**DQL et QueryBuilder**

**1.Le Doctrine Query Language (DQL)**

Le DQL n'est rien d'autre que du SQL adapté à la vision par objets que Doctrine utilise. Il s'agit donc de faire ce qu'on a l'habitude de faire, des requêtes textuelles comme celle-ci par exemple :

**public function** findAllOrdered()

{

$dql = 'SELECT cat FROM AppBundle\Entity\Category cat ORDER BY cat.name DESC';

$query = $this->getEntityManager()->createQuery($dql);

**return** $query->execute();

}

**2.Le QueryBuilder :**

Le QueryBuilder est un moyen plus puissant. Comme son nom l'indique, il sert à construire une requête, par étape.

**public function** getActiveBrands($active)

{

$qb = $this->createQueryBuilder('b');

$qb

->where('b.deleted =:active')

->setParameter('active', $active)

->orderBy('b.id', 'DESC');

;

**return** $qb->getQuery()

->getResult();

}

Exemple2 :

**public function** findByAuthorAndDate($author, $year)

{

$qb = $this->createQueryBuilder('a');

$qb->where('a.author = :author')

->setParameter('author', $author)

->andWhere('a.date < :year')

->setParameter('year', $year)

->orderBy('a.date', 'DESC')

;

**return** $qb

->getQuery()

->getResult()

;

}

**Exemple3 :**

**public function** count()

{

**return** $this->createQueryBuilder('a')

->select('COUNT(a)')

->getQuery()

->getSingleScalarResult();

}