Doctrine2 Einführung und Best Practices

Shopware Community Day 2013

Benjamin Eberlei, Qafoo GmbH

8. Juni 2013

Qafoo

passion for software quality

Me

- Working at Qafoo

- Doctrine Developer
- Symfony2 Contributor
- Twitter `@beberlei` and `@qafoo`

Copyright Qafoo GmbH; All Rights Reserved

Outline

The Doctrine Project

Why ORMs?

The ORM in ten steps

Lazy-Loading

Best Practices

Copyright Qafoo GmbH; All Rights Reserved

What comes next?

The Doctrine Project

Copyright Qafoo GmbH; All Rights Reserved

The Doctrine Project

www.doctrine-project.org

The Doctrine Project is the home of a selected set of PHP libraries primarily focused on providing persistence services and related functionality.

Copyright Qafoo GmbH; All Rights Reserved

Projects

- Common
  - Annotations
  - Cache
  - Collections
  - Persistence
- DBAL
- ORM
- ODMs: MongoDB, CouchDB, PHPCR

Copyright Qafoo GmbH; All Rights Reserved

Outline

The Doctrine Project

Why ORMs?

The ORM in ten steps

Lazy-Loading

Best Practices

Copyright Qafoo GmbH; All Rights Reserved

Why ORMs?

Copyright Qafoo GmbH; All Rights Reserved

Why ORMs?

- The Good
  - Allow object-oriented modellling
  - Abstract SQL and prevent injection
  - Standardize Data-Access for developers
- The Bad
  - Inefficient Queries
  - Slower compared to Raw SQL
  - Complex internals
- The Ugly
  - Leaky abstraction

Copyright Qafoo GmbH; All Rights Reserved

Outline

The Doctrine Project

Why ORMs?

The ORM in ten steps

Lazy-Loading

Best Practices

Copyright Qafoo GmbH; All Rights Reserved

The ORM in ten steps

Copyright Qafoo GmbH; All Rights Reserved

Database

```
1  CREATE TABLE Order (id INT AUTO_INCREMENT PRIMARY KEY,
2      orderNumber VARCHAR(255),
3      customer_id INT,
4      created DATETIME
5  );
6
7  CREATE TABLE OrderItems (id INT AUTO_INCREMENT PRIMARY KEY,
8      item_count INT NOT NULL,
9      order_id INT NOT NULL,
10     product_id INT NOT NULL
11 );
12
13 CREATE TABLE Product (id INT AUTO_INCREMENT PRIMARY KEY,
14     label VARCHAR(255),
15     price DECIMAL(10,2)
16 );
17
18 CREATE TABLE Customer (id INT AUTO_INCREMENT PRIMARY KEY,
19     email VARCHAR(255)
20 );
```


Copyright Qafoo GmbH; All Rights Reserved

1. Maps PHP objects to Database tables

```php
<?php
class Order
{
    /** @var int */
    protected $id;

    /** @var string */
    protected $orderNumber;

    /** @var DateTime */
    protected $created;

    public function __construct($orderNumber)
    {
        $this->orderNumber = $orderNumber;
        $this->created = new \DateTime("now");
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

2. Metadata mapping in Docblocks, XML, Yaml

```php
<?php
/** @Entity **/
class Order
{
    /**
     * @Id @GeneratedValue @Column(type="integer")
     * @var int
     */
    protected $id;
    /**
     * @Column(type="string")
     * @var string
     */
    protected $orderNumber;
    /**
     * @Column(type="datetime")
     * @var DateTime
     */
    protected $created;
}
```


Copyright Qafoo GmbH; All Rights Reserved

3. Object references map to foreign keys

```php
<?php
/** @Entity **/
class Order
{
    // ...
    /**
     * @ManyToOne(targetEntity="Customer")
     * @var Customer
     */
    protected $customer;

    public function __construct($orderNumber, Customer $customer)
    {
        $this->orderNumber = $orderNumber;
        $this->created = new \DateTime("now");

        $this->customer = $customer;
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

4. Collections contain many object references

```php
<?php
use Doctrine\Common\Collections\ArrayCollection;

class Order
{
    /**
     * @OneToMany(targetEntity="OrderItem", mappedBy="order")
     */
    protected $orderItems;

    public function __construct($orderNumber, Customer $customer)
    {
        // ...
        $this->orderItems = new ArrayCollection();
    }

    public function addItem(Product $product, $cnt)
    {
        $this->orderItems[] = new OrderItem($this, $product, $cnt);
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

5. Bootstrapping

