

SYMPHONY

Introduction

Créé par Étienne Dauvergne

L'ÉCOLE DE FORMATION PROFESSIONNELLE CRÉÉE PAR DES INFORMATICIENS

Créée en 1994, l'école de la filière numérique est actuellement implantée sur Nantes, Angers et Rennes. IMIE vous propose d'obtenir des diplômes reconnus, et de maîtriser les technologies les plus récentes.

HISTORIQUE

... Sensio Framework

2007 Symphony 1.0

2011 Symphony 2.0

2015 Symphony 3.0

FRAMEWORK

Un ensemble de bibliothèques et de composants permettant de développer plus **efficacement**.

*HttpFoundation, Form, Validation, Config,
DependencyInjection, EventDispatcher*

Symfony2 is a reusable set of standalone, decoupled, and cohesive PHP components that solve common web development problems.

Fabien Potencier «What is Symfony2?» October 25, 2011

Then, based on these components,
Symfony2 is also a full-stack web framework.

ICONVÉNIENTS

- Architecture imposée
- Courbe d'apprentissage

AVANTAGES

- Synthèse des bonnes pratiques
- Cadre de développement

PRE-REQUIS

Version PHP : 5.5.9+

RECOMMANDATIONS

Extensions : mb_string, intl, opcache, pdo_*

php.ini :

short_open_tag = Off

magic_quotes_gpc = Off

register_globals = Off

session.auto_start = Off

SYMFONY INSTALLER

Linux:

```
sudo curl -Ls https://symfony.com/installer \
  -o /usr/local/bin/symfony
sudo chmod a+x /usr/local/bin/symfony
symfony new project_name
```

Windows:

```
php -r "file_put_contents('symfony',
  file_get_contents('https://symfony.com/installer'));"
php symfony new project_name
```

CURL ERROR 60

SSL certificate problem: unable to get local issuer certificate

```
[curl]  
; https://curl.haxx.se/ca/cacert.pem  
curl.cainfo = C:\php\extras\ssl\cacert.pem
```

COMPOSER

Linux

```
composer create-project \  
    symfony/framework-standard-edition project_name
```

Windows:

```
php composer.phar create-project \  
    symfony/framework-standard-edition project_name
```


PREMIER DÉMARRAGE

Lancer le serveur PHP intégré :


```
cd /repertoire/racine/du/projet  
php bin/console server:run
```

http://127.0.0.1:8000

Welcome to Symfony 3.2.0


 Your application is now ready. You can start working on it at: ~


What's next?


 Read the documentation to learn
[How to create your first page in Symfony](#)


200

@ homepage 147 ms 2.0 MB

 anon.

