**一：创建项目环境**

**1. 查看系统环境：**

python和mysql

**2. 安装虚拟环境：**

virtualenv -p /usr/local/python3.6/bin/python3 --no-site-packages python\_env

激活虚拟环境：source activate

**3.安装django以及创建ironfort项目：**

django-admin startproject ironfort

**4. 配置数据库**

create database ironfort charset utf8;

**5. 在Pycharm中远程编辑代码**

**6. 创建和注册app**

**二：设计模型models和测试用例**



**1. 远程主机**

主机ip可重复，但是ip加端口不重复。联合唯一索引

**2. 远程主机账号**

**3.绑定远程主机账号**

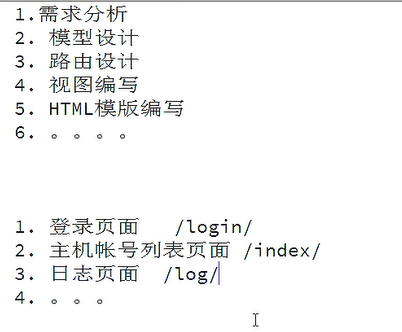
**4. 用户组：**

**5. 用户：**

用django自带的auth

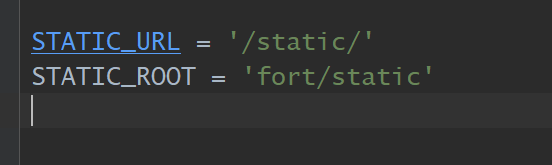
