目录

[个人博客设计 1](#_Toc510530051)

[数据库设计 1](#_Toc510530052)

[部署ckeditor 2](#_Toc510530053)

# 个人博客设计

### 数据库设计

用户：包含个人的基本信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 类型 | 注释 | 含义 |  |
| username |  | 用户名 |  | Not null |
| Password |  | 密码 |  | Not null |
| Firstname |  |  |  | Null |
| Lastname |  |  |  | Null |
| Email |  | 邮箱 |  | Null |
| Birth |  | 生日 |  | Null |
| Cellphone |  | 电话 |  | Null |
| QQ |  |  |  | Null |
| WorkState |  | 工作状态 |  | Null |
| Age |  | 年龄 |  | Null |
| WorkAge |  | 工作年龄 |  | Null |
| Address |  | 现居住地址 |  | Null |
| Hobby |  | 爱好 |  | Null |
|  |  |  |  |  |

## 部署ckeditor

#### 环境部署

#### Admin

**包安装**

* Pip install django-ckeditor
* Pillow是python的图像处理库，django-ckeditor需要依赖该库

Pip install pillow

**环境配置**

* Python manage.py startapp account
* 在settings.py中添加app：

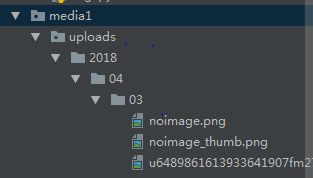
1. Account
2. Ckeditor，ckeditor\_uploader

* settings.py文件中，添加CKEDITOR\_UPLOAD\_PATH配置项:

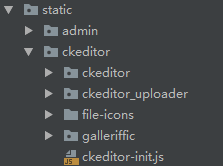
https://www.cnblogs.com/starof/p/4682812.html

1. MEDIA\_URL =’/media/’
2. MEDIA\_ROOT = os.path.join(BASE\_DIR,’media1’)(此处为了展示所以命名+1)
3. STATIC\_URL = ‘/static/’
4. STATIC\_ROOT = os.path.join(BASE\_DIR,’static’)
5. CKEDITOR\_UPLOAD\_PATH = ‘uploads/’

自动生成目录展示：



**需要指出的是**：在开发阶段，这样设置settings.py已经足够了。但是，到了正式部署你的应用时，你需要设置好STATIC\_ROOT和STATIC\_URL，并运行manage.py collectstatic命令，该命令会将ckeditor相关的静态资源拷贝到你的工程下。



注1：此处需要了解绝对路径和相对路径

绝对路径：指目录下的绝对位置，直接到达目标位置

相对路径：指由这个文件所在的路径引起的跟其它文件（或文件夹）的路径关系

[绝对路径](https://www.baidu.com/s?wd=%E7%BB%9D%E5%AF%B9%E8%B7%AF%E5%BE%84&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLuAmknHRzPju-nHmLm1D10ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHbzP161njfzP1bYnWn1njb3Ps)需要输入完整的文件路径，比如D:\images\郊游照片\2016年郊游照片\photo1.jpg，而[相对路径](https://www.baidu.com/s?wd=%E7%9B%B8%E5%AF%B9%E8%B7%AF%E5%BE%84&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLuAmknHRzPju-nHmLm1D10ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHbzP161njfzP1bYnWn1njb3Ps" \t "https://zhidao.baidu.com/question/_blank)的结构一般是：在同一个目录：photo1.jpg 或./photo1.jpg； 在下一个目录：xxx/photo1.jpg； 在下下个目录：xxx/xxx/photo1.jpg； 在上一个目录：../photo1.jpg； 在上上个目录：../../photo1.jpg

注2：此处需要了解django setting.py的路径配置

* （<http://www.mamicode.com/info-detail-1184722.html>）

BASE\_DIR=os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))  #代表mysite目录

STATIC\_URL = ‘/static/‘

STATIC\_ROOT = os.path.join(BASE\_DIR, ‘collected\_static‘)   #相当于 mysite/collected\_static

mysite/

├── collected\_static

├── manage.py

└── mysite

├── \_\_init\_\_.py

├── settings.py

├── urls.py

└── wsgi.py

* <https://blog.csdn.net/huobanjishijian/article/details/72669477>

<https://blog.csdn.net/fighter_yy/article/details/41249033>

MEDIA\_ROOT: 主要是为了存放上传的文件，比如在ImageField中，这个值加上upload\_to的值就是真实存放上传图片的文件位置；Django里边文件内容实际上是不会存放到数据库里边的，大多数数据库存放数据效率低，需要保存在文件系统里。PS：[FileUploads](https://docs.djangoproject.com/en/1.4/topics/http/file-uploads/" \t "_blank)(用来存储文件)

