

## 29. 模型的自动时间戳和只读字段

学习要点：

1. 模型自动时间戳
2. 模型只读字段

本节课我们来学习模型中用于记录时间的自动时间戳和不可更改只读字段。

### 一. 模型自动时间戳

1. 如果你想全局开启，在 `database.php` 中，设置为 `true`；

```
// 自动写入时间戳字段
'auto_timestamp' => true,
```

2. 如果你只想设置某一个模型开启，需要设置特有字段；

```
//开启自动时间戳
protected $autoWriteTimestamp = true;
```

3. 当然，还有一种方法，就是全局开启，单独关闭某个或某几个模型为 `false`；
4. 自动时间戳开启后，会自动写入 `create_time` 和 `update_time` 两个字段；
5. 此时，它们的默认的类型是 `int`，如果是时间类型，可以更改如下：

```
'auto_timestamp' => 'datetime', //或
protected $autoWriteTimestamp = 'datetime';
```

6. 都配置完毕后，当我们新增一条数据时，无须新增 `create_time` 会自动写入时间；
7. 同理，当我们修改一条数据时，无须修改 `update_time` 会自动更新时间；
8. 自动时间戳只能在模型下有效，数据库方法不可以使用；
9. 如果创建和修改时间戳不是默认定义的，也可以自定义；

```
protected $createTime = 'create_at';
protected $updateTime = 'update_at';
```

10. 如果业务中只需要 `create_time` 而不需要 `update_time`，可以关闭它；

```
protected $updateTime = false;
```

11. 也可以动态实现不修改 `update_time`，具体如下：

```
$user->isAutoWriteTimestamp(false)->save();
```

### 二. 模型只读字段

1. 模型中可以设置只读字段，就是无法被修改的字段设置；
2. 我们要设置 `username` 和 `email` 不允许被修改，如下：

```
protected $readonly = ['username', 'email'];
```

3. 除了在模型端设置，也可以动态设置只读字段；

```
$user->readonly(['username', 'email'])->save();
```

4. 同样，只读字段只支持模型方式不支持数据库方式；