创建超级用户并且在admin里面注册

**三：路由和登录界面**

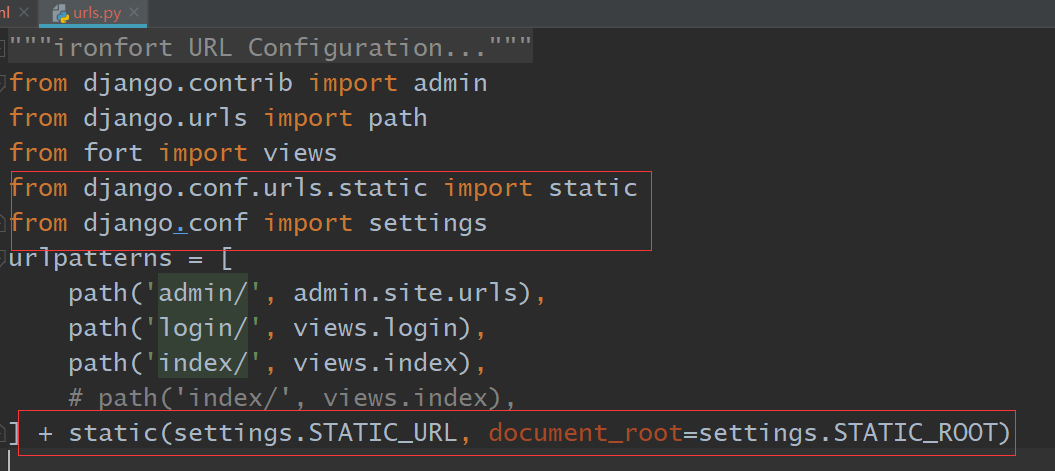


1. 如果不用nginx处理静态文件，则需要为静态文件编写路由

settings配置

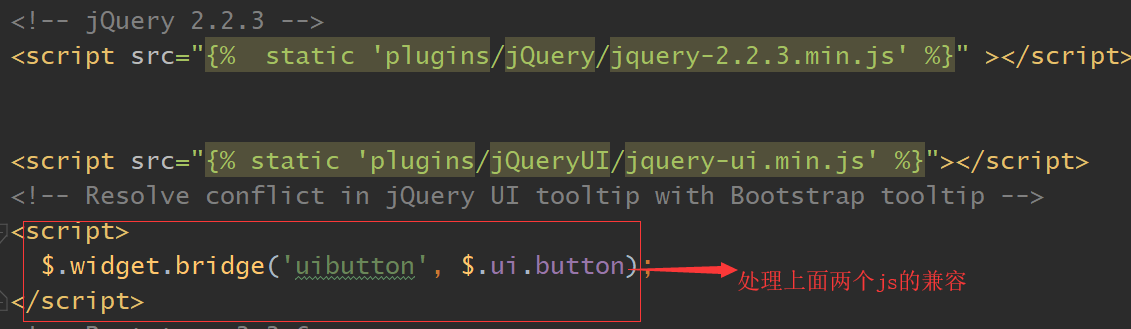


根路由配置



**四：AdminLTE框架使用：**

**静态文件导入是有顺序的，且很重要**



**五：Auth认证系统和用户登录：**

**用login装饰器来限制在url强制访问:即不能直接在url中访问**

django自带的Auth认证框架里面有：

from django.contrib.auth.decorators import login\_required

**没加括号，把没有合法登录的跳转到默认页面**

**通过加括号和指定参数来跳转到指定页面**

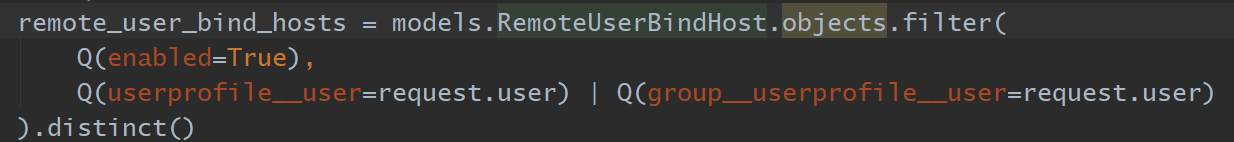
@login\_required(login\_url='/login/')

def index(request):

