

TD n°2



- Projet **td2**
- Application gestion des contacts (Version BDD)

Objectifs

1. Créer des routes, des contrôleurs, utiliser/structurer les templates
2. Utiliser l'injection de dépendances
3. Créer des services
4. Créer des Entities
5. Effectuer les opérations CRUD avec Doctrine

Containtes techniques

Flash messages

Vous utiliserez les [flash messages](#) pour tous les messages à afficher.

Intégrer Doctrine au projet :

```
composer require doctrine maker
```

Création de la base de données

Configurer la connexion à la base dans le fichier **.env** :

```
# customize this line!
```

```
DATABASE_URL="mysql://db_user:db_password@127.0.0.1:3306/contacts"  
  
# to use sqlite:  
# DATABASE_URL="sqlite:///kernel.project_dir%/var/app.db"
```

Créer la base de données **contacts** :

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

Création d'entity et génération de table

Créer la classe **Contact** :

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

Modifier la classe (ajouter les membres et annotations Doctrine)

Exemple de classe annotée :

```
// this use statement is needed for the annotations  
use Doctrine\ORM\Mapping as ORM;  
  
class Product  
{  
    /**  
     * @ORM\Id  
     * @ORM\GeneratedValue  
     * @ORM\Column(type="integer")  
     */  
    private $id;  
  
    /**  
     * @ORM\Column(type="string", length=100)  
     */  
    private $name;  
  
    /**  
     * @ORM\Column(type="decimal", scale=2, nullable=true)  
     */  
    private $price;  
}
```

Générer le script de création de la table :

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

Exécuter le script de migration pour effectuer les maj dans la BDD :

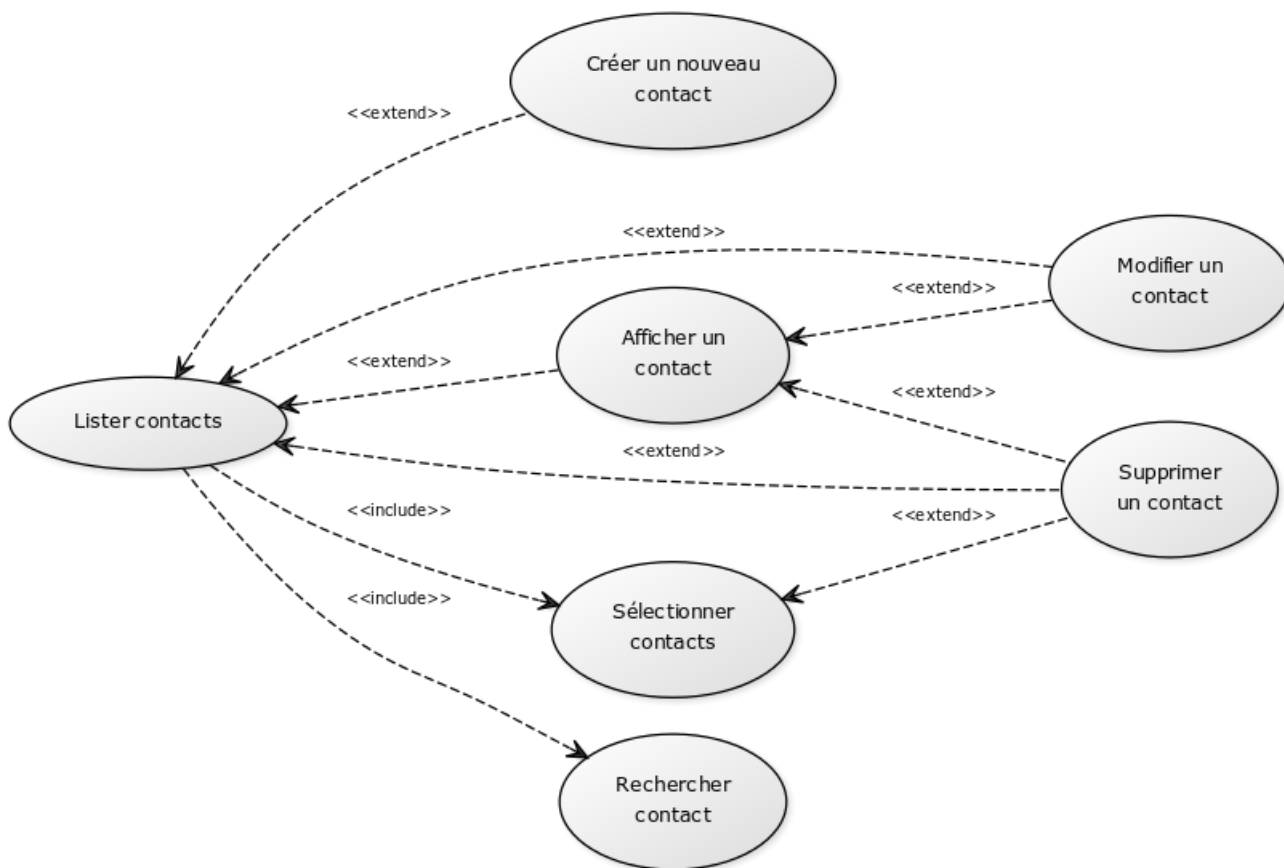
```
php bin/console doctrine:migrations:migrate
```

Réaliser à nouveau les opérations de migration après toute modification de la classe.

