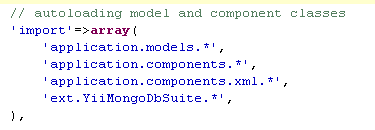
**使用yii增删改查mongodb**

**文档参考：**<http://canni.github.com/YiiMongoDbSuite/>

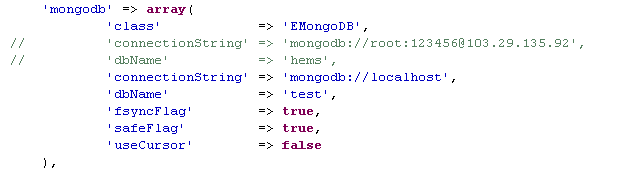
**Mongodb是非关系型数据库**

**Mian配置：**

1. 导入YiiMongoDBSuite支持包

****

1. 配置mongodb数据库

****

**Model定义方式：**

1. 静态Model方法,获得数据表实例
2. 模型名称必须继承EMongoDocument。
3. 定义表中字段,在新增数据时会在表中自动创建字段名称。
4. getCollectionName方法，定义表名。
5. rules方法，对表中字段进行验证
6. attributeNames，定义表中字段的名称，在页面可以通过label获得。

<?php

**class** Article **extends** EMongoDocument {

**public** $title;

**public** $content;

**public** $sortnum;

**public static function** *model*($className=\_\_CLASS\_\_) {

**return parent**::*model*($className);

}

**public function** getCollectionName() {

**return** 'article';

}

**public function** rules() {

**return array**(

**array**('title, content,sortnum', 'safe'),

);

}

**public function** attributeLabels() {

**return array**(

'title' => '文章标题',

'content' => '文章内容',

'sortnum' => '排序号'

);

}

}

**Mongodb增删改查：**

**EMongoModifier：mongodb修改类，用于“set”参数**

**EMongoCriteria：mongodb条件类，用于“where”条件语句**

1. **增**

**public function** actionAdd() {

$article = **new** Article();

$article->title = 'php MongoDB 增删改查';

$article->content = 'php MongoDB 增删改查 内容为300个字';

$article->sortnum = 1;

$result = $article->save();

**if**($result) {

**echo** '添加成功';

} **else** {

**echo** '添加失败';

}

}

**public function** actionAddAll() {

**for**($i=1; $i<=19; $i++) {

$article = **new** Article();

$article->title = 'php MongoDB 增删改查'.'第'.$i.'篇';

$article->content = 'php MongoDB 增删改查 内容为300'.$i.'个字';

$article->sortnum = $i;

$result = $$article->save();

}

**if**($result) {

**echo** '添加成功';

} **else** {

**echo** '添加失败';

}

}

1. **删**

**public function** actionDelete() {

$criteria = **new** EMongoCriteria();

$criteria->addCond("\_id", "==", "c4ca4238a0b923820dcc509a6f75849b");

$result = Article::*model*()->deleteAll($criteria);

**if**($result) {

**echo** '删除成功';

} **else** {

**echo** '删除失败';

}

}

1. **改**

**public function** actionUpdate() {

$mondifier = **new** EMongoModifier();

$mondifier->addModifier("title", "set", "修改后的文章标题");

$mondifier->addModifier("content", "set", "修改后的文章内容");

$mondifier->addModifier("sortnum", "set", "100");

$criterida = **new** EMongoCriteria();

$criterida->addCond("\_id", "==", "c4ca4238a0b923820dcc509a6f75849b");

$result = Article::*model*()->updateAll($mondifier, $criterida);

**if**($result['updatedExisting']) {

**echo** '修改成功';

} **else** {

**echo** '修改失败';

}

}

**public function** actionUpdateAll() {

$articles = Article::*model*()->findAll();

$count = 1;

**foreach** ($articles **as** $article) {

$modifier = **new** EMongoModifier();

$modifier->addModifier("sortnum", "set", $count);

$count++;

$criteria = **new** EMongoCriteria();

$criteria->addCond("sortnum", "==", $article->phonenum);

$result = Article::*model*()->updateAll($modifier, $criteria);

**if**($result['updatedExisting']) {

**echo** '修改成功';

} **else** {

**echo** '修改失败';

}

}

}

1. **查：**

**public function** actionFind() {

$criteri = **new** EMongoCriteria();

$criteri->addCond("sortnum", "==", "1");

$result = Article::*model*()->find($criteri);

**if**(**isset**($result)) {

**echo** $result->title.'被查出来了';

} **else** {

**echo** '查询失败';

}

}

**public function** actionFindAll() {

$articles = Article::*model*()->findAll();

**foreach** ($articles **as** $article) {

**echo** '文章标题：'.$article->title.'<br/>';

}

}

**public function** actionFindAllBySort() {

$criteria = **new** EMongoCriteria();

$criteria->sort('sortnum', EMongoCriteria::*SORT\_DESC*);

// $criteria->sort('sortnum', EMongoCriteria::SORT\_ASC);

$articles = Article::*model*()->findAll($criteria);

**foreach** ($articles **as** $article) {

**echo** '文章标题：'.$article->title.'<br/>';

}

}

**public function** actionFindAllByPaging() {

$totalCount = Article::*model*()->count();

$offsetnum = 0;

$limitnum = 10;

$totalPage = $this->getTotalPage($totalCount, $limitnum);

$count = 1;

**for**($i=0; $i<$totalPage; $i++) {

$criteria = **new** EMongoCriteria();

$criteria->offset($limitnum\*$i + $offsetnum);

$criteria->limit($limitnum);

$articles = Article::*model*()->findAll($criteria);

**foreach** ($articles **as** $article) {

**echo** $count.'用户名：'.$article->title.'<br/>';

}

$count++;

}

}

**public function** getTotalPage($totalCount, $limitnum) {

**if**($totalCount/$limitnum>0) {

**if**($totalCount%$limitnum == 0) {

**return** intval($totalCount/$limitnum);

} **else** {

**return** intval($totalCount/$limitnum) + 1;

}

} **else** {

**return** 0;

}

**return null**;

}