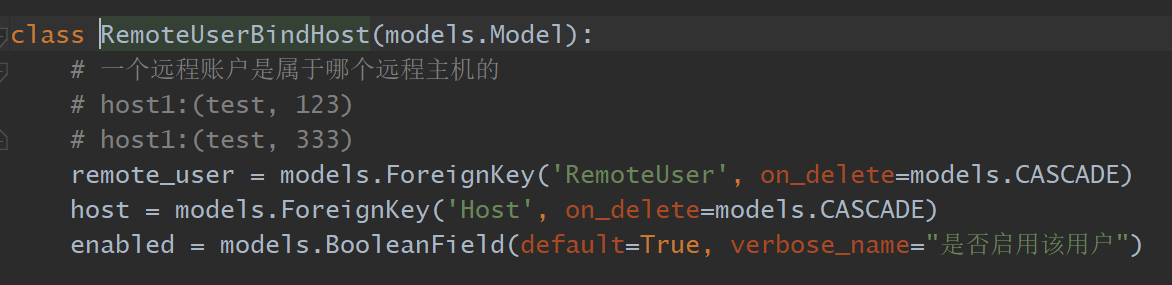
pass

**五：主机账号页面以及DataTable使用：**

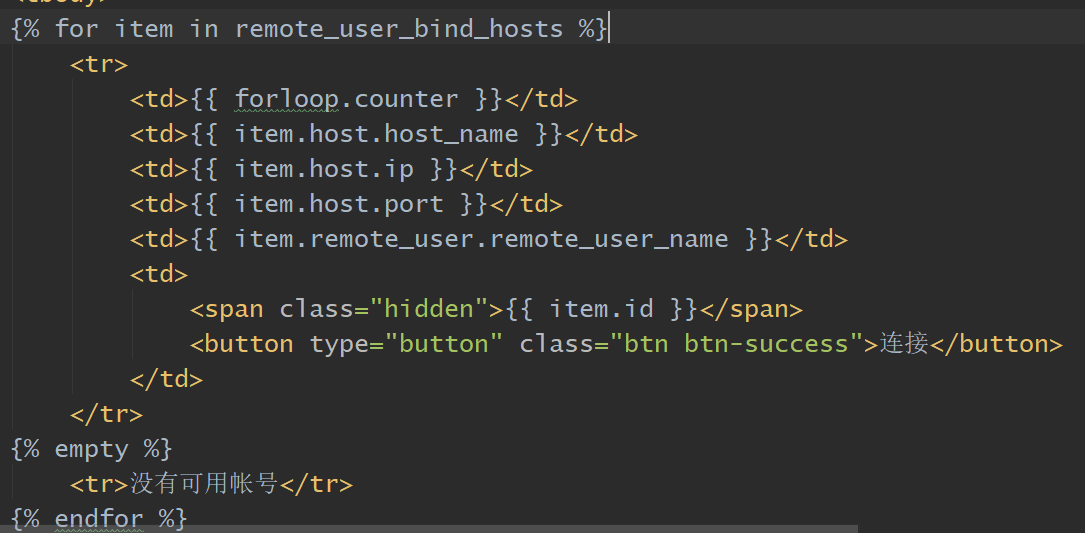
**多值联合查询：用Q查询**



Models表关系



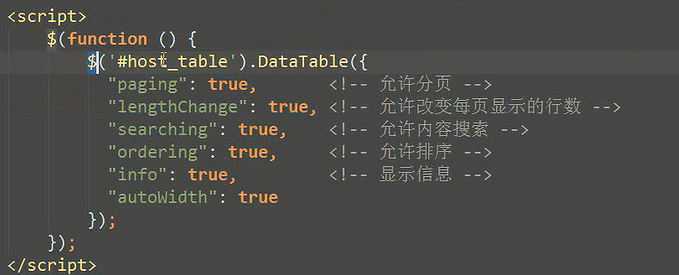
在html处理RemoteUserBindHost对象



**去除查询过滤：distinct()**

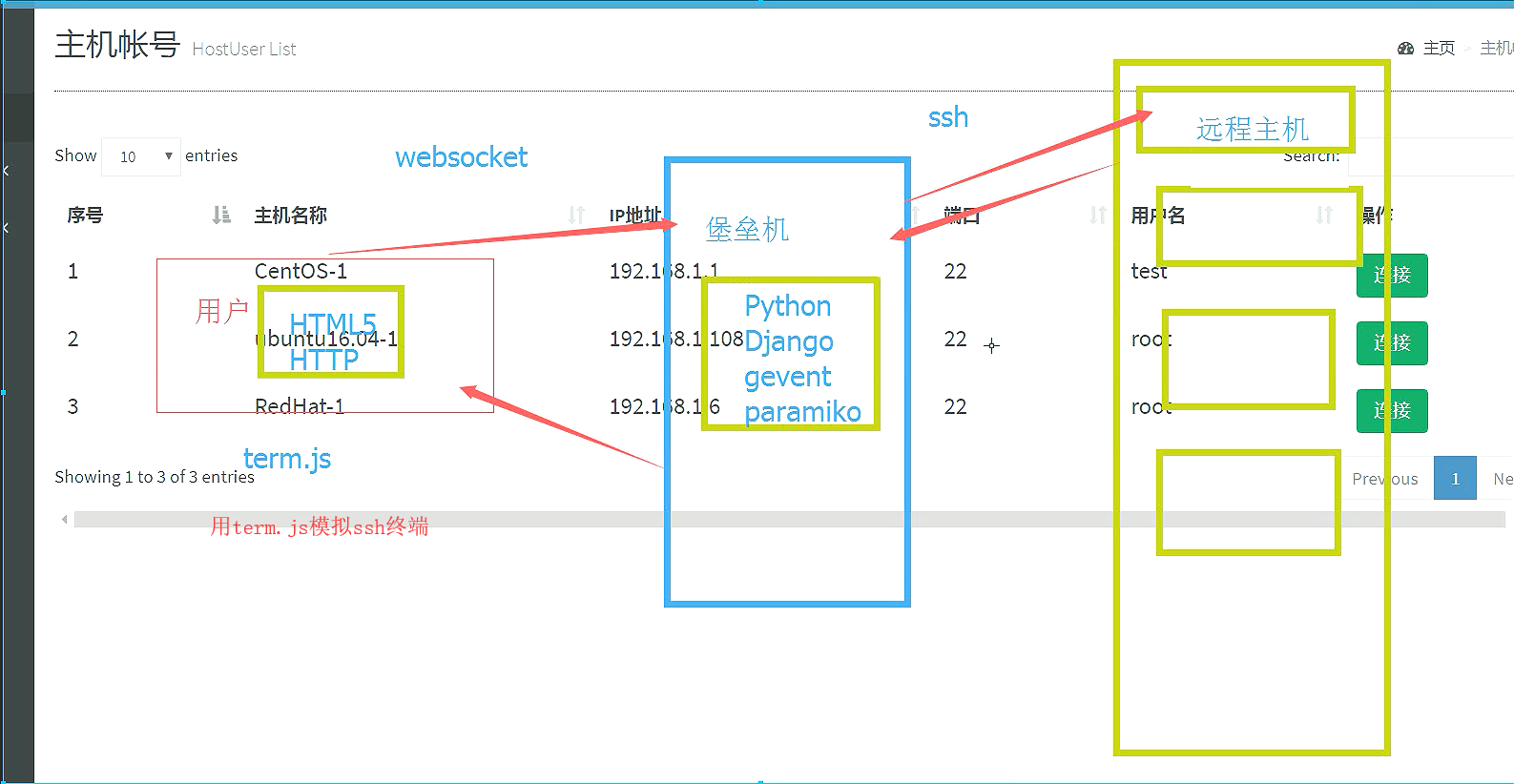
table-responsive：响应式表格

启动DataTable表：



**六：Websocket使用：**

1.打开链接要打开一个Websocket通道



