

17. 模型的定义

学习要点：

1. 默认设置
2. 模型定义

本节课我们来开始学习数据库的模型部分的定义和默认值的设置。

一. 默认设置

1. 框架可以使用 Eloquent ORM 进行数据库交互，也就是关系对象模型；
2. 在数据库入门阶段，我们已经创建了一个 User.php 模型，如下：

```
php artisan make:model Http/Models/User //默认在 app 目录
```

3. 而调用的时候，我们也知道表名要遵循它默认规则，修改为复数，或特定；

```
class User extends Model
{
    protected $table = 'user';
}
```

4. 系统假定你的主键为 id，如果你要修改默认主键，可以特定；

```
protected $primaryKey = 'xid';
```

5. 系统假定你的主键 id 为自增性，意味着是主键会自动转换 int 类型；

6. 如果你希望不是非自增，非数值类型主键，可以设置取消；

```
public $incrementing = false;
```

7. 如果你主键不是一个整数，那么需要 \$keyType 设置为 string；

```
protected $keyType = 'string';
```

8. 系统默认情况下会接管 created_at 和 updated_at 两个时间戳列；

9. 如果不想让系统干涉这两个列，可以设置 false 取消；

```
public $timestamps = false;
```

10. 如果你想自定义时间戳的格式，可以设置；

```
protected $dateFormat = 'U';
```

11. 可以更改创建时间 created_at 和更新时间 updated_at 字段名；

```
const CREATED_AT = 'create_time';
const UPDATED_AT = 'update_time';
```

12. 默认读取 database.php 配置的数据库连接，也可以在模型端局部更改；

```
protected $connection = 'mysql';
```

二. 模型定义

1. 之前在查询构造器部分，把常用的数据库操作基本讲完，模型大体相同；
2. 比如，我们要查询所有数据，直接使用模型::all()即可；

//查询所有记录

```
$users = User::get();           //或 all()  
return [$users];
```

3. 也可以像查询构造器一样，添加各种各样的条件，写法一样；

//查询性别为男，价格大于90，限制显示2条

```
$users = User::where([  
    ['gender', '=', '男'],  
    ['price', '>', 95]  
])->limit(2)->get();
```

4. 虽然安装了插件，但模型还是没有代码提示，可以通过安装插件解决；

```
composer require barryvdh/laravel-ide-helper
```

```
php artisan ide-helper:generate - 为 Facades 生成注释
```

```
php artisan ide-helper:models - 为数据模型生成注释
```

```
php artisan ide-helper:meta - 生成 PhpStorm Meta file
```

5. 其它查询方法基本和查询构造器一样，如果有不一样，参考错误提示；
6. 这里列出官网给出示例的方法，对照实验(结合详细文档，重复较多)；

- (1) .find(1) //通过主键查找
- (2) .first() //查找第一个
- (3) .firstWhere() //找到查询中的首个
- (4) .find([1,2,3]) //通过数组查找
- (5) .firstOr() //查找首个返回，支持闭包
- (6) .firstOrFail() //找不到时返回异常
- (7) .count()、max()等集合 //集合操作

PS: 还有很多在查询构造器中的方法，比如排序、分组子查询等等都可以使用(并未一一验证)。