

Django 制作留言板

一. 安装 django

1. 在虚拟环境 env 下安装 django，安装 django 需要与 python 版本进行匹配

2. 版本匹配表

Python环境与Django版本对应表：

Django version	Python versions
1.8	2.7, 3.2 (until the end of 2016), 3.3, 3.4, 3.5
1.9, 1.10	2.7, 3.4, 3.5
1.11	2.7, 3.4, 3.5, 3.6
2.0	3.4, 3.5, 3.6
2.1	3.5, 3.6, 3.7

3. 安装 django

```
pip install virtualenv -i http://mirrors.aliyun.com/pypi/simple --trusted-host mirrors.aliyun.com
```

二. 创建第一个 django 项目

1. 创建 django 项目 test2

```
django-admin.py startproject test2
```

2. 在 message 里面创建 test2app

```
python manage.py startapp test2app
```

3. 启动服务器，运行项目（127.0.0.1:8000）

```
python manage.py runserver
```

http://127.0.0.1:8000 (留言页面)

http://127.0.0.1:8000/index1 (查看留言页面)

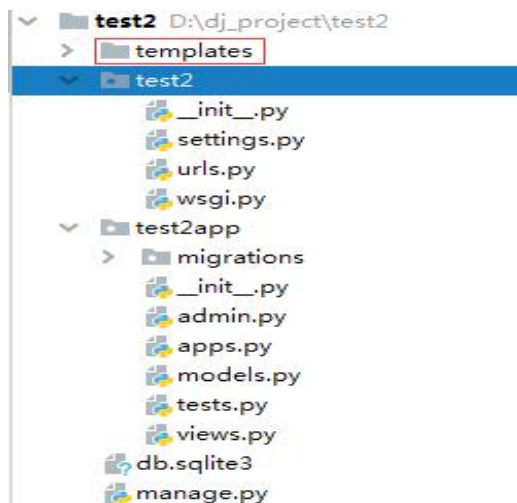
三. 在浏览器上显示页面

1. 配置项目 setting (可使用 pycharm 打开)

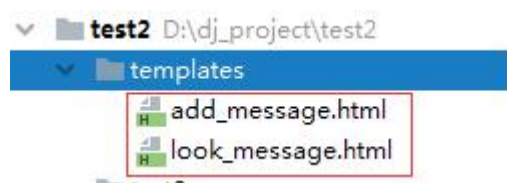
```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'test2app',  
]
```

2. 配置 templates 文件路径并在根目录下创建文件夹

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [os.path.join(BASE_DIR, 'templates')],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    },  
]
```



3. 将前期写好的 html 文件复制到 templates 文件夹



4. 配置 views 文件

```
from django.shortcuts import render
# Create your views here.
def add(request):
    return render(request, 'add_message.html')
def show(request):
    return render(request, 'look_message.html')
```

5. 配置 urls 文件

```
from django.contrib import admin
from django.urls import path
from test2app.views import add
from test2app.views import show