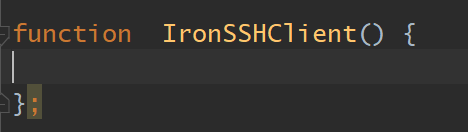
**Websocket**



**七：创建Websocket通道和Term模拟终端：**

**编写iornssh.js**

**0. 编写IronSSHClient函数：**

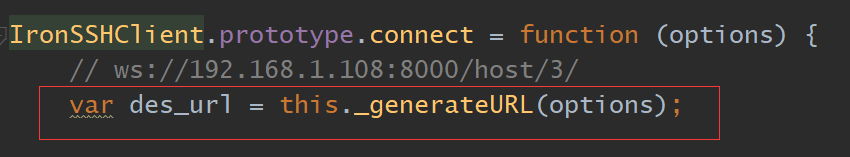


**1. 编写\_generateURL，目的是判断协议并返回构造的url:**

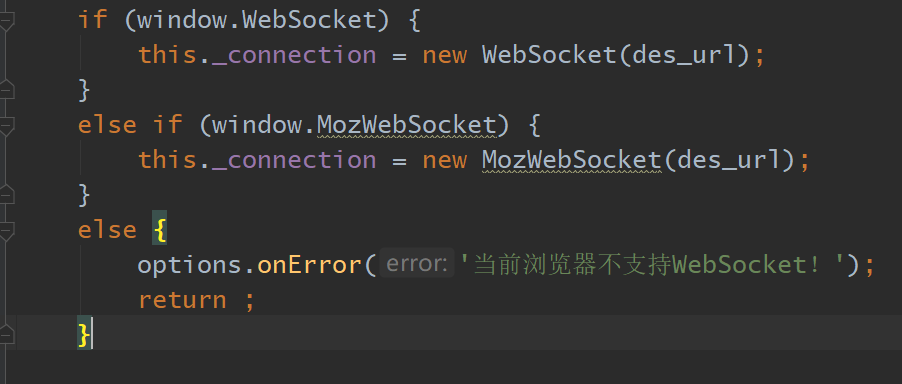


**2.编写IronSSHClient的connect方法：**

1.接受要连接的url地址



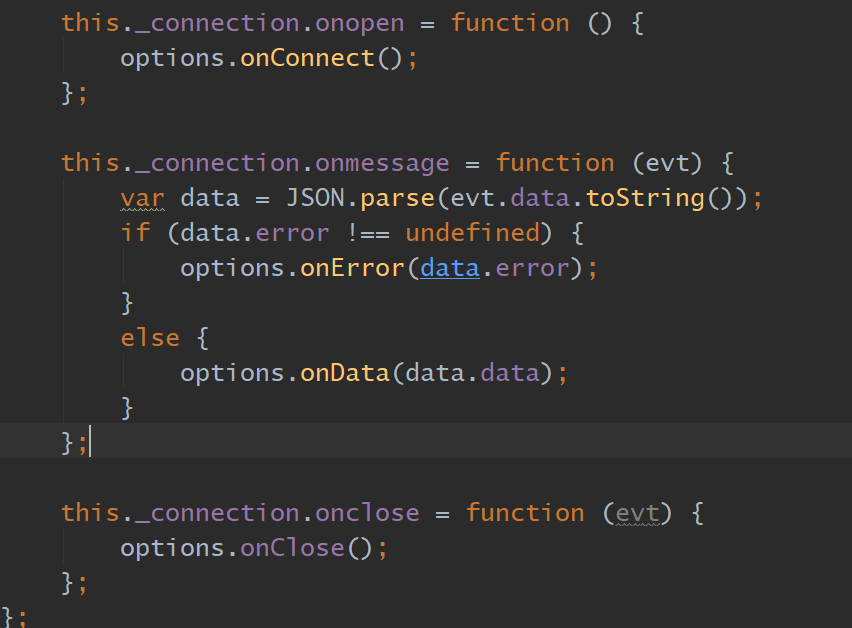
2.判断当前是啥浏览器，并为该浏览器创造Websocket通道



