

## 8-博客文章详情页

首页展示的是所有文章的列表，当用户看到感兴趣的文章时，他点击文章的标题或者继续阅读的按钮，应该跳转到文章的详情页面来阅读文章的详细内容。现在让我们来开发博客的详情页面，有了前面的基础，开发流程都是一样的了：首先配置 URL，即把相关的 URL 和视图函数绑定在一起，然后实现视图函数，编写模板并让视图函数渲染模板。

### 1、设计文章详情页的 URL

回顾一下我们首页视图的 URL，在 `blog/urls.py` 文件里，我们写了：

*【blog/urls.py】*

```
from django.conf.urls import url
from . import views

urlpatterns = [
    url(r'^$', views.index, name='index'),
]
```

首页视图匹配的 URL 去掉域名后其实就是一个空的字符串。对文章详情视图而言，每篇文章对应着不同的 URL。比如我们可以把文章详情页面对应的视图设计成这个样子：当用户访问 `<网站域名>/post/1/` 时，显示的是第一篇文章的内容，而当用户访问 `<网站域名>/post/2/` 时，显示的是第二篇文章的内容，这里数字代表了第几篇文章，也就是数据库中 Post 记录的 id 值。下面依照这个规则来绑定 URL 和视图：

*【blog/urls.py】*

```
from django.conf.urls import url
from . import views

app_name = 'blog'
urlpatterns = [
    url(r'^$', views.index, name='index'),
    url(r'^post/(?P<pk>[0-9]+)/$', views.detail, name='detail'),
]
primary key
```

Django 使用正则表达式来匹配用户访问的网址。这里 `r'^post/(?P<pk>[0-9]+)/$'` 整个正则表达式刚好匹配我们上面定义的 URL 规则。这条正则表达式的含义是，以 `post/` 开

头，后跟一个至少一位数的数字，并且以 / 符号结尾，如 post/1/、 post/255/ 等都是符合规则的，[0-9]+ 表示一位或者多位数。此外这里 (?P<pk>[0-9]+) 表示命名捕获组，其作用是从用户访问的 URL 里把括号内匹配的字符串捕获并作为关键字参数传给其对应的视图函数 detail。比如当用户访问 post/255/ 时（注意 Django 并不关心域名，而只关心去掉域名后的相对 URL），被括起来的部分 (?P<pk>[0-9]+) 匹配 255，那么这个 255 会在调用视图函数 detail 时被传递进去给到参数 pk，实际上视图函数的调用就是这个样子：detail(request, pk=255)。我们这里必须从 URL 里捕获文章的 id，因为只有这样才能知道用户访问的究竟是哪篇文章。

此外我们通过 app\_name='blog' 告诉 Django 这个 urls.py 模块是属于 blog 应用的，这种技术叫做视图函数命名空间（相当于 flask 的蓝本名+视图函数名）。我们看到 blog\urls.py 目前有两个视图函数，并且通过 name 属性给这些视图函数取了个别名，分别是 index、detail。但是一个复杂的 Django 项目可能不止这些视图函数，例如一些第三方应用中也可能有叫 index、detail 的视图函数，那么怎么把它们区分开来，防止冲突呢？方法就是通过 app\_name 来指定命名空间，命名空间具体如何使用将在下面介绍。如果你忘了在 blog\urls.py 中添加这一句，接下来你可能会得到一个 NoMatchReversed 异常。

为了方便地生成上述的 URL，我们在 Post 类里定义一个 get\_absolute\_url 方法（直接在模型中自己就生成了自己的 URL !!!），注意 Post 本身是一个 Python 类，在类中我们可以定义任何方法的。

【blog/models.py】

```
from django.db import models
from django.contrib.auth.models import User
from django.urls import reverse
from django.utils.six import python_2_unicode_compatible

@python_2_unicode_compatible
class Post(models.Model):
    ...

    def __str__(self):
        return self.title

    # 自定义 get_absolute_url 方法，用于模板中的 a 标签。
    # 记得从 django.urls 中导入 reverse 函数
    def get_absolute_url(self):
        return reverse('blog:detail', kwargs={'pk': self.pk}) key word
```

注意到 URL 配置中的 url(r'^post/(?P<pk>[0-9]+)/\$', views.detail, name='detail')，我们设定的 name='detail' 在这里派上了用场。看到这个 reverse 函数，它的第一个参数的值是 'blog:detail'，意思是 blog 应用下的 name=detail 的函数，由于我们在上面通

