

Séquence 4



Symfony

OBJECTIF DE LA SÉQUENCE



Cette séquence a pour objectifs de

- Configurer une base de données
- Créer une **Entity** et un **Repository**
- Mettre en œuvre l'architecture MVC



Configuration >>>



Doctrine a besoin de connaître les paramètres de connexion à la DB

Fichier .env



```
DATABASE_URL=mysql://db_user:db_password@127.0.0.1:3306/db_name
```



Création de la DB

 SymfonyCLI

⇒ CMD

symfony console doctrine:database:create



Fichier .env



Création d'une entité

 SymfonyCLI

=> CMD

symfony console make:entity NomEntity



SYMFONY

Entité Etudiant >>>



Example


SymfonyCLI

⇒ CMD

symfony console make:entity Etudiant

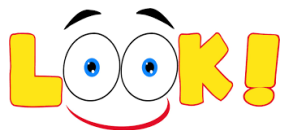


SYMFONY

Fichiers générés >>>



GENERATE



Dossier Entity



Etudiant.php



Dossier Repository



EtudiantRepository.php



annotations

LES MIGRATIONS

Définition >>>

- >>> **Doctrine** permet de créer directement une **table** associée à une **Entity**
- >>> Cette opération s'appelle une **migration**
- >>> **Doctrine** s'appuie sur les **annotations** de l' **Entity** pour générer une **migration**



Création d'une migration


SymfonyCLI

=> CMD

symfony console make:migration

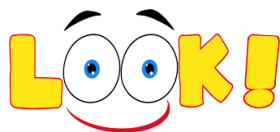
MIGRATION

Création >>>



 SymfonyCLI **=> CMD**

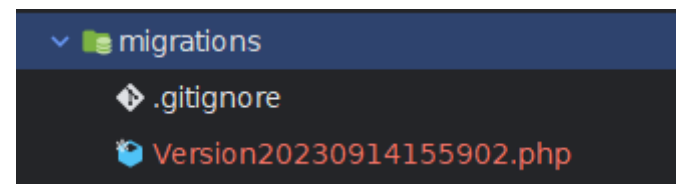
symfony console make:migration



annotations

Etudiant

GENERATE



**Classe avec une
méthode **up()****



Exécution d'une migration


SymfonyCLI

⇒ CMD

symfony console doctrine:migration:migrate

MIGRATION

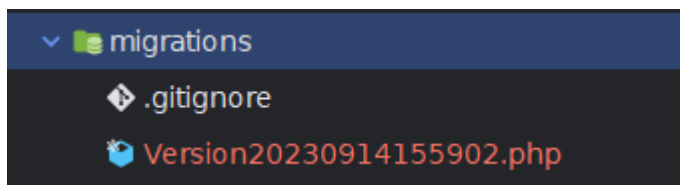
Exécution >>>



Symfony CLI

⇒ CMD

symfony console doctrine:migration:migrate



EXECUTION

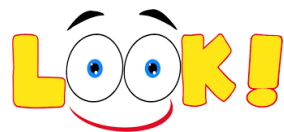
Méthode up()



CREATE TABLE ...

MIGRATION

Exécution >>>



doctrine_migration_versions



Contient toutes les
migrations déjà exécutées



Permet de générer les **tables dans la bases de données en fonction des **Entity** définies dans l'application Symfony**



Permet de garder à jour (synchroniser) le schéma de la base de données avec les classes du domaine (classes **Entity)**



Permet de suivre les changements apportés à la base de données

LES FIXTURES

Définition >>



Création d'un jeu de données dans une table à partir de l'application Symfony



Cette opération s'appelle une fixture

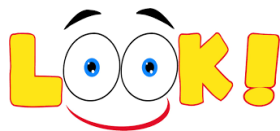
FIXTURES

Installation »»



Installation d'une dépendance avec composer

⇒ CMD `composer require orm-fixtures --dev`



Dossier DataFixtures

GENERATE



Création d'une fixture


SymfonyCLI

⇒ CMD

symfony console make:fixtures NomFixtures

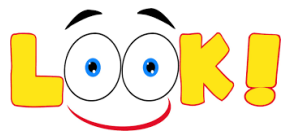
FIXTURES

Fixtures Etudiant »»



Example

⇒ CMD **symfony console make:fixtures EtudiantFixtures**



Dossier DataFixtures



EtudiantFixtures.php

GENERATE

1 Créer le jeu de données "à la main"



2 Utiliser un bundle (package, dependance) permettant de générer un jeu de données "fake"



FIXTURES

Faker »»