```php
<?php

use Doctrine\ORM\Tools\Setup;
use Doctrine\ORM\EntityManager;

require_once "vendor/autoload.php";

$dbParams = array(
    'driver'   => 'pdo_mysql',
    'user'     => 'root',
    'password' => '',
    'dbname'   => 'myproject'
);
$path      = 'src/MyProject/Entities';

$config = Setup::createAnnotationMetadataConfiguration($path);
$entityManager = EntityManager::create($dbParams, $config);
```


Copyright Qafoo GmbH; All Rights Reserved

6. EntityManager is notified about objects

```php
<?php
$product1 = new Product("Grill-Sauce", 9.95);
$product2 = new Product("Buch: BBQ fuer Profis", 14.99);

$customer = new Customer("benjamin@qafoo.com");

$order = new Order("1008123", $customer);
$order->addItem($product1, 2);
$order->addItem($product2, 1);

// Notify add entity
$entityManager->persist($customer);
$entityManager->persist($order);

// Notify remove entity
$entityManager->remove($order);
```


Copyright Qafoo GmbH; All Rights Reserved

7. EntityManager#flush() aggregates SQL

```
1  <?php
2
3  $entityManager->flush();
```


Copyright Qafoo GmbH; All Rights Reserved

8. Find object by ID

```php
<?php

$order = $entityManager->find("MyProject\Entities\Order", $id);
```


Copyright Qafoo GmbH; All Rights Reserved

9. Find objects by fields

```php
<?php

$customerRepository = $entityManager->getRepository("MyProject\Entities\Customer");
$customer = $customerRepository->find($customerId);

$orderRepository = $entityManager->getRepository("MyProject\Entities\Order");
$order = $orderRepository->findOneBy(array("orderNumber" => "12308M08234"));

$orders = $orderRepository->findBy(
    array("customer" => $customer),
    array("created" => "DESC")
);
```


Copyright Qafoo GmbH; All Rights Reserved

10. Doctrine Query Language (DQL)

```php
<?php

$dql = "SELECT c AS customer, count(o.id) AS orders
          FROM MyProject\Entities\Customer c
          JOIN c.orders o
      GROUP BY c";
$results = $entityManager->createQuery($dql)->getResult();

foreach ($results as $row) {
    echo $row['customer']->getName() . " (" . $row['orders'] . ")";
}
```


Copyright Qafoo GmbH; All Rights Reserved

Outline

The Doctrine Project

Why ORMs?

The ORM in ten steps

Lazy-Loading

Best Practices

Copyright Qafoo GmbH; All Rights Reserved

What comes next?

Lazy-Loading

Copyright Qafoo GmbH; All Rights Reserved

Lazy-Loading

- Doctrine allows traversal of complete object-graph
- On demand lazy-loading of unloaded objects
- Naive usage of Doctrine will cause N+1 problem
- Solution: Doctrine Query Language or Fetch Strategies

Copyright Qafoo GmbH; All Rights Reserved

Proxies and Lazy-Loading

```php
<?php
$order = $em->find('MyProject\Entities\Order', $myOrderId);

// 'Order::$customer' is a proxy
$me = $order->getCustomer();

// lazy load fired
echo "Customer: " . $me->getEmail();
```


Copyright Qafoo GmbH; All Rights Reserved

Generated Proxy Code

```php
<?php
class CustomerProxy extends Customer implements Proxy
{
    private function _load()
    {
        // lazy loading code
    }
    public function getEmail()
    {
        $this->_load();
        return parent::getEmail();
    }
    // .. other public methods of Customer
}
```


Copyright Qafoo GmbH; All Rights Reserved

Collections and Lazy-Loading

```php
<?php
$order = $entityManager->find('MyProject\Entities\Order',
    $myOrderId);
// 'Order::$items' is PersistentCollection

$items = $order->getItems();
$sum = 0;

// loop fires lazy load of all related Items
foreach ($items AS $item) {
    // product lazy load
    $price = $item->getProduct()->getPrice();

    $sum += $item->getAmount() * $price;
}
```


Copyright Qafoo GmbH; All Rights Reserved

DQL Fetch-Joins to the Rescue

```php
<?php
$dql = "SELECT o, oi, c
          FROM MyProject\Entities\Order o
          JOIN o.orderItems oi
          JOIN o.customer c
         WHERE o.id = ?1";
$query = $entityManager->createQuery($dql);
$query->setParameter(1, $myOrderId);

