



Jobeet FR

Le tutoriel pour Symfony2 en français

Le Routage // [Commentaires ↓](#)

URLs

Si vous cliquez sur une offre sur la page d'accueil Jobeet, l'URL ressemble à ceci: **/job/1/show**. Si vous avez déjà développé des sites web PH vous êtes probablement plus habitués à des URL comme **/job.php?id=1**. Comment Symfony les fait fonctionner? Comment Symfony déterminer l'action à appeler à partir de cette URL? Pourquoi l'id de l'offre est-il récupéré avec le paramètre **\$id** dans l'action? Ici, nous allons répondre toutes ces questions.

Vous avez déjà vu le code suivant dans le template **src/Ens/JobeetBundle/Resources/views/job/index.html.twig**:

```
{{ path('ens_job_show', { 'id': entity.id }) }}
```

Celui-ci utilise la fonction de l'helper template **path** pour générer l'url pour l'offre qui a l'id 1. **ens_job_show** est le nom de la route utilisé définie dans la configuration comme vous le verrez ci-dessous.

Configuration du routage

Dans Symfony2, la configuration du routage se fait habituellement dans le fichier **app/config/routing.yml**. Ceci importe la configuration de routage d'un paquet spécifique. Dans notre cas, le fichier **src/Ens/JobeetBundle/Resources/config/routing.yml** est importé:

```
1. # app/config/routing.yml
2.
3. EnsJobeetBundle:
4.     resource: "@EnsJobeetBundle/Resources/config/routing.yml"
5.     prefix:  /
```

Maintenant, si vous regardez dans le fichier **routing.yml** de **JobeetBundle** vous verrez qu'il importe un autre fichier de routage, celui pour le contrôleur **Job** et définit une route appelée **EnsJobeetBundle_homepage** pour le modèle d'URL **/hello/{name}**:

```
1. # src/Ens/JobeetBundle/Resources/config/routing.yml
2.
3. EnsJobeetBundle_job:
4.     resource: "@EnsJobeetBundle/Resources/config/routing/job.yml"
5.     prefix: /job
6.
7. EnsJobeetBundle_homepage:
8.     pattern: /hello/{name}
9.     defaults: { _controller: EnsJobeetBundle:Default:index }
```

```
1. # src/Ens/JobeetBundle/Resources/config/routing/job.yml
2.
3. ens_job:
4.     pattern: /
5.     defaults: { _controller: "EnsJobeetBundle:Job:index" }
6.
7. ens_job_show:
8.     pattern: /{id}/show
9.     defaults: { _controller: "EnsJobeetBundle:Job:show" }
10.
11. ens_job_new:
```

```
12.     pattern: /new
13.     defaults: { _controller: "EnsJobeetBundle:Job:new" }
14.
15. ens_job_create:
16.     pattern: /create
17.     defaults: { _controller: "EnsJobeetBundle:Job:create" }
18.     requirements: { _method: post }
19.
20. ens_job_edit:
21.     pattern: /{id}/edit
22.     defaults: { _controller: "EnsJobeetBundle:Job:edit" }
23.
24. ens_job_update:
25.     pattern: /{id}/update
26.     defaults: { _controller: "EnsJobeetBundle:Job:update" }
27.     requirements: { _method: post }
28.
29. ens_job_delete:
30.     pattern: /{id}/delete
31.     defaults: { _controller: "EnsJobeetBundle:Job:delete" }
32.     requirements: { _method: post }
```

Voyons de plus près la route **ens_job_show**. Le modèle défini par **ens_job_show** agit comme **/*/show** où le joker est l'**id**. Pour l'UF **/1/show**, la variable **id** obtient une valeur de 1, qui est disponible pour vous permettre de l'utiliser dans votre contrôleur. Le paramètre **_controller** est une entrée spéciale qui dit à Symfony quel contrôleur/action doit être exécuté quand une URL correspond à cette route. Dans notre cas, il doit exécuter **showAction** de **JobController** dans **EnsJobeetBundle**.

