



Symfony Framework Commands

installer **Symfony**

```
wget https://get.symfony.com/cli/installer -O - | bash
```

Check des requis de **Symfony**

```
symfony check:requirements
```

Cree projet **Symfony**

```
symfony new --version=lts --full projectname
```

Lancée son projet sur le web

```
cd my-project/  
symfony server:start
```

Installer un project **Symfony** existant

```
# clone the project  
cd projects/  
git clone ...  
# make composer Install  
cd my-project/  
composer install
```

Voir toutes les Commandes **Symfony**

```
php bin/console
```

Crée un Controller

```
php bin/console make:controller name
```

Crée une Entity

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

Crée une base de donnée **Symfony**

tout d'abord il vous faut crée le crud de vos entity

```
php bin/console make:crud
```

ensuite validee le shéma

```
php bin/console doctrine:schema:validate
```

puis regarder le status de votre mapping

```
php bin/console doctrine:migrations:status
```

si tout est ok crée votre fichier de migration

```
php bin/console make:migrations
```

Et faite votre migration

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

Mettre a jour les fichier css etc..

```
npm run dev
```

Fictures **Symfony**

Installation du composer fixtures

```
composer require --dev doctrine/doctrine-fixtures-bundle
```

Crée une fixtures (déjà crée dans l'installation du composer)

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

chargée une fixtures

```
php bin/console doctrine:fixtures:load
```

Commandes Dangereuse DataBase

Cette commande remet a zero les id
(a ne surtout pas faire sur une machine de prod)

```
php bin/console doctrine:database:drop --force
```

Ensuite recrée la database

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


Et re faire la migrations

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

Les Composers de fictures **Symfony**

Faker : <https://github.com/fzaninotto/Faker#fakerproviderlorem>

HauteLook : <https://github.com/hautelook/AliceBundle>

Création d'un script Automatisation

```
#!/bin/bash
php bin/console do:da:dr --force
php bin/console do:da:cr
php bin/console do:mi:mi --no-interaction
# selon le composer utiliser
php bin/console ha:fi:lo --no-interaction
```

Lancée script phpunit

```
bin/phpunit
```

Les Tests **Symfony**

Installation du composer

```
composer require --dev symfony/phpunit-bridge
```

Création du test

```
php bin/console make:unit-test name
```

Les Liens utile pour les tests

<https://github.com/symfony/panther>

<https://symfony.com/doc/current/testing.html>

Liens VM

<https://docs.docker.com/get-started/>

<https://www.virtualbox.org/>

Update 09/07/2020 Author: Foxy