Tutoriel Doctrine

Créer une base de données

- Paramétrage : mettre à jour les informations de la future base (utilisateur, nom de la base, ...) dans le fichier *app/config/parameters.yml*
- Création de la base de données : en mode console, se placer dans le dossier Symfony et exécuter la commande : *php bin/console doctrine:database:create*

Créer une entité

- Notion : une entité permet à l'ORM de manipuler et d'enregistrer vers la base de données l'objet à persister. Une entité correspond à une classe à laquelle sont rajoutées des annotations.
- Démarche : en mode console, se placer dans le dossier Symfony et exécuter la commande *php bin/console doctrine:generate:entity* puis répondre aux questions :

- nom de l'entité : NomBundle:NomEntité (Exemple : GSBCovoiturageBundle:Trajet)
- configuration : choix du format des commentaires de l'entité. On conserve celui par défaut (annotation)
- ajout automatique du premier id (clé primaire)
- définition d'un champ :
  - nom
  - type (string, integer, smallint, bigint, boolean, decimal, date ou datetime, time, text, object, array, float)
  - champ facultatif (Is nullable) ? par défaut non
  - champ unique (Unique) ? par défaut non (true est rare en dehors de la clé primaire)
- création des autres champs puis entrée pour terminer
- Now get to work :-)

Analyser l'entité créée

- Notion : la classe générée représente l'entité avec des informations de mapping. Elle comporte des attributs, des accesseurs, etc. Le rôle de Doctrine est de se charger de la persistance de vos données : on manipule des objets dans l'application et l'ORM s'occupe de les enregistrer dans la base de données.
- Entité : le fichier de l'entité générée se situe dans le dossier *nomBundle/Entity*
- Personnalisation : il est possible d'ajouter des méthodes pour coller à la logique métier. Exemple : *public function __construct() { // Par défaut, la date de l'annonce est la date d'aujourd'hui $this->date = new \Datetime();}*

Matérialiser l'entité en table dans la base de données

- But : Après la création de l'entité, Doctrine va mettre à jour la base de données en générant les tables.
- Génération des tables : en mode console, se placer dans le dossier Symfony et exécuter la commande : *php bin/console doctrine:schema:update --dump-sql* . Cette commande compare l'état actuel de la base de données avec ce qu'elle devrait être en tenant compte des entités. A partir de cette comparaison, elle affiche les requêtes SQL à exécuter pour mettre à jour la base.

- Validation des requêtes : en mode console, se placer dans le dossier Symfony et exécuter la commande : *php bin/console doctrine:schema:update --force* . La requête est exécutée et la base mise à jour.

Modifier une entité

- Création d'un attribut : dans l'entité concernée, ajouter l'attribut complété de son annotation.
- Création des accesseurs : exécuter la commande :
php bin/console doctrine:generate:entities nomBundle:nomEntité .
- Vérification de la requête : exécuter la commande *php bin/console doctrine:schema:update --dump-sql* .
- Mise à jour de la base : exécuter la commande *php bin/console doctrine:schema:update --force* .