

35. 一对一关联查询

学习要点：

1. `hasOne` 模式
2. `belongsTo` 模式

本节课我们来了解关联模型中，一对一关联查询的使用方法。

一. `hasOne` 模式

1. `hasOne` 模式，适合主表关联附表，具体设置方式如下：

```
hasOne('关联模型',['外键','主键']);  
return $this->hasOne(Profile::class,'user_id', 'id');
```

关联模型（必须）：关联的模型名或者类名

外键：默认的外键规则是当前模型名（不含命名空间，下同）+ `_id`，例如 `user_id`

主键：当前模型主键，默认会自动获取也可以指定传入

2. 在上一节课，我们了解了表与表关联后，实现的查询方案：

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

3. 使用 `save()` 方法，可以设置关联修改，通过主表修改附表字段的值：

```
$user = UserModel::find(19);  
$user->profile->save(['hobby'=>'酷爱小姐姐']);
```

4. `->profile` 属性方式可以修改数据，`->profile()` 方法方式可以新增数据；

```
$user->profile()->save(['hobby'=>'不喜欢吃青椒']);
```

二. `belongsTo` 模式

1. `belongsTo` 模式，适合附表关联主表，具体设置方式如下：

```
belongsTo('关联模型',['外键','关联主键']);  
return $this->belongsTo(Profile::class,'user_id', 'id');
```

关联模型（必须）：模型名或者模型类名

外键：当前模型外键，默认的外键名规则是关联模型名+ `_id`

关联主键：关联模型主键，一般会自动获取也可以指定传入

2. 对于 `belongsTo()` 的查询方案，上一节课已经了解过，如下：

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

3. 使用 `hasOne()` 也能模拟 `belongsTo()` 来进行查询;

```
//参数一表示的是 User 模型类的 profile 方法，而非 Profile 模型类
$user = UserModel::hasWhere('profile', ['id'=>2])->find();
return json($user);

//采用闭包，这里是两张表操作，会导致 id 识别模糊，需要指明表
$user = UserModel::hasWhere('profile', function ($query) {
    $query->where('profile.id', 2);
})->select();

return json($user);
```