

PHP Symfony

Un framework pour
faire des sites web depuis 2005

Alain Bidault
bidault.enseignement@neuf.fr

Symfony

Cadre de travail français né en 2005, revu en 2011 et finalisé par symfony 3.

plugin pour Eclipse : <https://marketplace.eclipse.org/>
Symfony Plugin 1.3.0

en console : télécharger l' installateur Symfony et le mettre dans C:\xampp\php\www

cours openclassrooms par son créateur :

<https://openclassrooms.com/courses/developpez-votre-site-web-avec-le-framework-symfony/vous-avez-dit-symfony>

Symfony

Une approche MVC :

→ un contrôleur frontal, plutôt deux :

. app.php pour le visiteur

. app_dev.php pour le développeur.

Il contrôle la requête de l'utilisateur, la communique éventuellement au kernel (notre métier) qui lui retourne une réponse qui sera mise en forme par la vue.

Symfony

On range le code :

répertoires

/app pour les configs de votre site (sa cohérence entière)

/src pour notre code

/web pour le site vu par les visiteurs (*dont les 2 contrôleurs*)

/var fichiers temporaires, les logs de Symfony

/vendor la boîte à outils (libraries)

Symfony

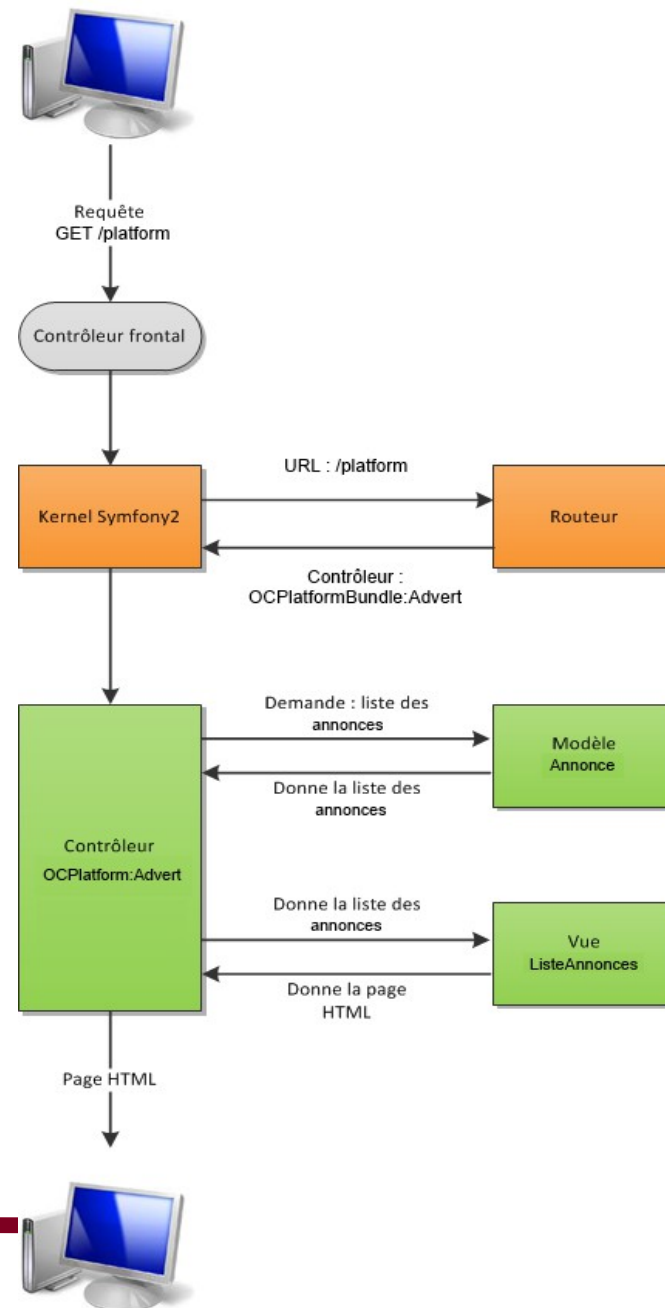
(source : *openclassrooms*)

Le visiteur demande la page /platform ;

Le contrôleur frontal reçoit la requête, charge le Kernel et la lui transmet ;

Le Kernel demande au Routeur
(composant Symfony qui fait la
correspondance entre URL et contrôleurs)
quel contrôleur exécuter pour l'URL
/platform.