 2 ms

 3.2.0



MISE A JOUR DES DÉPENDANCES

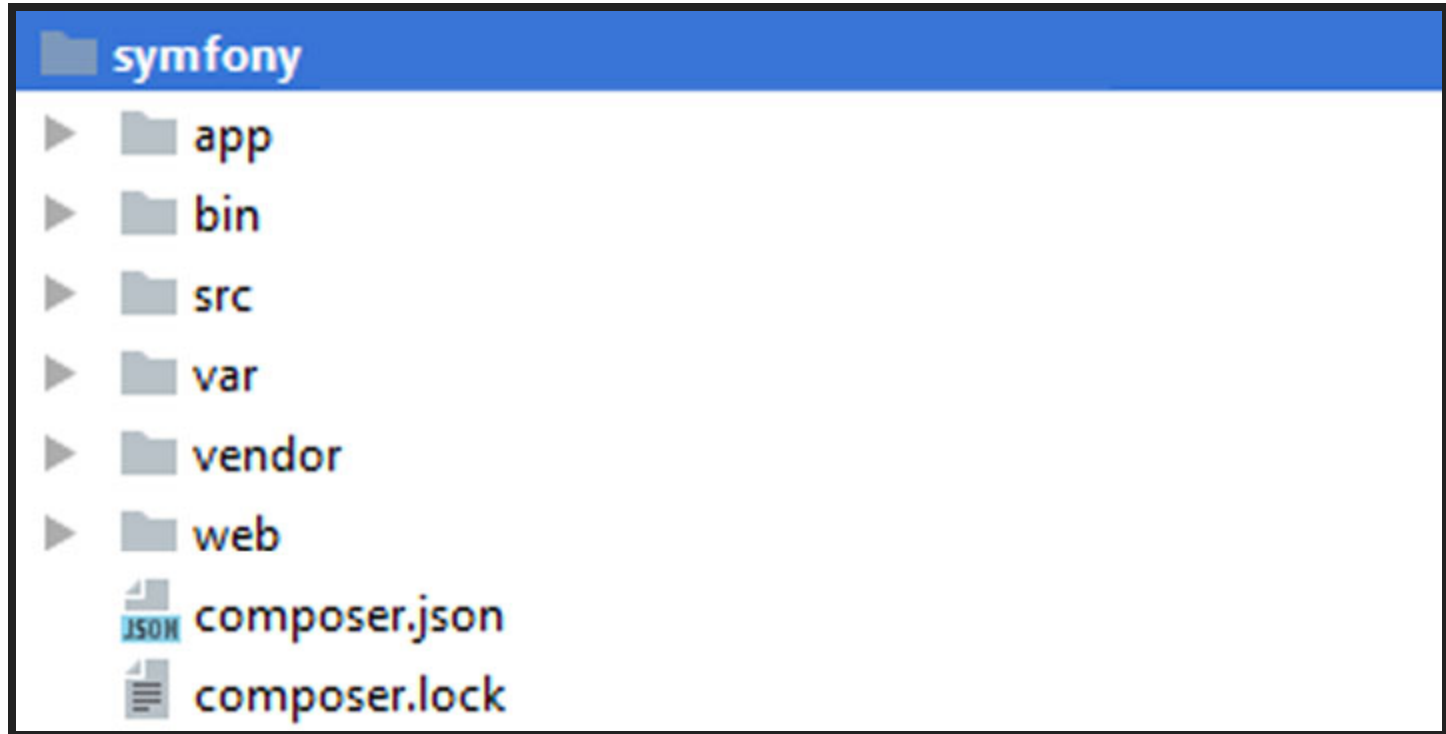
Linux :

```
