

Guide complet Symfony - BELGIANPROD

Création du projet Symfony

Étapes :

1. Ouvrir un terminal à l'emplacement désiré.
2. Créer le projet :

```
composer create-project symfony/skeleton BELGIANPROD
```

3. Accéder au dossier du projet :

```
cd BELGIANPROD
```

Création de la base de données

- Ouvrir phpMyAdmin ou tout autre outil MySQL.
- Créer une base : **belgianprod**
- Vérifier que l'utilisateur a les droits sur cette base.

`DATABASE_URL="mysql://root:@127.0.0.1:3306/clubplayer?serverVersion=10.4.32-MariaDB&charset=utf8mb4"`

pour creer la base de donn  dans le terminal
php bin/console doctrine:database:create

Configuration Symfony pour la base

- Ouvrir le fichier `.env`
- Modifier la ligne `DATABASE_URL` :

```
DATABASE_URL="mysql://root:@127.0.0.1:3306/belgianprod"
```

(adaptez user/password et port si n cessaire)

le Nom que aura la base donn 

Cr ation des entit s (tables)

Acteur

```
php bin/console make:entity Acteur
```

- Champs : nom, prenom, dateNaissance, sexe

Categorie et Film

- Créer de la même manière.

Création des tables

```
php bin/console make:migration
php bin/console doctrine:migrations:migrate
```

Création du contrôleur pour ajouter des données

Pour envoyer les données via sur les table via le terminal, il faut créer d'abord une classe dans le dossier controller → DataController Commande: php bin/console make:contro

```
<?php
namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Attribute\Route;
use Doctrine\ORM\EntityManagerInterface;
use App\Entity\Acteur;

final class DataController extends AbstractController
{
    #[Route('/add-acteur', name: 'add_acteur')]
    public function addActeur(EntityManagerInterface $em): Response
    {
        $act = new Acteur();
        $act->setNom("Diallo");
        $act->setPrenom("Alpha");
        $act->setDateNaissance(new \DateTime("2001-10-30"));
        $act->setSexe("M");

        $em->persist($act);
        $em->flush();

        return new Response("Acteur ajouté avec ID : " . $act->getId());
    }
}
```

il faut pas oublier d'ajouter les use pour les classe dont ont a besoin.

il faut toujours mettre la route avant la fonction qui ajpoute

la fonction doit acoir en paremetre un objet de type EntityManagerInterface

pour sauver les donnés dans la memoire

pour les envoyer vers la base de donnés

retour toujours une reposnse; on ajouter un message de reussite

Lancer le serveur Symfony

- Avec PHP intégré :

```
php -S 127.0.0.1:888 -t public
```

- Accéder au navigateur :

le que j'ai defini dans meds configuration

Pour presque tous les cas (Symfony, PHP, Apache, MySQL...
l'adresse locale = 127.0.0.1

<http://localhost:888/add-acteur>

~~<http://127.0.0.1:888/add-acteur>~~

[bin/console doctrine:fixtures:load](#) pour
executer les fixtures

Résumé

[php bin/console crud](#) cree le controller formulaire et template

1. Créer le projet Symfony
2. Créer la base dans phpMyAdmin
3. Configurer `.env`
4. Créer les entités
5. Créer les tables avec migrations
6. Créer un contrôleur avec routes pour ajouter les données
7. Lancer le serveur et appeler les routes dans le navigateur