

# Les Controller Symfony

Symfony repose sur le modèle MVC et utilise donc des Controllers. Nous en avons vu les bases dans la partie précédente, mais nous allons revenir dessus dans cette partie pour présenter certaines fonctionnalités.

## Génération d'URLs et Redirections

Dans la partie précédente, nous avons nommé nos routes. C'est ici que cela va devenir utile. En effet on peut utiliser le nom d'une route pour générer une URL de cette route, en remplissant ses paramètres.

Reprenons cet exemple de route paramétrée :

```
□ /**
 * @Route("/blog/{id}", name="blog_show", methods = {"GET"})
 */
public function show(int $id) { ... }
```

On peut donc générer une URL qui correspond à cette route comme ceci :

```
$url = $this->generateUrl('blog_show', ['id' => 10]);
```

On peut également faire des redirections vers des routes :

```
return $this->redirectToRoute('blog_show', ['id' => 10]);
```

Ou vers des URLs :

```
return $this->redirect('http://symfony.com/doc');
```

## Session

On peut également, via un Controller Symfony, accéder à la session et y lire/écrire des données. Pour cela, il suffit de prendre en paramètre de la méthode endpoint un paramètre de type `SessionInterface` qui sera fourni par le framework.

```
□ /**
 * @Route("/auth/login", name="login", methods = {"POST"})
 */
```

```
public function login(SessionInterface $session) { ... }
```

Il s'agit d'un tableau associatif avec le lequel vous pouvez interagir via les méthodes `set` et `get` :

```
□ /**
 * @Route("/auth/login", name="login", methods = {"POST"})
 */
public function login(SessionInterface $session) {
    $session->set('key', 'value');
    $value = $session->get('key');
}
```

## L'objet Request

L'objet Request qui peut être fourni en paramètre d'une méthode endpoint par le framework, permet d'accéder aux données de la requête :

```
public function myEndpoint(Request $request){ ... }
```

Il permet de récupérer des paramètres du Body de la requête (équivalent `$_POST`) :

```
$request->request->get('myParam');
```

Ou des paramètres GET (équivalent `$_GET`) :

```
$request->query->get('myUrlParam');
```

Vous pouvez aussi récupérer un fichier dans le Body de la requête (équivalent `$_FILES`) :

```
$request->files->get('myFile');
```

## Retourner un fichier

Au lieu de retourner un Template rendu, vous pouvez aussi retourner un fichier qui sera téléchargé au client :

```
return $this->file('/chemin/du/fichier.pdf');
```