Créer une méthode de contrôleur Symfony

1. Objectifs

- Etre capable de créer une méthode dans un contrôleur Symfony pour afficher une page d'accueil.

2. Présentation

- Un contrôleur Symfony peut contenir différentes méthodes, chacune correspondant généralement à une action à effectuer dans l'application. Ces méthodes peuvent accepter des paramètres, tels que la requête HTTP, pour accéder aux données envoyées par le client.
- Une méthode de contrôleur peut renvoyer différents types de réponses, selon les besoins de l'application. Voici quelques exemples de types de réponses couramment utilisés :
  - Texte brut
  - JSON
  - Vue HTML ou Twig
  - XML
  - Téléchargement de fichier
  - Génération de PDF ou d'image directement dans le navigateur, etc.

3. Exemples de types de réponses renvoyées par les méthodes de contrôleur Symfony

1. Exemple de rendu de texte brut

- Pour afficher du texte brut dans une méthode de contrôleur Symfony, vous pouvez utiliser la classe Response de Symfony pour retourner une réponse HTTP contenant le texte brut. Voici comment vous pouvez le faire dans votre exemple :

```
<?php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
```

apcpedagogie.com

```
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class UserController extends AbstractController
{
        #[Route("/exemples", name: "exemples")]

    public function exemple(): Response
    {
        return new Response("Hello, Mohamed SALAH");
    }
}
```

Dans cet exemple :

- Nous avons défini une méthode **exemple()** dans le contrôleur **UserController**.
- Nous avons annoté la méthode avec `@Route` pour définir l'URL à laquelle cette méthode doit répondre.
- À l'intérieur de la méthode, nous avons créé une nouvelle instance de Response en passant le texte brut "Hello, Mohamed SALAH" en tant que contenu de la réponse.
- La méthode retourne ensuite cette réponse.
- Maintenant, lorsque vous accédez à l'URL correspondante dans votre navigateur (par exemple, **http://localhost:8000/exemples**), vous verrez le texte "Hello, Mohamed SALAH" affiché à l'écran.

2. **Rendre un fichier de vue twig situé dans le dossier src/template**

   - Pour rendre un fichier de vue Twig situé dans le dossier src/template dans un contrôleur Symfony, vous pouvez utiliser la méthode **`render()`** comme suit :

```
<?php

namespace App\Controller;

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

class UserController extends AbstractController
{
   #[Route("/exemples", name: "exemples")]
public function dashboard()
```

```
    {
        return $this->render('template/dashboard.html.twig');
    }
}
```

3. **Dans cet exemple :**
   - Le contrôleur est dans le namespace **App\Controller**.
   - La méthode **example()** rend le fichier de vue Twig **dashboard.html.twig**.
   - Le chemin du fichier de vue est relatif au dossier templates du bundle actuel. Symfony utilise le dossier templates pour stocker les fichiers de vue par convention. Donc, si votre contrôleur est dans le bundle App, le chemin vers le fichier de vue dans votre exemple devrait être **templates/dashboard.html.twig**.

4. **Rendre le tableau en tant que tableau json ou json**

```
<? php

namespace App\Http\Controllers;
use Illuminate\Http\Request;

class UserController extends Controller
{
    function example()
    {
        // render array as json
        return [1, 2, 3];

        // render json array
        return response()->json([
            "name" => "Sandip Patel"
        ]);
    }
}
```

5. **Rendre un texte avec des en-têtes supplémentaires**

```
<? php
```

apcpedagogie.com

```
namespace App\Http\Controllers;
use Illuminate\Http\Request;

class UserController extends Controller
{
    function example()
    {
        return response('Hello World', 200)
                ->header('Content-Type', 'text/plain');

        return response($content)
            ->header('Content-Type', $type)
            ->header('X-Header-One', 'Header Value1')
            ->header('X-Header-Two', 'Header Value2');
    }
}
```

6. **Rendre un fichier ou un pdf**

```
<? php

namespace App\Http\Controllers;
use Illuminate\Http\Request;

class UserController extends Controller
{
    function example()
    {
        // render a file directly on broswer
        // i.e. image or pdf file
        return response()->file($pathToFile);
    }
}
```

7. **Télécharger le fichier de rendu**

```
<? php
```

apcpedagogie.com

```
namespace App\Http\Controllers;
use Illuminate\Http\Request;

class UserController extends Controller
{
    function example()
    {
        // download a file with given path
        return response()->download($pathToFile);

        // download a file with given path with different name and
        // pass some extra headers along with the new file name
        return response()->download($pathToFile, $name, $headers);

        // delete the file once download completes
        return response()->download($pathToFile)->deleteFileAfterSend();
    }
}
```

8. **Redirection vers des URL internes ou externes**

```
<? php

namespace App\Http\Controllers;
use Illuminate\Http\Request;

class UserController extends Controller
{
    function example()
    {
        // redirect to last page with
        // submitted form data
        return back()->withInput();

        // redirect to user/dashboard page
        return redirect('user/dashboard');

        // redirect with named route
        return redirect()->route('login');

        // redirect with named route by passing variables
```

apcpedagogie.com

```
        return redirect()->route('profile', ['id' => 1]);


        // redirect to specific controller method
        return redirect()->action('HomeController@index');


        // redirect to external url
        return redirect()->away('https://www.google.com');

        // redirect with flash session message
        return redirect('dashboard')->with('success', 'Profile updated!');
    }
}
```

Cours Symfony

Source : https://www.apcpedagogie.com/creer-une-methode-de-controleur-symfony · téléchargé le 05/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com