

## 34. 关联模型初探

学习要点：

### 1. 关联模型定义

本节课我们来了解模型中关联模型的定义方式，并实现两张表的关联。

#### 一. 关联模型定义

1. 关联模型，顾名思义，就是将表与表之间进行关联和对象化，更高效的操作数据；
2. 我们已经有了表 `tp_user`，主键为：`id`；我们需要一个附属表，来进行关联；
3. 附属表：`tp_profile`，建立两个字段：`user_id` 和 `hobby`，外键是 `user_id`；

对象	tp_user @grade (本地) - 表			
开始事务	文本	筛选	排序	导入
id	username	password	gender	
19	蜡笔小新	123	男	
20	路飞	123	男	
21	黑崎一护	456	男	
24	小明	123	男	
25	孙悟空	123	男	
26	孙悟空	123	男	
27	樱桃小丸子	123	女	
29	孙悟空	123	男	
79	辉夜	123	女	

对象

tp\_profile @grade (本地) - 表

开始事务

文本

筛选

排序

导入

id	user_id	hobby
1	19	喜欢大姐姐
2	20	特爱吃肉
3	21	朽木露琪亚
4	24	暗恋小红
5	25	毕迪丽
6	26	特兰克斯
7	27	埼玉爷爷
8	29	有空就修行
9	79	会长大人

4. 创建 `User` 模型和 `Profile` 模型，均为空模型；
5. `User` 模型端，需要关联 `Profile`，具体方式如下：

```
class User extends Model
{
    public function profile()
    {
        //hasOne 表示一对一关联，参数一表示附表，参数二外键，默认 user_id
        return $this->hasOne(Profile::class,'user_id');
    }
}
```

6. 创建一个控制器用于测试输出：`Grade.php`；

```
$user = UserModel::find(21);
return json($user->profile);
return $user->profile->hobby;
```

7. 对于关联方式，系统提供了 9 种方案，具体如下：

hasOne	一对一
belongsTo	一对一
hasMany	一对多
hasOneThrough	远程一对一
hasManyThrough	远程一对多
belongsToMany	多对多
morphMany	多态一对多
morphOne	多态一对一
morphTo	多态

8. 上面的例子，我们采用了一对一的关联模型，它还有相对的反向关联：

```
class Profile extends Model
{
    public function user()
    {
        return $this->belongsTo(User::class);
    }
}
```

```
$profile = ProfileModel::find(1);
return $profile->user->email;
```

9. 正反向关联也就是关联关系和相对的关联关系，具体如下表：

一对一	hasOne	belongsTo
一对多	hasMany	belongsTo
多对多	belongsToMany	belongsToMany
远程一对多	hasManyThrough	不支持
多态一对一	morphOne	morphTo
多态一对多	morphMany	morphTo