1) INTRODUCTION

Symfony 2, en 2011, est le premier framework un peu moderne de PHP. Il a une approche façon boîte à outils, un peu moins monolithique que ses concurrents. Il est composé de différents composants (comme Twig, le router, les formulaires etc) qui liées ensemble, permettent de créer une applications web de manière plus rapide et de manière « standardisée » en forçant les bonnes pratiques aux maximum.

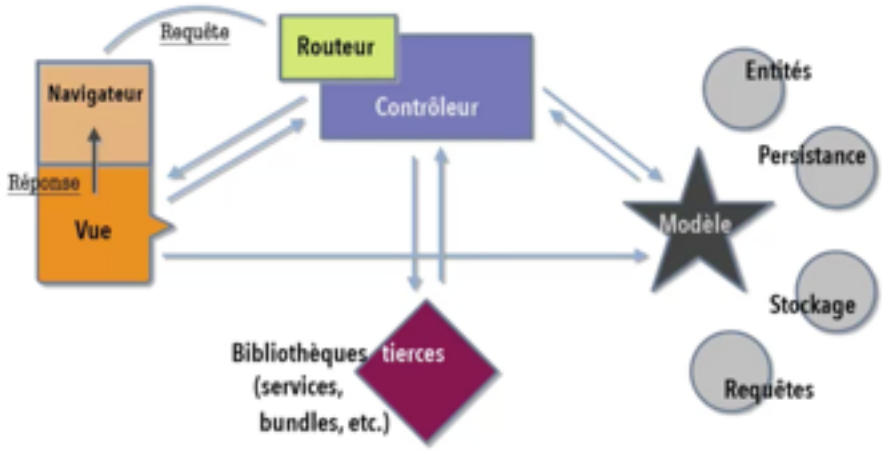
La version 4 pousse encore plus cette approche de boite à outils avec une installation de départ d'un squelette Symfony, presque vide de tout composant, que l'on peut ensuite ajouter au fur et à mesure si besoin.

Aujourd'hui de nombreuses applications utilisent des parties de cette boite à outils : Drupal, Laravel, Prestashop, Magento, PHPBB.

1) Piqure de rappel (en bref) :

- PHP 5.3 a introduit les espaces de nom, alias Namespace, pour éviter les collusions de noms. On peut aussi définir des alias / raccourcis de classe pour améliorer la lisibilité. On le définie dans la première ligne PHP et on respecte l'arborescence des dossiers. L'autoload permet de venir charger automatiquement ces classes.
- PSR : C'est une initiative pour normaliser le code PHP avec des recommandations, des conventions, mis en place par le FIG (Framework Interoperability Group).
  La plus importante est le PSR-4, elle précise comment se fait l'autoloading des classes. Mais il y a de nombreuses recommandations.
- MVC : architecture de code permettant de séparer et organiser les fichiers d'une applications en fonction de leur rôle.
  - Contrôleur : reçoit les requêtes. Chargé de gérer les interactions entre l'utilisateur et le modèle. C'est le chef d'orchestre de l'application.
  - Modele : gère la manipulation et le traitement des données.
  - Vue : présente les données du modèle à l'interface utilisateur. Enregistre les actions utilisateurs (envoi d'un formulaire etc, et les transmet au contrôleur.