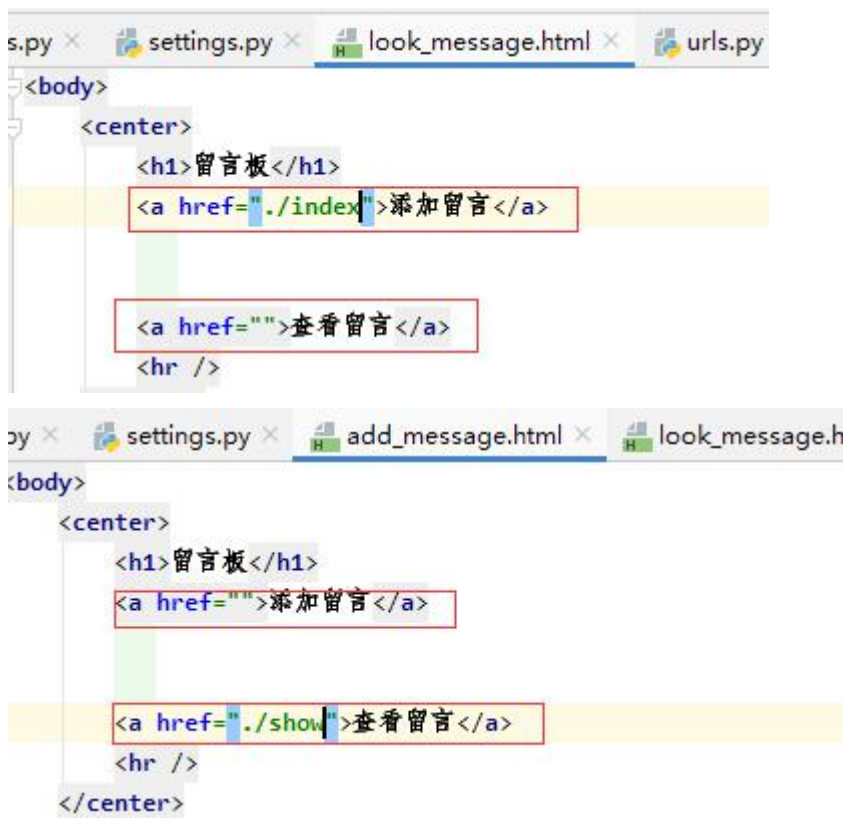
urlpatterns = [
    path('admin/', admin.site.urls),
    path('index', add),
    path('show', show),
]
```

6. 前台显示页面





7. 需改文件路径，完成页面切换



8. 创建数据库字段 (models.py)

```
s.py × settings.py × add_message.html × models.py × look_message.html × urls.py ×
from django.db import models

# Create your models here.
class Liuyan(models.Model):
    author = models.CharField(max_length=256)
    them = models.CharField(max_length=256)
    article = models.TextField(max_length=512)
    time = models.DateField()

# 显示内容
def __str__(self):
    tpl = '<Liuyan:[id={id}],author={author}, them={them}, article={article}, time={time}]>'
    return tpl.format(id = self.id,author=self.author, them=self.them,
                       article =self.article,time= self.time)
```

9. 初始化数据库（创建更改的文件并将生成的 py 文件应用到数据库,数据库默认使用 sqlite3,此处“guestbook”理解为“test2app”）

```
(env) D:\PythonProject\virtualenv\message>python manage.py makemigrations
Migrations for 'guestbook':
  guestbook\migrations\0001_initial.py
  - Create model Message

(env) D:\PythonProject\virtualenv\message>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, guestbook, 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 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 guestbook.0001_initial... OK
  Applying sessions.0001_initial... OK
```

10. 生成后台管理页面（本地化，语言和时区的本地化，修改 settings.py 文件）

```
settings.py × add_message.html × models.py ×  
  
LANGUAGE_CODE = 'zh-hans'  
  
TIME_ZONE = 'Asia/Shanghai'  
  
USE_I18N = True  
  
USE_L10N = True  
  
USE_TZ = True
```

11. 创建管理员: `python manage.py createsuperuser`
12. 登录后台 (地址后加/admin)
13. 注册模型类 (在 `admin.py` 中注册) 并自定义管理页面

```
settings.py × admin.py × add_message.html × models.py ×  
  
from django.contrib import admin  
from test2app.models import Liuyan  
class MessageAdmin(admin.ModelAdmin):  
    list_display = ['id', 'them', 'author', 'article', 'time']  
admin.site.register(Liuyan, MessageAdmin)
```

14. 实现查看留言功能 (前往后天页面添加测试数据)

The screenshot shows the Django admin interface in a web browser. The page title is "增加 liuyan | Django 站点管理". The URL is "127.0.0.1:8000/admin/test2app/liuyan/add/". The page has a blue header with "Django 管理" and a navigation bar with "首页", "Test2App", "Liuyans", and "增加 liuyan". The main content area is titled "增加 liuyan" and contains a form with the following fields:

- Author:
- Them:
- Article:
- Time:

At the bottom right, there are three buttons: "保存并增加另一个", "保存并继续编辑", and "保存".


