

### Routing : Documentation officielle

Retrouvez la documentation officielle sur le Routing dans Symfony ici : <https://symfony.com/doc/current/routing.html>

### Route sous forme d'annotation @Route

```
<?php
use Symfony\Component\Routing\Annotation\Route;

class MonContrôleur {
    /**
     * @Route(
     *   "/URL-VOLUME/{param1}/{param2}",
     *   name="nomDeLaRoute",
     *   methods={ "GET", "POST" },
     *   requirements={ "param1" : "regexVoulue", "param2" : "regexVoulue" },
     *   defaults={ "param1" : "ValeurParDefaut" },
     *   host="my.host",
     *   schemes={ "http", "https" }
     *   priority=1
     * )
     */
    public function myMethod($param1, $param2) {
        // ...
    }
}
```

### Les options possibles dans l'annotation @Route

|                                    |                                                                    |
|------------------------------------|--------------------------------------------------------------------|
| name="<routeName>"                 | Donner un nom à une Route                                          |
| methods={"<method1>", "<method2>"} | Restreindre l'accès à une Route sur une ou plusieurs méthodes HTTP |
| host="<host>"                      | Restreindre l'accès à une Route sur un host particulier            |

|                                                     |                                                                 |
|-----------------------------------------------------|-----------------------------------------------------------------|
| requirements={"param1":"regex1", "param2":"regex2"} | Préciser les règles qui s'appliquent aux paramètres de la Route |
| defaults={"param1":"default1", "param2":"default2"} | Préciser les valeurs par défaut des paramètres de la Route      |

Notez que les mêmes options sont disponibles au format YAML

### Les classes importantes de HttpFoundation

|                                           |                                                                         |
|-------------------------------------------|-------------------------------------------------------------------------|
| Symfony\Component\HttpFoundation\Request  | Classe qui représente une requête HTTP reçue sur l'application          |
| Symfony\Component\HttpFoundation\Response | Classe qui représente une réponse HTTP qu'on souhaite renvoyé au client |

### Les classes importantes de HttpFoundation (cont)

|                                                   |                                                            |
|---------------------------------------------------|------------------------------------------------------------|
| Symfony\Component\HttpFoundation\RedirectResponse | Classe qui représente une réponse HTTP de type redirection |
| Symfony\Component\HttpFoundation\JsonResponse     | Classe qui représente une réponse HTTP de type JSON        |

### La classe Request

|                                                                                                                              |                                                                                                                |
|------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| \$request->request                                                                                                           | Equivalent de la super globale \$_POST                                                                         |
| \$request->getQuery()                                                                                                        | Equivalent de la super globale \$_GET                                                                          |
| \$request->attributes                                                                                                        | Représente des données que le Framework a inclus dans la requête : paramètres de l'URL et autres choses utiles |
| \$request->get('nomParam', 'default')                                                                                        | Permet d'obtenir une information dans le \$_GET ou le \$_POST                                                  |
| Vous pouvez vous "faire livrer" la requête HTTP par le framework en la demandant en paramètre d'une méthode liée à une Route |                                                                                                                |

### La classe Response

|                                                                                                  |                                                                                                           |
|--------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| <code>\$response = new Response(&lt;contenu&gt;, &lt;statusCode&gt;, &lt;headers[]&gt;);</code>  | Permet de créer une réponse HTTP classique                                                                |
| <code>\$response = new RedirectResponse(&lt;url&gt;, &lt;headers[]&gt;);</code>                  | Permet de créer une réponse HTTP qui va provoquer une redirection                                         |
| <code>\$response = new JsonResponse(&lt;data&gt;, &lt;statusCode&gt;, &lt;headers[]&gt;);</code> | Permet de créer une réponse JSON                                                                          |
| <code>\$response-&gt;setContent(&lt;contenu&gt;);</code>                                         | Permet de modifier le contenu d'une réponse existante                                                     |
| <code>\$response-&gt;headers</code>                                                              | Représente les en-têtes de la réponse                                                                     |
| <code>\$response-&gt;setStatusCode(&lt;code&gt;)</code>                                          | Permet de modifier le statut HTTP de la réponse                                                           |
| <code>\$response-&gt;send()</code>                                                               | Permet d'envoyer la réponse au navigateur (en principe ce n'est pas à vous de le faire mais au Framework) |

### Controller et AbstractController

#### Qu'est-ce qu'un *controller* au sens strict ?

Un controller est une FONCTION qui va prendre en charge une Requête HTTP et renvoyer une Réponse HTTP adéquate

#### Qu'est-ce qu'une *classe Controller* ?

C'est une classe dont les méthodes sont des *controllers* dans le sens qu'elles sont destinées à répondre à une requête HTTP précise par une réponse adéquate

#### Pourquoi hériter de la classe *AbstractController* ?

La classe **AbstractController** vous est fournie par le Framework et va vous offrir énormément de raccourcis intéressants pour des comportements communs qu'on utilise souvent dans nos controllers

#### Quels sont les *pouvoirs magiques* qu'on peut utiliser dans une fonction controller ?

Chaque méthode liée à une Route peut demander au Framework de lui fournir différents paramètres comme la *Request* mais aussi les paramètres compris dans l'URL (entre autres)

### Commandes utiles de la CLI à ce stade

|                                                                      |                                                                  |
|----------------------------------------------------------------------|------------------------------------------------------------------|
| <code>php bin/console make:controller &lt;NomDuController&gt;</code> | Permet de créer un Controller                                    |
| <code>php bin/console debug:rout</code>                              | Permet de voir la liste des routes existantes dans l'application |