serveur Apache.

Nous allons ouvrir le panneau de contrôle de XAMPP puis, sur la ligne du module Apache, cliquer sur "Stop" puis "Start" (nous devrions avoir le texte "Apache" souligné en vert).

Nous allons ensuite ouvrir le fichier hosts ("C:\Windows\System32\drivers\etc\hosts" sur Windows et "/etc/hosts" sur Mac) et ajouter à la fin les lignes suivantes :

```
1 127.0.0.1      symfony.localhost
```

Nous pouvons maintenant ouvrir notre navigateur préféré et taper dans la barre d'adresse l'URL : `http://symfony.localhost13`.

Afficher notre premier « Hello world! »

À ce stade, nous n'avons pas encore créé de page : cette page par défaut n'est accessible que pendant le développement. Pour créer notre première page, nous avons besoin de modifier deux fichiers.

Dans le fichier `config/routes.yaml`, nous avons besoin de décommenter les lignes, comme ceci :

```
1 index:
2   path: /
3   controller: App\Controller\DefaultController::index
```

Rendons-nous ensuite dans le répertoire `src/Controller` et créons un fichier `DefaultController.php`, puis insérons le code suivant :

```
1 <?php
2
3 namespace App\Controller;
4
5 use Symfony\Component\HttpFoundation\Response;
6
7 class DefaultController
8 {
9     public function index()
10    {
11        return new Response('Hello world!');
12    }
13 }
```

Rafraîchissons la page : nous devrions obtenir notre premier « *Hello world!* ».

Hello world!

Syntaxe À retenir

Comme nous l'avons vu, il est très simple d'installer Symfony grâce à Composer, et tout aussi simple de **servir** l'application grâce à XAMPP. Nous pourrions modifier le texte affiché comme bon nous semble, en y ajoutant notre nom, par exemple.

Complément

The Big Picture¹

Exercice : Appliquez la notion

Exercice

Avec quel outil avons-nous créé notre projet Symfony ?

- ☐ Une ligne de commande
- ☐ Une archive
- ☐ Git
- ☐ Composer

Exercice

Comment est servie notre application Symfony ?

- ☐ Avec Apache
- ☐ Avec XAMPP
- ☐ Par magie

Exercice

Quelle version du framework Symfony avons-nous installé ?

- ☐ Symfony 3
- ☐ Symfony 4
- ☐ Symfony 5
- ☐ Symfony 6

V. Auto-évaluation

A. Exercice final

Exercice

Exercice

Une bibliothèque et un framework, c'est la même chose.

¹ https://symfony.com/doc/current/quick_tour/the_big_picture.html

- ☐ Vrai
- ☐ Faux

Exercice

Une bibliothèque permet de mettre en place des bonnes pratiques.

- ☐ Vrai
- ☐ Faux

Exercice

Si j'ai besoin d'une fonctionnalité, je peux :

- ☐ Créer ma propre bibliothèque
- ☐ Utiliser une bibliothèque existante
- ☐ Utiliser un framework

Exercice

Un framework permet de :

- ☐ Adopter des bonnes pratiques
- ☐ Travailler à plusieurs plus facilement
- ☐ Accéder à un ensemble de composants afin de ne pas réinventer la roue
- ☐ Ne pas développer

Exercice

Symfony est une bibliothèque PHP.

- ☐ Vrai
- ☐ Faux

Exercice

L'utilisation d'un framework ne comporte que des avantages.

- ☐ Vrai
- ☐ Faux

Exercice

Le framework Symfony est un projet maintenu par :

- ☐ Une personne
- ☐ Une dizaine de personnes
- ☐ Plusieurs centaines de personnes
- ☐ Plusieurs milliers de personnes

Exercice

Combien faut-il de commandes pour créer un projet Symfony avec Composer ?

- ☐ 0
- ☐ 1
- ☐ 2
- ☐ Plus de 5

Exercice

De quoi avez-vous besoin pour faire fonctionner un projet Symfony ?

- ☐ Un serveur HTTP
- ☐ PHP
- ☐ Un système de gestion de base de données

Exercice

Quelle est la dernière version de Symfony ?

- ☐ 4
- ☐ 5
- ☐ 6

B. Exercice : Défi

L'exercice final de ce cours porte sur la mise en place d'un projet Symfony.

Question

Installez votre projet Symfony en utilisant **Composer** sur votre ordinateur. Pour cela, rien de plus simple : vous pouvez reproduire pas à pas les différentes étapes de la partie précédente.

Pour changer un petit peu, affichez votre prénom.


Solutions des exercices

Exercice p. 4 Solution n°1

Exercice

Qu'est-ce qui permet d'envoyer des e-mails ?


☒ Une bibliothèque☐ Un framework

 Pour envoyer un e-mail, vous aurez besoin d'une bibliothèque intégrant cette fonctionnalité, telle que Symfony Mailer.

Exercice

Qu'est-ce qui permet de mettre en place des bonnes pratiques de développement ?


☐ Une bibliothèque☒ Un framework

 Une bibliothèque est une boîte noire permettant d'intégrer une fonctionnalité, elle n'impose aucune bonne pratique de développement, contrairement à un framework.

Exercice

Qu'est-ce qui permet de ne pas réinventer la roue ?

☒ Une bibliothèque☒ Un framework


 Une bibliothèque évite de redévelopper à chaque projet la même fonctionnalité.
Un framework permet de mettre en place les mêmes bonnes pratiques sur chaque projet.
Ils sont donc tous les deux utiles pour ne pas avoir à réinventer la roue !