```
views.py x views.py x admin.py x add_message.html x models.py
from django.shortcuts import render
from test2app.models import Liuyan
from django.http import HttpResponseRedirect
from . import models
# Create your views here.
def add(request):
    return render(request, 'add_message.html')
def show(request):
    liuyan = Liuyan.objects.all()
    return render(request, 'look_message.html', {'Liuyan': liuyan})
```

```
views.py x views.py x admin.py x look_message.html x add_message.html x models.py x
<td align="center"><b>操作</b></td>
</tr>
{% if Liuyan %}
    {% for Liuyan in Liuyan %}
        <tr class="{% cycle 'active' 'success' 'warning' 'info' %}">
            <td align="center">{{ Liuyan.id }}</td>
            <td align="center">{{ Liuyan.title }}</td>
            <td align="center">{{ Liuyan.username }}</td>
            <td align="center">{{ Liuyan.publish|date:'Y-m-d H:i:s' }}</td>
            <td align="center">{{ Liuyan.content }}</td>
            <td align="center"><a href="/delete?id={{ Liuyan.id }}">删除</a></td>
        </tr>
    {% endfor %}
{% else %}
    <tr>
        <td colspan="6">无数据</td>
    </tr>
{% endif %}
```

15. 实验添加留言功能

自行对照代码完成

16. 实验删除留言功能

```
views.py x views.py x admin.py x look_message.html x add_message.html
from django.shortcuts import render
from test2app.models import Liuyan
from django.http import HttpResponseRedirect
from . import models
# Create your views here.
def add(request):
    return render(request, 'add_message.html')
def show(request):
    liuyan = Liuyan.objects.all()
    return render(request, 'look_message.html', {'Liuyan': liuyan})
def delete(request):
    ly_id = request.GET.get('id')
    liuyan = Liuyan.objects.get(id=ly_id)
    liuyan.delete()
    return HttpResponseRedirect('/show')
```

ings.py × views.py × admin.py × urls.py × look_message

<https://docs.djangoproject.com/en/2.1/topics/http/urls/>

Examples:

Function views

1. Add an import: `from my_app import views`
2. Add a URL to `urlpatterns`: `path('', views.home, name='home')`

Class-based views

1. Add an import: `from other_app.views import Home`
2. Add a URL to `urlpatterns`: `path('', Home.as_view(), name='home')`

Including another `URLconf`

1. Import the `include()` function: `from django.urls import include`
2. Add a URL to `urlpatterns`: `path('blog/', include('blog.urls'))`

"""

```
from django.contrib import admin
from django.urls import path, re_path
from test2app.views import add
from test2app.views import show
from test2app.views import delete
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('index', add),
    path('show', show),
    re_path('delete', delete)
```

```
]
```

```
{% if Liuyan %}
    {% for Liuyan in Liuyan %}
        <tr class="{% cycle 'active' 'success' 'warning' 'info' %}">
            <td align="center">{{ Liuyan.id }}</td>
            <td align="center">{{ Liuyan.title }}</td>
            <td align="center">{{ Liuyan.username }}</td>
            <td align="center">{{ Liuyan.publish|date:'Y-m-d H:i:s' }}</td>
            <td align="center">{{ Liuyan.content }}</td>
            <td align="center"><a href="/delete?id={{ Liuyan.id }}">删除</a></td>
        </tr>
    {% endfor %}
{% else %}
    <tr>
        <td colspan="6">无数据</td>
    </tr>
{% endif %}
```


17. 访问页面查看效果

留言板

[添加留言](#)[查看留言](#)

添加留言

留言标题：

留言人：

留言内容：

请输入留言内容...

提交

重置

留言板

[添加留言](#)[查看留言](#)

查看留言

留言ID	留言标题	留言人	留言时间	留言内容	操作
9	sdfsd	fdsf	2020-06-23 11:30:06	请输入留言内容...	删除
10	好困呀	CZXCZX	2020-06-23 13:00:16	请输入留言内容...	删除
11	CZXCZXC	XCZCZX	2020-06-23 13:00:23	ZXCZCX	删除