

74. 关联预载入

学习要点：

1. 关联预载入

本节课我们来了解关联模型中，关联预载入的使用方法。

一. 关联预载入

1. 在普通的关联查询下，我们循环数据列表会执行 $n+1$ 次 SQL 查询；

```
$list = UserModel::all([19, 20, 21]);
foreach ($list as $user) {
    dump($user->profile);
}
```

2. 上面继续采用一对一的构建方式，打开 **trace** 调试工具，会得到四次查询；
3. 如果采用关联预载入的方式，将会减少到两次，也就是起步一次，循环一次；

```
$list = UserModel::with('profile')->all([19, 20, 21]);
foreach ($list as $user) {
    dump($user->profile);
}
```

4. 关联预载入减少了查询次数提高了性能，但是不支持多次调用；
5. 如果你有主表关联了多个附表，都想要进行预载入，可以传入多个模型方法即可；
6. 为此，我们再创建一张表 **tp_book**，和 **tp_profile** 一样，关联 **tp_user**；

```
$list = UserModel::with('profile,book')->all([19, 20, 21]);
foreach ($list as $user) {
    dump($user->profile.$user->book);
}
```

7. 上面的方式，还有一种简要写法，如下：

```
UserModel::all([19, 20, 21], 'profile,book');
```

8. **with()**是 **IN** 方式的查询，如果想使用 **JOIN** 查询，可以使用 **withJoin()**：

```
UserModel::withJoin('profile')->all([19, 20, 21])
```

9. 在使用 **JOIN** 查询方案下，限定字段，可以使用 **withField()**方法来进行；

```
$list = UserModel::withJoin(['profile'=>function ($query) {
    $query->withField('hobby');
}])->all([19, 20, 21]);
```

或者：

```
UserModel::withJoin(['profile'=>['hobby']])->all([19, 20, 21])
```

10. 关联预载入还提供了一个延迟预载入，就是先执行 `all()`再 `load()`载入；

```
$list = UserModel::all([19, 20, 21]);  
$list->load('profile');  
foreach ($list as $user) {  
    dump($user->profile);  
}
```