

thinkphp3.2.3 Doctrine\ORM

参考文章: thinkphp5 Doctrine\ORM

参考文章地址: <http://www.thinkphp.cn/extend/1066.html>

1, 安装Doctrine

1. composer `require doctrine/orm`

2, 初始化配置Doctrine

(1) cli-config.php

```
1. <?php
2. // E:/wamp/www/tp3.2.3/cli-config.php
3. use Doctrine\ORM\Tools\Setup;
4. use Doctrine\ORM\EntityManager;
5. require_once 'vendor/autoload.php';
6. $config = require_once 'Application/Common/Conf/config.php';
7. $isDevMoe = true;
8.
$configuration = Setup::createAnnotationMetadataConfiguration(array(__DIR__
__ . '/Application/Entity'), $isDevMoe);
9. $conn = array(
10.     'driver' => 'pdo_mysql',
11.     'user'   => $config['DB_USER'] ? $config['DB_USER'] : 'root',
12.     'password' => $config['DB_PWD'] ? $config['DB_PWD'] : '',
13.     'dbname' => $config['DB_NAME'] ? $config['DB_NAME'] : 'symfony',
14.     'port' => $config['DB_PORT'] ? $config['DB_PORT'] : 3306,
15.     'charset' => 'utf8'
16. );
17. $entityManager = EntityManager::create($conn, $configuration);
18.
return \Doctrine\ORM\Tools\Console\ConsoleRunner::createHelperSet($entit
yManager);
```

(2)Entity.php

```
1. <?php
2. // E:/wamp/www/tp3.2.3/ThinkPHP/Library/Vendor/Entity/
3.
4. use Doctrine\ORM\Tools\Setup;
5. use Doctrine\ORM\EntityManager;
6.
7. class Entity
8. {
9.     public static function getEntityManager()
10.    {
11.        $isDevMoe = true;
12.        $config =
Setup::createAnnotationMetadataConfiguration(array(APP_PATH . '/Entity'),
$isDevMoe);
13.        $conn = array(
14.            'driver' => 'pdo_mysql',
15.            'user'   => 'user' => C('DB_USER') ? C('DB_USER') : ' root',
16.            'password' => 'password' => C('DB_PWD') ? C('DB_PWD') : '',
17.            'dbname' => 'dbname' => C('DB_NAME') ? C('DB_NAME') :
'thinkphp50',
18.            'port' => C('DB_PORT') ? C('DB_PORT') : 3306,
19.            'charset' => 'utf8'
20.        );
21.        return EntityManager::create($conn, $config);
22.    }
23. }
```

定义Entity信息

(1) Product.php

```
1. <?php
2. // .../Application/Entity/Product.php
3. namespace Application\Entity;
4. /**
```

```
5. * Class Product
6. * @package Application\Entity
7. *
8. * @Entity
9. * @Table(name="products")
10. */
11. class Product
12. {
13.     /**
14.      * @var int
15.      *
16.      * @Id
17.      * @Column(type="integer")
18.      * @GeneratedValue
19.      */
20.     protected $id;
21.     /**
22.      * @var string
23.      *
24.      * @Column(type="string")
25.      */
26.     protected $name;
27.     /**
28.      * @param mixed $id
29.      */
30.     public function setId($id)
31.     {
32.         $this->id = $id;
33.     }
34.     /**
35.      * @param mixed $name
36.      */
37.     public function setName($name)
38.     {
39.         $this->name = $name;
40.     }
41.     /**
42.      * @return mixed
```

```
43. */
44. public function getId()
45. {
46.     return $this->id;
47. }
48. /**
49.  * @return mixed
50. */
51. public function getName()
52. {
53.     return $this->name;
54. }
55. }
```

(2) User.php

```
1. <?php
2. namespace Application\Entity;
3. /**
4.  * Class User
5.  * @package Application\Entity
6.  *
7.  * @Entity
8.  * @Table(name="users")
9. */
10. class User
11. {
12.     /**
13.      * @var int
14.      *
15.      * @Id
16.      * @Column(type="integer")
17.      * @GeneratedValue
18.      */
19.     protected $id;
20.     /**
21.      * @var string
```

```

22.  *
23.  * @Column(type="string")
24.  */
25.  protected $username;
26.  /**
27.  * @return int
28.  */
29.  public function getId()
30.  {
31.      return $this->id;
32.  }
33.  /**
34.  * @param int $id
35.  */
36.  public function setId($id)
37.  {
38.      $this->id = $id;
39.  }
40.  /**
41.  * @return string
42.  */
43.  public function getUsername()
44.  {
45.      return $this->username;
46.  }
47.  /**
48.  * @param string $username
49.  */
50.  public function setUsername($username)
51.  {
52.      $this->username = $username;
53.  }
54. }

```

4, 生成数据表, 不用直接操作数据库, 执行该命令后, 会自动在数据库生成 products表和相关定义的字段id, name及字段属性

1. vendor/bin/doctrine orm:schema-tool:update --force --dump-sql

```
D:\TEST\start\codes>vendor\bin\doctrine orm:schema-tool:update --force --dump-sql

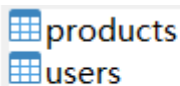
The following SQL statements will be executed:

CREATE TABLE products (id INT AUTO_INCREMENT NOT NULL, name VARCHAR(255) NOT NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci ENGINE = InnoDB;
CREATE TABLE users (id INT NOT NULL, username VARCHAR(255) NOT NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci ENGINE = InnoDB;

Updating database schema...

2 queries were executed

[OK] Database schema updated successfully!
```



5, 插入数据

```
1. <?php
2. namespace Home\Controller;
3.
4. use Think\Controller;
5.
6. class IndexController extends Controller
7. {
8.     public function index()
9.     {
10.         return 'Hello world';
11.     }
12.     public function create()
13.     {
14.         require_once APP_PATH . 'Entity/Product.php';
15.
16.         $product = new \Application\Entity\Product();
17.         $product->setName('北京油鸡');
18.
19.         vendor('Entity.Entity');
20.         $entityManager = \Entity\Entity::getEntityManager();
21.         $entityManager->persist($product);
22.         $entityManager->flush();
23.     }
}
```

```
24.     $this->show('Created Product with ID ' . $product->getId());  
25. }  
26. }
```