Rutas

Symfony

Fernando Díaz-Alonso Dorado

Symfony

¿Qué es el enrutamiento?

Cuando la aplicación recibe una solicitud, la app llama a una acción de un determinado controlador para generar la respuesta que la aplicación mostrará.

Es la configuración del enrutamiento donde se define qué acción se debe ejecutar para cada una de las URL recibidas.

Symfony

Rutas en Symfony

Para crear las rutas podemos hacerlo de diferentes formas, todas igual de efectivas y con los mismos rendimientos:

- Usando archivos de configuración YAML, XML o PHP.
- Definiendo las rutas en las clases de los controladores mediante los atributos de PHP.

Información:

https://symfony.com/doc/current/routing.html#creating-routes

Symfony

Enrutamiento con YAML, XML o PHP

Este tipo de definiciones se realizan en un archivo de formato YAML, XML o PHP separado de los controladores.

- Su principal ventaja es que no requiere ninguna dependencia adicional.
- Y su principal inconveniente es que hay que trabajar con varios archivos para poder verificar el enrutamiento de las acciones de los controladores.

Importante: Por defecto, Symfony carga las rutas de archivos en formato YAML. Si deseas cambiarlo debes hacerlo en el archivo *src/Kernel.php*

Información:
https://symfony.com/doc/current/routing.html#creating-routes-in-yaml-xml-or-php-files

Symfony

Enrutamiento con Atributos PHP

Los atributos de PHP permiten definir las rutas junto al código de los controladores, asociando las rutas a los controladores.

Los atributos son nativos a partir de PHP 8, y para versiones posteriores. Pudiendo usarlos dentro del propio código de la clase del controlador.

Symfony recomienda esta opción porque es conveniente colocar las rutas y el controlador en el mismo lugar.

Información:

https://symfony.com/doc/current/routing.html#creating-routes-as-attributes

Symfony

Enrutamiento con Atributos PHP II

Para poder usar los atributos de PHP se debe configurar mediante el archivo de configuración attributes.yaml dentro de config/routes/

Esta configuración sigue el estándar PSR-4.

```yaml
# config/routes/attributes.yaml
controllers:
    resource:
        path: ../../src/Controller/
        namespace: App\Controller
    type: attribute

kernel:
    resource: App\Kernel
    type: attribute
```

Symfony

Enrutamiento con Atributos PHP III

Las rutas siempre van asociadas a una determinada acción dentro del controlador. Crearemos una anotación para la función que define la acción.

Podemos tener diferentes acciones a una misma ruta pero mediante métodos, condiciones o requerimientos definir cual es la acción correcta para la ruta.

El formato de la ruta será:

```
#[Route(
    'path',
    name: 'nombre_ruta' ,
    methods:['GET, 'POST',… ],
    requirements[ '_locale' => 'pais', _format => 'json', …],
),]
```

Symfony

Enrutamiento con Atributos PHP IV

Opciones de la anotación:

- *path:* la ruta que se indica en la URL
- *name:* opción que define como llamamos a nuestra ruta por 'nombre_ruta'
- *methods:* sirve para definir los diferentes métodos HTTP (GET, POST, PUT, …) y utilizar diferentes acciones con una misma. Más info: https://symfony.com/doc/current/routing.html#matching-http-methods
- *condiction:* sirve para definir las condiciones mediante parámetros usando condiciones. Más info:

  https://symfony.com/doc/current/routing.html#matching-expressions
- *requirements:* sirve para definir las condiciones mediante expresiones de PHP.Más info:

https://symfony.com/doc/current/routing.html#parameters-validation

Symfony

Enrutamiento con Atributos PHP V

Ejemplo:

Para la ruta: /blog la definición sería como en el código inferior dentro de la clase del controlador BlogController

```php
<?php
//scr/Controller/BlogController.php
namespace App\Controller;

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

class BlogController extends AbstractController{
    #[Route('/blog', name: 'blog_list', methods: ['GET'])]
    public function list(): Response{
        //...
    }

    #[Route('/blog/{id}', name: 'blog_list', methods:['POST'])]
    public function edit(int $id): Response{
        //...
    }

    #[Route('/blog/{id}', name: 'blog_list', methods:['POST'], condition: "params['id'] < 0")]
    public function delete(int $id): Response{
        //...
    }
}
```

Symfony

Debugging Rutas

Podemos listar todas las rutas que tiene definida nuestra aplicación mediante el comando en la consola → debug:router

php bin/console debug:router

```
---------------- -------- -------- ------ --------------------------
 Name             Method   Scheme   Host   Path
---------------- -------- -------- ------ --------------------------
 _preview_error   ANY      ANY      ANY    /_error/{code}.{_format}
 blog_list        GET      ANY      ANY    /blog
 blog_post        POST     ANY      ANY    /blog/{id}
 blog_delete      POST     ANY      ANY    /blog/{id}
---------------- -------- -------- ------ --------------------------
```

Symfony

Debugging Rutas II

O conocer la definición de alguna de las rutas en concreto por su nombre.

php bin/console debug:router blog_delete

```
+--------------+-------------------------------------------------------------+
| Property     | Value                                                       |
+--------------+-------------------------------------------------------------+
| Route Name   | blog_delete                                                 |
| Path         | /blog/{id}                                                  |
| Path Regex   | {^/blog/(?P<id>[^/]++)$}sDu                                 |
| Host         | ANY                                                         |
| Host Regex   |                                                             |
| Scheme       | ANY                                                         |
| Method       | POST                                                        |
| Requirements | NO CUSTOM                                                   |
| Class        | Symfony\Component\Routing\Route                             |
| Defaults     | _controller: App\Controller\BlogController::delete()        |
| Options      | compiler_class: Symfony\Component\Routing\RouteCompiler     |
|              | utf8: true                                                  |
| Condition    | params['id'] < 0                                            |
+--------------+-------------------------------------------------------------+
```

Symfony

Debugging Rutas III

E incluso podemos saber cual es el nombre de la ruta a partir de una de las rutas, para verificar que tiene una acción asociada. → route:match

php bin/console route:match /blog

```
[OK] Route "blog_list" matches

+--------------+-----------------------------------------------------------+
| Property     | Value                                                     |
+--------------+-----------------------------------------------------------+
| Route Name   | blog_list                                                 |
| Path         | /blog                                                     |
| Path Regex   | {^/blog$}sDu                                              |
| Host         | ANY                                                       |
| Host Regex   |                                                           |
| Scheme       | ANY                                                       |
| Method       | GET                                                       |
| Requirements | NO CUSTOM                                                 |
| Class        | Symfony\Component\Routing\Route                           |
| Defaults     | _controller: App\Controller\BlogController::list()        |
| Options      | compiler_class: Symfony\Component\Routing\RouteCompiler   |
|              | utf8: true                                                |
+--------------+-----------------------------------------------------------+
```

Symfony

Más opciones de trabajo con rutas

Symfony tiene un sinfín de documentación y de diferentes casos de uso.

Es recomendable antes de abordar un nuevo proyecto que se revise la documentación de Symfony.

https://symfony.com/doc/current/routing.html#routing-route-parameters