

model update 常规用法

假设表结构为:

```
class User(models.Model):
    username = models.CharField(max_length=255, unique=True, verbose_name='用户名')
    is_active = models.BooleanField(default=False, verbose_name='激活状态')
```

方法一:

```
User.objects.filter(id=1).update(username='nick',is_active=True)
```

方法二:

```
_t = User.objects.get(id=1) _t.username='nick' _t.is_active=True _t.save()
```

方法一适合更新一批数据, 类似于mysql语句update user set username='nick' where id = 1

方法二适合更新一条数据, 也只能更新一条数据, 当只有一条数据更新时推荐使用此方法, 另外此方法还有一个好处, 我们接着往下看

具有 auto_now 属性字段的更新

我们通常会给表添加三个默认字段

- 自增 ID, 这个 django 已经默认加了, 就像上边的建表语句, 虽然只写了 `username` 和 `is_active` 两个字段, 但表建好后也会有一个默认的自增 id 字段
- 创建时间, 用来标识这条记录的创建时间, 具有 `auto_now_add` 属性, 创建记录时会自动填充当前时间到此字段
- 修改时间, 用来标识这条记录最后一次的修改时间, 具有 `auto_now` 属性, 当记录发生变化时填充当前时间到此字段

就像下边这样的表结构:

```
class User(models.Model):
    create_time = models.DateTimeField(auto_now_add=True, verbose_name='创建时间')
    update_time = models.DateTimeField(auto_now=True, verbose_name='更新时间')
    username = models.CharField(max_length=255, unique=True, verbose_name='用户名')
    is_active = models.BooleanField(default=False, verbose_name='激活状态')
```

当表有字段具有 `auto_now` 属性且你希望他能自动更新时, 必须使用上边方法二的更新, 不然 `auto_now` 字段不会更新

Reference

[Django model update的各种用法介绍](#) [Django model update的各种用法介绍](#)

2019.4.9