

# Reverse Engineering d'une base avec Doctrine

Cette manipulation est à faire en **début de projet** pour repartir d'une base existante.

Elle écrase tout ce qui existe et ne doit donc pas être faite en cours de développement.

## Prérequis

Pour que la manipulation fonctionne, on doit au préalable :

- Disposer des outils doctrine :

```
> composer require symfony/orm-pack
> composer require symfony/maker-bundle --dev
```

- Avoir défini les éléments de connexion dans le fichier .env

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

## Import de la structure

Pour importer la structure, on utilisera :

```
php bin/console
doctrine:mapping:import
"App\Entity"
annotation --
path=src/Entity
```

Les entités sont créées dans le dossier défini par l'option **-path** (ici **src/Entity**).

Comme indiqué dans la documentation ([https://symfony.com/doc/current/doctrine/reverse\\_engineering.html](https://symfony.com/doc/current/doctrine/reverse_engineering.html)), cette commande récupère 70 à 80% de la structure, mais pas :

- les associations inverses,
- l'héritage,
- les entités avec des clés étrangères comme clé primaire
- les opérations sémantiques sur les associations telles que *cascade* ou les événements sur le cycle de vie (*on drop, on create, etc*)

## Génération des getters et setters

La structure est importée sans les méthodes d'accès à la base de données.

Il faut donc régénérer l'application entière **ce qui écrase tout l'existant** avec :

```
php bin/console make:entity --regenerate App
```

## Finalisation

L'importation ne génère pas les liens **One-To-Many** qui doivent être redéfinis manuellement.

Toutes les options spécifiques (actions en cascade ou autres) doivent être réécrites dans les classes générées.

From:  
<https://wiki.sio.bts/> - **WIKI SIO : DEPUIS 2017**

Permanent link:  
<https://wiki.sio.bts/doku.php?id=doctrine4>

Last update: **2020/07/29 20:22**

