



链滴

Django_ 学员管理后台系统开发

作者: [yuanhenglizhen](#)

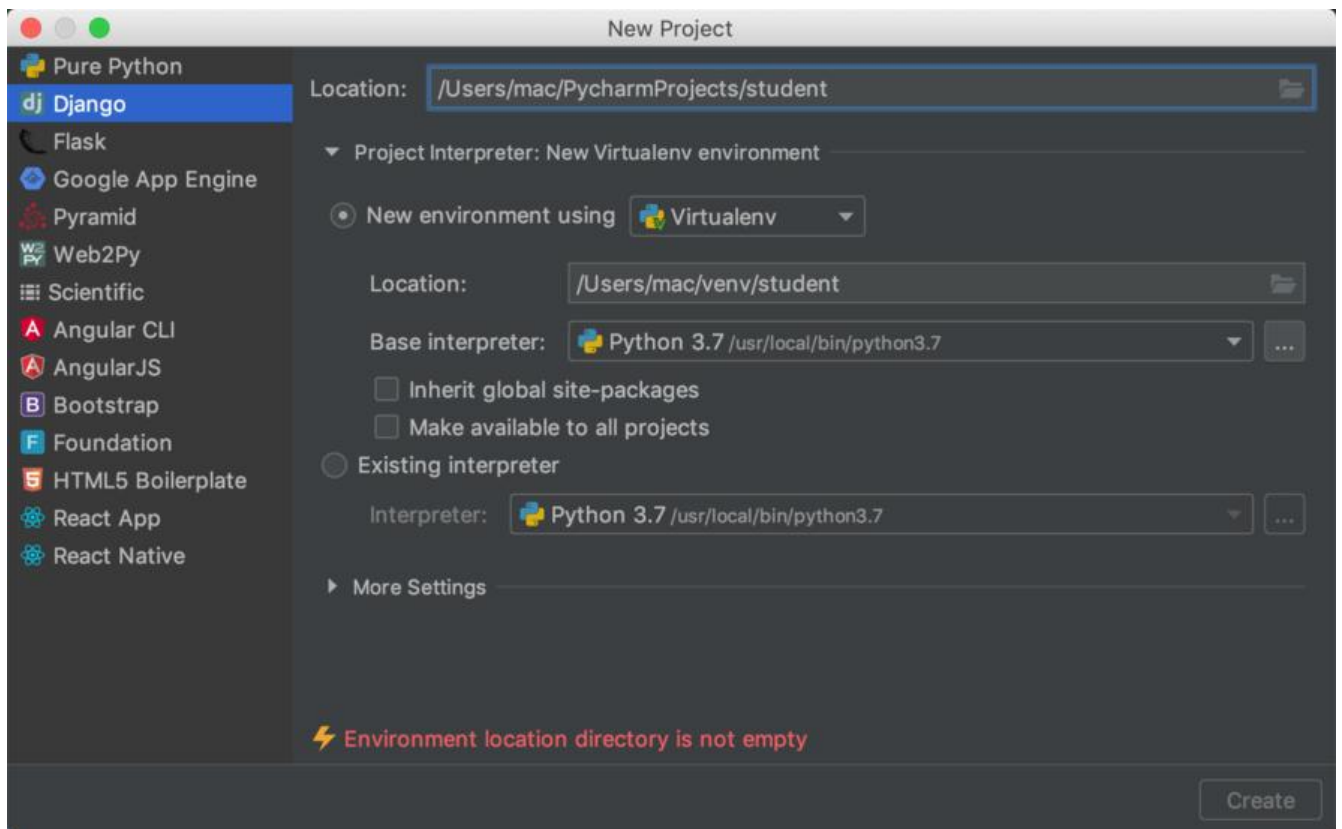
原文链接: <https://ld246.com/article/1563287510777>

