

Django| 给你博客装个Markdown编辑器

原创 JAP君 Python进击者

2020-02-28原文

- 小白学Django - 给你博客装个Markdown编辑器

小白学Django系列:

- 小白学Django第一天| MVC、MVT以及Django的那些事
- 小白学Django第二天| Django原来是这么玩的!
- 小白学Django第三天| 一文带你快速理解模型Model
- 小白学Django第四天| Django后台管理及配置MySQL数据库
- 小白学Django第五天| 视图View的初步使用
- 小白学Django第六天| 一文快速搞懂模板的使用
- Django实战小型图书人物信息网页(MVT的综合运用)
- 小白学Django第七天| 模型类Model进阶学习
- 小白学Django第八天| 登录案例实战
- 持续更新中...

前言

最近自己在开发一个blog，因为比较喜欢用Markdown来写文章，而且目前很多平台都支持Markdown的语法，所以想给blog装个Markdown的编辑器。于是，就搜了一下，发现了django-mdeditor这个库，给大家分享一下。

简单介绍

Github地址：

<https://github.com/pylixm/django-mdeditor>

Django-mdeditor是基于Editor.md的一个django Markdown 文本编辑插件应用。

Django-mdeditor的灵感参考自项目 [django-ckeditor](#)

<https://github.com/django-ckeditor/django-ckeditor>

后端编辑器使用

1.安装相关库

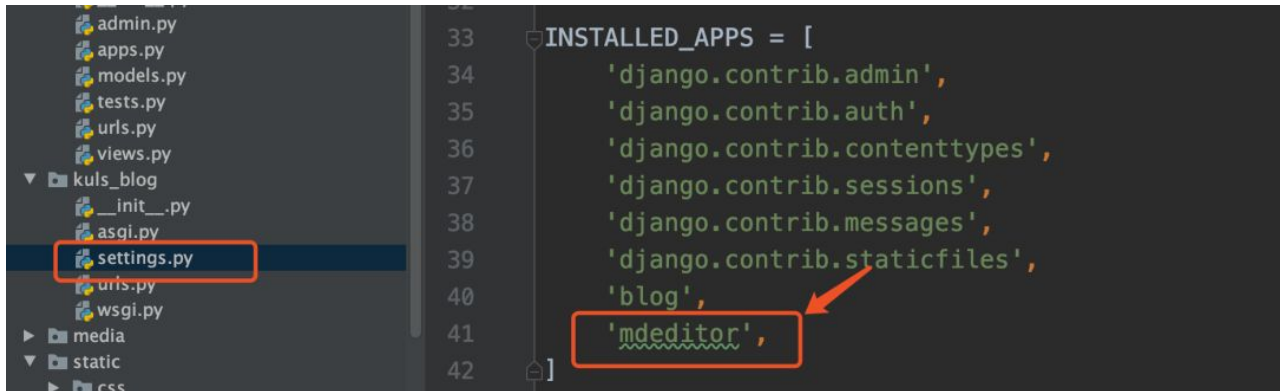
```
pip install django-mdeditor # 用于后台编辑
```

```
pip install markdown # 用于前端显示
```

首先大家先安装这两个库，django-mdeditor库就是用在管理后台的md编辑器，markdown则是在前端显示时使用。

2.配置

安装完两个库后，我们需要进行相关的配置。



新增setting配置：

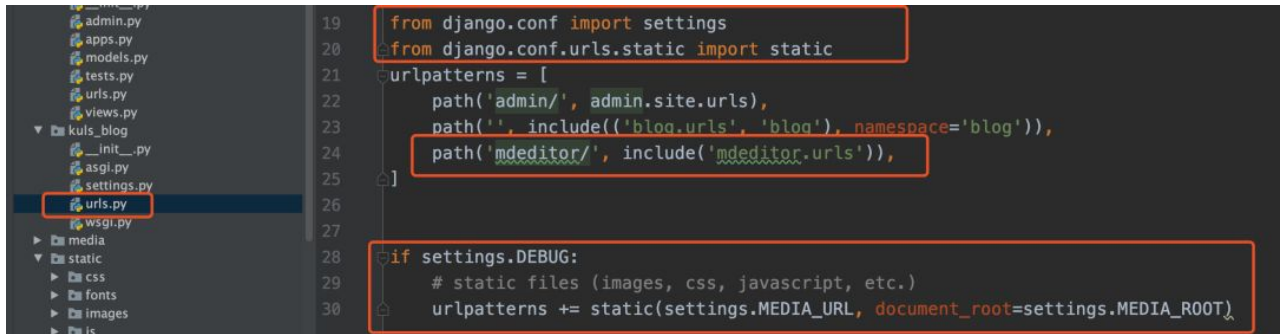
```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'blog',  
    'mdeditor',  
]
```

除了配置上面的信息，还需要配置资源文件夹：

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

```
MEDIA_URL = '/media/' #你上传的文件和图片会默认存在/media/editor下
```

以为就完了？不，你还需要去url进行配置：



```
19 from django.conf import settings
20 from django.conf.urls.static import static
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('', include(('blog.urls', 'blog'), namespace='blog')),
24     path('mdeditor/', include('mdeditor.urls')),
25 ]
26
27
28 if settings.DEBUG:
29     # static files (images, css, javascript, etc.)
30     urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

点击放大

大家把我打红框的代码弄上去就ok了

这时，我们就大概配置完成了。

3.使用Markdown