oopen:打开的时候执行什么方法

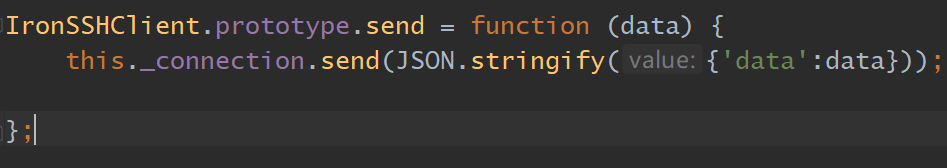
onmessage: 有消息过来

onclose: 关闭



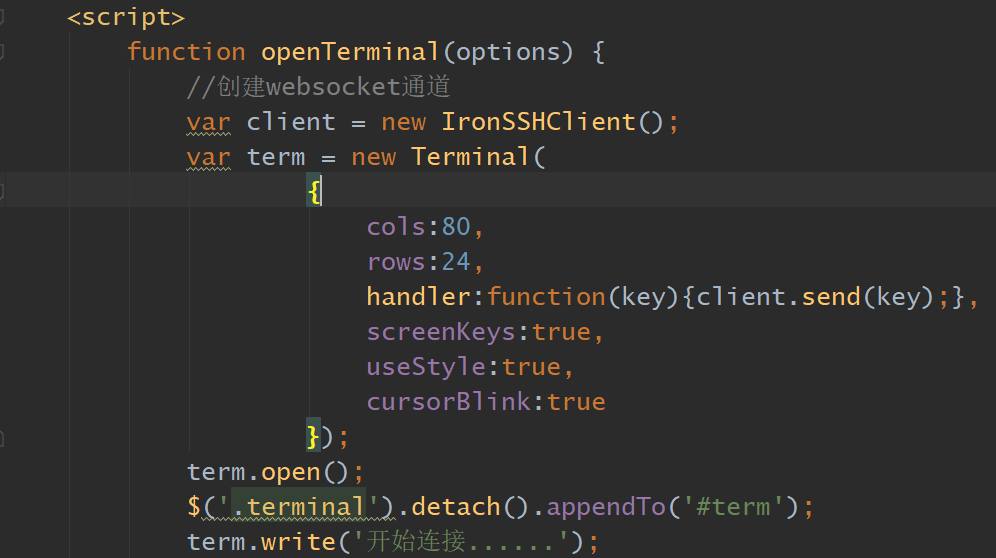
**3. 编写IronSSHClient的send方法：**

send：服务器往外发送数据并打包成JSON格式的字典



**在index.html编写<script></script>**

通过Term.js：即浏览器模拟终端



1. 初始IronSSHClient对象

2. 初始Terminal终端

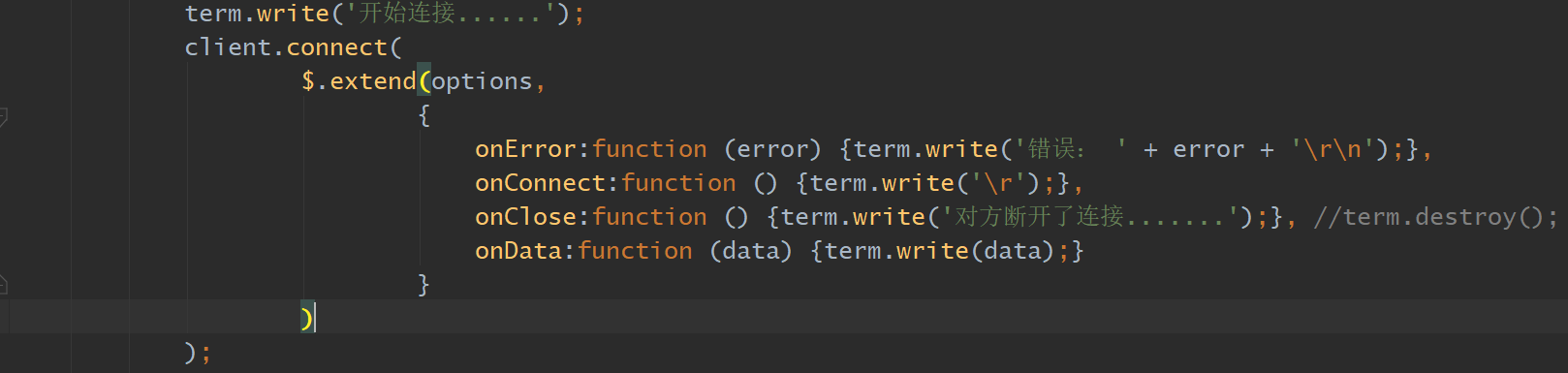
3. 调用term.open()开启终端