来源网站: 链滴

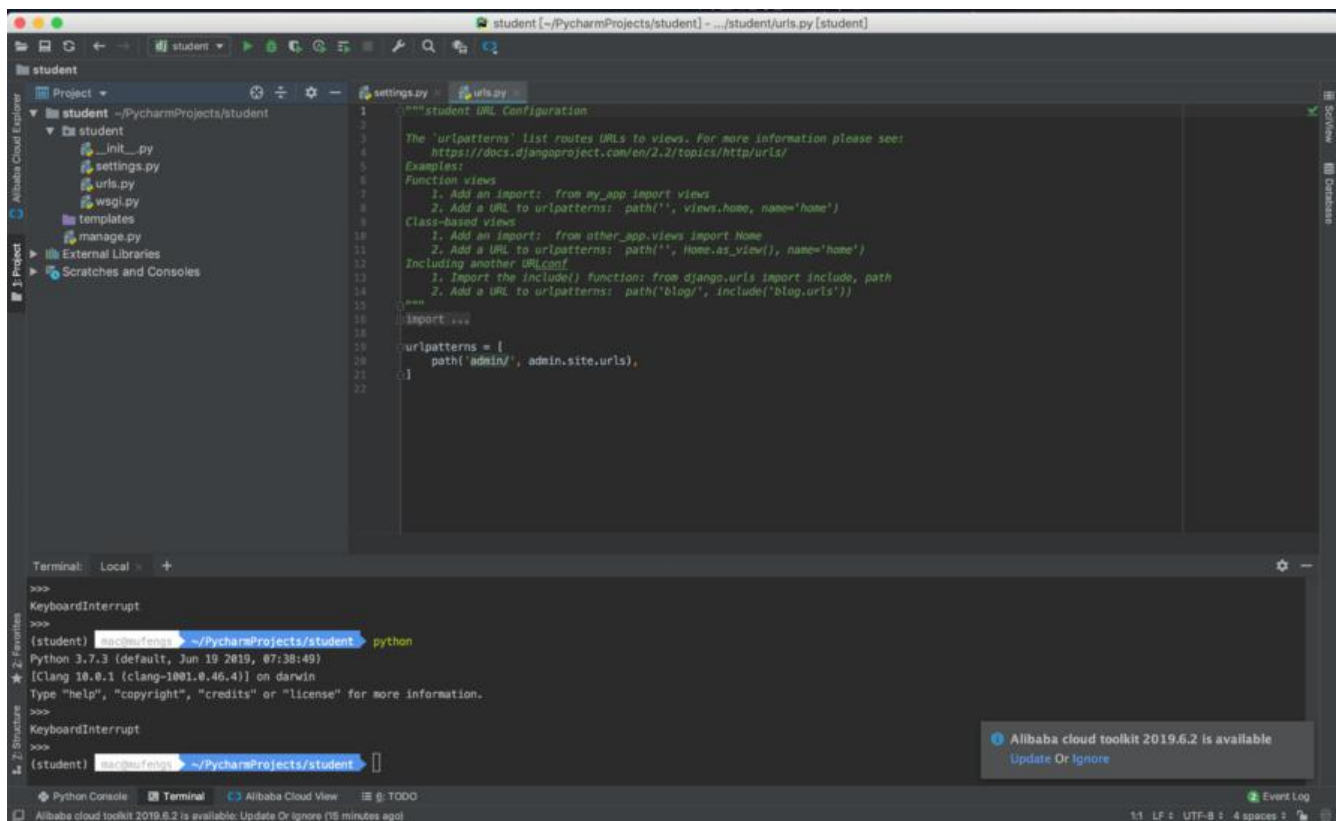
许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1.初始化环境

我这边用的是pycharm，选择Django，新建一个虚拟环境。如下图所示



等待一丢丢会，项目初始化成功



在命令行中执行python，可以看到已经是python3.7了

项目结构如下：

```
.
├── manage.py # python命令管理
├── student # 项目配置相关
│   ├── __init__.py
│   ├── settings.py # 配置项
│   ├── urls.py # 路由规则
│   └── wsgi.py # Web Server Gateway Interface
└── templates # 模版
```

2.创建一个应用 student_sys

使用manage创建一个app

```
python manage.py startapp student_sys
```

这时候我们的项目结构已经发生了变化

```
.
├── manage.py
├── student
│   ├── __init__.py
│   ├── __pycache__
│   │   ├── __init__.cpython-37.pyc
│   │   └── settings.cpython-37.pyc
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
├── student_sys
│   ├── __init__.py
│   ├── admin.py
│   ├── apps.py
│   ├── migrations
│   │   └── __init__.py
│   ├── models.py
│   ├── tests.py
│   └── views.py
└── templates
```

下面就开始扣代码吧， "cheat is cheaper, show me code"

ok,let`s go!

编写代码

model.py

修改model，定义相关的字段。这边是用的python3.7，如果你用python2版本的需要加上相关编码明

```
from django.db import models

# Create your models here.

class Student(models.Model):

    SEX_ITEMS = [

        (1, 'boy'),

        (2, 'girl'),

        (0, 'unknow'),

    ]

    STATUS_ITEMS = [

        (0, 'apply'),

        (1, 'pass'),

        (2, 'refuse')

    ]

    name = models.CharField(max_length=128, verbose_name='name')
    sex = models.IntegerField(choices=SEX_ITEMS, verbose_name='sex')
    profession = models.CharField(max_length=128, verbose_name='job')
    email = models.EmailField(verbose_name='email')
    qq = models.CharField(max_length=128, verbose_name='QQ')
    phone = models.CharField(max_length=128, verbose_name='phone')
    status = models.IntegerField(choices=STATUS_ITEMS, default=0, verbose_name='check_status')

    created_time = models.DateTimeField(auto_now_add=True, editable=False, verbose_name='created_time')

    def __str__(self):

    return '<Student: {}>'.format(self.name)

    class Meta:

        verbose_name = verbose_name_plural = 'student_info'
```

