

个人站点首页搭建



1、功能需求:

- 使用Bootstrap框架，2-8布局
- 左边栏为文章的分类，标签，日期归档列表。
 - 点击不同的内容，页面会自动跳转。
- 右边栏为页面的主体内容，是一个列表形式，内容是各种各样的文章
- 点击文章的标题可以跳转到文章的详情页面

2、技术实现-前端:

○ 左边栏实现:

- 使用Bootstrap的面板组件。
- 组件中的每一个元素就是一个标签（超链接）
 - 标签内容是：标签的分类名(文章数量)
 - 例如：python(32)
 - java(30)
 - 2022年8月(20)
 - 当点击时可以跳转不同的页面
 - 但是始终在个人站点内跳转。页面的左边栏需要保持一致不变，的是右边栏的内容。
 - 所以将左边栏做成inclusion_tag标签。
 - <a>标签href的地址形式为：
 - href="用户名/category/category的id号 #文章分类
 - href="用户名/tag/tag的id号 #文章标签
 - href="用户名/archive/日期归档的id号 #文章日期归档
 - 左边栏中组件的中的元素的具体显示内容，由{% form item in obj_list %}标签控制

○ 右边栏实现。

- 和首页一样，使用Bootstrap框架中的媒体对象，显示。
- 具体的媒体对象的显示由{% for article in article_list %}标签控制。
- 文章标题使用超链接<a>点击可以实现跳转。

○ 模板继承:

- 由于个人站点首页，和文章归类的页面及其相似。

- 所以创建了一个base.html 这个页面，
 - 拥有基础的导航条功能，
 - 使用2-10布局
 - 左边栏直接嵌入inclusion_tag标签
 - 右边栏为自主修改区域{% block content %}
- 同时个人站点首页也继承这个页面

3、技术实现-后端：

○ url配置：

- 个人站点首页： `re_path(r"^(?P<username>\w+)/$", views.site, name="site")`
 - 使用用户名称作为个人站点的路径
- 侧边栏筛选： `re_path(r"^(?P<username>\w+)/(?P<condition>category|tag|archive)/(?P<param>.*)/", views.site),`
 - 此url可以同时匹配文章的分类、标签、日期归档等做个不同url。降低了冗余。

○ 左边栏：

○ inclusion_tag实现：

- 构造左边栏需要的数据，使用跨表查询
- 1、查询当前用户所有的分类，及分类下的文章数
- 2、查询当前用户所有的标签以及标签下的文章数
- 3、按照年月统计所有文章
 - 其中提取日期中的年月使用到了TruncMonth模块，此模块可以专门从日期对象中年月信息。
 - `from django.db.models.functions import TruncMonth`
 - 与之类似还有以下模块：
 - TruncMonth #提取年月
 - TruncDate
 - TruncDay
 - TruncHour
 - TruncMinute
 - TruncQuarter
 - TruncSecond
 - TruncTime

- TruncWeek
- TruncYear
- **右边栏:**
- 由于个人站点和侧边栏筛选，都都是用了同一个视图函数site(),所以视图函数可以接收多个参数。
 - set(request,username,**kwargs)
 - username #个人站点根路劲不可为空。
 - **kwargs值为空: #表示访问的是个人站点首页
 - **kwargs值不为空 #表示访问的是侧边栏
 - 如果访问个人站点首页，直接查询所有文章，然后返回html页面
 - 如果访问侧边栏，还需要判断访问的是侧边栏中的那个类型，
 - 然后根据类型，过滤对应的文章数据，然后返回html页面。