4. 打开终端并添加到容器里面，即添加到id为元素为term里面里

clipboard.png

5. 调用IronSSHClient.prototype.connect方法

6. 给options(本身为字典)extend(扩展内容)，以字典的方式拓展内容，成为一个新的options字典

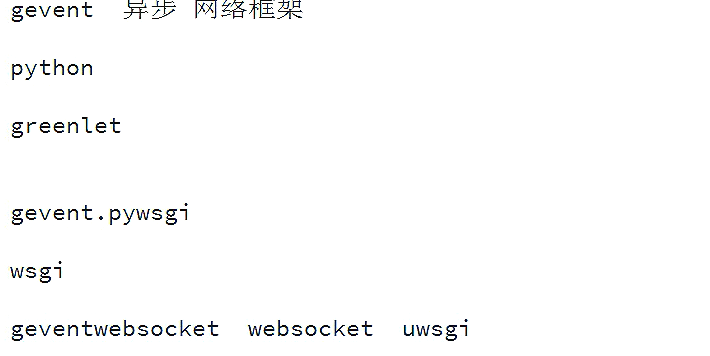


7. 在网页上点击连接按钮时执行如下



**八：使用gevent实现异步通信服务器：**

**django是一个 同步web框架，不支持异步请求**



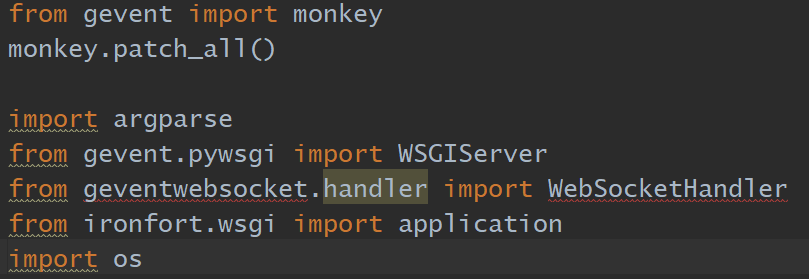
**在项目里面创建start\_ironfort.py**

1. pip install gevent 异步框架

