



django实战项目 内容管理系统cms (四)视图层views.py1/4

D.Roger • 2016年6月28日 • 7 条评论

春分：春分之日玄鸟至，又五日雷乃发声，又五日使电。



[TOC]

你看，这个标题有个1/4，说明views.py有不少麻烦事，毕竟网站的大部分业务逻辑都是在这个文件里实现的。

views.py用于封装负责处理用户请求及返回响应的逻辑。

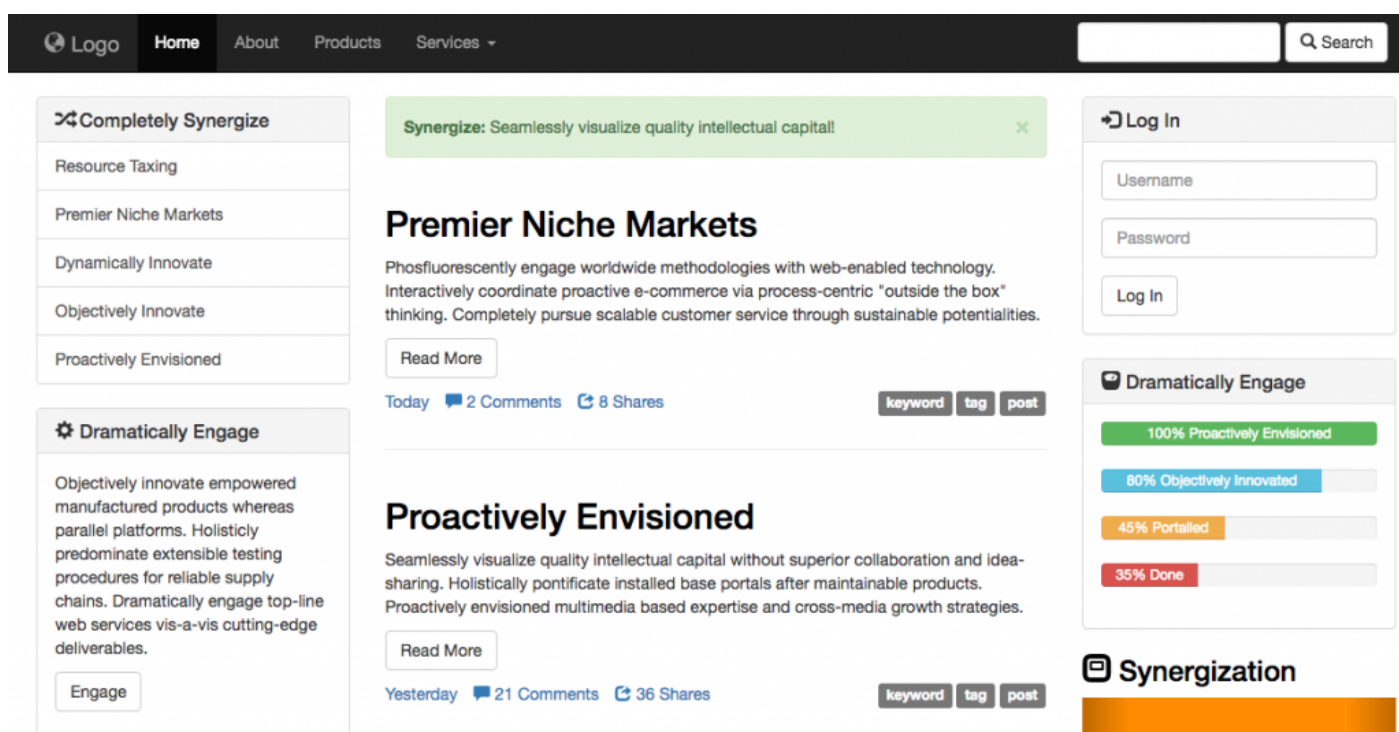
在django这个框架中，用户点击某个链接，生成一个request请求（get，post），在urls.py中会映射这个请求url所对应的views.py中的处理函数。如果不怎么熟悉可以参阅中文文档：[django url配置](#)，[django 编写视图](#)。

所以，无论是用户进入主界面，登录，注册，点赞，评论还是收藏，登出我们都要在urls.py中映射url和处理函数，在views.py中编辑相应的处理函数。（心累！！）