composer update
```

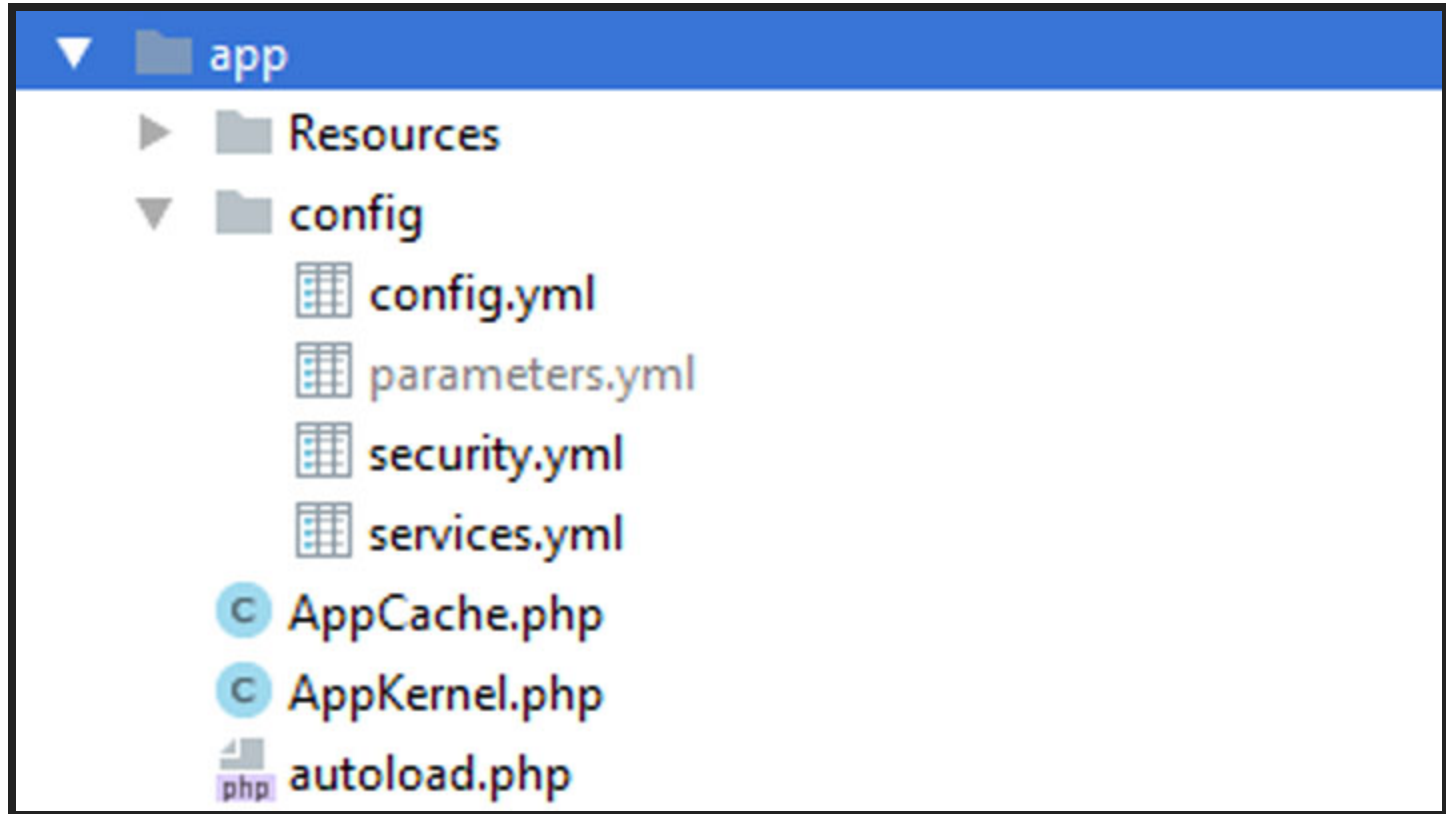
Windows

```
php composer.phar update
```