Les paramètres de route (par exemple, **{id}**) sont particulièrement importants parce que chacun est mis à disposition en tant qu'argument à méthode du contrôleur:

```
1. public function showAction($id)
2. {
3.     // ...
4. }
```

Configuration du routage en environnement dev

L'environnement de dev charge le fichier **app/config/routing_dev.yml** qui contient les routes utilisées par la **Web Debug Toolbar**, entre autres. Ce fichier charge, à la fin, le fichier de configuration principal **routing.yml**.

Si vous allez à http://jobeet.local/app_dev.php, vous verrez la page par défaut de Symfony. Nous allons changer cela pour qu'à l'avenir l'environnement de dev affiche les mêmes pages que celui de production. Ouvrez votre **app/config/routing_dev.yml** et enlevez les 3 premières routes qui sont relatives au paquet de démonstration de Symfony. Maintenant, le fichier **routing_dev.yml** devrait ressembler à ceci:

```
1. # app/config/routing_dev.yml
2.
3. _assetic:
4.     resource: .
5.     type: assetic
6.
7. _wdt:
8.     resource: "@WebProfilerBundle/Resources/config/routing/wdt.xml"
9.     prefix: /_wdt
10.
11. _profiler:
12.     resource: "@WebProfilerBundle/Resources/config/routing/profiler.xml"
13.     prefix: /_profiler
14.
15. _configurator:
16.     resource: "@SensioDistributionBundle/Resources/config/routing/webconfigurator.xml"
```

```
17.     prefix:    /_configurator
18.
19.     _main:
20.         resource: routing.yml
```

N'oubliez pas de vider le cache après cette modification.

```
php app/console cache:clear --env=prod
php app/console cache:clear --env=dev
```

Personnalisation des routes

Pour l'instant, lorsque vous demandez l'URL `/` dans un navigateur, vous obtenez une erreur 404. C'est parce que cette URL ne correspond aucune des routes définies. Nous avons une route **EnsJobeetBundle_homepage** qui correspond à l'URL **/hello/jobeeet** et nous envoie l'action **index** de **DefaultController**. Nous allons la changer pour correspondre à l'URL `/` et appeler l'action **index** de **JobController**. Pour faire le changement, modifiez comme ci-dessous:

```
1. # src/Ens/JobeetBundle/Resources/config/routing.yml
2. # ...
3.
4. EnsJobeetBundle_homepage:
5.     pattern:    /
6.     defaults: { _controller: EnsJobeetBundle:Job:index }
```

Maintenant, si vous effacez le cache et allez à <http://jobeeet.local>, vous verrez la page d'accueil Job. Nous pouvons maintenant modifier le lien c logo Jobeet dans le layout pour utiliser la route **EnsJobeetBundle_homepage**:

```
1. <!-- src/Ens/JobeetBundle/Resources/views/layout.html.twig -->
2. <!-- ... -->
3. <h1><a href="{{ path('EnsJobeetBundle_homepage') }}">
4.     
5. </a></h1>
6. <!-- ... -->
```

Pour quelque chose d'un peu plus compliqué, nous allons changer l'URL d'une offre avec quelque chose de plus significatif:

```
/job/sensio-labs/paris-france/1/web-developer
```

Sans rien savoir de Jobeet, et sans regarder la page, vous pouvez comprendre à partir de l'URL que Sensio Labs est à la recherche d'un Développeur Web pour travailler à Paris, France.

Le schéma ci-dessous correspond à une telle URL:

```
/job/{company}/{location}/{id}/{position}
```

Modifiez la route **ens_job_show** à partir du fichier **job.yml**:

```
1. # src/Ens/JobeetBundle/Resources/config/routing/job.yml
2. # ...
3.
4. ens_job_show:
5.     pattern:    /{company}/{location}/{id}/{position}
6.     defaults: { _controller: "EnsJobeetBundle:Job:show" }
```

Maintenant, nous avons besoin de passer tous les paramètres de la route modifiée pour que cela fonctionne:

```
1. <!-- src/Ens/JobeetBundle/Resources/views/Job/index.html.twig -->
2. <!-- ... -->
3. <a href="{{ path('ens_job_show', { 'id': entity.id, 'company': entity.company, 'location': entity.location, 'posit
```

```

4.     ion': entity.position }) }}">
5.     {{ entity.position }}
6. </a>
7. <!-- ... -->