MEDIT\_URL: URL的映射，前后要加上‘/’ 表示从根目录开始，比如“/site\_media/”，加上这个属性之后，静态文件的链接前面会加上这个值。

STATIC\_ROOT: 在这个文件里边的目录会当成静态文件处理。但是不能把自己辛苦写的Javascript或者图片等静态文件放进去。

STATIC\_URL: URL映射，指定静态目录的URL，默认的是”/static/”。

* STATICFILES\_DIRS：指定一个工程里边哪个目录存放了与这个工程相关的静态文件，是一个列表。如果列表中有一个是 “/home/shishang/test/static”，其中有一个文件内容是productlist.html, 我们只要访问<http://localhost:8000/static/productlist.html>就可以直接访问界面了。
* 转：Django提供了一个方法自动地将所有的静态文件放在一起。只要在写App的时候创建一个static子目录专门保存静态文件就行了。在开发阶段，不必费心去做映射，不需要配置urls.py。在布署到生产环境的时候，只需要配置Apache把/static/映射到STATIC\_ROOT。然后运行manage.py collectstatic，自动地STATICFILES\_DIRS列出的目录以及各个App下的static子目录的所有文件复制到STATIC\_ROOT。因为复制过程可能会覆盖掉原来的文件，所以，一定不能把我们辛苦做出来静态文件放这边！在开发阶段，Django把/static映射到django.contrib.staticfiles这个App。Staticfiles自动地从STATICFILES\_DIRS、STATIC\_ROOT以及各个App的static子目录里面搜索静态文件。一旦布署到开发环境上，settings.py不需要重新编写，只要在Apache的配置文件里面写好映射，/static将会被Apache处理。Django.contrib.staticfiles虽然仍然存在，但因为不会接收到以/static/开始的路径，所以将不会产生作用。不必担心Django会使用处理速度变慢。另外，当settings.DEBUG is False的时候，staticfiles将自动关闭。

[html] [view plaincopy](http://blog.csdn.net/fuwencaho/article/details/41146871)

1. #setting.py 中加入：
3. import os
4. HERE = os.path.dirname(os.path.dirname(\_\_file\_\_))
6. MEDIA\_ROOT = os.path.join( HERE, “../media”).replace(‘\\’, ‘/’)
7. MEDIA\_URL = “/site\_media/”
9. STATIC\_ROOT = os.path.join(HERE, “../static”).replace(‘\\’, ‘/’)
10. STATIC\_URL = “/static/”
12. STATICFILES\_DIRS = ( os.path.join(HERE, “../app名字/static/”).replace(‘\\’, ‘/’), )  #和STATIC\_ROOT最好路径不要重叠
13. #建议将静态文件保存在app的static目录中
15. #urls.py中加入：
17. from django.conf.urls.static import static
18. from django.conf import settings
20. urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)
21. urlpatterns += static(settings.STATIC\_URL, document\_root=settings.STATIC\_ROOT)
23. #这样使用
24. <link  type=”text/css”  rel=”stylesheet” href=”/static/css/bootstrap.min.css”/>
25. <link  type=”text/css”  rel=”stylesheet” href=”/static/css/bootstrap-responsive.css”/>

* 修改urls.py：

添加：url(r'ckeditor/', include('ckeditor\_uploader.urls'))

Urlpatterns +=static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)（若不写，上传的图片将不能正常显示C:\Users\xucheng\Desktop\noimage.png）为静态文件分配多个不同的网址

* RichTextUploadingField():允许上传图片

from ckeditor\_uploader.fields import RichTextUploadingField  
from django.db import models  
class Blog(models.Model):  
 title = models.CharField(max\_length=50, verbose\_name="标题")  
 content = RichTextUploadingField(blank=True, null=True, verbose\_name="内容")  
  
 def \_\_str\_\_(self):  
 return self.title