Application, fonctionnalités

L'application devra permettre d'effectuer les mêmes opérations de base sur les contacts que dans le précédent projet (votre précédent code - contrôleurs et vues - peut donc être intégralement repris) :

- Lister les contacts
- Rechercher un contact
- Afficher un contact
- Supprimer un contact
- Modifier un contact
- Ajouter un contact
- Sélectionner des contacts



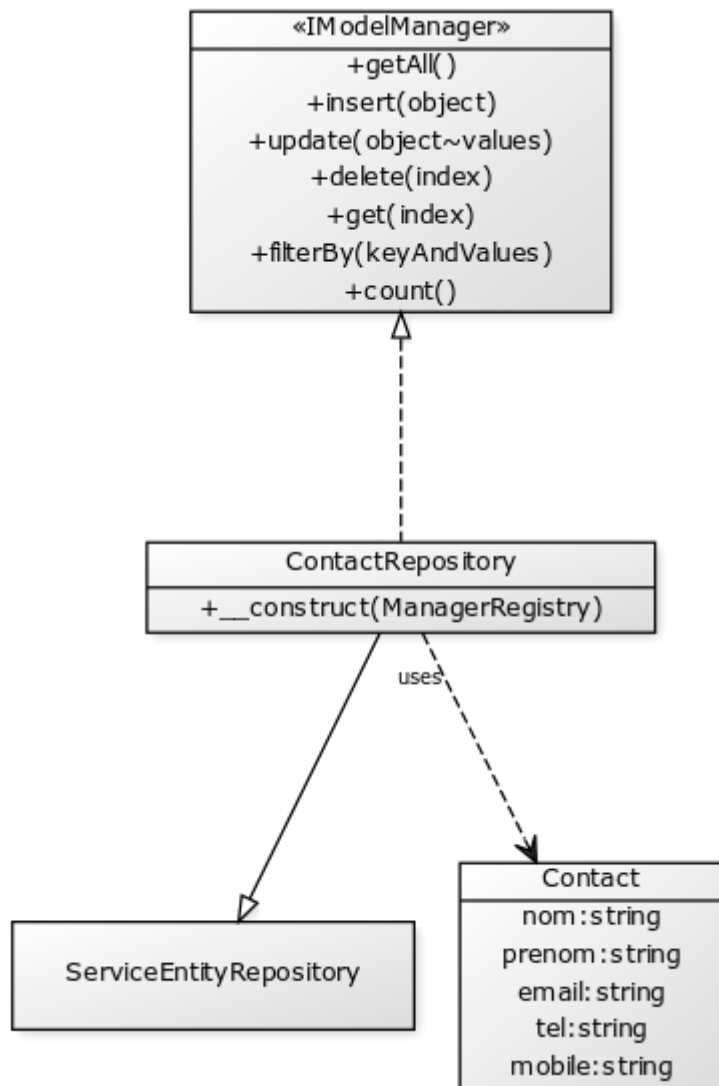
Routes

URL	Méthode	paramètre(s)	Fonctionnalité
/contacts	GET		Liste des contacts
/contact/new	GET		Formulaire de création d'un contact
/contact/edit/1	GET	index:entier	Formulaire de modification du contact d'index 1
/contact/update	POST		Validation du formulaire d'ajout ou de modification
/contact/display/1	GET	index:entier	Affichage du contact d'index 1

URL	Méthode	paramètre(s)	Fonctionnalité
/contacts/search	POST		Recherche et affiche les contacts correspondant au critère
/contacts/select	POST		Sélectionne le ou les contacts dont l'index est posté
/contacts/delete	POST		Supprime des contacts celui ou ceux dont l'index est posté

Classes

Namespace	Classe/interface
App\Entity	Contact
App\Repository	ContactRepository
App\Service	IModelManager



From: <http://slamwiki2.kobject.net/> - SlamWiki 2.1

Permanent link: <http://slamwiki2.kobject.net/framework-web/symfony/td2?rev=1518573170>

Last update: 2019/08/31 14:43