个人站点回顾

```

1 # 由于url方法第一个参数是正则表达式，所有当路由特别多的时候，可能会出现被顶替的情况，针对这种情况有两种解决方式
2 1.修改正则表达式
3 2.调整url方法的位置
4
5 """
6 1.页面布局不再是282 变成左右布局
7 2.个人站点每个人的样式都不一样内部大致如何实现的
8 内部给每个人开设了可以自定义css和js的文件接口并且用户自定义之后会将用户的文件保存下来，之后在打开用户界面的时候会自动加载用户自己写的css和js从而实现每个用户界面不一样的情况
9 3.侧边栏展示问题
10 只要你的orm学的没有问题一般情况下不会有难度
11 只要是queryset对象就可以无限制的点击queryset对象方法
12         filter().filter().filter().filter()
13
14         # 1 查询当前用户所有的分类及分类下的文章数
15         # 2 查询当前用户所有的标签及标签下的文章数
16         # 3 按照年月统计所有的文章
17         1.需要截取日期字段
18 参考官网提供的方法
19         from django.db.models.functions import TruncMonth

```

```

20         date_list =
models.Article.objects.filter(blog=blog).annotate(month=TruncMonth('create_time')).value
s('month').annotate(count_num=Count('pk')).values_list('month','count_num')
21
22         2.如果按照上述写法出现报错,那么需要你去配置文件中修改时区
23         TIME_ZONE = 'Asia/Shanghai'
24         USE_TZ = False
25 4.侧边栏筛选功能
26     1.多个url公用一个视图函数
27     2.当多个url公用一个视图函数的时候 你应该思考着多个url能不能优化一下
28     # 个人站点页面搭建
29     url(r'^(?P<username>\w+)/$',views.site,name='site'),
30     # 侧边栏筛选功能
31     # url(r'^(?P<username>\w+)/category/(\d+)/',views.site),
32     # url(r'^(?P<username>\w+)/tag/(\d+)/',views.site),
33     # url(r'^(?P<username>\w+)/archive/(\w+)/',views.site),
34     # 上面的三条url其实可以合并成一条
35     url(r'^(?P<username>\w+)/(?P<condition>category|tag|archive)/(?
P<param>.*)/',views.site)
36
37
38     if kwargs:
39     # print(kwargs) # {'condition': 'tag', 'param': '1'}
40     condition = kwargs.get('condition')
41     param = kwargs.get('param')
42     # 判断用户到底想按照哪个条件筛选数据
43     if condition == 'category':
44         article_list = article_list.filter(category_id=param)
45     elif condition == 'tag':
46         article_list = article_list.filter(tags__id=param)
47     else:
48         year,month = param.split('-') # 2020-11 [2020,11]
49         article_list =
article_list.filter(create_time__year=year,create_time__month=month)
50
51 5.手动补全侧边栏url即可
52 """

```

urls.py

```
1 #个人站点页面搭建
2 re_path(r"^(?P<username>\w+)/$", views.site, name="site"),
3 #侧边栏筛选
4 re_path(r"^(?P<username>\w+)/(?P<condition>category|tag|archive)/(?P<param>.*)/", views.site),
```

inclusion_tag标签

app01.template_tags.mytag.py

```
1 from django import template
2 from django.db.models.functions import TruncMonth
3
4 from app01 import models
5 from django.db.models import Count
6 register=template.Library()
7
8 #自定义inclusion_tag
9 @register.inclusion_tag("mytemplates/left_menu.html")
10 def left_menu(username):
11     user_obj=models.UserInfo.objects.filter(username=username).first()
12     blog=user_obj.blog
13     #构造侧边栏需要的数据
14     #1、查询当前用户所有的分类，及分类下的文章数
15
16     category_list=models.Category.objects.filter(blog=blog).annotate(count_num=Count("article__pk")).values_list("name", "count_num", "pk")
17     # print(category_list)
18     #2、查询当前用户所有的标签以及标签下的文章数
19
20     tag_list=models.Tag.objects.filter(blog=blog).annotate(count_num=Count("article__pk")).values_list("name", "count_num", "pk")
21     # print(tag_list)
22
23     #3、按照年月统计所有文章
24
25     date_list=models.Article.objects.filter(blog=blog).annotate(month=TruncMonth("create_time")).values("month").annotate(count_num=Count("pk")).values_list("month", "count_num")
```

```
22     # print(date_list)
23     return locals()
```

