

Building an app with



Symfony 3

Part1 Introduction Install First page

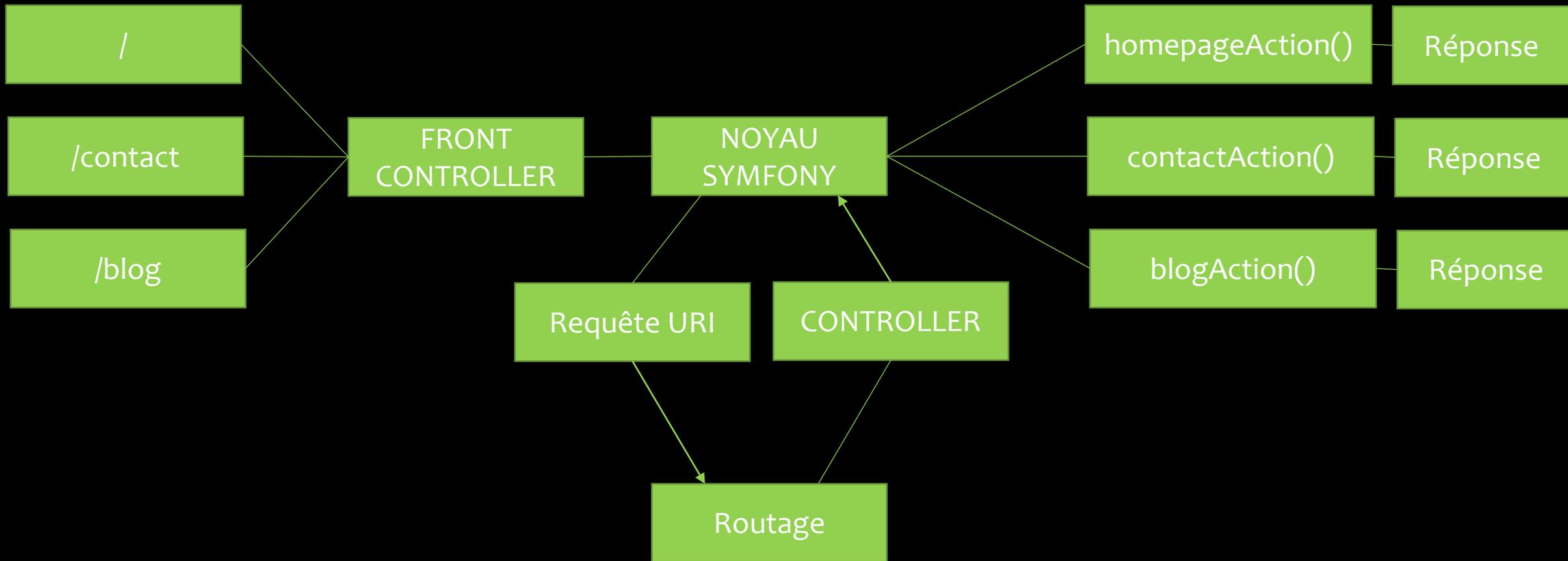
URI Uniform Resource Identifier

- HTTP->Request->URI adresse ou location unique qui identifie la ressource désirée par le client `$_SERVER['REQUEST_URI']`

FRONT CONTROLLER

- Habituellement on a `index.php` `contact.php` et `blog.php`
- Avec un contrôleur `/index.php` `/index.php/contact` `/index.php/blog`
- Pour utiliser un contrôleur il faut active `mod_rewrite` sur apache

MODELE SYMFONY



Chaque requête http exécute un fichier Front Controller.

Le système de routage détermine quelle fonction PHP doit être exécuté en fonction de la requête et de la configuration créée du routage.

La bonne fonction PHP est exécutée, où votre code est exécuté et retourne un objet Response approprié.