```

Si vous regardez les URLs générées, elles ne sont pas encore tout à fait comme nous voulons qu'elles soient:

```
http://jobeet.local/job/Sensio Labs/Paris, France/1/Web Developer
```

Nous avons besoin de "remodeler" les valeurs de chaque colonne en remplaçant tous les caractères non ASCII par un tiret (-). Ouvrez le fichier **Job.php** et ajoutez les méthodes suivantes à la classe:

```

1. // src/Ens/JobeetBundle/Entity/Job.php
2. // ...
3.
4. use Ens\JobeetBundle\Utils\Jobeet as Jobeet;
5. class Job
6. {
7.     // ...
8.     public function getCompanySlug()
9.     {
10.         return Jobeet::slugify($this->getCompany());
11.     }
12.
13.     public function getPositionSlug()
14.     {
15.         return Jobeet::slugify($this->getPosition());
16.     }
17.
18.     public function getLocationSlug()
19.     {
20.         return Jobeet::slugify($this->getLocation());
21.     }
22. }

```

Ensuite, créez le fichier **src/Ens/JobeetBundle/Utils/Jobeet.php** et ajoutez la méthode **slugify()**:

```

1. // src/Ens/JobeetBundle/Utils/Jobeet.php
2.
3. namespace Ens\JobeetBundle\Utils;
4. class Jobeet
5. {
6.     static public function slugify($text)
7.     {
8.         // replace all non letters or digits by -
9.         $text = preg_replace('/\W+/', '-', $text);
10.
11.         // trim and lowercase
12.         $text = strtolower(trim($text, '-'));
13.
14.         return $text;
15.     }
16. }

```

Nous avons défini trois nouveaux **accesseurs** "virtuels": **getCompanySlug()**, **getPositionSlug()**, et **getLocationSlug()**. Ils retournent la valeur correspondant à leur colonne après avoir appliqué la méthode **slugify()**. Maintenant, vous pouvez remplacer les noms réels des colonnes par les virtuels dans le modèle:

```

1. <!-- src/Ens/JobeetBundle/Resources/views/Job/index.html.twig -->
2. <!-- ... -->
3. <a href="{{ path('ens_job_show', { 'id': entity.id, 'company': entity.companySlug, 'location': entity.locationSlug

```

```
4. , 'position': entity.positionSlug}) }}">
5. {{ entity.position }}
6. </a>
7. <!-- ... -->
```

Vous aurez maintenant l'URL attendue:

```
http://jobeet.local/app_dev.php/job/sensio-labs/paris-france/1/web-developer
```

Les conditions dans les routes

Le système de routage a une fonction de validation intégrée. Chaque modèle de variable peut être validé par une expression régulière définie en utilisant l'entrée **requirements** d'une définition de route:

```
1. # src/Ens/JobeetBundle/Resources/config/routing/job.yml
2. # ...
3.
4. ens_job_show:
5.     pattern:  /{company}/{location}/{id}/{position}
6.     defaults: { _controller: "EnsJobeetBundle:Job:show" }
7.     requirements:
8.         id: \d+
```

L'entrée **requirements** ci-dessus oblige l'**id** à être une valeur numérique. Sinon, la route ne fonctionne pas.

Débugage des routes

Bien que l'ajout et la personnalisation de routes soit parlant, il est utile de pouvoir visualiser et obtenir des informations détaillées sur vos routes. Un excellent moyen de voir toutes les routes dans votre application se fait via la commande **router:debug**. Exécutez la commande en la lançant à partir de la racine de votre projet:

```
php app/console router:debug
```

La commande permet d'afficher une liste utile de toutes les routes configurées dans votre application. Vous pouvez également obtenir de plus amples renseignements très précis sur une seule route en incluant le nom de la route après la commande:

```
php app/console router:debug ens_job_show
```

Réflexions finales

Pour en savoir plus sur le système de routage Symfony2, lisez le chapitre [Routing](#) de la documentation officielle.

Une question ? Une réaction ?

6 Commentaires [Jobeet Symfony2 FR](#)

 [S'identifier](#)

 [Recommander](#) 1

 [Partager](#)

[Les meilleurs](#)



Participer à la discussion...