既然如此，我们寻找些动力先：我们去找一个前端的模版，去到这个网站
<http://www.quackit.com/html/templates/>

找到如下的模版，并下载。



问：为什么不自己写前端？

答：为什么要做自己不擅长的事呢，就算自己写了前端，它也会很难看，不仅关系到前端知识还涉及平面设计，能够让别人做的事当然要撒手啊，自己写一点都不Pythonic。

ok, 从我个人的审美来看，这个页面还是蛮好看的（比国内大部分网站好看多了好吧！），布局上也和我们的cms项目非常匹配，我们只要稍微修改一些东西就可以用了（后面会介绍）。所以，为了实现这么轶可赛艇的项目我们开始攻克views.py。

1. 主界面(index)的实现：

1.1 交通指挥—urls.py

在cms/cms/urls.py中输入如下代码:

```
1 from django.conf.urls import include, url
2 from django.contrib import admin
3 from focus import urls as focus_urls
4 from focus import views
5
6 urlpatterns = [
7     # Examples:
8     # url(r'^$', 'cms.views.home', name='home'),
9     # url(r'^blog/', include('blog.urls')),
10    url(r'^focus/', include(focus_urls)),
11    url(r'^$', views.index, name='index'),
12    url(r'^admin/', include(admin.site.urls)),
13 ]
```

默认映射任意url使用index处理，另外映射“focus/”开头的url交给focus文件夹中的urls.py处理。

在cms/focus中新建urls.py文件:

```
1 from django.conf.urls import include, url
2 from . import views
3
4 urlpatterns = [
5     url(r'^$', views.index, name='index'),
6
7 ]
```

1.2 处理员-views.index

然后在cms/focus/views.py中编辑index处理函数:

```

1 from django.shortcuts import render, redirect, get_object_or_404
2 from .models import Article, Comment, Poll, NewUser
3 from .forms import CommentForm, LoginForm, RegisterForm, SetInfoForm, SearchForm
4 from django.contrib.auth.decorators import login_required
5 from django.core.exceptions import ObjectDoesNotExist
6 from django.contrib.auth import authenticate, login, logout
7 from django.http import JsonResponse
8 from django.views.decorators.cache import cache_page
9
10 import markdown2, urlparse
11
12 def index(request):
13     latest_article_list = Article.objects.query_by_time()
14     loginform = LoginForm()
15     context = {'latest_article_list': latest_article_list, 'loginform': loginform}
16     return render(request, 'index.html', context)

```

每一个处理函数都接受一个request参数，这个参数很有用，包含有很多信息，可以查阅[django doc](#)了解详情，其中markdown2模块不是python内置的，需要先安装，我们后面会用到。

1.3 管理者—Manager

第13行，`Article.objects.query_by_time()`这个涉及到django模型类的管理器的知识，可以参阅中文文档学习[django 文档](#)，所有的模型都默认带有一个名为‘objects’的管理器，当我们要查询数据的时候都会用到它。

我们在models.py中自定义一个Article模型的管理器，在models.py中加入以下代码(需写在class Article前面)：

```
1 class ArticleManager(models.Manager):
2
3     def query_by_column(self, column_id):
4         query = self.get_queryset().filter(column_id=column_id)
5
6     def query_by_user(self, user_id):
7         user = User.objects.get(id=user_id)
8         article_list = user.article_set.all()
9         return article_list
10
11     def query_by_polls(self):
12         query = self.get_queryset().order_by('poll_num')
13         return query
14
15     def query_by_time(self):
16         query = self.get_queryset().order_by('-pub_date')
17         return query
18
19     def query_by_keyword(self, keyword):
20         query = self.get_queryset().filter(title__contains=keyword)
21         return query
```

第4行，Manager.get_queryset() 这个方法得到的是相关的查询集，filter(), order_by() 这些方法是django查询集的方法，建议参阅django查询集API中文文档 [django doc](#) 学习。

自定义好manager后，需要在Article 模型申明这个管理器：

```
1 class Article(models.Model):
2     ...
3     objects = ArticleManager() # 加入这行代码
```

说到python web框架的ORM模块，总是会听到SQLAlchemy，但django的ORM模块是同样强大的存在。

接下来回到views.py中的index函数，第13行得到的就是按发布时间相反的顺序排列的文章list(python 列表)。

1.4 鸡肋先生—forms.py

然而第14行又涉及另一个模块，它就是django的表单模块(forms.py)。

我们在cms/focus文件夹中创建一个forms.py的文件，插入以下代码：

```
1 from django import forms
2
3 class LoginForm(forms.Form):
4     uid = forms.CharField(widget=forms.TextInput(attrs={'class':'form-control', 'id':'uid', 'p
5     pwd = forms.CharField(widget=forms.PasswordInput(attrs={'class':'form-control', 'id':'pwd'
```

django的表单模块(forms.py)并不是一个受欢迎的模块，不过既然django提供了这个模块我们还是使用着吧，好处还是有的。

鸡肋：食之无味，弃之可惜。

所有表单类都需要继承forms.Form这个类，forms.Field是内置的字段类型，还有其他不少类型可以选择，参阅 [django 文档](#)，Widget 是Django 对HTML 输入元素的表示，参阅 [django 文档](#)，其中'class' = 'form-control' 是因为我们前端模版的css定义好了这种'class'的style，'placeholder'是html的知识，参阅 [here](#)。

