

19. 批量赋值和软删除

学习要点：

1. 批量赋值
2. 软删除

本节课我们来开始学习数据库的批量赋值的原因和软删除的方法。

一. 批量赋值

1. 上一节增删改中，新增中我们发现需要进行批量赋值的许可；
2. 一般情况下，是为了防止提交过来的字段在部分场景中不需要或不能；
3. 所以，我们需要通过黑白名单机制进行过滤掉必要的字段；

```
//通过提交过来的数据一次性新增
User::create(\Request::all());
```

二. 软删除

1. 什么叫软删除？它相对于真实的删除，而并非真正的删除，只是隐藏了；
2. 首先，需要在数据库创建一个字段 `deleted_at`(默认)，用于判断是否被软删除；
3. 默认设置这个字段为空(`null`)，如果写入数据，成为非空状态，则说明被删除；
4. 开启软删除的功能，需要在模型端设置一下：

```
//开启软删除功能
use SoftDeletes;
```

5. 当开启了软删除功能，之前的删除操作，都会变成更新操作，给 `deleted_at` 赋值；

```
//删除一
$users = User::find(82);
$users->delete();
```

```
//删除二
User::destroy(81);
```

6. 而当我们开启了软删除的功能后，此时通过正常的的数据获取列表，会自动隐藏；

```
//软删除的数据不可见
$users = User::get();
return [$users];
```

```
//单独获取被软删除的数据不行
$users = User::find(82);
return [$users];
```

7. 如果需要查询包含软删除的数据，通过 `withTrashed()` 方法实现：

```
//获取包含软删除的数据
$users = User::withTrashed()->get();
return [$users];

//获取某个被软删除的数据(即使不是软删除的也可以搜索到)
$users = User::withTrashed()->find(82);
return [$users];
```

8. 如果只想搜索出被软删除的数据，通过 `onlyTrashed()` 方法实现：

```
//获取所有软删除的数据
$users = User::onlyTrashed()->get();
return [$users];

//获取某个被软删除的数据(只有软删除的数据才可以被搜索到)
$users = User::onlyTrashed()->find(82);
return [$users];
```

9. 如果要对这个数据进行软删除的判断，是否是被软删除的，可以使用 `trashed()`；

```
//判断是否是被软删除的数据
$users = User::withTrashed()->find(81);
return $users->trashed();
```

10. 如果想将软删除的数据恢复正常(类似从回收站还原)，使用 `restore()` 方法；

```
//将被软删除的数据回复正常
$users = User::onlyTrashed()->find(82);
$users->restore();
```

11. 如果开启了软删除，还需要强行真实的永久删除，可以使用 `forceDelete()` 方法；

```
//开启软删除时的真实永久删除
$users = User::onlyTrashed()->find(82);
$users->forceDelete();
```