[med](#) • il y a 3 ans



add in Job.php

```

use Ens\JobeetBundle\Utils;
//.....
public function getCompanySlug() {

    return Utils\Jobeet::slugify($this->getCompany());

}

public function getPositionSlug() {

    return Utils\Jobeet::slugify($this->getPosition());

}

public function getLocationSlug() {

    return Utils\Jobeet::slugify($this->getLocation());

}

```

1 ^ | v • Répondre • Partager ›

**Thomas Caron** • il y a un mois

Hello, beaucoup plus simple:

Twig à une fonction toute prête pour remplacer les , et les espaces vides

comme ceci :

```
'id': job.id, 'company': job.company|replace({' ': '-'}), 'location': job.location|replace({' ': '-'}), 'position': job.position
```

^ | v • Répondre • Partager ›

**Ignition** • il y a un an

J'ai peut-être pas tout suivi correctement, mais si vous avez mis votre classe Jobeet.php dans src/JobeetBundle/Utils et que même en respectant la casse dans le twig vous avez une erreur, essayez d'importer votre classe Jobeet.php dans votre fichier Job.php :

```

namespace Ens\JobeetBundle\Entity;

use Doctrine\ORM\Mapping as ORM;

use Ens\JobeetBundle\Utils\Jobeet as Jobeet;

```

J'avais une erreur floue qui m'indiquait que sf2 ne trouvait pas Jobeet :)

Si ça peut vous aider... =)

^ | v • Répondre • Partager ›

**N.** • il y a 4 ans

Désolé pour le double post, je retente de poster le code html.

```
{{ path('ens_job_show', { 'id': entity.id, 'company': entity.companySlug, 'location': entity.locationSlug, 'position': entity.positionSlug }) }}
```

Si ça ne passe pas : il faut mettre en majuscule le S de slug dans le twig => entity.companySlug, entity.locationSlug et entity.positionSlug.

^ | v • Répondre • Partager ›

**Jon'** Mod ➔ N. • il y a 4 ans

Merci c'est corrigé :)

^ | v • Répondre • Partager ›

**N.** • il y a 4 ans

Un léger problème de casse dans le twig pour faire appel aux accesseurs virtuels qui peut poser problème :

```
{{ path('ens_job_show', { 'id': entity.id, 'company': entity.companySlug, 'location': entity.locationSlug, 'position': entity.positionSlug }) }}
```

```
{ path: 'site_ges_ges', { id: 'entity_id', company: 'entity:company_id', location: 'entity:location_id', position: 'entity:position_id' },
```

++

[^](#) | [v](#) • [Répondre](#) • [Partager](#) ›

ÉGALEMENT SUR JOBEET SYMFONY2 FR

Flux de données

1 commentaire • il y a 4 ans•



Jean-François E. — Pour voir la suite du tutoriel Jobeet Symfony2 en français : - Symfony2 - Jobeet - Jour 15 - Services Web : <http://blog.erlem.fr/programma...> - Symfony2 - Jobeet - Jour 16 ...

Les tests unitaires

2 commentaires • il y a 4 ans•



Iorant — Bonjour,En installant phpunit par composer à l'aide de :php composer.phar require "phpunit/phpunit"J'ai bien un dossier phpunit dans les vendors et un fichier phpunit.xml.dist dans ...

Sécurité

1 commentaire • il y a 4 ans•



philippe — Bonjour,Le app/config.security.yml tel qu'il est paramétré au début génère une erreur "InvalidConfigurationException: Unrecognized options "users" ...

Le projet

1 commentaire • il y a 4 ans•



Toofff — Bonjour, j'aimerais avoir plus d'information au sujet des applications car je n'arrives pas à voir la correspondance entre SF et SF2. Par exemple dans SF1 je pouvais faire 3 appli : ...

[✉ S'abonner](#) [D Ajoutez Disqus à votre site web ! Add Disqus Add](#) [🔒 Vie Privée](#)Tutoriel original par [Dragos Holban](#) - Traduit par [Jonathan Thuau](#) - Le tuto sur [github](#)