

## TD n°6



- Projet **boards**
- Application gestion de projets SCRUM

### Objectifs

1. Factorisation du code
2. Réutilisation

### Prise en main

Vous pouvez au choix :

- Partir de votre propre projet et y inclure les éléments du projet Github
- Partir du projet GitHub pour y intégrer vos propres fonctionnalités (recommandé)

A partir du dossier du projet, exécuter :

```
composer update
```

Dans le fichier **.env**, ajuster la valeur de la variable **DATABASE\_URL**.

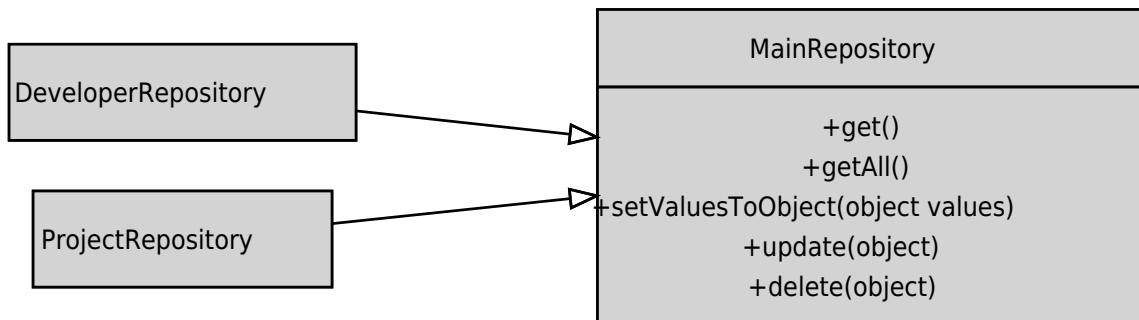
Démarrer le serveur (Mysql) et le serveur http de développement :

```
php bin/console server:run
```

### Factorisation du code

## Doctrine repositories

Les repositories de chacun des models héritent de **MainRepository**, définissant les opérations de base CRUD sur les objets.



CREATED WITH YUML

Chaque Repository héritant de MainRepository a juste à définir dans son constructeur la classe métier auquel il correspond :

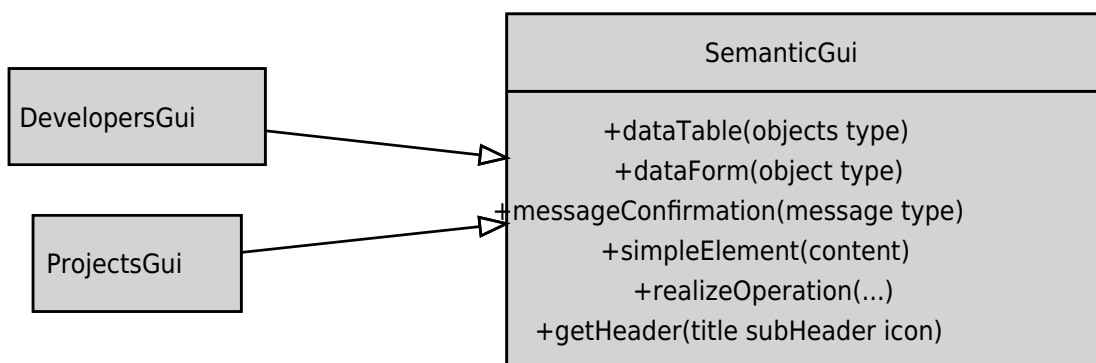
```
<?php
namespace App\Repository;

use Symfony\Bridge\Doctrine\RegistryInterface;
use App\Entity\Developer;

class DeveloperRepository extends MainRepository{
    public function __construct(RegistryInterface $registry){
        parent::__construct($registry, Developer::class);
    }
}
```

## Services Semantic

Les services Semantic de chacun des models héritent de **SemanticGui**, classe définissant les opérations utilisables dans tous les contrôleurs.



CREATED WITH YUML

Chaque Classe Gui dérivée de **SemanticGui** doit logiquement redéfinir les méthodes **dataTable** et **dataForm**.

From:

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

Permanent link:

<http://slamwiki2.kobject.net/framework-web/symfony/td6?rev=1521572265>

Last update: **2019/08/31 14:43**