templates.mytemplates.left_menu.html

```
1  <div class="panel panel-danger">
2      <div class="panel-heading">
3          <h3 class="panel-title">文章分类</h3>
4      </div>
5      <div class="panel-body">
6          {% for category in category_list %}
7              <p><a href="/{{ username }}/category/{{ category.2 }}">{{ category.0 }}({{
category.1 }})</a></p>
8          {% endfor %}
9      </div>
10 </div>
11 <div class="panel panel-info">
12     <div class="panel-heading">
13         <h3 class="panel-title">文章标签</h3>
14     </div>
15     <div class="panel-body">
16         {% for tag in tag_list %}
17             <p><a href="/{{ username }}/tag/{{ tag.2 }}">{{ tag.0 }}({{ tag.1 }})</a>
</p>
18         {% endfor %}
19     </div>
20 </div>
21 <div class="panel panel-default">
22     <div class="panel-heading">
23         <h3 class="panel-title">日期归档</h3>
24     </div>
25     <div class="panel-body">
26         {% for date in date_list %}
27             <p><a href="/{{ username }}/archive/{{ date.0|date:"Y-m" }}">{{
date.0|date:"Y年m月" }}({{ date.1 }})</a> </p>
28         {% endfor %}
29     </div>
30 </div>
```

templates.base.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Site</title>
6     <link href="https://cdn.bootcss.com/twitter-bootstrap/3.4.1/css/bootstrap.min.css"
7     rel="stylesheet">
8     <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
9     <script src="https://cdn.bootcss.com/twitter-bootstrap/3.4.1/js/bootstrap.min.js">
10    </script>
11    <link rel="stylesheet" href="/media/css/{{ blog.site_theme }}"/>
12    {% block css %}
13    {% endblock %}
14 </head>
15 <body>
16 <!-- *****导 航 条***** -->
17 <nav class="navbar navbar-inverse">
18     <div class="container-fluid">
19         <!-- Brand and toggle get grouped for better mobile display -->
20         <div class="navbar-header">
21             <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
22             data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
23                 <span class="sr-only">Toggle navigation</span>
24                 <span class="icon-bar"></span>
25                 <span class="icon-bar"></span>
26                 <span class="icon-bar"></span>
27             </button>
28             <a class="navbar-brand" href="#">{{ blog.site_title }}</a>
29         </div>
30         <!-- Collect the nav links, forms, and other content for toggling -->
31         <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
32             <ul class="nav navbar-nav">
```

```

31         <li class="active"><a href="#">博客 <span class="sr-only">(current)
    </span></a></li>
32         <li><a href="#">文章</a></li>
33         <li class="dropdown">
34             <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">更多 <span class="caret">
    </span></a>
35             <ul class="dropdown-menu">
36                 <li><a href="#">Action</a></li>
37                 <li><a href="#">Another action</a></li>
38                 <li><a href="#">Something else here</a></li>
39                 <li role="separator" class="divider"></li>
40                 <li><a href="#">Separated link</a></li>
41                 <li role="separator" class="divider"></li>
42                 <li><a href="#">One more separated link</a></li>
43             </ul>
44         </li>
45     </ul>
46     <form class="navbar-form navbar-left">
47         <div class="form-group">
48             <input type="text" class="form-control" placeholder="Search">
49         </div>
50         <button type="submit" class="btn btn-default">Submit</button>
51     </form>
52     <ul class="nav navbar-nav navbar-right">
53         {% if request.user.is_authenticated %} <!--判断用户是否登录-->
54         <li><a href="#">{{ request.user.username }}</a></li> <!--如果登录显
示用户名已经相关的用户操作-->
55         <li class="dropdown">
56             <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">更多操作<span class="caret">
    </span></a>
57             <ul class="dropdown-menu">
58                 <li><a href="#" data-toggle="modal" data-target=".bs-
example-modal-lg">修改密码</a></li><!--当点击修改密码时，跳出模态框-->
59                 <li><a href="#">修改头像</a></li>
60                 <li><a href="#">后台管理</a></li>
61                 <li role="separator" class="divider"></li>
62                 <li><a href="{% url 'logout' %}">退出登录</a></li>
63             </ul>
64         <!-- Large modal 模态框-->

```


[illegible]

