

39. 多对多关联查询

学习要点：

1. 多对多关联

本节课我们来了解关联模型中，多对多的关联查询以及新增删除等操作。

一. 多对多关联

1. 复习下一对一，一个用户对应一个用户档案资料，是一对一关联；
2. 复习下一对多，一篇文章对应多个评论，是一对多关联；
3. 多对多怎么理解，分解来看，一个用户对应多个角色，而一个角色对应多个用户；
4. 那么这种对应关系，就是多对多关系，最经典的应用就是权限控制；
5. 首先，我们来看多对多关系的三张表，具体如下：

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

id	type
1	超级管理员
2	评论审核专员
3	图片监察员
4	帐户处理专员
5	广告投放专员

id	user_id	role_id
1	19	2
2	20	4
3	21	5
4	24	1
5	25	3
6	19	3
7	21	4
8	19	1

6. tp_user: 用户表; tp_role: 角色表; tp_access: 中间表;
7. access 表包含了 user 和 role 表的关联 id, 多对多模式;
8. 在 User.php 的模型中, 设置多对多关联, 方法如下:

```
public function roles()
{
    return $this->belongsToMany(Role::class, Access::class);
}
```

9. 在 roles 方法中, belongsToMany 为多对多关联, 具体参数如下:

```
belongsToMany('关联模型', '中间表', ['外键', '关联键']);
$this->belongsToMany(Role::class, Access::class, 'role_id', 'user_id');
```

10. role.php 和 access.php 创建一个空模型即可, 无须创建任何;
11. 注意: Role 继承 Model 即可, 而中间表需要继承 Pivot;
12. 在 user.php 中, 创建 many() 方法, 用于测试, 查询方式如下:

```
public function many()
{
    //得到一个用户: 蜡笔小新
    $user = UserModel::find(21);
    //获取这个用户的所有角色
    $roles = $user->roles;
    //输出这个角色所具有的权限
```

```
        return json($roles);  
    }  
}
```

13. 当我们要给一个用户创建一个角色时，用到多对多关联新增；

14. 而关联新增后，不但会给 `tp_role` 新增一条数据，也会给 `tp_access` 新增一条；

```
$user->roles()->save(['type'=>'测试管理员']);  
$user->roles()->saveAll([[...],[...]]);
```

15. 一般来说，上面的这种新增方式，用于初始化角色比较合适；

16. 也就是说，各种权限的角色，并不需要再新增了，都是初始制定好的；

17. 那么，我们真正需要就是通过用户表新增到中间表关联即可；

```
$user->roles()->save(1);  
或：  
$user->roles()->save(Role::find(1));  
$user->roles()->saveAll([1,2,3]);  
或：  
$user->roles()->attach(1);  
$user->roles()->attach(2, ['details'=>'测试详情']);
```

18. 除了新增，还有直接删除中间表数据的方法：

```
$user->roles()->detach(2);
```