过 `app_name = 'blog'` 告诉了 Django 这个 URL 模块是属于 blog 应用的，因此 Django 能够顺利地找到 blog 应用下 name 为 detail 的视图函数，于是 `reverse` 函数会去解析这个视图函数对应的 URL，我们这里 detail 对应的规则就是 `post/(?P<pk>[0-9]+)/` 这个正则表达式，而正则表达式部分会被后面传入的参数 `pk` 替换，所以，如果 `Post` 的 `id`（或者 `pk`，这里 `pk` 和 `id` 是等价的）是 255 的话，那么 `get_absolute_url` 函数返回的就是 `/post/255/`，这样 `Post` 自己就生成了自己的 URL。

## 2、编写 detail 视图函数

接下来就是实现我们的 detail 视图函数了：

*【blog/views.py】*

```
from django.shortcuts import render, get_object_or_404
from .models import Post

def index(request):
    # ...

def detail(request, pk):
    post = get_object_or_404(Post, pk=pk)
    return render(request, 'blog/detail.html', context={'post': post})
```

视图函数很简单，它根据我们从 URL 捕获的文章 `id`（也就是 `pk`，这里 `pk` 和 `id` 是等价的）获取数据库中文章 `id` 为该值的记录，然后传递给模板。注意这里我们用到了从 `django.shortcuts` 模块导入的 `get_object_or_404` 方法，其作用就是当传入的 `pk` 对应的 `Post` 在数据库存在时，就返回对应的 `post`，如果不存在，就给用户返回一个 404 错误，表明用户请求的文章不存在。

## 3、编写详情页模板

我们看到 `index.html` 文件和 `detail.html` 文件除了 `main` 标签包裹的部分不同外，其它地方都是相同的，我们可以把相同的部分抽取出来，放到 `base.html` 里。首先在 `templates\` 目录下新建一个 `base.html` 文件，这时候你的项目目录应该变成了这个样子：

```
blogproject\
  manage.py
  blogproject\
    __init__.py
    settings.py
    ...
  blog\
    __init__.py
    models.py
```

```
""
templates\
base.html
blog\
index.html
detail.html
```

## ( 1 ) 设置基模板

对 index.html 做如下更改，作为基模板 base.html：

*【templates/base.html】*

```
<div class="content-body">

    <div class="container">

        <div class="row">

            <main class="col-md-8">

                {% block main %}

                {% endblock main %}

            </main>

            <aside class="col-md-4">

                {% block toc %}

                {% endblock toc %}

            <div class="widget widget-recent-posts">
```

## ( 2 ) index 模板

*【templates/blog/index.html】*

```
{% extends 'base.html' %}

{% block main %}

    {% for post in post_list %}

        <article class="post post-{{ post.pk }}">
```

```

        <header class="entry-header">

            <h1 class="entry-title">

                <a
href="{{ post.get_absolute_url }}">{{ post.title }}</a>

            </h1>

            <div class="entry-meta">

                <span class="post-category"><a
href="#">{{ post.category.name }}</a></span>

                <span class="post-date"><a href="#"><time
class="entry-date"

datetime="{{ post.created_time }}">{{ post.created_time }}</time></a
></span>

                <span class="post-author"><a
href="#">{{ post.author }}</a></span>

                <span class="comments-link"><a href="#">4 评论
</a></span>

                <span class="views-count"><a href="#">588 阅读
</a></span>

            </div>

        </header>

        <div class="entry-content clearfix">

            <p>{{ post.excerpt }}</p>

            <div class="read-more cl-effect-14">

                <a href="{{ post.get_absolute_url }}"
class="more-link">继续阅读 <span class="meta-nav">→</span></a>

            </div>

        </div>

    </article>

    {% empty %}

```

```
<div class="no-post">暂时还没有发布的文章！</div>
```

```
{% endfor %}
```

```
<!-- 简单分页效果
```

```
<div class="pagination-simple">
```

```
<a href="#">上一页</a>
```

```
<span class="current">第 6 页 / 共 11 页</span>
```

```
<a href="#">下一页</a>
```

```
</div>
```

```
-->
```

```
<div class="pagination">
```

```
<ul>
```

```
<li><a href="">1</a></li>
```

```
<li><a href="">...</a></li>
```

```
<li><a href="">4</a></li>
```

```
<li><a href="">5</a></li>
```

```
<li class="current"><a href="">6</a></li>
```

```
<li><a href="">7</a></li>
```

```
<li><a href="">8</a></li>
```

```
<li><a href="">...</a></li>
```

```
<li><a href="">11</a></li>
```

```
</ul>
```

```
</div>
```

```
{% endblock main %}
```

