



jackyrong

浏览: 156038 次

性别:

来自: 广州

[详细资料](#)[留言簿](#)

搜索本博客

最近访客 [>>更多访客](#)[hzh0725](#)[wojingwu](#)[mmBlue](#)[ietong](#)

博客分类

- [全部博客 \(876\)](#)
- [软件工程 \(36\)](#)
- [JAVA 相关 \(140\)](#)
- [ajax/web 相关 \(36\)](#)
- [数据库相关/oracle \(67\)](#)
- [PHP \(44\)](#)
- [UNIX/LINUX/FREEBSD/solaris \(33\)](#)
- [音乐探讨 \(1\)](#)
- [闲话 \(5\)](#)
- [网络安全等 \(6\)](#)
- [.NET \(140\)](#)
- [ROR和GOG \(3\)](#)
- [\[网站分类\]4.其他技术区 \(176\)](#)

2007-03-08

[九型人格,工作中的好帮手](#) | [很高兴能入住博客园](#)

[php里ezpdo orm框架初探](#)

java里有hibernate,php有啥,这一直困扰着大家.最近发现了一个还可以的php的orm框架 ezpdo, 网站是<http://www.ezpdo.net>,有手册和帮助文件下载,今天看了下,现将其中一篇导学文大致翻译出来 (<http://www.ezpdo.net/blog/2005/03/03/tutorial/>),我是自己的话写出来的,不大喜欢一句句翻译,供参考

首先,当然是下载这玩意了,下载完之后,解压缩到一个目录,比如起名为ezpdo..要注意的是,这个框架必须在mysql 4.1以上(支持主外键)和php 5.04以上的(PHP版本5以上,越新越好),注意,在下面的例子中,是用mysql做数据库的,原文是用sqlite的,所以PHP5里,要配置好mysql,我今天试的是mysql5的

我们讲解下examples\books这个项目,.在books子目录下,有两个子目录,分别是classes目录和compiled目录,其中classes子目录是放类文件的,compiled目录是放ezpdo自己生成的文件,要求该目录要有写权限,而在每个项目下,比如books的根目录下,有一个配置文件config.inc,里面是ezpdo的一些配置项,其中我们要注意的是

```
; The default DSN to the database you want to store your objects
; This will be the default DSN for classes that do not have DSN specified
;default_dsn = sqlite://books.db
```

```
default_dsn = mysql://root:123456@localhost:3309/books
```

```
; default_dsn = pgsql://ezpdo_ex:pdoiseasy@localhost/ezpdo_ex
```

这里,我们用default_dsn指定了数据库连接串了,这里用的是MYSQL.,并且我们先建立了一个空的数据库 books.

接下来我们设计实体类.这里以作者和其著作来说明,一个作者可以写多本书,一本书可以由多个作者合写,构成典型的多对多关系了,先看一个基类

```
class Base {

    /**
     * tracking id (used by bookstore)
     * @var string
     * @orm char(64)
     */
    public $trackId;

    /**
     * Constructor
     */
    public function __construct() {
        $this->trackId = uniqid('track-');
    }
}
```

这里没啥的,构造一个序列号而已,留意这里的注释了, @orm char(64),说明映射到数据库是string类型,64位的,这有点象JAVA里JPA最新的annoatation的标记了,十分方便

接着是Author 类

```
class Author extends Base {

    /**
     * Name of the author
     * @var string
     * @orm char(64)
     */
    public $name;

    /**
     * Books written by the author
```

- [算法等 \(3\)](#)
- [\[随笔分类\]SOA \(6\)](#)
- [收藏区 \(71\)](#)
- [金融证券 \(4\)](#)
- [\[网站分类\]5.企业信息化 \(1\)](#)
- [c&c++学习 \(1\)](#)
- [读书区 \(7\)](#)
- [其它 \(4\)](#)
- [收藏夹 \(1\)](#)
- [设计模式 \(1\)](#)
- [FLEX \(10\)](#)
- [Android \(1\)](#)

我的留言簿 [>>更多留言](#)

- 好厉害
-- by [qddy7217286](#)
- 不是，用UC-CENTER-JAVA
API这个做的，但有点问题
-- by [jackyrong](#)
- 请问您在整合Java与Ucenter的
时候是通过CAS来做的吗？请
教问题
-- by [chris_zcl](#)

其他分类

- [我的收藏 \(83\)](#)
- [我的论坛主题贴 \(4\)](#)
- [我的所有论坛贴 \(30\)](#)
- [我的精华良好贴 \(0\)](#)

最近加入圈子

- [作书译书圈子](#)
- [广州JavaEyer饭局群](#)

存档

- [2010-07 \(40\)](#)
- [2010-06 \(24\)](#)
- [2010-05 \(25\)](#)
- [更多存档...](#)

最新评论

- [一个日志分析工具splunk](#)
每天才500m啊，处理量太小了
-- by [juliashine](#)
- [mysql的show index和opti...](#)
whitesock 写道>Cardinality >索

```
* @var array of Book
* @orm has many Book
*/
public $books = array();

/**
 * Constructor
 * @param string $name author name
 */
public function __construct($name = "") {
    parent::__construct();
    $this->name = $name;
}

// the rest of the code in the class omitted...
}
```

要留意的是\$books是一个数组,因为一个作者有多本著作,而@orm has many Book则表明,一个作者类关联到多个book类,而其中的关联等都不用我们搞了,也不象hibernate那样去设置.hbm配置文件了.