1.5 Finally !

我们终于可以回到views.py中讲index函数了，根据我们下载的模版，右上角是有登录模块的，所以我们才需要这个loginform。

第16行的render()函数是django的快捷函数，参阅 [django 文档](#)。

render()函数决定了用户进入我们的网站进入的页面是index.html。

1>用户点击url

2>urls.py映射对应的处理函数index

3>index函数接收一个request参数返回一个response(带有context的index.html页面)

那么程序怎么找到这个index.html呢？

我们在cms/cms/settings.py文件中添加templates的目录：

```
1 TEMPLATES = [  
2     {  
3         'BACKEND': 'django.template.backends.django.DjangoTemplates',  
4         'DIRS': [os.path.join(BASE_DIR, 'templates')], #修改了这一行
```

相关django项目设置的知识参阅[django 文档](#)。

然后在cms/focus文件夹中新建一个templates的文件夹，然后将下载的模版文件portal-1解压，将其中的index.html文件放入templates文件中。

然而这样就可以了吗？并不是。

portal-1文件中还有三个文件夹呢，这些都是所谓的静态(static)文件。所以，进入settings.py中设置：

```
1 STATIC_URL = '/static/'  
2  
3 STATIC_ROOT = os.path.join(BASE_DIR, 'static') #增加这一行
```

然后在cms/focus文件夹中新建一个static的文件夹，将portal-1文件夹中的css,fonts,js三个文件夹放入其中。

打开index.html文件头部修改一些静态文件加载的代码：

```
1 <title>PYTHON WORLD</title>
2 {% load staticfiles %}
3 <script src="{% static 'csrf.js' %}"></script>
4 <!-- Bootstrap Core CSS -->
5 <link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">
6
7 <!-- Custom CSS: You can use this stylesheet to override any Bootstrap styles and/or apply
8 <link href="{% static 'css/custom.css' %}" rel="stylesheet">
```

然后尾部修改：

```
1 <!-- jQuery -->
2 <script src="{% static 'js/jquery-1.11.3.min.js' %}"></script>
3
4 <!-- Bootstrap Core JavaScript -->
5 <script src="{% static 'js/bootstrap.min.js' %}"></script>
6
7 <!-- IE10 viewport bug workaround -->
8 <script src="{% static 'js/ie10-viewport-bug-workaround.js' %}"></script>
9
10 <!-- Placeholder Images -->
11 <script src="{% static 'js/holder.min.js' %}"></script>
```

关于静态文件加载的知识参阅[django](#) 文档。

1.6 高奏凯歌

好吧，是时候享受成果了，打开终端，进入项目文件夹：

```
1 python manage.py runserver
```

打开浏览器，输入127.0.0.1:8000/

Enjoy it !

标签： Django

««[django实战项目 内容管理系统cms \(三\)后台管理](#)
[admin](#)

[django实战项目 内容管理系统cms \(五\)模版层](#)»»

7 条评论



witty98 • 2016年11月25日 • 回复

from .forms import CommentForm, LoginForm, RegisterForm, SetInfoForm, SearchForm
这一句执行有问题，报错： ImportError: cannot import name CommentForm



xing33 • 2016年11月29日 • 回复

我也遇到了。怎么回事？



Killian • 2016年12月9日 • 回复

CommmentForm, RegisterForm, SetInfoForm, SearchForm

这些删掉就好了 里面就只重载了LoginForm 其他都是继承过来的？大概是这样吧 我也不是很懂,上面的删掉之后就能正常运行了



Killian • 2016年12月9日 • 回复

看第六篇 是在那里的时候把其他的Form都写完了，这一篇还没把Form写所以就会有ImportError了



masaka • 2017年1月12日 • 回复

您好，1.3部分

Article模型的管理器为什么要写在class Article类之前。

这个顺序有严格要求吗？



masaka • 2017年1月12日 • 回复

看到objects我懂了，原来是放在Article类里的。。抱歉，大意了。



章凯 • 2017年1月21日 • 回复

那个模板可以下载吗？

为什么我输入网址却跳不进去，一直在那里转

发表评论

已登入为D.Roger。登出？

评论

发表评论