$order = $query->getSingleResult();
```

Qafoo
passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

Outline

The Doctrine Project

Why ORMs?

The ORM in ten steps

Lazy-Loading

Best Practices

Copyright Qafoo GmbH; All Rights Reserved

Best Practices

Copyright Qafoo GmbH; All Rights Reserved

Entities

- Applying the Single Responsibility Principle (SRP)
- Business logic in Plain PHP Objects
- Interaction through object references
- No direct database access
- “persistence-ignorance”

Copyright Qafoo GmbH; All Rights Reserved

Repositories - Encapsulation of Queries

```
1  $orders = $entityManager->getRepository("MyProject\Entities\Order")
2                  ->findOneBy(array("customer" => $customer));
```

Qafoo
passion for software quality
Copyright Qafoo GmbH; All Rights Reserved

Repositories - Introduce Interface

```
1  use Doctrine\Common\Persistence\ObjectRepository;
2
3  interface OrderRepository extends ObjectRepository
4  {
5      public function findByCustomer(Customer $customer);
6  }
```


Copyright Qafoo GmbH; All Rights Reserved

Repositories - Implement Interface

```
1  use Doctrine\ORM\EntityRepository;
2
3  class OrderOrmRepository extends EntityRepository
4                              implements OrderRepository
5  {
6      public function findByCustomer(Customer $customer)
7      {
8          return $this->findOneBy(array("cusotmer" => $customer));
9      }
10 }
```


Copyright Qafoo GmbH; All Rights Reserved

Repositories

```
1  $orders = $entityManager
2    ->getRepository("MyProject\Entities\Order")
3    ->findByCustomer($customer)
4  ;
```


Copyright Qafoo GmbH; All Rights Reserved

Query + QueryBuilder

- Complex API
- Hardly usable in Unit-Tests
- Solution: Encapsulate in Repositories

Copyright Qafoo GmbH; All Rights Reserved

Query + QueryBuilder

```
1  class OrderOrmRepository extends EntityRepository
2  {
3      public function findRecent()
4      {
5          $query = $this->getEntityManager()->createQuery(
6              "SELECT o, oi
7                 FROM MyProject\Entities\Order o
8                 JOIN o.orderItems oi
9             ORDER BY o.created DESC"
10         );
11         return $query->setMaxResults(10)
12                      ->getResult();
13     }
14 }
```


Copyright Qafoo GmbH; All Rights Reserved

Manager

- Three responsibilities
  - Query API
  - INSERT, UPDATE, DELETE
  - Transaction Management
- Split them between different ”layers”

Copyright Qafoo GmbH; All Rights Reserved

Service Layer

```
1  use Doctrine\Common\Persistence\ObjectManager;
2
3  class OrderService
4  {
5      private $orderRepository;
6      private $customerRepository;
7      private $productRepository;
8
9      public function __construct($orderRepository,
           $customerRepository, $productRepository)
10     {
11         $this->orderRepository = $orderRepository;
12         $this->customerRepository = $customerRepository;
13         $this->productRepository = $productRepository;
14     }
15 }
```


Copyright Qafoo GmbH; All Rights Reserved

Service Layer

```php
class OrderService
{
    public function create($customerId, array $products)
    {
        $customer = $this->customerRepository->find($customerId);
        $orderNumber = $this->orderRepository->generateNext();

        $order = new Order($orderNumber, $customer);

        foreach ($products as $productId => $amount) {
            $product = $this->productRepository->find($productId);
            $order->addItem($product, $amount);
        }

        $this->orderRepository->add($order);

        return $orderNumber;
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

Controller

```
1  class OrderController
2  {
3      public function createAction($request)
4      {
5          // ...
6          $orderNumber = $orderService->create(
7              $request->get('customerId'),
8              $request->get('products')
9          );
10         $entityManager->flush();
11
12         return $this->redirect('order', array(
13             'number' => $orderNumber)
14         );
15     }
16 }
```


Copyright Qafoo GmbH; All Rights Reserved

Questions?

Stay in touch

- Benjamin Eberlei
- benjamin@qafoo.com
- @beberlei / @qafoo

Copyright Qafoo GmbH; All Rights Reserved

THANK YOU

Rent a quality expert
qafoo.com

Copyright Qafoo GmbH; All Rights Reserved