Exercice p. 6 Solution n°2

Exercice

Il est obligatoire d'utiliser un framework pour bien développer.


☐ Vrai☒ Faux

 L'utilisation d'un framework est conseillé pour diverses raisons, mais il n'est pas obligatoire pour bien développer.

Exercice

Le framework Symfony est connu pour :

☒ Sa grande communauté☐ Sa difficulté de mise à jour☒ Sa compatibilité sur les différents systèmes d'exploitation et versions de PHP

 Le framework Symfony facilite la mise à jour des applications construites avec lui, offre aussi une bonne compatibilité avec les différents systèmes, et sa communauté lui assure un support et une pérennité.

Exercice p. 10 Solution n°3

Exercice

Avec quel outil avons-nous créé notre projet Symfony ?

- ☒ Une ligne de commande
- ☐ Une archive
- ☐ Git
- ☒ Composer



Nous avons utilisé Composer en ligne de commande pour créer notre projet Symfony. Sachez qu'il existe d'autres façons de faire.

Exercice

Comment est servie notre application Symfony ?

- ☒ Avec Apache
- ☐ Avec XAMPP
- ☐ Par magie



XAMPP ne sert pas réellement notre application. Il s'agit d'un outil comprenant Apache (qui est un serveur HTTP), PHP (qui interprète nos scripts PHP) et MySQL (qui gère la persistance de nos données). Le seul élément obligatoire pour servir une application web est un serveur HTTP. Ce n'est évidemment pas de la magie.

Exercice

Quelle version du framework Symfony avons-nous installé ?

- ☐ Symfony 3
- ☐ Symfony 4
- ☒ Symfony 5
- ☐ Symfony 6



La version du framework Symfony utilisée est indiquée sur la page d'accueil par défaut. Dans notre cas, il s'agit de la version 5.

Exercice p. 10 Solution n°4

Exercice

Une bibliothèque et un framework, c'est la même chose.

- ☐ Vrai
- ☒ Faux




Une bibliothèque implémente une fonctionnalité. Un framework utilise des bibliothèques, mais offre un cadre de travail et des bonnes pratiques.

Exercice

Une bibliothèque permet de mettre en place des bonnes pratiques.


- ☐ Vrai
- ☒ Faux

 C'est le framework qui permet de mettre en place de bonnes pratiques.

Exercice

Si j'ai besoin d'une fonctionnalité, je peux :


- ☒ Créer ma propre bibliothèque
- ☒ Utiliser une bibliothèque existante
- ☐ Utiliser un framework

 Lorsque vous avez besoin d'une fonctionnalité, telle que l'envoi d'un e-mail, vous pouvez choisir de développer votre propre bibliothèque ou en utiliser une existante. L'utilisation d'un framework, même s'il comporte un composant permettant d'envoyer des e-mails, ne permettra pas d'envoyer un e-mail directement.

Exercice

Un framework permet de :


- ☒ Adopter des bonnes pratiques
- ☒ Travailler à plusieurs plus facilement
- ☒ Accéder à un ensemble de composants afin de ne pas réinventer la roue
- ☐ Ne pas développer

 Un framework n'a rien de magique. Il vous permettra de développer plus facilement vos applications, mais ne fera pas tout le travail à votre place.

Exercice

Symfony est une bibliothèque PHP.


- ☐ Vrai
- ☒ Faux

 Symfony est un framework PHP, qui est composé d'un ensemble de composants (bibliothèque).

Exercice

L'utilisation d'un framework ne comporte que des avantages.


- ☐ Vrai
- ☒ Faux

 L'utilisation d'un framework peut parfois imposer un formalisme, tel qu'une arborescence de fichiers, un nommage des dossiers, l'utilisation de certaines conventions... Ce qui peut vous empêcher de faire ce que vous voulez. De plus, l'apprentissage peut prendre un certain temps au début, mais il sera largement rentabilisé sur la durée.

Exercice


Le framework Symfony est un projet maintenu par :

- ☐ Une personne
- ☐ Une dizaine de personnes
- ☐ Plusieurs centaines de personnes

- ☒ Plusieurs milliers de personnes
-  Plus de 2000 personnes ont contribué au framework Symfony.


Exercice

Combien faut-il de commandes pour créer un projet Symfony avec Composer ?

- ☐ 0
- ☒ 1
- ☐ 2
- ☐ Plus de 5
-  Avec Composer, une seule commande suffit pour créer un projet : *composer create-project symfony/skeleton monprojet*.


Exercice

De quoi avez-vous besoin pour faire fonctionner un projet Symfony ?

- ☒ Un serveur HTTP
- ☒ PHP
- ☐ Un système de gestion de base de données
-  Pour faire fonctionner un projet Symfony, vous n'avez besoin que de PHP et d'un serveur HTTP. Vous pouvez ajouter un système de gestion de base de données si vous voulez stocker des données (telles que vos utilisateurs), mais cela n'est pas obligatoire.

Exercice

Quelle est la dernière version de Symfony ?

- ☐ 4
- ☒ 5
- ☐ 6
-  La dernière version de Symfony est la version 5.

Exercice p. Solution n°5

config/routes.yaml

```
1 index:
2   path: /
3   controller: App\Controller\DefaultController::index
```

src/Controller/DefaultController.php

```
1 <?php
2
3 namespace App\Controller;
4
5 use Symfony\Component\HttpFoundation\Response;
6
7 class DefaultController
```



```
8 {  
9   public function index()  
10  {  
11     return new Response('Bonjour Nicolas !');  
12  }  
13 }  
14
```