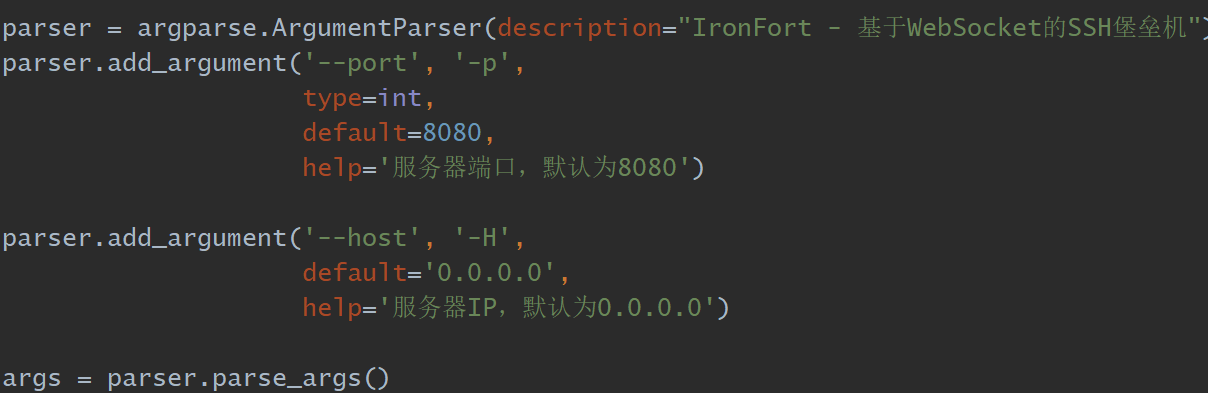
pip gevent-websocket gevent WSGIweb 服务器

pip install argparse 命令运行参数分析

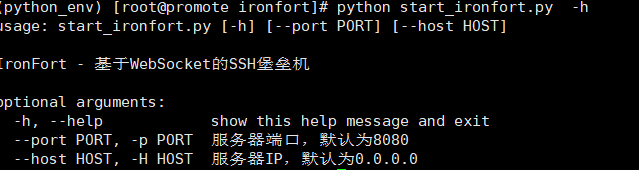
2.



**monkey.patch\_all()**  把同步阻塞通信全部变为异步非阻塞通信，放首位



通过parser分析传递过来的参数



以0.0.0.0：8080运行我们的服务器



用gevent连接上application

创建WSGIweb 服务器：

参数以元组的形式提供，1. 提供host和port 2. 指定用哪个应用来处理这些请求application 3. log记录 4.处理方式 handler——class来处理这些