Une requête symfony en action

Exemple pour ajouter une adresse main/contact

Dans app/config/routing.yml

Contact:

```
path: /contact
defaults: { _controller: AppBundle:Main:contact }
```

Pour pointer sur la méthode contactAction dans la classe MainController

Dans /src/AppBundle/Controller/MainController.php

```
namespace AppBundle\Controller;
use Symfony\Component\HttpFoundation\Response;
class MainController{
    public function contactAction(){
        return new Response('<h1>Contactez nous:</h1>');
    }
}
```

Symfony Components

Plus de 20 librairies indépendantes utilisables dans n'importe quel projet PHP

- **HttpFoundation:** Classes Request et Response pour gérer des sessions et l'upload de fichiers.
- **Routing:** Système de routage rapide et puissant.
- **Form:** complet et flexible framework pour créer des formulaires et gérer leur soumission.
- **Validator:** Système pour créer des règles sur les données pour valider ou pas celles soumises par l'utilisateur.
- **Templating:** Un toolkit pour le rendu des templates, gère leur héritage (un template est décoré par un layout) et exécute les autres tâches communes des templates.
- **Security:** Une librairie puissante pour gérer toutes les sécurités nécessaires dans une application
- **Translation:** Un framework pour traduire les chaînes de caractères dans votre application.

Installation & configuration de Symfony

Installation

La meilleure manière d'installer Symfony est d'utiliser Symfony Installer qui nécessite php 5.4 ou +.

Sous Mac OS X et Linux:

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

Ca créera un commande globale symfony sur le système.

Sous Windows:

```
php -r "readfile('https://symfony.com/installer');" > symfony
```

Déplacer le fichier symfony téléchargé dans votre répertoire de projet et exécuter ce qui suit:

```
move symfony c:\wamp\www\synfoProjects  
c:\wamp\www\synfoProjects\>php symfony
```

Créer une application Symfony:

Nouvelle commande

Sous Mac OS X et Linux:

```
symfony new my_project_name
```

Sous Windows:

```
c:\wamp\www\synfoProject\>php symfony new my_project_name
```

Cette commande crée un nouveau répertoire appelé my_project_name contenant un nouveau projet basé sur la dernière version stable. De plus, l'installateur vérifie que votre système est compatible.

Pour installer une version spécifique, vous devez préciser un deuxième argument optionnel à la commande new:

```
symfony new my_project_name 2.8
```

```
symfony new my_project_name 3.0
```

```
symfony new my_project_name 2.7.3
```

```
symfony new my_project_name 2.8.1
```

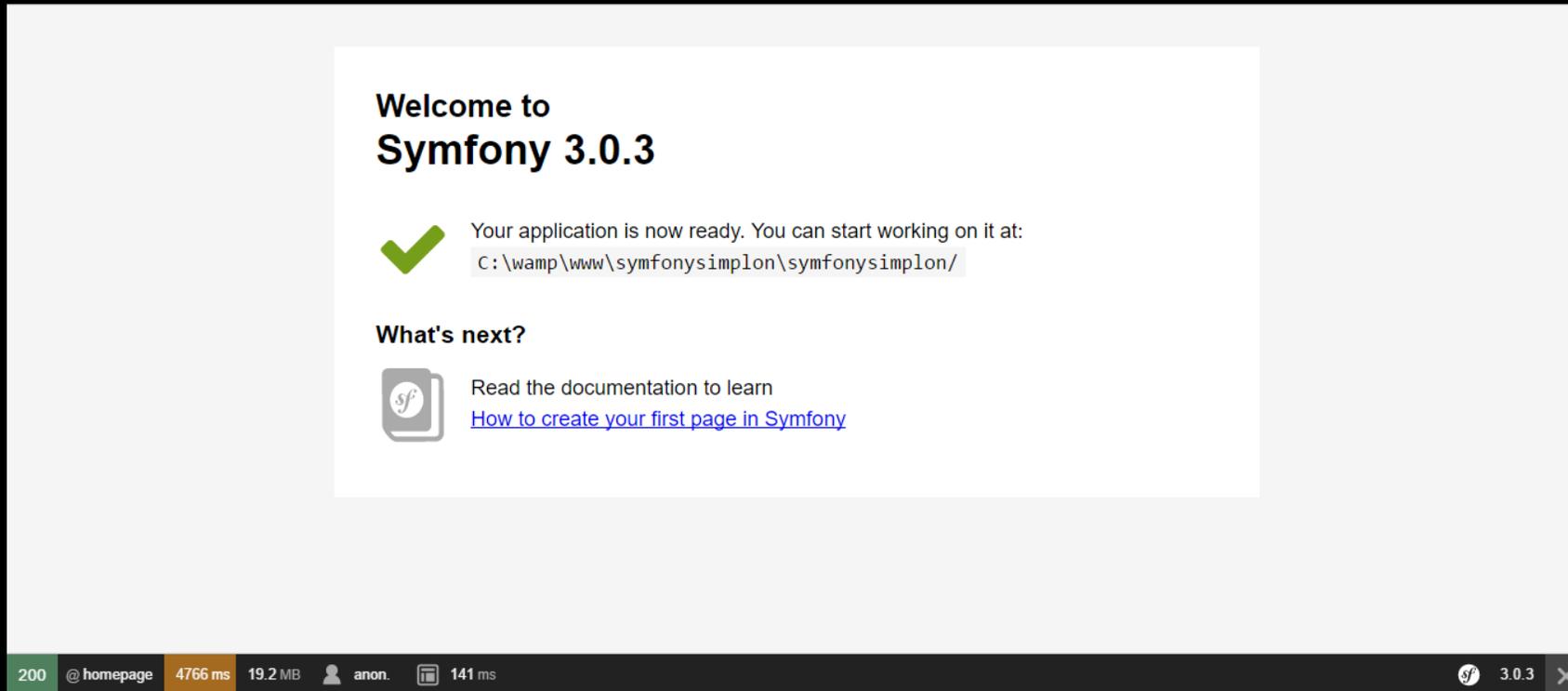
On peut installer sans l'installer avec composer (faites des recherches sur le net si nécessaire)

