

Doctrine

- Ce n'est pas sensio Lab
- Comprendre l'utilité d'un ORM
- Savoir créer, modifier, sauvegarder et supprimer des données
- Savoir récupérer des données Le : CRUD

Doctrine

- S'adapte à tous les types de base de données
- Une promesse : Plus de requêtes SQL (ou bien moins ...)
- Mais des objets à manipuler
- Faire abstraction de la base de données

Installation de Doctrine

```
composer req doctrine
```

- Nouveaux dossiers :
 - src/ Entity
 - src/Migrations/
 - src/Repository/
- Configuration :
 - env
 - /config/ doctrine.yaml

Configuration de Doctrine

1. Paramètres de base de données dans .env
2. `php bin/console doctrine:database:create`
3. Modifier l'interclassement dans phpMyAdmin

Les Entités

Une classe PHP représentant les données

Générés des entités :

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

Le shema

- Ensemble des configurations des entités
- Le cœur du modèle
- Pour mettre à jour la base de données par rapport au *schema*:

```
php bin/console doctrine:schema:update --force
```

L' Entity Manager \$em

- EntityManager: gestionnaire d'entités
- Pour les C, U et D du CRUD
- méthodes pour le récupérer depuis un contrôleur :
 - `$em = $this->getDoctrine()->getManager()`
- Un EntityManager pour toutes les entités

Le Repository de Doctrine

- Repository : dépôt
- Pour Le R CRUD
- 2 méthodes pour le récupérer depuis un contrôleur :
 - `$this->getDoctrine()->getRepository(Objet::class)`
 - `$em->getRepository(Objet::class)`
- Un Repository par entité

Les méthodes principales du Repository

```
$repo->findAll()
```

```
$repo->find($id)
```

```
$repo->findOneByTitle($title)
```

```
$repo->findOneBy(["name" => "lorem"])
```

```
$repo->findBy([], ["price" => "DESC"], 30, 0)
```

```
$repo->count()
```