

# L'ENTITYMANAGER

Gérer le lien entre entité, et base de données

# SAUVEGARDE / MISE À JOUR

```
<?php

$user = new User();
$user->setName( 'Test' );

$entityManager->persist($user);
$entityManager->flush();
```

---

# SUPPRESSION

```
<?php

$entityManager->remove($user);
$entityManager->flush();
```

# RÉCUPÉRATION VIA ID

```
<?php  
  
$test = $entityManager->find('Test', 1);  
  
// ou  
$test = $entityManager->getRepository('Test')->find(1);
```

## DOCUMENTATION

<http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/reference/working-with-objects.html>

# DOCTRINE QUERY LANGUAGE (DQL)

- Langage de requête de Doctrine
  - SQL sur objets : Manipulation des attributs, pas de tables, uniquement des classes
- 

## EXEMPLE

```
$query = $entityManager->createQuery(  
    'SELECT t FROM Test t WHERE t.text = "coucou"  
');  
  
// table de tous les tests ayant comme text 'coucou'  
$users = $query->getResult();
```

## DOCUMENTATION

<http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/reference/dql-doctrine-query-language.html>

# LE QUERYBUILDER

- Abstraction **objet** de la construction de requêtes DQL
- 

## EXEMPLE

```
<?php
// Récupération du query builder
$qqb = $em->createQueryBuilder();

$qqb
    ->select('u')                // select un alias
    ->from('User', 'u')          // déclaration de la classe cible et alias
    ->where('u.id = :identifiant') // condition
    ->orderBy('u.name', 'ASC')    // tri des résultats
    ->setParameter('identifiant', 100) // set un paramètre
;

return $qqb->getQuery()->getResult(); // retourne le résultat
```

# DOCUMENTATION

<http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/reference/query-builder.html>

# REPOSITORY

- Un repository par classe déclaré dans l'annotation **@Entity**
  - Classe contenant l'ensemble des requêtes DQL d'une entité
  - Par défaut, toutes les entités sont gérées par **Doctrine\ORM\EntityRepository**
- 

## DOCUMENTATION

<http://doctrine-orm.readthedocs.org/projects/doctrine-orm/en/latest/reference/working-with-objects.html#custom-repositories>



# EXEMPLE

```
<?php
/**
 * Déclaration du repository
 * @Entity(repositoryClass="TestRepository")
 */
class Test {}

class TestRepository extends \Doctrine\ORM\EntityRepository
{
    public function getAll()
    {
        return $this
            ->createQueryBuilder('t')
            ->getQuery()
            ->getResult()
        ;
    }
}
```

# USAGE

```
<?php
// On peut appeler ses propres méthodes
$allTests = $entityManager->getRepository('Test')->getAll();
```