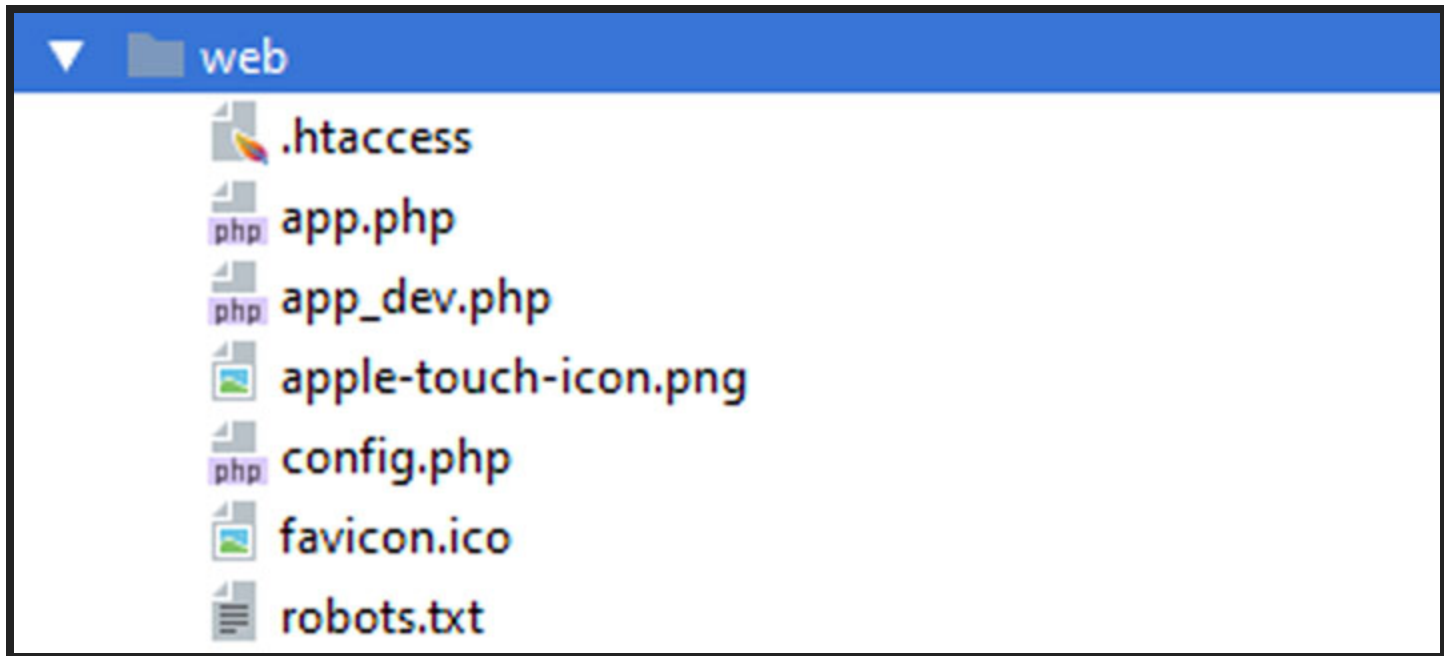
STRUCTURE D'UN PROJET



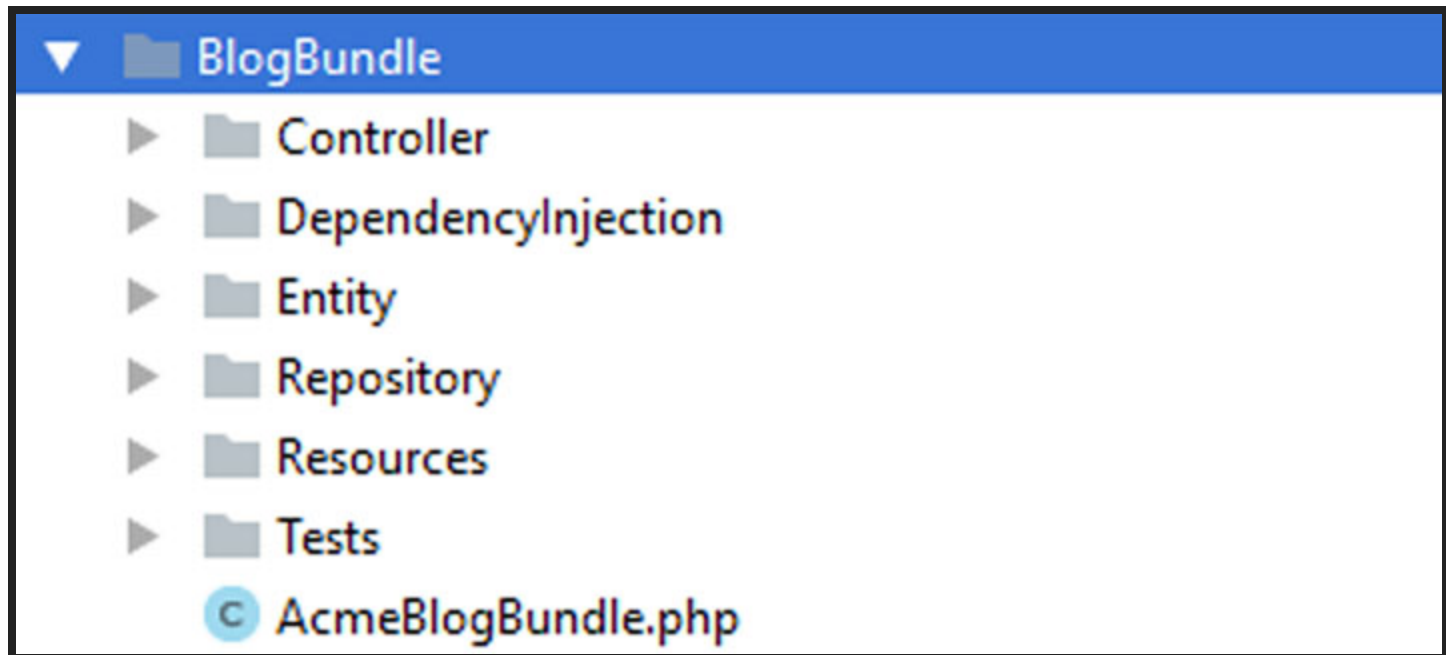
/app : Configuration



/web : Document root



Structure d'un bundle



HELLO WORLD!

CLI : Création du bundle HelloBundle

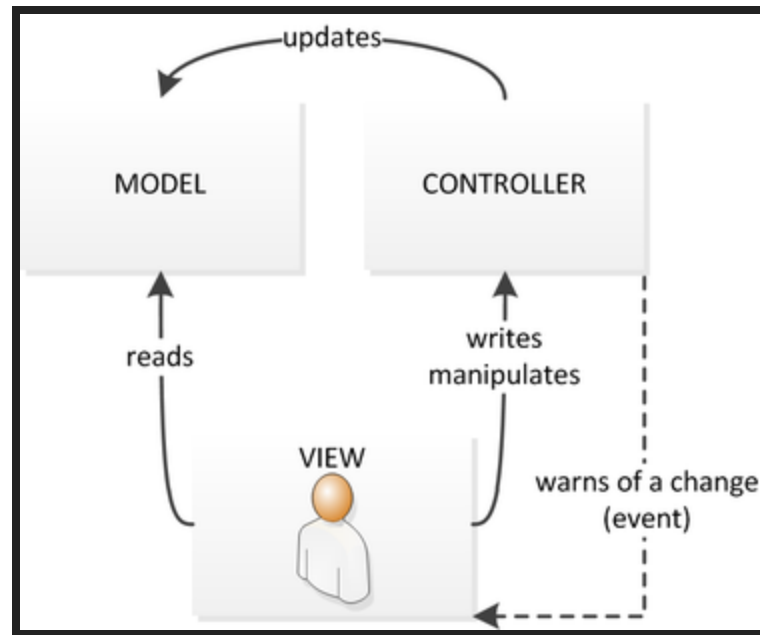
```
php bin/console generate:bundle
```

```
1. // src/HelloBundle/Controller/DefaultController
2. namespace HelloBundle\Controller;
3.
4. use Symfony\Bundle\FrameworkBundle\Controller\Controller;
5.
6. class DefaultController extends Controller
7. {
8.     public function indexAction()
9.     {
10.         return $this->render('HelloBundle:Default:index.html.twig');
11.     }
12. }
```

```
1. {% src/HelloBundle/Resources/views/Default/index.html.twig %}  
2. <!DOCTYPE html>  
3. <html>  
4.     <body>  
5.         <p>Hello world!</p>  
6.     </body>  
7. </html>
```

MVC

Un patron d'architecture logicielle :