再来看book类

```
class Book extends Base {

/**
 * Bool title
 * @var string
 * @orm title char(80)
 */
public $title;

/**
 * Number of pages
 * @var integer
 * @orm integer
 */
public $pages = -1;

/**
 * Book author (assuming many co-authors)
 * @var Author
 * @orm has many Author
 */
public $authors = array();

/**
 * Constructor
 * @param string
 */
public function __construct($title = "") {
    parent::__construct();
    $this->title = $title;
}

// the rest of the code in the class omitted...
}
```

可以看到,book类和author类差不多,也通过@orm has many Author类进行设置了关联

好了，接下来我们可以开始设置一个add.php的文件里,用来给数据库增加数据,在目录里是add.php文件
首先要把ezpdo的类库API包含进来

```
include_once(dirname(__FILE__) . '/../..../ezpdo_runtime.php');
```

// get the persistence manager (a singleton)

```
$m = epManager::instance();
```

引中唯一值的数目 ...

-- by [jackyrong](#)

- [mysql的show index和opti...](#)

>Cardinality>索引中唯一值的数目的估计值。通过运行ANAL ...

-- by [whitesock](#)

- [一个日志分析工具splunk](#)

plantegg 写道这样的话也可以直接某度桌面搜索吧？ 什么意思？

-- by [jackyrong](#)

- [一个日志分析工具splunk](#)

这样的话也可以直接某度桌面搜索吧？

-- by [plantegg](#)

评论排行榜

- [如何能调动团队的积极性](#)
- [如何带平均年龄小的团队](#)
- [团队中不能只有梅西呀](#)
- [惊闻广州政协做调查为广州电视台取消粤语节...](#)
- [一个日志分析工具splunk](#)



[\[什么是RSS?\]](#)

这里获得持久管理器类的实例了,有点象hibernate的sessionfactory,接着我们创建作者类的实例了

```
// create authors
```

```
$a1 = $m->create('Author');  
$a1->name = 'Erich Gamma';
```

```
$a2 = $m->create('Author');  
$a2->name = 'Richard Helm';
```

```
$a3 = $m->create('Author');  
$a3->name = 'Ralph Johnson';
```

```
$a4 = $m->create('Author');  
$a4->name = 'John Vlissides';
```

我们创建了4个作者了

如果你习惯了java的getter/setter,也可以这样

```
// setter
```

```
$a1->setName('Erich Gamma');
```

```
// getter
```

```
echo $a1->getName();
```

但是这不是必要的,因为在实体类里,你可以不必写getter/setter,ezpdoZ会帮你自动搞好,这点比hibernate要强些哦,

接下来是书本类

```
// create books
```

```
$b1 = $m->create('Book');  
$b1->title = 'Design Patterns';  
$b1->pages = 395;
```

```
$b2 = $m->create('Book');  
$b2->title = 'Contributing to Eclipse: Principles, Patterns, and Plugins';  
$b2->pages = 320;
```

最后是把Author类和Books类关联起来

```
// add authors to books
```

```
$b1->authors = array($a1, $a2, $a3, $a4);  
$b2->authors = array($a1);  
$b3->authors = array($a2);  
$b4->authors = array($a3);  
$b5->authors = array($a4);
```

```
// add books to authors
```

```
$a1->books = array($b1, $b2);  
$a2->books = array($b1, $b3);  
$a3->books = array($b1, $b4);  
$a4->books = array($b1, $b5);
```

最后,提交到数据库

`$m->flush();`(要注意的是,如果config.inc里设置了auto_flush,则不用写这句话了)

如果我们运行add.php后,再运行print.php,可以看到数据库中的确建立了3个表,一个是books表, 一个是author表, 另一个是自动生成的用来做多对多的关联表了。

要把对象状态提取出来, 也很容易

```
// get the persistence manager  
$m = epManager::instance();
```

```
// get all authors and books  
$authors = $m->get('Author');  
$books = $m->get('Book');
```

查找对象也很容易,比如找Eric Gamma写的说,如下

```
// get the persistence manager  
$m = epManager::instance();
```

```
// create the example object
```

```
$ea = $m->create('Author');
```

```
// set name to search
$ea->name = 'Erich Gamma';

// null variable is ignored in searching
// !!!important if the class constructor set non-null values!!!
$ea->trackId = null;
$ea->books = null;

// use the example object to find
if (!$as = $m->find($ea)) {
    echo "Cannot find author [" . $ea->name . "]\n";
    exit();
}

// go through each author and print
foreach($as as $a) {
    echo $a; echo "\n";
}

甚至可以用象HIBERNATE中的HQL语句写,这里成了EZPDO SQL了,呵呵
// use EZOQL to find objects
$as = $m->query("from Author as author where author.name = 'Erich Gamma'");
if (!$as) {
    echo "Cannot find author [Erich Gamma]\n";
    exit();
}

// ...
这次先说这么多，接下来继续研究
```

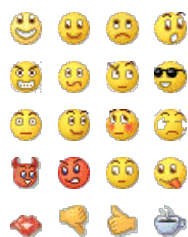
[九型人格,工作中的好帮手](#) | [很高兴能入住博客园](#)

15:13 | [浏览 \(166\)](#) | [评论 \(0\)](#) | 分类: [\[网站分类\]4.其他技术区](#) | [相关推荐](#)

评论

发表评论

表情图标



字体颜色:

字体大小:

对齐:

提示: 选择您需要装饰的文字, 按上列按钮即可添加上相应的标签

您还没有登录，请[登录](#)后发表评论(快捷键 **Alt+S** / **Ctrl+Enter**)

声明：JavaEye 文章版权属于作者，受法律保护。没有作者书面许可不得转载。若作者同意转载，必须以超链接形式标明文章原始出处和作者。

© 2003-2010 JavaEye.com. All rights reserved. 上海炯耐计算机软件有限公司 [沪ICP备05023328号]