

Installer composer

(vérifier avec composer -v)

```
symfony new --full mon_projet
```

```
cd mon_projet
```

```
symfony server:start
```

Création d'une base de données DOCTRINE

. Se positionner dans votre projet

. Déclarer les accès BDD ainsi que le nom de la base de données dans .env (ici par exemple animaux).

. Création de la base de données

```
php bin/console doctrine:database:create
```

. Création d'une table

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

création des champs (pas encore dans la BDD)

l'Id est créé automatiquement (pas de setter sur ce champ)

. Migration

Le principe de la migration est de mettre à jour la structure de la BDD si il y a un changement. Ce principe est aussi utilisé la 1ère fois. Un fichier php est généré à chaque migration.

```
php bin/console make:migration
```

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

. Fixtures

Les fixtures permettent de remplir les tables pour faire de tests.

```
composer require orm-fixtures --dev -> creation dossier Fixture
```

```
php bin/console make:fixtures AnimalFixtures
```

modifier AnimalFixtures.php

```
use App\Entity\Animal
```

dans la fonction load() {

```
    $a1 = new Animal();
```

```
    $a1->setNom('cheval')->setDescription("lorem ipsum...") ;
```

```
    $manager->persist($a1);
```

```
    $a2 = new Animal();
```

```
    $a2->setNom('chat')->setDescription("lorem ipsum 2...") ;
```

```

        $manager->persist($a2);

        $a3 = new Animal();
        $a3->setNom('chien')->setDescription("lorem ipsum 2...") ;
        $manager->persist($a3);

        $manager->flush;
    }

```

php bin/console doctrine:fixtures:load

. Je crée mon contrôleur AnimalController

php bin/console make:controller AnimalController

. Je change la route de Animal en / (dans annotations)

dans les annotations

```

/**
 * @Route("/", name="animal")
 */

```

. Je modifie le template correspondant

```

<ul>
    {% for animal in animaux %}
        <li>{{animal.nom}} de la couleur {{animal.color}}</li>
    {% endfor %}
</ul>

```

on peut donner un nom à une route (name dans annotation) et l'utiliser avec path() dans notre template menu avec la commande {{ path('lenom') }}

pour accéder aux données dans le répertoire public utiliser asset('chemin/vers/images/ou/css')

Dans la classe **AnimalController**

y insérer la classe Animal

```
$repository = $this->getDoctrine()->getRepository(Animal::class)
```

Dans Repository il y a des méthodes déjà définies (exemple find.All() et d'autres.)

Après le getRepository

```
$animaux = $repository->findAll()
```