Cheatography

Bases de Symfony Cheat Sheet

by LiorChamla via cheatography.com/128746/cs/25281/

Routing : Documentation officielle

Retrouvez la documentation officielle sur le Routing dans Symfony ici : https://symfony.com/doc/current/routing.html

Route sous forme d'annotation @Route

```
<?php
use Symfony\Component\Routing\Annotation\Route;
class MonController {
  /**
   * @Route(
   * "/URL-VOULUE/{param1}/{param2}",
   * name="nomDeLaRoute",
   * methods={"GET", "POST"},
   * requirements={"param1":"regexpVoulue", "param2":"regexpVoulue"},
   * defaults={"param1":"ValeurParDefaut"}
   * host="my.host"
   * schemes={"http", "https"}
   * priority=1
   * )
   */
  public function myMethod($param1, $param2) {
     // ...
  }
}
```

Les options possibles dans l'annotation @Route

name="<routeName>"	Donner un nom à une Route
methods={"<method1>", "<method2>"}	Restreindre l'accès à une Route sur une ou plusieurs méthodes HTTP
host="<host>"	Restreindre l'accès à une Route sur un host particulier
requirements={"param1":"regexp1", "param2":"regexp2"}	Préciser les règles qui s'appliquent aux paramètres de la Route
defaults={"param1":"default1", "param2":"default2"}	Préciser les valeurs par défaut des paramètres de la Route

Notez que les mêmes options sont disponibles au format YAML

Les classes importantes de HttpFoundation

Symfony\Component\HttpFoundation\Request	Classe qui représente une requête HTTP reçue sur l'application
Symfony\Component\HttpFoundation\Response	Classe qui représente une réponse HTTP qu'on souhaite renvoyé au client

Les classes importantes de HttpFoundation (cont)

Symfony\Component\HttpFoundation\RedirectResponse	Classe qui représente une réponse HTTP de type redirection
Symfony\Component\HttpFoundation\JsonResponse	Classe qui représente une réponse HTTP de type JSON

La classe Request

$request->request	Equivalent de la super globale $_POST
$request->query	Equivalent de la super globale $_GET
$request->attributes	Représente des données que le Framework a inclus dans la requête : paramètres de l'URL et autres choses utiles
$request->get('nomParam', 'default')	Permet d'obtenir une information dans le $_GET ou le $_POST

Vous pouvez vous "faire livrer" la requête HTTP par le framework en la demandant en paramètre d'une méthode liée à une Route

By **LiorChamla**
cheatography.com/liorchamla/

Published 19th November, 2020.
Last updated 19th November, 2020.

La classe Response

$response = new Response(<contenu>, <statusCode>, <headers[]>);	Permet de créer une réponse HTTP classique
$response = new RedirectResponse(<url>, <headers[]>);	Permet de créer une réponse HTTP qui va provoquer une redirection
$response = new JsonResponse(<data>, <statusCode>, <headers[]>);	Permet de créer une réponse JSON
$response->setContent(<contenu>);	Permet de modifier le contenu d'une réponse existante
$response->headers	Représente les en-têtes de la réponse
$response->setStatusCode(<code>)	Permet de modifier le statut HTTP de la réponse
$response->send()	Permet d'envoyer la réponse au navigateur (en principe ce n'est pas à vous de le faire mais au Framework)

Controller et AbstractController

Qu'est-ce qu'un *controller* au sens strict ?

> Un controller est une FONCTION qui va prendre en charge une Requête HTTP et renvoyer une Réponse HTTP adéquate

Qu'est-ce qu'une *classe Controller* ?

> C'est une classe dont les méthodes sont des *controllers* dans le sens qu'elles sont destinée à répondre à une requête HTTP précise par une réponse adéquate

Pourquoi hériter de la classe *AbstractController* ?

> La classe **AbstractController** vous est fournie par le Framework et va vous offrir énormément de raccourcis intéressants pour des comportements communs qu'on utilise souvent dans nos controllers

Quels sont les *pouvoirs magiques* qu'on peut utiliser dans une fonction controller ?

> Chaque méthode liée à une Route peut demander au Framework de lui fournir différents paramètres comme la *Request* mais aussi les paramètres compris dans l'URL (entre autres)

Commandes utiles de la CLI à ce stade

php bin/console make:controller <NomDuController>	Permet de créer un Controller
php bin/console debug:router	Permet de voir la liste des routes existantes dans l'application