Routage sous Symfony

[sommaire]

1. ## Objectifs

- Comprendre ce qu'est le routage dans le contexte d'une application web.
- Apprendre comment les routes permettent de faire correspondre les URL aux contrôleurs de votre application Symfony.

- ## Qu'est-ce que le routage sous Symfony

    - **Symfony** fournit un système de routage flexible. Déclarer une route consiste à associer une URI (Uniform Resource Identifier, c'est-à-dire la partie de l'adresse qui suit le nom de domaine) à un code à exécuter.
    - Le routage permet de créer des URL de requête pour votre application. Ce qui rend le routage dans **Symfony** puissant, c'est que vous avez la liberté de définir vos routes selon vos besoins.
    - Le routage est l'un des composants essentiels du framework Symfony. Il s'agit simplement d'un mécanisme qui mappe vos requêtes vers une action spécifique du contrôleur.
    - Toutes les routes de Symfony sont définies dans des fichiers spécifiques, généralement sous la forme de **`config/routes.yaml`** ou **`config/routes/annotations.yaml`**.
    - Dans Symfony, les routes sont généralement définies dans des fichiers de configuration distincts. Pour une application web traditionnelle, vous pouvez définir vos itinéraires dans **routes.yaml**, tandis que pour une API, vous pouvez utiliser **routes/api.yaml**.

- ## Présentation

- Lorsque Symfony est installé et configuré, vous disposerez de tous les fichiers nécessaires dans votre dossier pour démarrer un projet Symfony.
- En naviguant sur une page Web Symfony dans le navigateur, l'URL ne se terminera pas par le nom d'un fichier .html ou .php. Au lieu de cela, elle suivra une structure de chemins ressemblant à des répertoires et sous-répertoires, par exemple **http://127.0.0.1:8000/inscription**.
- Toutes les routes de l'application sont enregistrées dans un fichier de configuration spécifique, tel que `routes.yaml` ou `annotations.yaml`. Ce fichier indique à Symfony les URI auxquelles il doit

apcpedagogie.com

répondre et les actions associées à ces URI.

- Par défaut, Symfony est livré avec un dossier de routes dans son répertoire racine pour gérer divers besoins en matière d'itinéraires. Ce dossier peut contenir différents fichiers, mais les plus courants sont **routes.yaml** et **annotations.yaml**.
- Pour créer une route, vous pouvez utiliser la configuration dans le fichier de routes ou des annotations dans vos contrôleurs. La configuration YAML ou les annotations indiquent le chemin de l'URI et l'action à exécuter pour cette route.
- Dans Symfony, les routes peuvent être définies de manière déclarative dans un fichier de configuration ou de manière programmatique dans vos contrôleurs.
- Dans Symfony, les routes peuvent également être définies via des annotations dans les contrôleurs, offrant une façon pratique de spécifier les routes directement au niveau des actions des contrôleurs.
- Le routage dans Symfony peut être géré à travers diverses méthodes, y compris les annotations dans les contrôleurs et les fichiers de configuration, offrant une grande flexibilité dans la gestion des itinéraires.

• Routage de base

1. **Comprendre le concept de routage :**

- Le concept de routage dans le contexte d'une application web, notamment avec Symfony, est essentiel pour comprendre comment les requêtes des utilisateurs sont gérées et dirigées vers les différentes parties de votre application.

1. **Définition du Routage :**

- Le routage est le processus de correspondance des URL (Uniform Resource Locator) demandées par les utilisateurs avec les actions spécifiques que votre application doit effectuer pour répondre à ces demandes. En d'autres termes, le routage détermine quel code (généralement des contrôleurs dans le contexte de Symfony) doit être exécuté en réponse à une URL particulière.

- **Fonctionnement du Routage :**

- Dans Symfony, le routage est géré par un composant dédié qui examine l'URL demandée et la compare à une liste de routes définies dans votre application. Chaque route est associée à une

apcpedagogie.com

action spécifique, généralement un contrôleur et une méthode de ce contrôleur. Lorsqu'une URL correspond à une route définie, Symfony exécute l'action associée à cette route.

- **Définition des Routes :**
  - Dans Symfony, vous pouvez définir vos routes de différentes manières. Vous pouvez les configurer dans un fichier de configuration (par exemple, routes.yaml), utiliser des annotations directement dans vos contrôleurs, ou même les définir dynamiquement dans votre code. Chaque route est définie par un URI (Uniform Resource Identifier) et une action à exécuter.

- **Illustrer le concept de routage dans Symfony evec exemple**
  - Supposons que vous développiez une application de blog avec Symfony. Vous voudriez probablement avoir des routes pour afficher des articles, ajouter de nouveaux articles, et gérer les commentaires. Voici comment vous pourriez configurer les routes pour ces fonctionnalités :

1. **Définition des Routes :**

- Dans votre fichier de configuration YAML ou via des annotations dans vos contrôleurs, vous définissez les routes pour chaque fonctionnalité de votre application :

```yaml
# config/routes.yaml

article_show:
    path: '/article/{id}'
    controller: App\Controller\ArticleController::show

article_new:
    path: '/article/new'
    controller: App\Controller\ArticleController::new

comment_add:
    path: '/article/{id}/comment/add'
    controller: App\Controller\CommentController::add
```

- **Correspondance des Routes :**
  - Lorsqu'un utilisateur accède à une URL spécifique, Symfony parcourt la liste des routes définies pour trouver celle qui correspond à l'URL demandée. Par exemple, si un utilisateur

apcpedagogie.com

accède à l'URL /article/1, Symfony identifiera la route article_show comme correspondant à cette URL.