Exécuter l'application Symfony

Symfony influence le serveur web interne PHP pour exécuter des applications pendant leurs développements.

```
cd my_project_name  
php bin/console server:run
```

Dans votre navigateur, accédez à l'adresse localhost:8000 pour obtenir:



A la fin du développement, pour arrêter le serveur php bin/console server:stop

Première Page Symfony

Pour créer une nouvelle page ou du JSON, il faut 2 étapes:

- Créer une route pour gérer l'URL et la pointer vers le contrôleur
- Créer un contrôleur qui construira votre page par une fonction. Vous recevez une requête d'information et utilisé le contrôleur pour créer un objet Response Symfony qui peut envoyer du HTML, JSON ou autres.

Comme pour tout le web, chaque interaction est initié par une requête HTTP. Votre boulot est de comprendre cette requête et renvoyer une réponse.

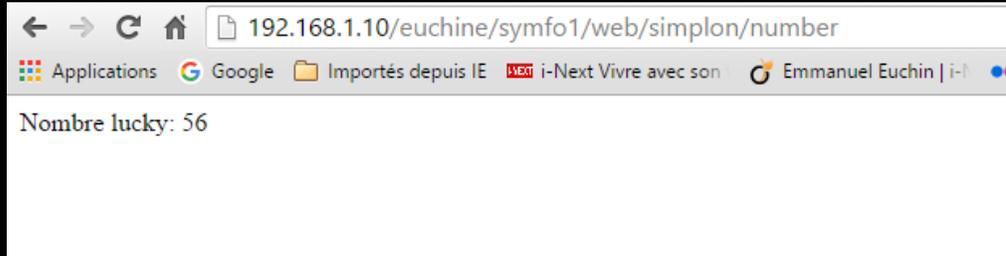
Création d'une page: Route et Controller

Nous désirons créer une page qui affiche un nombre aléatoire accessible à l'adresse /simplon/number
Donc nous allons créer le contrôleur `src/AppBundle/Controller/SimplonController.php`
namespace AppBundle\Controller;

```
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;  
use Symfony\Component\HttpFoundation\Response;
```

```
class SimplonController {  
  
    /**  
     * @Route("/manu/number")  
     * @return Response  
     */  
    public function numberAction() {  
        $number = rand(0, 100);  
        return new Response('<html><body>Nombre lucky: ' . $number . '</body></html>');  
    }  
}
```

Résultat



Fin du cours 1
La suite au prochain épisode