KylinBlog 开发注解-Part3

DJango中的ORM数据CURD接口

- Django中前台的表单数据是如何提交到后台呈现的?
- Django中前端页面所展示的数据是如何从后台获取的?

Django的对象关系映射系统(Object-Relational Mapper, ORM)提供了丰富的数据查询接口,

无需使用原生SQL语句即可通过对模型的简单操作实现对数据库里的数据进行增删改查。

查询得到的结果叫查询集(QuerySet), 所以这个接口被称为QuerySet API。

以一个简要的Article模型为例来讲解，这里之讲解最基础的CURD方法

示例模型

```
from django.db import models

class Article(models.Model):
    title = models.CharField('标题', max_length=200, unique=True)
    body = models.TextField('正文')
    created = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.title
```

增

对于新增数据，Django提供了 `save()` 和 `create()` 两种方法

save()

```
from .models import Article

article = Article(title="My first article", body="My first article body")
article.save()
```

注意，调用完Article（）方法后必须使用.save()，否则记录不会被添加到数据库中

create()

```
article = Article.objects.create(title="My first article", body="My first article body")
```

删

删即从数据表中删除一个已有条目。Django也允许同时删除一条或多条数据。

删除单条数据

```
# 删除第5篇文章
Article.objects.get(pk=5).delete()
```

删除部分数据

```
# 删除标题含有python的文章
Article.objects.filter(title__icontains="python").delete()
```

删除所有数据

```
# 慎用
Article.objects.all().delete()
```

查

查主要使用get, filter及exclude方法，而且这些方法是可以联用的。

查询所有数据

```
# QuerySet类型，实例对象列表
Article.objects.all()
# 字典列表
Article.objects.all().values()
# 只获取title-字典形式
Article.objects.all().values('title')
# 只获取title列表- 元组形式，只有value，没有key
Article.objects.all().values_list('title')
# 只获取title列表，只有value
Article.objects.all().values_list('title', flat=True)
```

查询单条数据

```
article = Article.objects.get(id=11)
```

当上述查询有个问题，如果id不存在，会抛出错误。还有一种方式是使用`filter`方法, 这样即使id不存在也不会报错。

```
article = Article.objects.filter(id=1).first()
```

一个更好的方式是使用Django提供的`get_object_or_404`方法，如下所示：

```
from django.shortcuts import get_object_or_404
article = get_object_or_404(Article, pk=1)
```

查询多条数据

按大于、小于及不等于查询

```
# gte：大于等于，lte：小于等于
articles = Article.objects.filter(id__gte=2).filter(id__lte=11)
# 不等于
articles = Article.objects.exclude(id=10)
```

按范围查询

```
# 按范围查询，in或者range
articles = Article.objects.filter(id__range=[2, 11])
articles = Article.objects.filter(id__in=[3, 6,9])
```

改

改既可以用save方法，也可以用update方法。

其区别在于save方法不仅可以更新数据中现有对象数据，还可以创建新的对象。

而update方法只能用于更新已有对象数据。

一般来说，如果要同时更新多个对象数据，用update方法或bulk_update方法更合适。

save方法

```
article = Article.objects.get(id=1)
article.title = "New article title"
article.save()
```