```
2 class Acticle(models.Model):
3     title = models.CharField(max_length=128, name="article title", default='')
4     category = models.ForeignKey(Category, on_delete=models.CASCADE, null=True, blank=True)
5     tags = models.ManyToManyField(Tag, blank=True)
6     body = MDTextField()
7     user = models.ForeignKey(User, on_delete=models.CASCADE, null=True, verbose_name="author")
8     read_nums = models.PositiveIntegerField(verbose_name="阅读数", default=0)
9     recommend = models.ForeignKey(Recommend, on_delete=models.CASCADE, null=True, verbose_name="推荐")
10    submit_time = models.DateTimeField(verbose_name="提交时间", auto_now_add=True)
11    upload_time = models.DateTimeField(verbose_name="上传时间", blank=True)
```

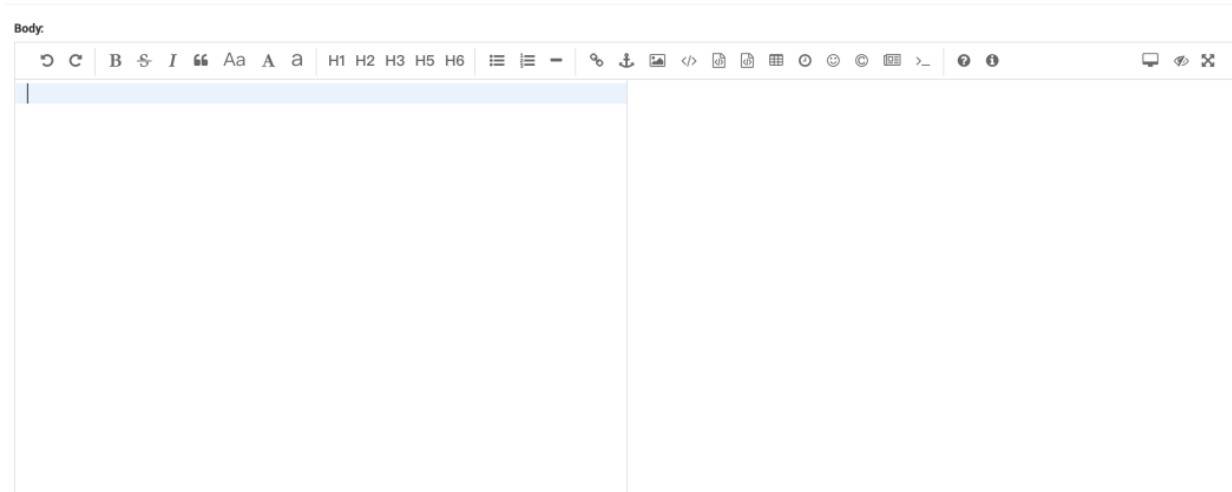
点击放大

此时只需要在model中填写相应的属性，即可调用该编辑器。

当然，在进入管理页面之前，你需要在admin中进行注册

```
admin.site.register(Acticle) # Acticle 是我文章的model名
```

打开后台之后，我们就会发现Markdown编辑器出现了：



我们在这里插入的图片或者上传的文件都会在`media`文件夹中，这个文件夹在上面配置中提到，必须要有！！！！

前端使用

我们使用了Markdown编辑器编写的文章在前端显示时，必须得将Markdown语法“翻译”成富文本的形式，所以这里我们需要使用到`markdown`这个库。

视图函数

```
pip install markdown
```

我们书写的博客文章内容存在`Post`的`body`属性里，回到我们的详情页视图函数，对`post`的`body`的值做一下渲染，把Markdown文本转为HTML文本再传递给模板：

```
import markdown

from django.shortcuts import render, get_object_or_404
```

```

from .models import Post

def post_article(request, pk):
    post = get_object_or_404(Post, pk=pk)
    # 记得在顶部引入 markdown 模块
    post.body = markdown.markdown(post.body,
                                    extensions=[
                                        'markdown.extensions.extra',
                                        'markdown.extensions.codehilite',
                                        'markdown.extensions.toc',
                                    ])
    return render(request, 'blog/detail.html', context={'post': post})

```

可能有些朋友不懂`get_object_or_404`方法，给大家简单介绍一下`get_object_or_404`：我们原来调用django的`get`方法(`model.object.get()`)，如果查询的对象不存在的话，会抛出一个`DoesNotExist`的异常，现在我们调用django`get_object_or_404`方法，它会默认的调用django的`get`方法，如果查询的对象不存在的话，会抛出一个`Http404`的异常，我感觉这样对用户比较友好，如果用户查询某个产品不存在的话，我们就显示404的页面给用户，比直接显示异常好。

markdown.extensions.extra： 用于标题、表格、引用这些基本转换

markdown.extensions.codehilite： 用于语法高亮

markdown.extensions.toc： 用于生成目录

替换网页模板

在模板中找到展示博客文章的地方加上如下代码，注意这里需要使用`safe`过滤器。

```
<div>
    {{ post.body|safe }}
</div>
```

通过这样就能够显示md语法的文章了

总结

在开发过程中遇到的一个小需求就分享给大家，当然我介绍的只是mdeditor的一部分知识，mdeditor还有一些相关的配置，这里我就没给大家说了，大家可以去GitHub上自行看他们的官方文档，顺便也可以去star一下！

原创不易！转发在看！

- 小白学Django第一天| MVC、MVT以及Django的那些事
- 小白学Django第二天| Django原来是这么玩的！
- 小白学Django第三天| 一文带你快速理解模型Model
- 小白学Django第四天| Django后台管理及配置MySQL数据库
- 小白学Django第五天| 视图View的初步使用
- 小白学Django第六天| 一文快速搞懂模板的使用
- Django实战小型图书人物信息网页(MVT的综合运用)
- 小白学Django第七天| 模型类Model进阶学习
- 小白学Django第八天| 登录案例实战
- 持续更新中...

设为星标 ★



JAVAandPython君

📍 扫码关注不迷路

技术干货 · 经验分享 · 成长心得 · 系统教程

精选留言

暂无...