Le Routeur indique au Kernel qu'il faut
exécuter le contrôleur
OCPlatform:Advert



Symfony

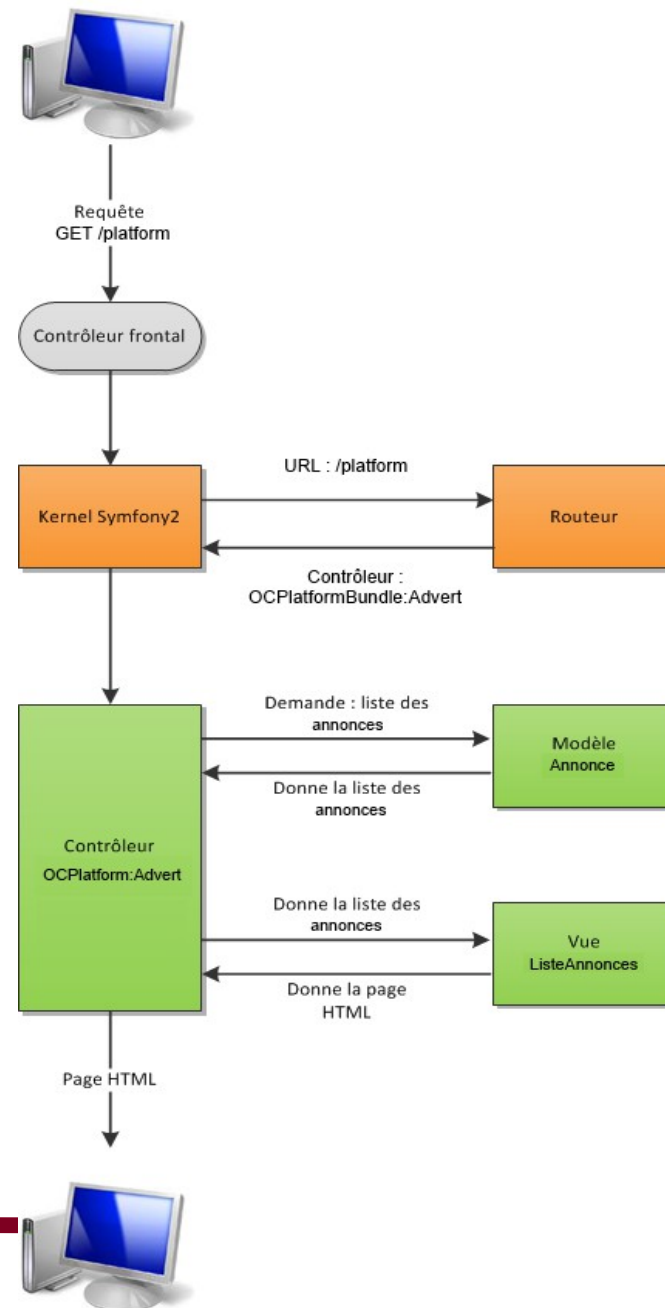
(source : *openclassrooms*)

Le Kernel exécute le contrôleur
OCPlatform:Advert.

Le contrôleur demande au modèle
Annonce la liste des annonces,

Il la donne à la vue ListeAnnonces
pour qu'elle construise et lui retourne
la page HTML,

Le contrôleur envoie enfin la page
HTML complète.



Premier Projet Symfony

Créer un projet Symfony sous Eclipse

Si vous ne pouvez pas le mettre directement sous xampp/htdocs, faites le ensuite avec un move.

Affichez dans un navigateur le controleur du développeur (et sa barre outils en bas :

http://localhost:8012/Symfony/web/app_dev.php/

et le contrôleur grand public :

<http://localhost:8012/Symfony/web/>

Symfony : les bundles

Un bundle est une brique de l'application, par exemple la gestion des utilisateurs, ou d'un blog, ou d'une page d'e-commerce.

Bibliothèque de bundles : <http://knpbundles.com>

Tout est bundle en Symfony, même notre propre code.
Par convention, l'espace de noms de chacun de vos bundles se termine par Bundle.

