

创建一个django项目（一）

创建project:

终端命令:

```
django-admin startproject.djangostart
```

djangostart就是项目名

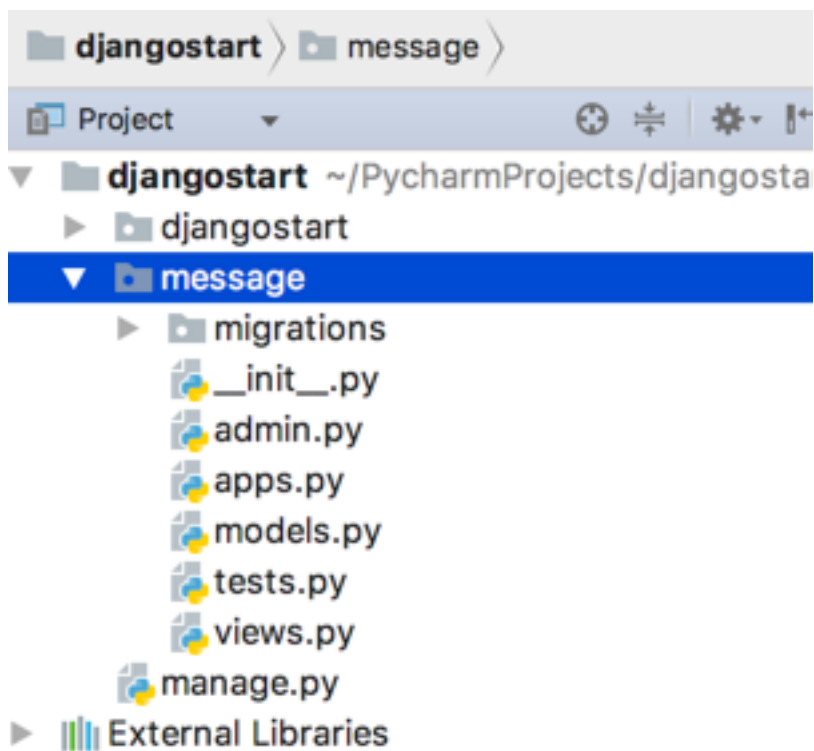
创建app:

cd到项目目录中

```
python manage.py startapp message
```

这个命令就创建了名字为message的app

该app下，会自动生成一些文件 / 文件夹



新建常用目录:

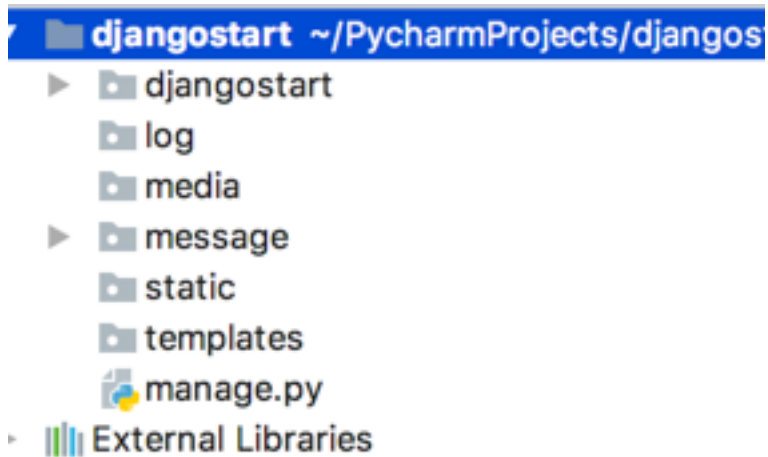
templates: 放置html文件

static: 放置css/js / 静态图片文件等

log: 存放日志

media: 用户上传文件等

如图:



在templates中新建message_form.html,并拆分css文件:

将css代码转移到static/css/style.css中,并在form.html中引入style.css
<link rel="stylesheet" href="/static/css/style.css">

在message/views.py中写getform()方法:

```
def getform(request):  
    return render(request, 'message_form.html')
```

配置url映射:

【当我们访问<http://127.0.0.1:8000/message/> 的时候,我们让它访问message/views.py中的getform()方法】

修改djangotest/url.py:

```
from django.conf.urls import url,include  
from django.contrib import admin  
  
urlpatterns = [  
    url(r'^admin/', admin.site.urls),  
    url(r'^message/', include('message.urls')),  
]
```

在message文件夹下新建url.py,并修改:

```
from django.conf.urls import url,include  
from django.contrib import admin  
from .views import getform  
  
urlpatterns = [  
    url(r'^$', getform),  
]
```

设置settings.py:

```
添加STATICFILES_DIRS = [  
    os.path.join(BASE_DIR, 'static'),  
]  
修改TEMPLATES配置项中'DIRS': [os.path.join(BASE_DIR, 'templates')],
```

运行:

```
python manage.py runserver  
在浏览器中访问http://127.0.0.1:8000/message 就能看到页面了
```

我们再升级一下难度：用户通过前端页面提交信息到mysql数据库

配置mysql数据源:

```
安装mysql客户端: pip3 install mysqlclient  
配置settings.py中的DATABASES选项:  
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'testdjango', #数据库名称  
        'USER': 'root', #数据库连接用户名  
        'PASSWORD': 'mima', #数据库连接密码  
        'HOST': 'xx.xx.xxx.xx'  
    }  
}
```

新建表user_message

在message/models.py中, 写入对应的类:

```
class UserMessage(models.Model):  
    # object_id = models.CharField(max_length=50, default='', primary_key=True)  
    name = models.CharField(max_length=50, null=True, blank=True, default='', verbose_name=u'用户名')  
    email = models.EmailField(verbose_name=u'邮箱')  
    address = models.CharField(max_length=100, verbose_name=u'联系地址')  
    message = models.CharField(max_length=100, verbose_name=u'留言信息')  
  
    class Meta:  
        verbose_name = u'用户留言信息'  
        verbose_name_plural = verbose_name  
        db_table = 'user_message'
```

将message加入settings.py的INSTALLED_APPS中，完成注册

```
python manage.py makemigrations message
python manage.py migrate
这样就生成表了
```

用户填写信息提交到表：

前端：

```
<form action="/message/newmessage/"
{% csrf_token %}
.....
</form>
```

后端message/views.py中：

```
from .models import UserMessage
```

```
def newmessage(request):
    if request.method == 'POST':
        user_message = UserMessage()
        user_message.name = request.POST.get('name')
        user_message.email = request.POST.get('email')
        user_message.address = request.POST.get('address')
        user_message.message = request.POST.get('message')
        user_message.save()
    return render(request, 'message_form.html')
```

url配置

- (1) 项目全局配置urls.py的urlpatterns中，添加：
url(r'^message/', include('message.urls')),
- (2) message/urls.py的urlpatterns中，添加：
from .views import getform,newmessage
url(r'^newmessage/\$', newmessage),

python manage.py runserver启动服务：

在浏览器中，访问<http://127.0.0.1:8000/message/> 填写信息，并提交
在mysql中查看数据是否被添加~

所有有中文出现的文件中，需要注明编码方式：

```
# -*- coding: utf-8 -*-
```

