

## 21. 模型的定义方式

学习要点：

1. 定义模型
2. 设置模型

本节课我们来学习模型篇章中的定义方法，设置以及一些基本的操作。

### 一. 定义模型

1. 定义一个和数据库表向匹配的模型；

```
class User extends Model
```

2. 模型会自动对应数据表，并且有一套自己的命名规则；
3. 模型类需要去除表前缀(tp\_)，采用驼峰式命名，并且首字母大写；

```
tp_user(表名)      => User
tp_user_type(表名) => UserType
```

4. 创建空模型后，我们可以在控制器调用，创建控制器 DataModel.php；
5. 可以直接使用模型的名称 User::\*调用查询方法，比如 select()等；
6. 如果没有代码提示，请和上一节一样，把 5.1 的注释复制过来；

```
class DataModel
{
    public function index()
    {
        return json(User::select());
    }
}
```

7. 如果担心设置的模型类名和 PHP 关键字冲突，可以开启应用类后缀；
8. 比如设置：Class UserModel 这种，需要更改文件名为 UserModel.php；
9. 然后设置一下\$name 属性为指定 user(表名)即可实现；

```
class UserModel extends Model
{
    protected $name = 'user';
}
```

### 二. 设置模型

1. 默认主键为 id，你可以设置其它主键，比如 uid；

```
protected $pk = 'uid';
```

2. 从控制器端调用模型操作，如果和控制器类名重复，可以设置别名；

```
use app\model\User as UserModel;
```

3. 在模型定义中，可以设置其它的数据表；

```
protected $table = 'tp_one';
```

4. 模型和控制器一样，也有初始化，在这里必须设置 `static` 静态方法；

```
//模型初始化
```

```
protected static function init()  
{  
    //第一次实例化的时候执行 init  
    echo '初始化 User 模型';  
}
```