**文件用途 和 部分重要代码 的注释**

**\ser**

**manage.py 管理文件**

execute\_from\_command\_line(sys.argv) #用于执行从命令行传入的参数

**\ser\ser**

**\_\_init\_\_.py 一个空文件，告诉 Python 该目录是一个 Python 包**

**wsgi.py 基于Python的网站服务器接口（本实验不会修改该文件）**

**settings.py 配置文件**

INSTALLED\_APPS 中添加 ’data’

DATABASES 如django.db.backends.sqlite3 表示本项目中使用的数据库

**urls.py一份由 Django 驱动的网站"目录"**

url(r'^admin/', admin.site.urls) #将web视图与URL绑定

**\ser\data**

**urls.py \_\_init\_\_.py 同上**

**admin.py apps.py 无重要内容**

**models.py 数据库中表的属性**

time = models.CharField(max\_length=100) #表示表中的一个属性特征

list\_display=('time','ident','value') #表示网页端显示参数

search\_fields = ['time',] #表示网页端搜索功能的搜索范围，可以修改

**views.py 交互功能代码**

**receive函数**

t = request.GET.get('time') #获取客户端的参数

d = models.Data(time=t,ident=i,value=v) #生成一条数据记录

d.save() #存入服务器数据库

**send函数**

data = models.Data.objects.filter(time\_\_gte=time1) #过滤早于time1的数据

data = data.filter(time\_\_lte=time2) #过滤晚于time2的数据

for i in range(0,5):

if type1[i] == '1': #判断查询数据的类型

l.append(data.filter(ident=i)) #只选择要查询的数据

s=""

for data in l:

for d in data:

s+=d.time

s+="\_"

s+=d.ident

s+="\_"

s+=d.value

s+="\n" #返回消息构造 time\_id\_value+’\n’ 格式

s+="aaa" #用于判断数据是否已经结束