

Routing

Resumen básico

En symfony, una ruta es un mapeo entre una url y un controlador. Las rutas no se programan, sino que se configuran o definen.

Hay 4 formas distintas de configurar las rutas en symfony:

- Con anotaciones en los propios controladores
- En un fichero de configuración YML
- En un fichero de configuración XML
- En un fichero de configuración PHP

El orden de las rutas importa. La primera ruta que haga *match* con la petición, será la ruta escogida por symfony.

Si no conoces cómo configurar rutas básicas o rutas con parámetros, puedes acudir a la documentación oficial de symfony:

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

Routing

Restricciones de los parámetros (requirements)

```
user_edit:
  path: /user/edit/{id}
  defaults: { _controller: App\Controller\UsersController:
edit }
  requirements:
    id: '\d+'
```

El `\d+` es una expresión regular.

Otro ejemplo con expresiones regulares podría ser:

```
homepage:
  path: /{_locale}
  defaults: { _controller: App\Controller\MainController:
homepage }
  requirements:
    _locale: es|fr
```

Valor por defecto de un

parámetro

```
user_edit:
  path: /user/edit/{id}
  defaults: { _controller: App\Controller\UsersController:
edit, id: 1 }
  requirements:
    id: '\d+'

homepage:
  path: /{_locale}
  defaults: { _controller: App\Controller\MainController:
homepage, _locale: en }
  requirements:
    _locale: en|fr
```

Aquí podemos ver unos ejemplos:

```
/      =>  {_locale} = "en"
/en     =>  {_locale} = "en"
/fr     =>  {_locale} = "fr"
/es     =>  No hace match
```

Con anotaciones:

```
class MainController extends Controller
{
```

```

/**
 * @Route("/{_locale}", defaults={"_locale": "en"}, re
quirements={
 *     "_locale": "en|fr"
 * })
 */
public function homepageAction($_locale)
{
}

```

Sin hacer nada más, symfony aplicará automáticamente el idioma correspondiente.

Parámetros extra

Es posible pasar parámetros extra en la ruta:

```

# config/routes.yaml
blog:
    path:      /usuarios/{page}
    controller: App\Controller\UsuariosController:index
    defaults:
        page: 1
        title: "Listado de usuarios"

```

Routing

Parámetros especiales

El componente de routing tiene estos 4 parámetros especiales:

- `_controller`: Este parámetro determina qué controlador se ejecutará. La sintaxis es `bundle:controller:action`
- `_locale`: Establece el idioma de la petición
- `_format`: Establece el formato de la request (Ej: `Content-Type:application/json`).
- `_fragment`: Establece el *fragment* de la url

Ejemplo:

```
article_show:
  path:      /articles/{slug}.{_format}
  defaults: { _controller: App\Controller\Article:show, _f
ormat: html }
  requirements:
    _format:  html|rss
```

```
class ArticleController extends Controller
```

```
{  
    /**  
    * @Route(  
    *     "/articles/{slug}.{_format}",  
    *     defaults={"_format": "html"},  
    *     requirements={  
    *         "_format": "html|rss",  
    *     }  
    * )  
    */  
    public function showAction($_locale, $year, $slug)  
    {  
    }  
}
```

Routing

Redirecciones

Redireccionar a otra url:

```
# redirecting the homepage
homepage:
    path: /
    controller: Symfony\Bundle\FrameworkBundle\Controller\
RedirectController::urlRedirectAction
    defaults:
        path: /app
        permanent: true
```

Redireccionar a otra ruta:

```
admin:
    path: /wp-admin
    controller: Symfony\Bundle\FrameworkBundle\Controller\
RedirectController::redirectAction
    defaults:
        route: sonata_admin_dashboard
        permanent: true
```

Ejercicio: investigar la diferencia entre poner permanent: true o false.

Routing

Depuración de rutas

Listado de las todas las rutas:

```
php bin/console debug:router
```

Información sobre una ruta específica:

```
php bin/console debug:router article_show
```

Testear una URL:

```
php bin/console router:match /blog/my-latest-post
```


Routing

Restricción de rutas por Host

```
# config/routes.yaml

mobile_homepage:
    path:      /
    host:      m.example.com
    controller: App\Controller\MainController::mobileHomepage

homepage:
    path:      /
    controller: App\Controller\MainController::homepage
```

```
// src/Controller/MainController.php

namespace App\Controller;

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

class MainController extends Controller
{
    /**
     * @Route("/", name="mobile_homepage", host="m.example.com")
     */
}
```

```

    */
    public function mobileHomepage()
    {
        // ...
    }

    /**
     * @Route("/", name="homepage")
     */
    public function homepage()
    {
        // ...
    }
}

```

Se pueden utilizar parámetros en la sección *host*:

```

# config/routes.yaml
projects_homepage:
    path:      /
    host:      "{project_name}.example.com"
    controller: App\Controller\MainController::projectsHomepage

homepage:
    path:      /
    controller: App\Controller\MainController::homepage

```

Si se usa algún parámetro, puede tener valor por defecto, restricciones, etc.

Routing

Restricción de rutas por método

```
api_user_show:
  path:      /api/users/{id}
  defaults: { _controller: App\Controller\UsersApi:show
}
  methods:   [GET, HEAD]
```

```
api_user_edit:
  path:      /api/users/{id}
  defaults: { _controller: App\Controller\UsersApi:edit
}
  methods:   [PUT]
```

```
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Method;
// ...

class UsersApiController extends Controller
{
    /**
     * @Route("/api/users/{id}")
     * @Method({"GET", "HEAD"})
     */
```

```
public function showAction($id)
{
    // ... return a JSON response with the post
}

/**
 * @Route("/api/users/{id}")
 * @Method("PUT")
 */
public function editAction($id)
{
    // ... edit a post
}
}
```

Routing

Restricción de rutas por protocolo

```
# config/routes.yml

secure:

    path:          /secure

    controller: App\Controller\MainController::secure

    schemes:       [https]
```

```
// src/Controller/MainController.php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route
;

class MainController extends Controller
{

    /**
     * @Route("/secure", name="secure", schemes={"https"})
     */
    public function secure()
    {

        // ...
    }
}
```

```
}  
}
```

(se puede gestionar también con el componente de seguridad de symfony)

https://symfony.com/doc/current/security/force_https.html