

# Opérations CRUD

## Entity manager & Repository

### Entity manager

Récupération de l'entity manager doctrine dans un contrôleur :

```
// you can fetch the EntityManager via $this->getDoctrine()  
// or you can add an argument to your action: index(EntityManagerInterface  
$em)  
$em = $this->getDoctrine()->getManager();
```

Comme indiqué, l'entity manager peut aussi être injecté.

### Repository

```
$productRepository = $this->getDoctrine()->getRepository(Product::class);
```

Pour une bonne séparation du code, il est conseillé de créer des classes Repository pour chacune des classes métier utilisées :

```
namespace App\Repository;  
//...  
class UserRepository extends ServiceEntityRepository {  
  
    public function __construct(ManagerRegistry $registry) {  
        parent::__construct($registry, User::class);  
    }  
  
    //...  
}
```

Il est ensuite possible d'injecter ce Repository pour l'utiliser dans un contrôleur :

```
class UsersController extends Controller{  
    /**  
     * @var UserRepository  
     */  
    private $repository;  
  
    public function __construct(UserRepository $userRepo){
```

```
        $this->repository=$userRepo;  
    }  
}
```

## Read : chargement d'enregistrements

Récupération du Repository :

```
$repository = $this->getDoctrine()->getRepository(Product::class);
```

Chargement d'une instance (par sa clé primaire, généralement 'id')

```
$product = $repository->find($id);
```

Chargement d'une instance sur critères variés :

```
// query for a single Product by name  
$product = $repository->findOneBy(['name' => 'Keyboard']);  
// or find by name and price  
$product = $repository->findOneBy(  
    'name' => 'Keyboard',  
    'price' => 19.99,  
]);
```

Chargement de plusieurs instances

```
// query for multiple Product objects matching the name, ordered by price  
$products = $repository->findBy(  
    ['name' => 'Keyboard'],  
    ['price' => 'ASC']  
);  
  
// find *all* Product objects  
$products = $repository->findAll();
```

From:

<http://slamwiki2.kobject.net/> - **Broken SlamWiki 2.0**

Permanent link:

<http://slamwiki2.kobject.net/framework-web/symfony/models/crud>

Last update: **2019/08/31 14:21**

