

Abstração de Banco de Dados com Doctrine ORM

@eduardobona

O que podemos tratar
em 15 minutos em
uma mini-talk sobre
Doctrine e Abstração
de Banco de Dados?

Quais os tipos de
desenvolvedores aqui
presentes?

Características

Conhece o Doctrine/ORM

Não conhece

utiliza

Não utiliza

Sabe usar

Não sabe usar

Gosta

Não gosta

Estuda o Doctrine

Não estuda

**Algumas junções
destas características
formam grande parte
dos devs...**

Não conhece
Não sabe usar
Não utiliza

- Entenda ORM
- Use Doctrine
c/ framework
- Use DBAL + ORM
- Repita N vezes

```
use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Entity(repositoryClass="\App...\Repo...\PostRepository")
 * @ORM\Table(name="post")
 */
class Post
{
    /**
     * @ORM\Id
     * @ORM\Column(name="id", type="integer")
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    protected $id;

    /**
     * @ORM\Column(name="title", type="string", length=255)
     */
    protected $title;
}
```

```
/**
 * @ORM\OneToMany(targetEntity="\Application\Entity\Comment",
mappedBy="post")
 * @ORM\JoinColumn(name="id", referencedColumnName="post_id")
 */
protected $comments;
```

```
/**
 * @ORM\ManyToMany(targetEntity="\Application\Entity\Tag",
inversedBy="posts")
 * @ORM\JoinTable(name="post_tag",
 *      joinColumns={@ORM\JoinColumn(name="post_id",
referencedColumnName="id")},
 *      inverseJoinColumns={@ORM\JoinColumn(name="tag_id",
referencedColumnName="id")})
 *      )
 */
protected $tags;
```


Mapeou?

Rode um comando e...

#XABLAU!

Depois crie
Repository para
consultas e Services
para Manipulação

```
use Doctrine\ORM\EntityRepository;

class PostRepository extends EntityRepository
{
    public function findPublishedPosts()
    {
        $entityManager = $this->getEntityManager();

        $queryBuilder = $entityManager->createQueryBuilder();

        $queryBuilder->select('p')
            ->from(Post::class, 'p')
            ->where('p.status = ?1')
            ->orderBy('p.dateCreated', 'DESC')
            ->setParameter('1', Post::STATUS_PUBLISHED);

        return $queryBuilder->getQuery();
    }
}
```

```
public function addNewPost($data)
{
    $post = new Post();
    $post->setTitle($data['title']);
    $post->setContent($data['content']);
    $post->setStatus($data['status']);

    $currentDate = new \DateTime();
    $post->setDateCreated($currentDate);

    $this->entityManager->flush();
}

public function removePost($post)
{
    $this->entityManager->remove($post);
    $this->entityManager->flush();
}
```

Exemplo

Zend + Doctrine + Forms +
Filter + Validation +
Persist + Repository



Quais são
algumas
vantagens?

- DBALess*
- Controle no código
- Multi BD
- Produtividade
- Todas as vantagens de Orientação a
- ~~---~~Objetos

Quais são algumas desvantagens?

- Mais código
- Muito muito annotation
- Memória e Performance
- Todas as desvantagens de OO

E o que está além do uso básico do Doctrine?

- Cache
- Transações
- Partials
- Segurança
- Eventos

E o que está além do uso não tão básico do Doctrine?

- Second Level
Cache
- Performance
- Migrations
- Fixtures

Além de ser tão
completo, afinal,
tem vantagem em
ser Orientado a
Objetos?

- Adicione
interfaces e
abstracts
- **Use traits *****

@eduardobona