```

98         </div>
99     </div>
100
101     </li>
102     {% else %} <!--如果没有登录，显示注册和登录字样-->
103         <li><a href="{% url 'reg' %}">注册</a></li> <!--使用的是url别名-->
104         <li><a href="{% url 'login' %}">登录</a></li>
105     {% endif %}
106 </ul>
107 </div><!-- /.navbar-collapse -->
108 </div><!-- /.container-fluid -->
109 </nav>
110
111
112 <!--*****网页内容
*****-->
113 <div class="container-fluid">
114     <!--*****侧边栏区域*****-->
115     <div class="row">
116         <div class="col-md-2">
117             {% load mytag %}
118             {% left_menu username %}
119         </div>
120         <!--*****右边栏区域*****-->
121         <div class="col-md-10">
122             {% block content %}
123
124             {% endblock %}
125         </div>
126
127     </div>
128 </div>
129 {% block js %}
130
131 {% endblock js %}
132
133 </body>
134 </html>

```

templates.site.html

```
1 {% extends "base.html" %}
2
3 {% block content %}
4     {% for article_obj in article_list %}
5         <!--媒体对象-->
6         <ul class="media-list">
7             <li class="media">
8                 <h4 class="media-heading"><a href="/{{ username }}/article/{{
9 article_obj.pk }}">{{ article_obj.title }}</a> </h4>
10                 <div class="media-left">
11                     <a href="#">
12                         
14                     </a>
15                 </div>
16                 <div class="media-body">
17                     {{ article_obj.desc }}
18                 </div>
19                 <div class="pull-right">
20                     <span>posted&nbsp;&nbsp;&nbsp;</span>
21                     <span>@&nbsp;&nbsp;&nbsp;</span>
22                     <span>{{ article_obj.create_time|date:"Y-m-d"}}&nbsp;&nbsp;&nbsp;</span>
23                     <span>{{ article_obj.blog.userinfo.username }}&nbsp;&nbsp;&nbsp;</span>
24                     <span class="glyphicon glyphicon-comment">评论数{{
25 article_obj.comment_num }}&nbsp;&nbsp;&nbsp;</span>
26                     <span class="glyphicon glyphicon-thumbs-up">点赞数{{
27 article_obj.up_num }}&nbsp;&nbsp;&nbsp;</span>
28                     <span><a href="#">编辑</a></span>
29                 </div>
30             </li>
31         </ul>
32         <hr>
33     {% endfor %}
34 {% endblock %}
```

app01.views.site()

```
1 def site(request,username,**kwargs):
2     '''
3     kwargs :如果该参数有值,也就意味着,需要对article_list做额外的筛选操作
4     '''
5     #先校验当前用户名是否存在
6     user_obj=models.UserInfo.objects.filter(username=username).first()
7     #用户名如果不存在返回一个404页面
8     if not user_obj:
9         return render(request,"errors.html")
10    blog=user_obj.blog
11    #查询当前个人站点下的所有文章
12    article_list=models.Article.objects.filter(blog=blog)
13    if kwargs:
14        print(kwargs)
15        condition=kwargs.get("condition")
16        param=kwargs.get("param")
17        #判断用户到底想按照哪个条件筛选数据
18        if condition=="category":
19            article_list=article_list.filter(category_id=param)
20        elif condition=="tag":
21            article_list=article_list.filter(tags__id=param)
22        else:
23            year,month=param.split("-") #2022-11 [2022,11]
24    article_list=article_list.filter(create_time__year=year,create_time__month=month)
25
26    return render(request,"site.html",locals())
```