2) **Cycle de vie d'une requête HTPP dans une application Symfony :**

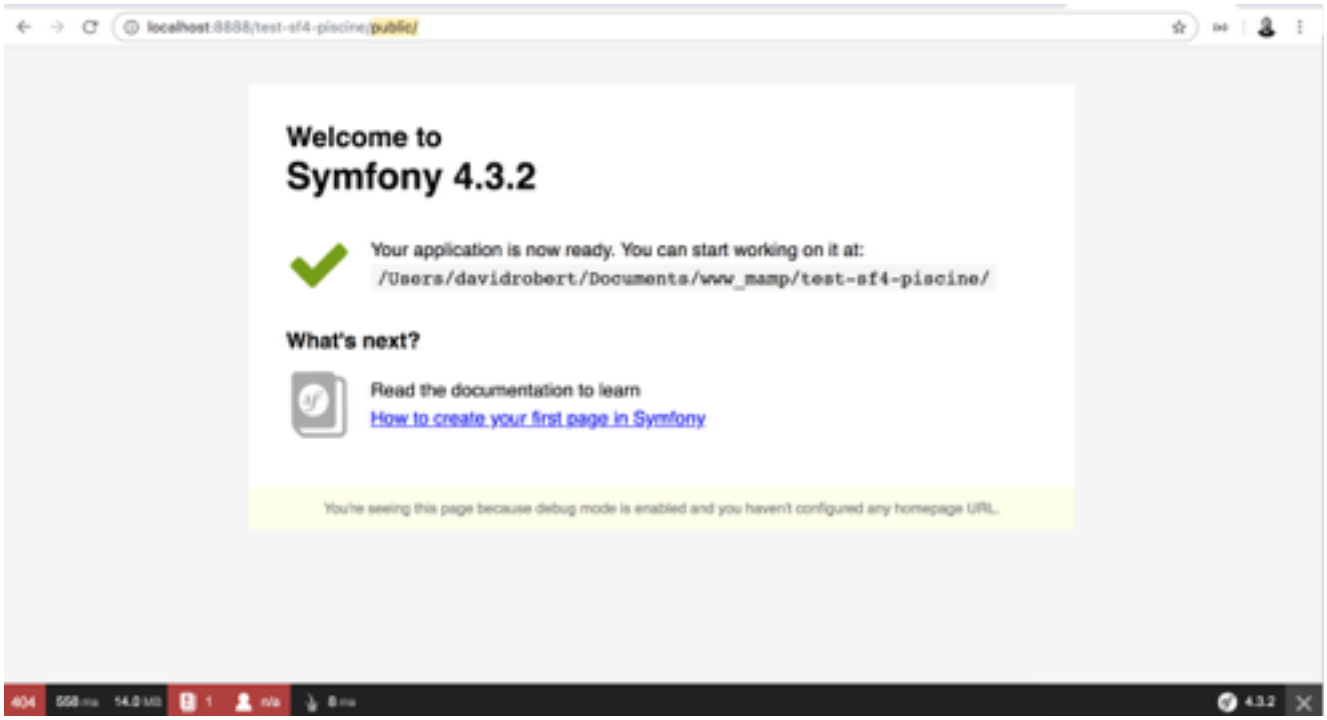


3) Installation de Symfony

En ligne de commande :
composer create-project symfony/website-skeleton nomduprojet

Si Apache est utilisé (notamment avec MAMP), il faut configurer le .htacces Pour ça on va utiliser :
composer require symfony/apache-pack

Une fois installé, un message de bienvenu devrait s'afficher en allant sur votre projet via votre serveur local, suivi de « public » :



3) Architecture

- Le dossier « public » est le seul dossier accessible par le navigateur. Il contient un fichier index.php qui est le point d'entrée de l'application.

- Le dossier config, contient les fichiers de configuration de chacun des packages de l'application.
- Le dossier src contient votre code
- Le dossier templates contient les fichiers Twig (la partie « vue » de votre application)
- Le dossier var contient les logs et les caches
- Le dossier tests contient les tests écrits pour votre application
- Le dossier bin contient le fichier d'entrée pour la ligne de commande
- Le dossier vendor contient les bibliothèques
- Le dossier translations contient les traductions de vos pages (si votre applications est en plusieurs langues)
- Le fichier .env contient la configuration pour l'accès à la base de données et d'autres informations sens

Environnements disponibles dans Symfony :

- prod : le site visible par les utilisateurs. N'affiche pas les erreurs et utilise des fichiers en cache pour optimiser les performances.
- dev : pour développer. Permet notamment d'afficher les erreurs et la barre de débug.
- test : utilisé pour les jeux de tests.

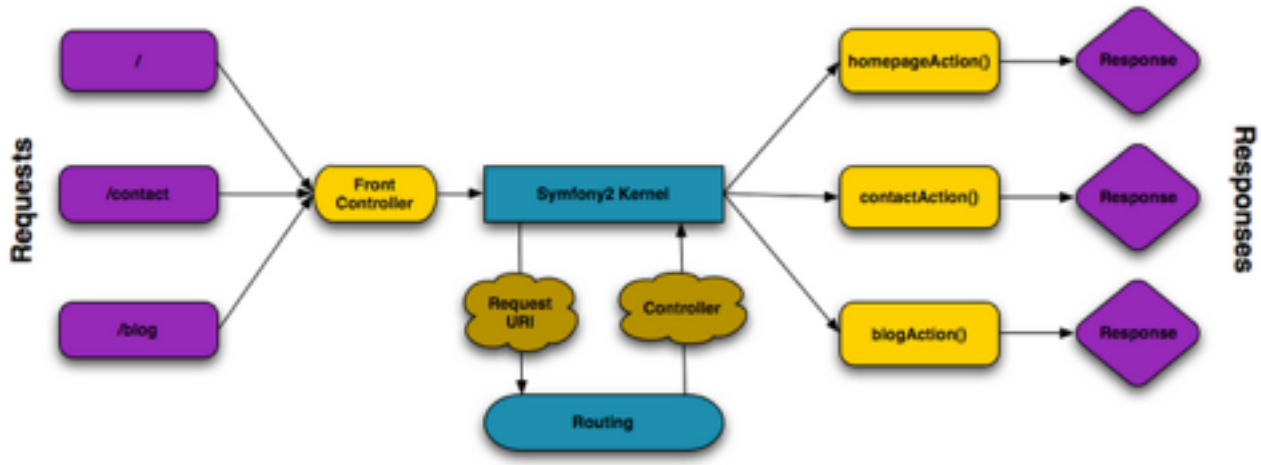
Chaque environnement peut avoir une configuration différente (via des fichiers de configurations spécifiques dans le dossier config).

2) LES ROUTES

Les routes dans Symfony permettent de définir une URL, d'identifier une ressource.

Une URL se compose de plusieurs éléments :

- protocole : http
- nom de domaine : www.example.com
- un port : le serveur écoute sur un port
- un chemin : spécifie le contenu auquel on accède via une arborescence
- un nom de ressource : index.html, image.jpg etc
- des options : ?nom=david&prenom=robert



Le composant routing dans symfony :
- gère les collections de routes
- détermine les contextes de requêtes (analyse l'url pour savoir le chemin demandé etc)
- relie une url avec une route

Les routes sont un peu l'API de notre appli. A chaque route fait référence à une méthode de controller.

Une route contient deux propriétés principales :
- id : un identifiant unique qui permet à l'appli d'accéder à la route.
- path : le schéma de l'url qui permettra d'analyser la requête du navigateur.

Les routes s'écrivent en « annotations » au dessus des méthodes de contrôleurs (il est possible de les écrire aussi dans un fichier Yaml, XML ou même PHP). Une annotation est un commentaire PHP qui permet d'exécuter du code.

Pour installer le composant Annotations :
composer require annotations