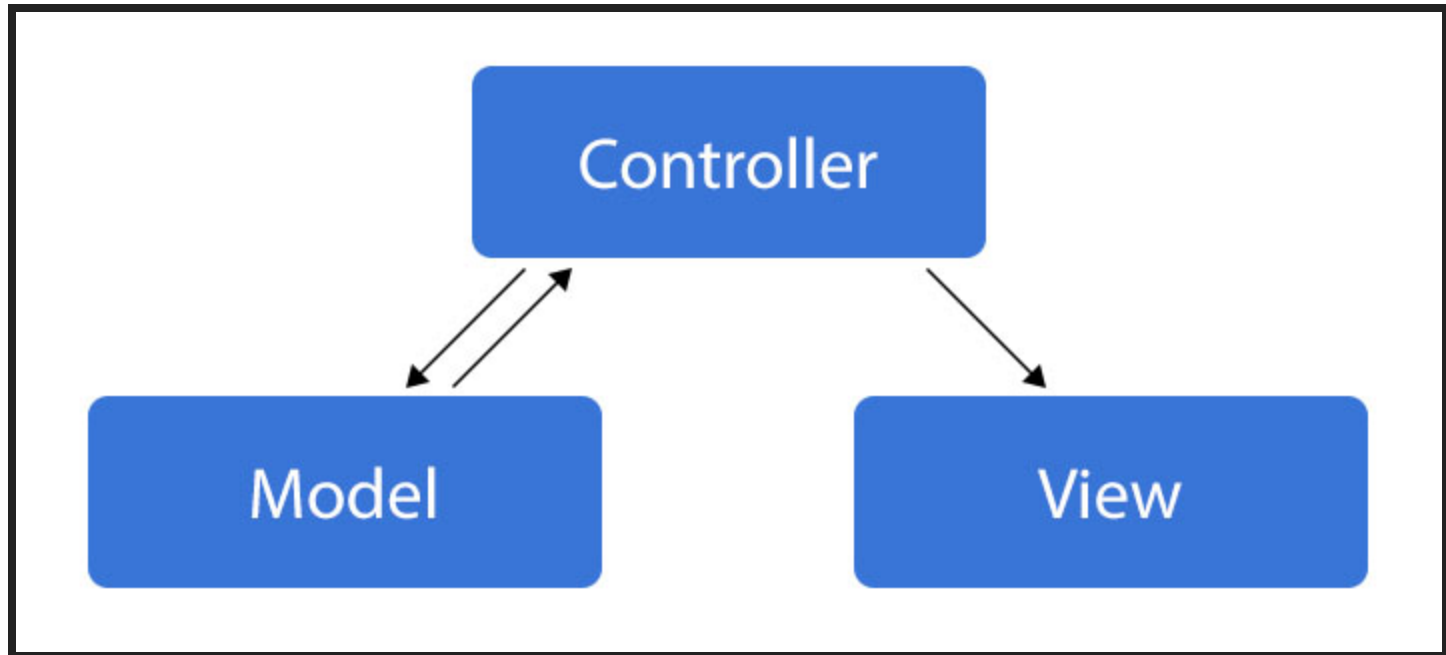


MODEL-VUE-CONTROLLER

Model	Modèle de données
Vue	Présentation, interface utilisateur
Controller	Logique de contrôle, gestion des événements, synchronisation

MVC

Appliqué à symfony :



MODEL-VUE-CONTROLLER

Model	Données et traitements
Vue	Templates
Controlleur	Manipule le modèle pour générer une vue

Du patron d'architecture **MVC**, **Symfony** n'applique que le principe «**Separation of concerns**», en séparant les données et traitements de la représentation graphique.

SYMFONY : UN FRAMEWORK HTTP

The web in action

The **User** asks for a **Resource** in a **Browser**

The **Browser** sends a **Request** to the **Server**

Symfony gives the **Developer** a **Request Object**

The **Developer** “converts” the **Request Object** to a **Response Object**

The **Server** sends back a **Response** to the **Browser**

The **Browser** displays the **Resource** to the **User**

HTTP KERNEL

```
1. namespace Symfony\Component\HttpKernel;
2.
3. use Symfony\Component\HttpFoundation\Request;
4.
5. interface HttpKernelInterface
6. {
7.     /**
8.      * @return Response
9.      */
10.     public function handle(Request $request);
11. }
```

De la Requête à la Réponse