### ( 3 ) detail 模板

*【templates/blog/detail.html】*

```
{% extends 'base.html' %}

{% block main %}

    <article class="post post-{{ post.pk }}">

        <header class="entry-header">

            <h1 class="entry-title">{{ post.title }}</h1>

            <div class="entry-meta">

                <span class="post-category"><a
href="#">{{ post.category.name }}</a></span>

                <span class="post-date"><a href="#"><time
class="entry-date"
datetime="{{ post.created_time }}">{{ post.created_time }}</time></a>
</span>

                <span class="post-author"><a
href="#">{{ post.author }}</a></span>

                <span class="comments-link"><a href="#">4 评论
</a></span>

                <span class="views-count"><a href="#">588 阅读
</a></span>

            </div>

        </header>

        <div class="entry-content clearfix">

            {{ post.body }}

        </div>

    </article>

    <section class="comment-area" id="comment-area">
```

```
<hr>
```

```
<h3>发表评论</h3>
```

```
<form action="#" method="post" class="comment-form">
```

```
<div class="row">
```

```
<div class="col-md-4">
```

```
<label for="id_name">名字 :</label>
```

```
<input type="text" id="id_name" name="name"
required>
```

```
</div>
```

```
<div class="col-md-4">
```

```
<label for="id_email">邮箱 :</label>
```

```
<input type="email" id="id_email" name="email"
required>
```

```
</div>
```

```
<div class="col-md-4">
```

```
<label for="id_url">网址 :</label>
```

```
<input type="text" id="id_url" name="url">
```

```
</div>
```

```
<div class="col-md-12">
```

```
<label for="id_comment">评论 :</label>
```

```
<textarea name="comment" id="id_comment"
required></textarea>
```

```
<button type="submit" class="comment-btn">发表
</button>
```

```
</div>
```

```
</div> <!-- row -->
```

```
</form>
```



```
<div class="comment-list-panel">
```

```
<h3>评论列表，共 <span>4</span> 条评论</h3>
```

```
<ul class="comment-list list-unstyled">
```

```
<li class="comment-item">
```

```
<span class="nickname">追梦人物</span>
```

```
<time class="submit-date" datetime="2012-11-09T23:15:57+00:00">2017 年 3 月 12 日 14:56</time>
```

```
<div class="text">
```

文章观点又有道理又符合人性，这才是真正为了表达观点而写，不是为了迎合某某知名人士粉丝而写。我觉得如果琼瑶是前妻，生了三孩子后被一不知名的女人挖了墙角，我不信谁会说那个女人是追求真爱，说同情琼瑶骂小三的女人都是弱者。

```
</div>
```

```
</li>
```

```
<li class="comment-item">
```

```
<span class="nickname">zmrenwu</span>
```

```
<time class="submit-date" datetime="2012-11-09T23:15:57+00:00">2017 年 3 月 11 日 23:56</time>
```

```
<div class="text">
```

本能有可能会冲破格局，但格局有时候也会拘住本能。

```
</div>
```

```
</li>
```

```
<li class="comment-item">
```

```
<span class="nickname">蝙蝠侠</span>
```

```
<time class="submit-date" datetime="2012-11-09T23:15:57+00:00">2017 年 3 月 9 日 8:56</time>
```

```
<div class="text">
```

其实真理一般是属于沉默的大多数的。那些偏激的观点只能吸引那些同样偏激的人。前几年琼瑶告于妈抄袭，大家都表示大快人心，说明吃瓜观众都只是就事论事，并不是对琼瑶有偏见。

</div>

</li>

<li class="comment-item">

<span class="nickname">长江七号</span>

<time class="submit-date" datetime="2012-11-09T23:15:57+00:00">2017 年 2 月 12 日 12:56</time>

<div class="text">

观点我很喜欢！就是哎嘛本来一清二楚的，来个小三小四乱七八糟一团乱麻夹缠不清，简直麻烦要死

</div>

</li>

</ul>

</div>

</section>

{% endblock main %}

{% block toc %}

<div class="widget widget-content">

<h3 class="widget-title">文章目录</h3>

<ul>

<li>

<a href="#">教程特点</a>

</li>

<li>

```
<a href="#">谁适合这个教程</a>
```

```
</li>
```

```
<li>
```

```
<a href="#">在线预览</a>
```

```
</li>
```

```
<li>
```

```
<a href="#">资源列表</a>
```

```
</li>
```

```
<li>
```

```
<a href="#">获取帮助</a>
```

```
</li>
```

```
</ul>
```

```
</div>
```

```
{% endblock toc %}
```