Le bundle propose son contrôleur, ses vues, ses modèles

Bundles : structure initiale

/Controller	Contient les contrôleurs
/DependencyInjection	Contient des informations sur le bundle (chargement automatique de la configuration par exemple)
/Entity	Contient les modèles
/Form	Contient les éventuels formulaires
/Resources	
-- /config	Contient les fichiers de configuration du bundle (les routes par exemple)
-- /public	Contient les fichiers publics du bundle : fichiers CSS et JavaScript, images, etc.
-- /views	Contient les vues du bundle, les templates Twig...

Premier Bundle Symfony

Dans le répertoire src, reproduisez la structure de AppBundle pour votre propre bundle, par exemple HelloBundle.

- création du répertoire
- création du fichier php au même nom

```
<?php namespace HelloBundle;  
use Symfony\Component\HttpFoundationBundle\Bundle;  
class HelloBundle extends Bundle  
{ }
```

- création du sous-répertoire controller
- création d'un fichier contrôleur

Premier Bundle Symfony

On déclare notre bundle dans AppKernel (dans /app).

```
new HelloBundle\HelloBundle()
```

Bundles mis par défaut pour Symfony.

Première série de bundles = pour la prod(uction), on y ajoute le notre.

Seconde série de bundles = pour le dév(eloppement)

Premier Bundle : les routes

Une route permet de relier une URL et son traitement.

On fait une route dans
src/HelloBundle/Resources/config/routing.yml

```
hello_coucou:  
    path:      /coucou  
    defaults: { _controller: HelloBundle:Hello:index }
```

hello_coucou → notre référence (identifiant) interne

/coucou → l'url pour cette page

controller → Bundle/contrôleur/méthode (sans suffixe)

Premier Bundle : les routes

On déclare ensuite notre route générale dans
app/config/routing.yml :

```
hello_coucou:  
  resource: "@HelloBundle/Resources/config/routing.yml"  
  prefix:   /
```

Premier Bundle : le contrôleur

Au bout de la route, il faut le contrôleur (sous Controller).

```
<?php
namespace HelloBundle\Controller;

use Symfony\Component\HttpFoundation\Response;

class HelloController
{
    public function indexAction()
    {
        return new Response("Hello Coucou !");
    }
}
```

Premier Bundle : le contrôleur

dans le namespace des contrôleurs du bundle.

le contrôleur utilise l'objet Response (use).

nom du contrôleur = nom du fichier (pour autoload).

la méthode `indexAction()`, avec le suffixe Action !

Réponse toute simple. L'argument de l'objet Response est le contenu de la page envoyée.

→ aller voir sur http://localhost/Symfony/web/app_dev.php/coucou

Template, pour MVC

Code générique qui propose un modèle à suivre sans se préoccuper de typage.

En programmation web, un Template propose une vue prédéfinie d'un site. Il permet de séparer le contenu de la forme.

Il assure une parfaite cohérence visuelle entre les pages.

Nous pouvons utiliser le template TWIG, écriture intuitive, pour respecter le MVC.

Page html du bundle

src/HelloBundle/**Resources/views**/Hello/index.html.twig

→ recommandation : on utilise le même nom.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Notre premier Template !</title>
  </head>
  <body>
    <p>
      Hello Coucou !
    </p>
  </body>
</html>
```

Lien dans le contrôleur

Appeler la page depuis le contrôleur.

```
namespace HelloBundle\Controller;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Symfony\Component\HttpFoundation\Response;

class HelloController extends Controller
{
    public function indexAction()
    {
        $content =
$this->get('templating')->render('HelloBundle:Hello:index.html.twig');

        return new Response($content);
        //return new Response("Hello Coucou !");
    }
}
```

Page html.twig du bundle

src/HelloBundle/**Resources/views**/Hello/index.html.twig

→ recommandation : on utilise le même nom.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Notre premier Template !</title>
  </head>
  <body>
    <p>
      Hello Coucou mais dans du html {{genre}} !
    </p>
  </body>
</html>
```

Lien dans le contrôleur

On ajoute un argument.

```
$content = $this  
    -> get('templating')  
    -> render('HelloBundle:Hello:index.html.twig',  
        array('genre' => 'coool'));
```

Symfony

C'est tout pour le moment...