Installation d'une dépendance avec composer : faker



Packagist
The PHP Package
Repository



FIXTURES

Faker »»



`$faker = Factory::create('fr_FR');`

```
class EtudiantFixtures extends Fixture
{
    public function load(ObjectManager $manager): void
    {
        $faker = Factory::create('fr_FR');
        for ($i = 0; $i < 100; $i++) {
            $etudiant = new \App\Entity\Etudiant();
            $etudiant->setPrenom($faker->firstName());
            $etudiant->setNom($faker->lastName());
            $etudiant->setEmail($faker->email());
            $etudiant->setDateNaissance($faker->dateTimeBetween('-30 years', '-20 years'));
            $manager->persist($etudiant);
        }

        $manager->flush();
    }
}
```



Exécution d'une fixture


SymfonyCLI

=> CMD

symfony console doctrine:fixtures:load

FIXTURES

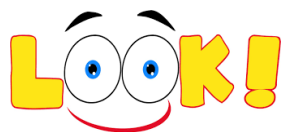
Exécution >>>



 SymfonyCLI

=> CMD

symfony console doctrine:fixtures:load



Dossier DataFixtures



EtudiantFixtures.php



EXECUTION

Méthode load()



INSERT INTO

Intégration MVC

INTÉGRATION MVC

Création d'un contrôleur »»



 Symfony CLI

=> CMD

symfony console make:controller Etudiant

```
class EtudiantController extends AbstractController
{
    #[Route('/etudiants', name: 'app_etudiants')]
    public function index(EtudiantRepository $etudiantRepository): Response
    {
        // Appel du modèle
        $etudiants = $etudiantRepository->findAll();

        // Appel de la vue
        return $this->render('etudiant/index.html.twig', [
            'etudiants' => $etudiants,
        ]);
    }
}
```



INTÉGRATION MVC

Injection de dépendance »»



**Symfony, STP,
"donne-moi" une
instance de type
EtudiantRepository**



**Injection de
dépendance**

```
class EtudiantController extends AbstractController
{
    #[Route('/etudiants', name: 'app_etudiants')]
    public function index(EtudiantRepository $etudiantRepository): Response
    {
        // Appel du modèle
        $etudiants = $etudiantRepository->findAll();

        // Appel de la vue
        return $this->render('etudiant/index.html.twig', [
            'etudiants' => $etudiants,
        ]);
    }
}
```

Injection de dépendance »»

- »» La création d'une instance de type **EtudiantRepository** n'a pas besoin d'être explicite dans le **contrôleur** !
- »» **Symfony** est capable de nous mettre à disposition cette instance de manière "automatique" dans le **contrôleur** sous la forme d'un paramètre



INTÉGRATION MVC

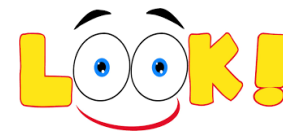
Appel du modèle >>>



```
class EtudiantController extends AbstractController
{
    #[Route('/etudiants', name: 'app_etudiants')]
    public function index(EtudiantRepository $etudiantRepository): Response
    {
        // Appel du modèle
        $etudiants = $etudiantRepository->findAll();

        // Appel de la vue
        return $this->render('etudiant/index.html.twig', [
            'etudiants' => $etudiants,
        ]);
    }
}
```

Le **contrôleur**
appelle la méthode
findAll() du
Repository



Dossier Repository



EtudiantRepository.php

INTÉGRATION MVC

Appel de la vue



```
class EtudiantController extends AbstractController
{
    #[Route('/etudiants', name: 'app_etudiants')]
    public function index(EtudiantRepository $etudiantRepository): Response
    {
        // Appel du modèle
        $etudiants = $etudiantRepository->findAll();

        // Appel de la vue
        return $this->render('etudiant/index.html.twig', [
            'etudiants' => $etudiants,
        ]);
    }
}
```

Le contrôleur

appelle la **vue** et
lui "passe" la liste
des **étudiants**



INTÉGRATION MVC

La vue »»



```
{% extends 'base.html.twig' %}

{% block title %}Etudiants{% endblock %}

{% block body %}
    {# Lister le prénom et le nom de tous les étudiants #}
    <h1>Etudiants</h1>
    <ul>
        {% for etudiant in etudiants %}
            <li>{{ etudiant.prenom }} {{ etudiant.nom }}</li>
        {% endfor %}
    </ul>
{% endblock %}
```

