

Le Routage | Symfony 4

Doc de référence : <https://symfony.com/doc/current/routing.html>

Suite à installation nous n'avons pour le moment aucune route de défini dans notre application. A nous de faire le nécessaire.

En réalité, symfony nous offres plusieurs possibilité selon nos préférences pour gérer les routes (URL) de notre application :

Annotations, YAML, PHP, XML

Le YAML

YAML, est un langage simple qui décrit les données. Il permet de représenter des données structurées, comme le ferait XML par exemple, mais de manière plus naturelle et moins verbeuse. C'est à dire de manière aisément modifiable et compréhensibles par un humain, quel que soit le langage de programmation".

La plupart des fichiers de configuration dans symfony sont au format YAML.

Rendez-vous sur :

config / routes.yaml

Nous allons retirer les commentaires et créer notre premier contrôleur.

```
index:
  path: /
  defaults: { _controller: 'App\Controller\DefaultController::index' }
```

Explication du fonctionnement des routes en YAML

Création de IndexController

Doc de référence : <https://symfony.com/doc/current/controller.html>

1. Créer le dossier TechNews dans src/Controller.
2. Créer une nouvelle classe 'IndexController'

Nous allons écrire ensemble notre première action :

```
public function index() {
    return new Response("<html><body><h1>PAGE D' ACCUEIL</h1></body></html>");
}
```

Actualisons maintenant notre page pour voir le résultat.

Testons maintenant dans notre console les commandes suivantes :

```
php bin/console debug:router --env=prod
php bin/console debug:router
```

- Que pouvez-vous constater ?
- D'où proviennent les routes complémentaires ? A quoi servent-elles ?

```
C:\xampp\htdocs\SYMFONY\02-PROJECT\TechNews (master -> origin)
λ php bin/console debug:router
```

Name	Method	Scheme	Host	Path
index_category	GET	ANY	ANY	/category/{label}
_twig_error_test	ANY	ANY	ANY	/_error/{code}.{_format}
_wdt	ANY	ANY	ANY	/_wdt/{token}
_profiler_home	ANY	ANY	ANY	/_profiler/
_profiler_search	ANY	ANY	ANY	/_profiler/search
_profiler_search_bar	ANY	ANY	ANY	/_profiler/search_bar
_profiler_phpinfo	ANY	ANY	ANY	/_profiler/phpinfo
_profiler_search_results	ANY	ANY	ANY	/_profiler/{token}/search/results
_profiler_open_file	ANY	ANY	ANY	/_profiler/open
_profiler	ANY	ANY	ANY	/_profiler/{token}
_profiler_router	ANY	ANY	ANY	/_profiler/{token}/router
_profiler_exception	ANY	ANY	ANY	/_profiler/{token}/exception
_profiler_exception_css	ANY	ANY	ANY	/_profiler/{token}/exception.css
index	ANY	ANY	ANY	/
accueil	ANY	ANY	ANY	/home
mentions_legales	ANY	ANY	ANY	/privacy

Les Annotations

Dans symfony, pour plus de facilité, nous pouvons écrire nos routes avec des annotations.

Doc de référence : <https://symfony.com/doc/current/routing.html#creating-routes>

Dans la console :

```
composer require annotations
```

Nous pouvons maintenant écrire notre première route grâce aux annotations.

```
Mise en place des routes
```

◀ ▶

```
php bin/console router:match /categorie/business
```

Testons les deux contrôleurs suivants :

Doc de référence : https://symfony.com/doc/current/routing/redirect_in_config.html#redirecting-using-a-route

```

accueil:
  path: /accueil
  controller: Symfony\Bundle\FrameworkBundle\Controller\RedirectController::redirectAction
  defaults:
    route: index
    permanent: true

```

Grâce à **redirectAction** du **RedirectController**, je peux créer une redirection sans avoir dû créer un contrôleur personnalisé...

TemplateController

Doc de référence : https://symfony.com/doc/current/templating/render_without_controller.html

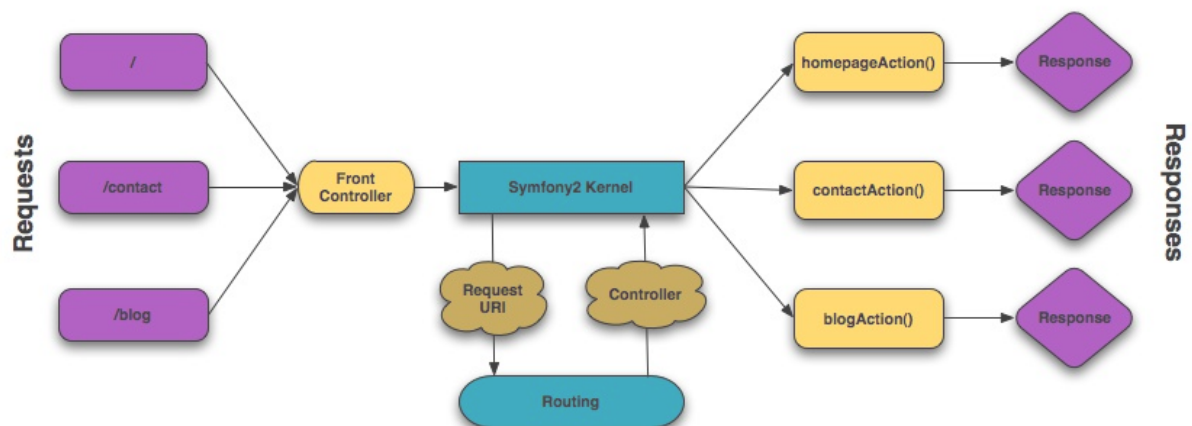
Il se peut que dans certains cas, nous ayons besoin d'afficher une simple page, sans pour autant passer de paramètre. Nous pouvons alors grâce au TemplateController éviter la création d'un contrôleur dédié.

```

mentions_legales:
  path: /mentions-legales
  controller: Symfony\Bundle\FrameworkBundle\Controller\TemplateController::templateAction
  defaults:
    template: static/mentionslegales.html.twig

```

Le parcours d'une requête



Written with ❤️ by [Hugo LIEGEARD](#).

Screenshots by [Frogg's web tools](#).