admin.py

```
from django.contrib import admin

# Register your models here.

from .models import Student

class StudentAdmin(admin.ModelAdmin):

    list_display = ('id', 'name', 'sex', 'profession',
                    'email', 'qq', 'phone', 'status',
                    'created_time'
                    )

    list_filter = ('name', 'status', 'created_time')

    search_fields = ('name', 'profession')

    fieldsets = (
        (None, {
            'fields': (
                'name',
                ('sex', 'profession'),
                ('email', 'qq', 'phone'),
                'status'
            )
        }),
    )

    admin.site.register(Student, StudentAdmin)
```

settings.py

添加student到INSTALLED_APPS

```
INSTALLED_APPS = [
```

```
'student',  
'django.contrib.admin',  
'django.contrib.auth',  
'django.contrib.contenttypes',  
'django.contrib.sessions',  
'django.contrib.messages',  
'django.contrib.staticfiles',  
,  
.....
```

迁移创建表

在包含manage.py那一级目录下执行

```
python manage.py makemigrations
```

```
python manage.py migrate
```

迁移创建表，输出如下

Operations to perform:

Apply all migrations: admin, auth, contenttypes, sessions

Running migrations:

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

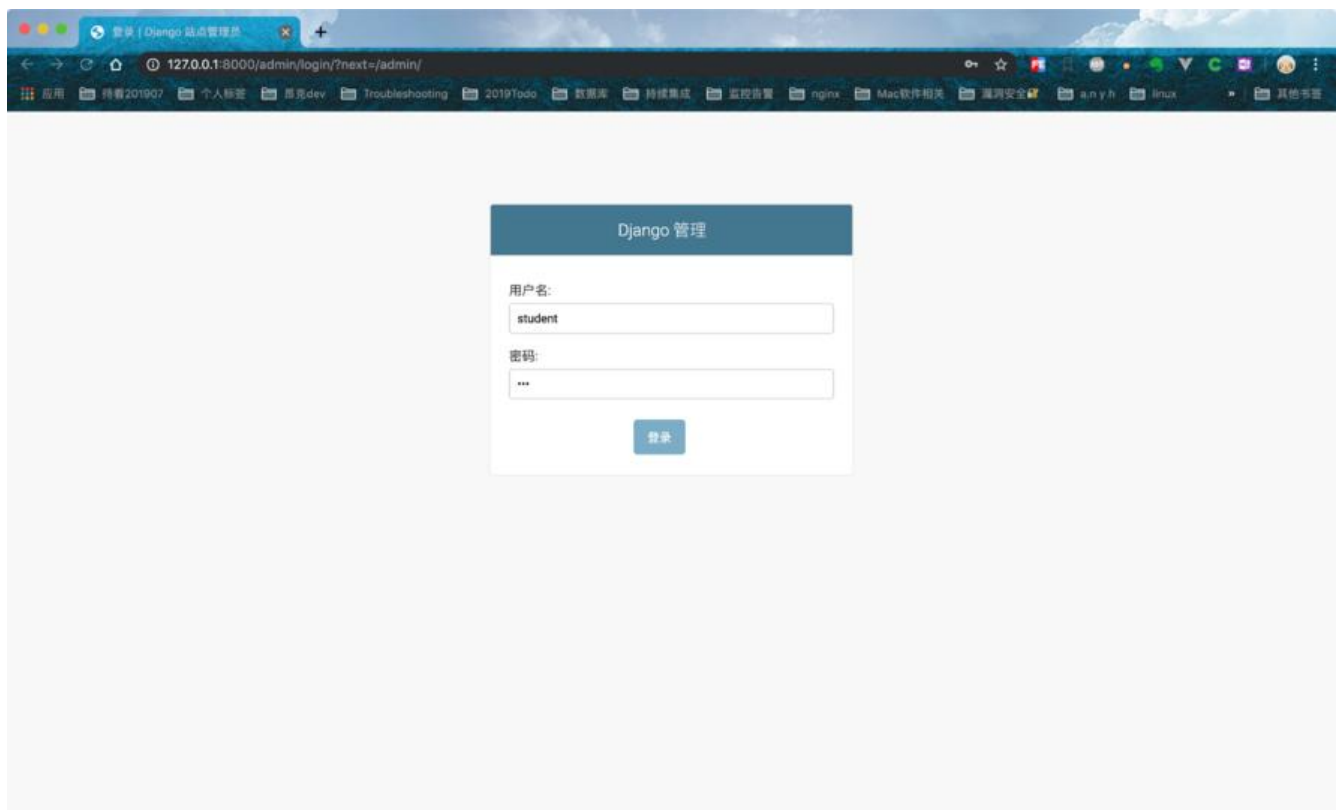
创建后台用户

`python manage.py createsuperuser`

到这里我们简单的管理后台就写完了

启动访问

`python manage.py runserver 8000`





今天就到这里，下面的功能后面在实现