

## TD Symfony

### Étape 2 : Une première route.

Il existe 2 façons de créer des routes en Symfony, soit dans un fichier config, soit à l'aide d'annotations (voir : [https://symfony.com/doc/current/page\\_creation.html](https://symfony.com/doc/current/page_creation.html))

Nous allons utiliser la seconde méthode, pour cela il faut installer le package « annotations » à l'aide de composer.

```
\td_symfony> composer require annotations
```

```
Generating optimized autoload files
```

...

```
Executing script cache:clear [OK]  
Executing script assets:install public [OK]
```

Une fois le package installé, nous allons pouvoir créer un premier contrôleur, dans le dossier src/Controller, y définir une action (function) et indiquer quelle est la route vers cette action.

```
src > Controller > TestController.php > TestController > index  
1  <?php  
2  
3  namespace App\Controller; 1  
4  
5  use Symfony\Component\HttpFoundation\Response;  
6  use Symfony\Component\Routing\Annotation\Route; 2  
7  
8  class TestController  
9  {  
10  
11      /**  
12      3      * @Route("/test")  
13      */  
14      public function index(): Response  
15      {  
16          return new Response(  
17      4          '<h1>TestController index</h1>'  
18          );  
19      }  
20  }
```

1. Le namespace dans lequel se trouve notre classe TestController
2. L'import avec use des classes utilisées dans notre classe TestController : la classe Response (représentant une réponse HTTP) et la classe Route du package annotations, que nous avons installé, permettant de définir les routes (vers l'action d'un contrôleur) directement dans nos contrôleur avec l'annotation @Route
3. La définition de la route avec une annotation directement au dessus de l'action.
4. La réponse HTTP de notre action, ici une balise h1.

Testons cette première route, lancer le serveur avec la commande de symfony-cli :

`symfony server:start`

Puis ouvrez la page dans le navigateur (ctrl + clic sur le lien de l'encadré vert), tapez la route dans la barre d'url du navigateur, et là ... C'est le drame ! Une belle page d'erreur Symfony :-(  
Autant vous y habituer tout de suite, vous risquez d'en voir souvent :-)

Par défaut dans le squelette de notre application, Symfony est configuré pour utiliser une base de données postreSql, et nous n'en n'avons pas.

Arrêter le serveur (ctrl+c) dans le terminal VSCode, puis utilisez la commande `symfony server:stop`

Lancez votre serveur de base de données MySQL (WAMP, MAMP, XAMPP, etc ...) et ouvrez phpMyAdmin.

Ouvrez le fichier `.env` de votre projet Symfony, commentez la ligne `DATABASE_URL` pour posgreSql et dé-commentez la ligne pour mySql (# en début de ligne pour commenter).

```
.env
28 #
29 # DATABASE_URL="sqlite:///kernel.project_dir%/var/data.db"
30 DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/db_name?serverVersion=5.7"
31 # DATABASE_URL="postgresql://db_user:db_password@127.0.0.1:5432/db_name?serverVersion=13&charset=utf8"
32 ###< doctrine/doctrine-bundle ###
33
```

Renseignez cette ligne mySql avec les paramètres de votre serveur de bases de données (db\_user, db\_pass, port, version). Nous allons appeler notre base de données « td\_symfony\_db » (utiliser ce nom pour db\_name).

```
.env
28 #
29 # DATABASE_URL="sqlite:///kernel.project_dir%/var/data.db"
30 DATABASE_URL="mysql://root:@127.0.0.1:3306/td_symfony_db?serverVersion=8.0"
31 # DATABASE_URL="postgresql://db_user:db_password@127.0.0.1:5432/db_name?serverVersion=13&charset=utf8"
32 ###< doctrine/doctrine-bundle ###
33
```

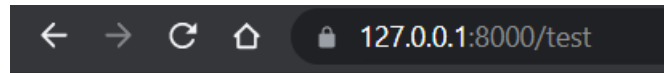
Retourner sur le terminal de VSCode et tapez la commande suivante :

```
td_symfony> php bin/console doctrine:database:create
```

```
Created database `td_symfony_db` for connection named default
PS C:\Users\LAURENT\Workspace\td_symfony> 
```

Nous pouvons nous connecter à notre serveur de bases de données sur phpMyAdmin et vérifier que notre base de données (td\_symfony\_db) a bien été créée. On peut en profiter pour changer l'encodage (→ utf8mb4\_general\_ci)

Relancez ensuite le serveur pour Symfony (symfony server:start) et faites un refresh de la page dans le navigateur. Notre route fonctionne :-)



## TestController index