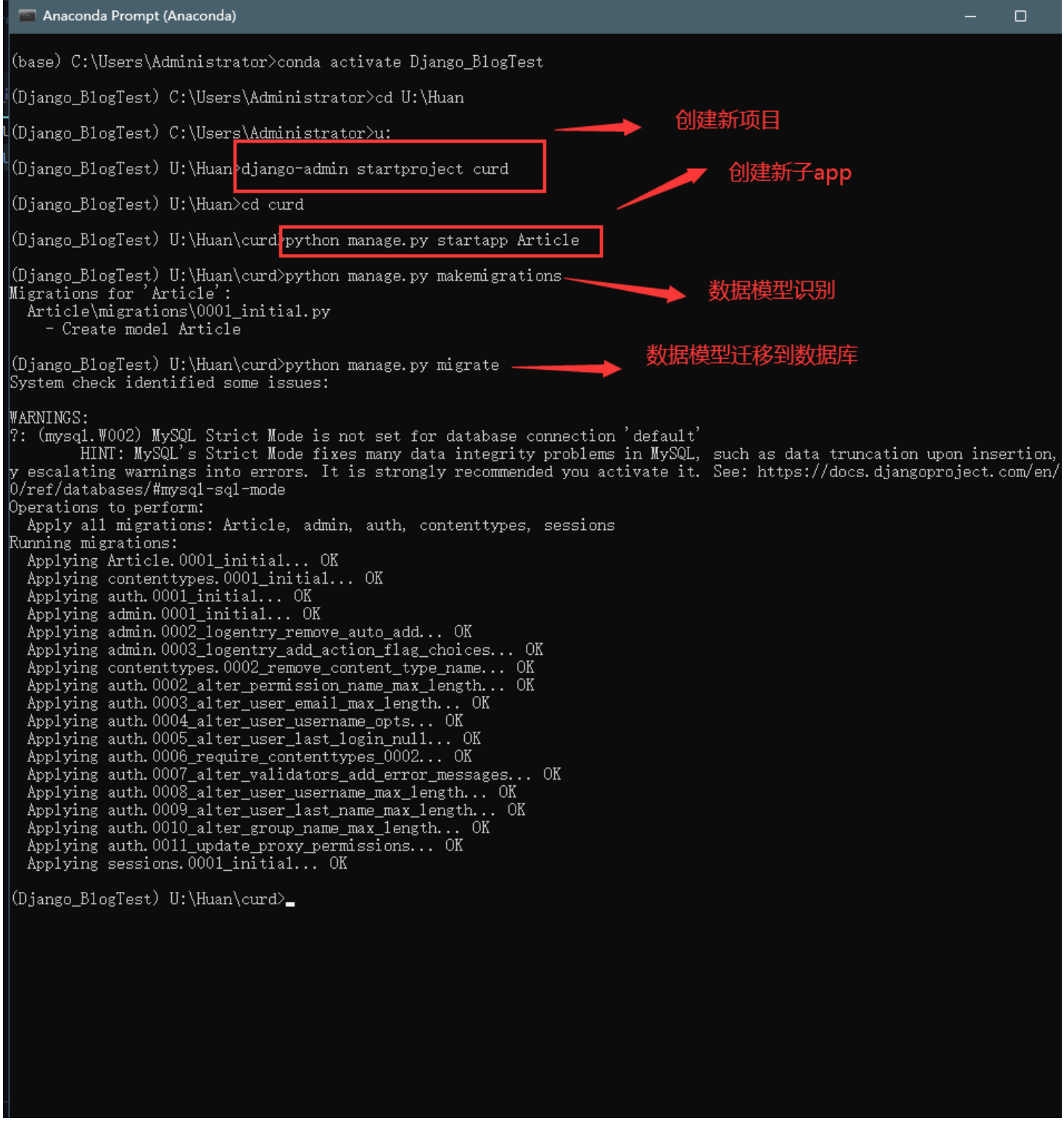
update方法更新单篇文章

```
article = Article.objects.get(id=1).update(title='new title')
```

update方法同时更新多篇文章

```
# 更新所有文章标题
article = Article.objects.filter(title__icontains='python').update(title='Django')
```

实例演示


Anaconda Prompt (Anaconda)
(base) C:\Users\Administrator>conda activate Django_BlogTest
(Django_BlogTest) C:\Users\Administrator>cd U:\Huan
(Django_BlogTest) C:\Users\Administrator>u:
(Django_BlogTest) U:\Huan>django-admin startproject curd
(Django_BlogTest) U:\Huan>cd curd
(Django_BlogTest) U:\Huan\curd>python manage.py startapp Article
(Django_BlogTest) U:\Huan\curd>python manage.py makemigrations
Migrations for 'Article':
Article\migrations\0001_initial.py
- Create model Article
(Django_BlogTest) U:\Huan\curd>python manage.py migrate
System check identified some issues:
WARNINGS:
?: (mysql.W002) MySQL Strict Mode is not set for database connection 'default'
HINT: MySQL's Strict Mode fixes many data integrity problems in MySQL, such as data truncation upon insertion,
y escalating warnings into errors. It is strongly recommended you activate it. See: https://docs.djangoproject.com/en/
0/ref/databases/#mysql-sql-mode
Operations to perform:
Apply all migrations: Article, admin, auth, contenttypes, sessions
Running migrations:
Applying Article.0001_initial... OK
Applying contenttypes.0001_initial... OK
Applying auth.0001_initial... OK
Applying admin.0001_initial... OK
Applying admin.0002_logentry_remove_auto_add... OK
Applying admin.0003_logentry_add_action_flag_choices... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying auth.0010_alter_group_name_max_length... OK
Applying auth.0011_update_proxy_permissions... OK
Applying sessions.0001_initial... OK
(Django_BlogTest) U:\Huan\curd>
创建新项目
创建新子app
数据模型识别
数据模型迁移到数据库