

## 25. 模型的字段设置

学习要点：

### 1. 字段设置

本节课我们来学习模型中的字段属性的设置方式。

#### 一. 字段设置

1. 模型的数据字段和表字段是对应关系，默认会自动获取，包括字段的类型；
2. 自动获取会导致增加一次查询，如果在模型中配置字段信息，会减少内存开销；
3. 可以在模型设置 `$schema` 字段，明确定义字段信息，字段需要对应表写完整；

//设置字段信息，需要写完整的数据表字段

```
protected $schema = [  
    'id'          =>    'int',  
    'username'    =>    'string',  
    'status'      =>    'int',  
    'create_time' =>    'datetime',  
    '...'         =>    '...' ];
```

4. 系统提供了一条命令，生成一个字段信息缓存，可以自动生成；

```
php think optimize:schema
```

5. 生成后的字段缓存文件在 `runtime` 下 `schema` 文件加下；
6. 我们可以先把这里的键值对复制到 `$schema` 属性上，开启 `trace` 测试效果；
7. 这时，在控制器执行查询，会发现减少了一次 SQL 查询；
8. 只不过，大可不必设置 `$schema` 属性，因为它只对模型有效；
9. 如果想模型和数据库 `Db` 类同时有效，直接运用字段缓存文件即可；
10. 默认情况下字段缓存文件是关闭状态，需要在 `config/database.php` 开启；

// 开启字段缓存

```
'fields_cache' => true,
```

11. 当数据获取到后，想要单独获取数据可以用 `->` 和数组方式来获取；

```
$user = UserModel::find(19);  
echo $user->username;  
echo $user['email'];
```

12. 如果我们在模型端把数据整理好，交给控制器直接调用，如下方式：

//模型端

```
public function getUsername($id)  
{  
    $obj = $this->find($id);
```

```
        return $obj->getAttr('username');  
    }  
}
```

```
//控制器端调用  
$user = new UserModel();  
return $user->getUsername(19);
```

13. 字段的赋值操作，也可以是->和数组方式，作用就是提交给模型处理；

```
$user = new UserModel();  
$user->username = 'Mr.Lee';  
$user['email'] = 'lee@163.com';
```

14. 默认情况下，字段是严格区分大小写的，也就是需要和数据表字段保持一致；

```
echo $user->create_time;
```

15. 我们可以在模型属性\$strict 设置为 false 即可实现非严格字段；

```
echo $user->createTime; //并非肆无忌惮的不严格，只能首字母大写
```