- **Actions des Routes :**
  - Une fois que la correspondance de route est trouvée, Symfony exécute l'action associée à cette route. Dans notre exemple, si un utilisateur accède à l'URL /article/1, Symfony appelle la méthode show du contrôleur ArticleController.

```
// src/Controller/ArticleController.php

namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class ArticleController
{
    /**
     * @Route("/article/{id}", name="article_show")
     */
    public function show($id): Response
    {
        // Logique pour afficher l'article avec l'identifiant $id
        // Retourne une réponse HTML avec les détails de l'article
    }

    // Autres méthodes du contrôleur pour d'autres fonctionnalités
}
```

- **Gestion des Paramètres :**
  - Vous pouvez également utiliser des paramètres dynamiques dans les URL pour capturer des valeurs spécifiques. Par exemple, dans notre route article_show, {id} est un paramètre dynamique qui représente l'identifiant de l'article à afficher.
  - Ce processus permet à Symfony de gérer les requêtes des utilisateurs de manière efficace en dirigeant les demandes vers les bonnes parties de votre application, en fonction des URLs demandées et des routes définies.

- **Connaître la structure du fichier de routage :**

apcpedagogie.com

- Apprendre à localiser et comprendre le fichier de routage principal de votre application Symfony (config/routes.yaml ou config/routes.php).
- Comprendre la syntaxe utilisée pour définir des routes dans Symfony.
- La structure du fichier de routage dans Symfony peut varier en fonction de la version de Symfony que vous utilisez et de la méthode de configuration que vous préférez (fichier YAML, annotations, PHP, etc.). Cependant, je vais vous expliquer la structure générale et vous montrer un exemple utilisant le fichier config/routes.yaml, qui est couramment utilisé dans Symfony.

1. **Localisation du fichier de routage principal :**

- Dans Symfony, le fichier de routage principal est généralement situé dans le répertoire config de votre application. Si vous utilisez la convention par défaut, le fichier de routage principal est config/routes.yaml. Cependant, vous pouvez également utiliser d'autres fichiers de routage ou même définir vos routes directement dans vos contrôleurs via des annotations.

- **Syntaxe pour définir des routes dans Symfony (fichier YAML) :**
  - Voici un exemple de la syntaxe utilisée dans le fichier config/routes.yaml pour définir des routes :

```
# Exemple de fichier de routage YAML

# Définition d'une route simple
app_homepage:
    path: /
    controller: App\Controller\HomeController::index

# Définition d'une route avec des paramètres
app_blog_show:
    path: /blog/{id}
    controller: App\Controller\BlogController::show
    methods: GET
    requirements:
        id: '\d+'

# Définition d'une route avec une méthode HTTP spécifique
app_contact_submit:
```

apcpedagogie.com

```
    path: /contact
    controller: App\Controller\ContactController::submit
    methods: POST

# Définition d'une route avec des options supplémentaires
app_admin_dashboard:
    path: /admin/dashboard
    controller: App\Controller\Admin\DashboardController::index
    options:
        expose: true
```

- Dans cet exemple :
  - Chaque route est identifiée par une clé unique (par exemple, app_homepage, app_blog_show, etc.).
  - Pour chaque route, vous spécifiez le chemin de l'URL (par exemple, /, /blog/{id}, etc.).
  - Vous indiquez également le contrôleur et la méthode à appeler lorsque la route est atteinte.
  - Vous pouvez également définir des options supplémentaires pour chaque route, telles que les méthodes HTTP acceptées, les exigences pour les paramètres de l'URL, etc.
  - Cette syntaxe YAML est l'une des façons courantes de définir des routes dans Symfony, mais vous pouvez également utiliser d'autres méthodes telles que les annotations dans vos contrôleurs ou même définir vos routes directement en PHP dans le fichier config/routes.php.

- **Exemple simple pour créer une route dans Symfony**
  - Voici un exemple simple pour créer une route dans Symfony :
  - Supposons que vous souhaitez créer une route pour afficher une page d'accueil. Voici comment vous pouvez le faire :
  - Ouvrez le fichier config/routes.yaml dans votre projet Symfony.
  - Ajoutez la définition de la route pour la page d'accueil. Par exemple :

```
# config/routes.yaml

app_homepage:
    path: /
    controller: 'App\Controller\HomeController::index'
```

Dans cet exemple :

app_homepage est le nom de la route. Vous pouvez choisir n'importe quel nom significatif pour votre route. path: / définit le chemin de l'URL pour lequel cette route doit être activée. Dans ce cas, c'est la racine de votre site Web. controller: 'App\Controller\HomeController::index' spécifie le contrôleur et la méthode qui doivent être appelés lorsque cette route est atteinte. Dans cet exemple, la méthode index du contrôleur HomeController sera appelée.

- Créez le contrôleur HomeController et la méthode index :

```
<?php
// src/Controller/HomeController.php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class HomeController extends AbstractController
{
    /**
     * @Route("/", name="app_homepage")
     */
    public function index(): Response
    {
        return $this->render('home/index.html.twig');
    }
}
```

- Dans ce contrôleur, nous avons une méthode index qui correspond à la route que nous avons définie précédemment. Cette méthode renvoie simplement une réponse en rendant un modèle Twig (home/index.html.twig).
- Créez le template Twig pour la page d'accueil :
- Créez un fichier templates/home/index.html.twig et ajoutez-y le contenu que vous souhaitez afficher sur la page d'accueil.
- Avec ces étapes, vous avez créé une route simple dans Symfony pour afficher une page d'accueil. Vous pouvez étendre ce modèle pour créer d'autres routes et pages dans votre application Symfony.

Cours Symfony

apcpedagogie.com

Source : https://apcpedagogie.com/routage-sous-symfony · téléchargé le 03/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com