ÉCOLE
DE LA FILIÈRE
NUMÉRIQUE

SYMFONY

Routing

Créé par Étienne Dauvergne

L'ÉCOLE DE FORMATION PROFESSIONNELLE CRÉÉE PAR DES INFORMATICIENS

Créée en 1994, l'école de la filière numérique est actuellement implantée sur Nantes, Angers et Rennes. IMIE vous propose d'obtenir des diplômes reconnus, et de maîtriser les technologies les plus récentes.

ROUTING

Associer un **Contrôleur** à une **Requête** (URL) pour obtenir une **Réponse**.

[**GET**] **/blog**	Afficher la liste des articles
[**GET**] **/blog/{slug}**	Afficher le detail de l'article

ANNOTATION

Route simple

```php
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Method;

class BlogController extends Controller
{
    /**
     * @Route("/blog", name="blog_list")
     */
    public function listAction()
    {
        // ...
    }
}
```

ANNOTATION

Route avec paramètre

```php
class BlogController extends Controller
{
    /**
     * @Route("/blog/{slug}", name="blog_show")
     */
    public function showAction($slug)
    {
        // Ex: /blog/super-article, $slug = 'super-article'
    }
}
```

ANNOTATION

Contrainte et valeur par défaut d'un paramètre

```php
class BlogController extends Controller
{
    /**
     * @Route("/blog/{page}",
     *     name="blog_list",
     *     requirements={"page": "\d+"}
     * )
     */
    public function listAction($page = 1)
    {
        // ...
    }
}
```

ANNOTATION

Contrainte et valeur par défaut d'un paramètre

```php
class BlogController extends Controller
{
    /**
     * @Route("/blog/{slug}.{_format}",
     *      name="blog_show",
     *      defaults={"_format": "html"},
     *      requirements={
     *          "slug": "[a-z0-9-]+",
     *          "_format": "html|json"
     *      }
     * )
     */
    public function showAction($slug) {}
}
```

ANNOTATION

Contrainte sur la méthode HTTP

```php
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Method;

class BlogController extends Controller
{
    /**
     * @Route("/blog/{slug}")
     * @Method("GET")
     */
    public function showAction($slug) {}
}
```

YAML

```yaml
# app/config/routing.yml
blog_list:
    path:           /blog/{page}
    methods:        [GET]
    defaults:
        _controller: AppBundle:Blog:list
        page:        1
    requirements:
        page:        \d+
```

YAML

```yaml
# app/config/routing.yml
blog_show:
    path:           /blog/{slug}.{_format}
    methods:        [GET]
    defaults:
        _controller: AppBundle:Blog:show
        _format:     html
    requirements:
        slug:        "[a-z0-9-]+"
        _format:     html|json
```

XML

```xml
<!-- app/config/routing.xml -->
<?xml version="1.0" encoding="UTF-8" ?>
<routes xmlns="http://symfony.com/schema/routing"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://symfony.com/schema/routing
        http://symfony.com/schema/routing/routing-1.0.xsd">

  <route id="blog_list" path="/blog/{page}" methods="GET">
    <default key="_controller">AppBundle:Blog:list</default>
    <default key="page">1</default>
    <requirement key="page">\d+</requirement>
  </route>

</routes>
```

XML

```xml
<!-- app/config/routing.xml -->
<?xml version="1.0" encoding="UTF-8" ?>
<routes xmlns="http://symfony.com/schema/routing"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://symfony.com/schema/routing
        http://symfony.com/schema/routing/routing-1.0.xsd">

  <route id="blog_show" path="/blog/{slug}.{_format}"
         methods="GET">
    <default key="_controller">AppBundle:Blog:show</default>
    <default key="_format">html</default>
    <requirement key="_format">html|rss</requirement>
    <requirement key="slug">[a-z0-9]+</requirement>
  </route>

</routes>
```

CONTROLLER

```php
namespace AppBundle\Controller;

use Symfony\Component\HttpFoundation\Response;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;

class BlogController extends Controller
{
    public function indexAction()
    {
        return new Response(
            '<html><body>Best blog ever !</body></html>'
        );
    }
}
```

REQUEST

```php
use Symfony\Component\HttpFoundation\Request;

$request = Request::createFromGlobals();

$request->isXmlHttpRequest();       // Ajax ?
$request->query->get('page');       // GET variables
$request->request->get('page');     // POST variables
$request->server->get('HTTP_HOST'); // SERVER variables
$request->files->get('foo');        // FILES variables
$request->headers->get('host');     // HTTP headers
```

RESPONSE

```php
use Symfony\Component\HttpFoundation\Response;

// Simple réponse avec un code HTTP 200 (succès)
$response = new Response('It works :D', Response::HTTP_OK);
// Modifier le contenu
$response->setContent('Or not :s');
// Modifier le code HTTP
$response->setStatusCode(Reponse::HTTP_NOT_FOUND);
// Renvoyer la réponse au navigateur
$response->prepare($request)->send();
```

SESSION

```php
$session = $request->getSession();

// Stocker une donnée
$session->set('foo', 'bar');

// Récupérer la donnée
$foobar = $session->get('foobar', 'default value');
```

RÉCUPÉRER LES PARAMÈTRES D'UNE ROUTE

```php
public function indexAction(Request $request)
{
    $slug = $request->attributes->get('slug', null);
    $_format = $request->attributes->get('_format', 'html');
}
```

GÉNÉRER UNE URL

```php
class MyController extends Controller
{
    public function showAction($slug)
    {
        $url = $this->generateUrl('blog_show', [
            'slug' => 'my-blog-post'
        ]);
        // $url = /blog/my-blog-post
    }
}
```

REDIRECTIONS

```php
public function indexAction()
{
    // Rediriger vers la route "home"
    return $this->redirectToRoute('home');

    // Redirection permanente (301)
    return $this->redirectToRoute('homepage', [], 301);

    // Redirider vers une route avec paramètres
    return $this->redirectToRoute('blog_show', [
        'slug' => 'my-page'
    ]);

    // Rediriger vers une URL
    return $this->redirect('http://symfony.com');
}
```

FAIRE SUIVRE LA REQUÊTE À UN AUTRE CONTRÔLEUR

```php
public function indexAction($name)
{
    // Réponse revoyée par la sous-requête
    $response = $this->forward('AppBundle:Something:fancy', [
        'name'  => $name,
        'color' => 'green',
    ]);

    return $response;
}
```

EFFECTUER LE RENDU D'UN GABARIT

```php
public function showAction($slug)
{
    $post = $this->findPost($slug);

    $content = $this->renderView('AppBundle:Blog:show', [
        'post' = $post,
    ]);

    return new Response($content);
}
```

AJOUTER DES MESSAGES FLASH

```php
public function showAction($slug)
{
    $this->addFlash(
        'notice',
        'Vos modifications ont été sauvegardées !'
    );

    // Ou $request->getSession()->getFlashBag()->add( ... )
}
```

CRÉER UNE EXCEPTION "PAGE INTROUVABLE"

```php
public function showAction($slug)
{
    return $this->createNotFoundException(
        "J'ai pourtant fouillé partout !"
    );
}
```