clipboard.png

通过运行脚本来运行Django服务器

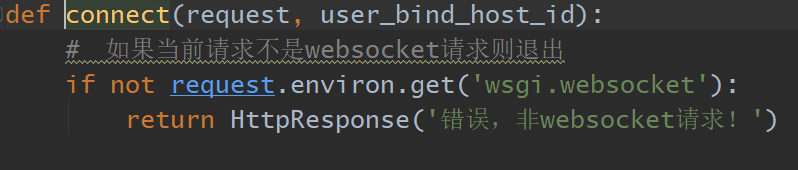
核心是gevent的WSGI服务器和WebSocketHandler处理器

**九：Django接受websocket请求**

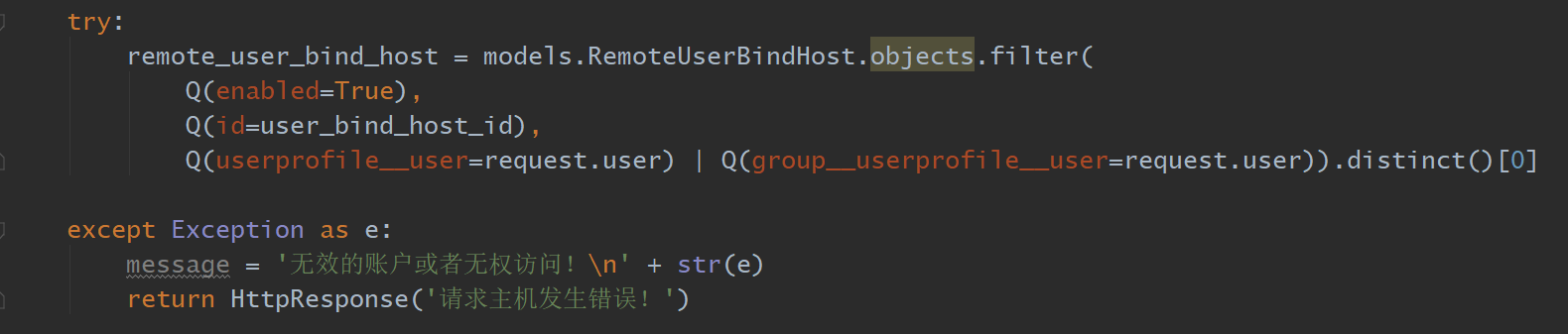
**1. 在urls里面添加如下url来处理websocket请求：**

path('host/<int:user\_bind\_host\_id>/', views.connect),

2. 判断请求方式

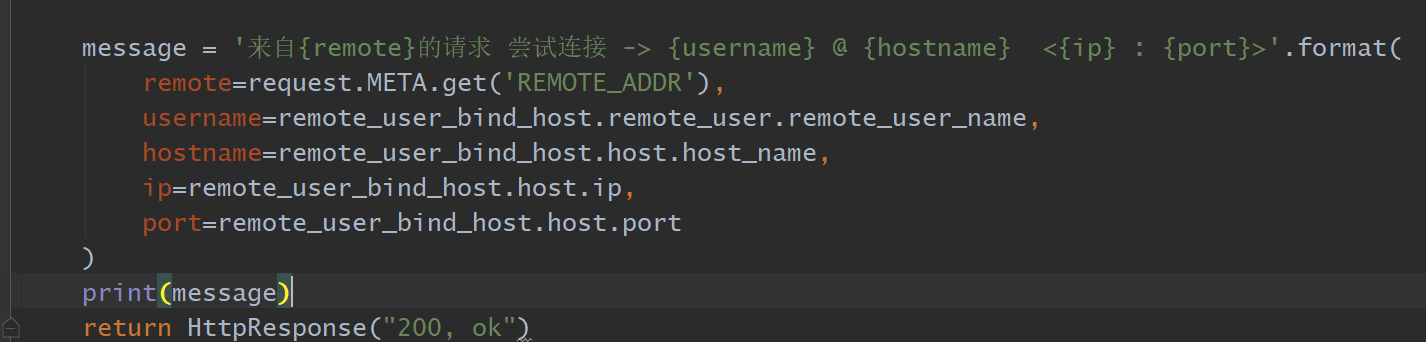


3. 限制只有自己访问权限的用户才可以访问

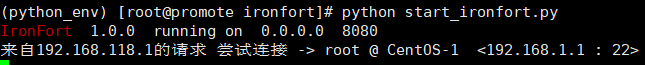


4. 输出访问信息，访问的是哪一个主机账号

REMOTE\_ADDR:浏览器用户的IP地址



5. 运行如下：



**十：桥接websocket和ssh实现异步通信**

1. 用户从浏览器通过html5的websocket在走http通道发送一条命令，比如：ls

2. 到了堡垒机，堡垒机通过gevent的WSGIServer服务器转发给Django，Django通过路由urls找到

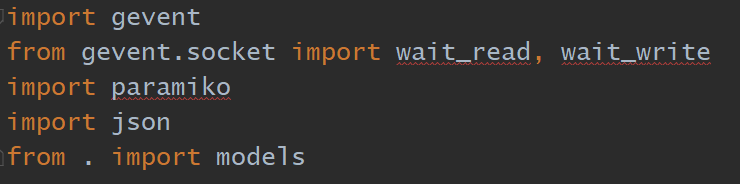
path('host/<int:user\_bind\_host\_id>/', views.connect),然后到views下的connect里面

**需要我们在堡垒机上开通一个ssh通道——Paramiko模块**

**用python实现ssh，一般用Paramiko模块**

**1.在ironfort这个app里面编写server.py**

wait\_read等待读和wait\_write等待写，只能在linux下才能看见

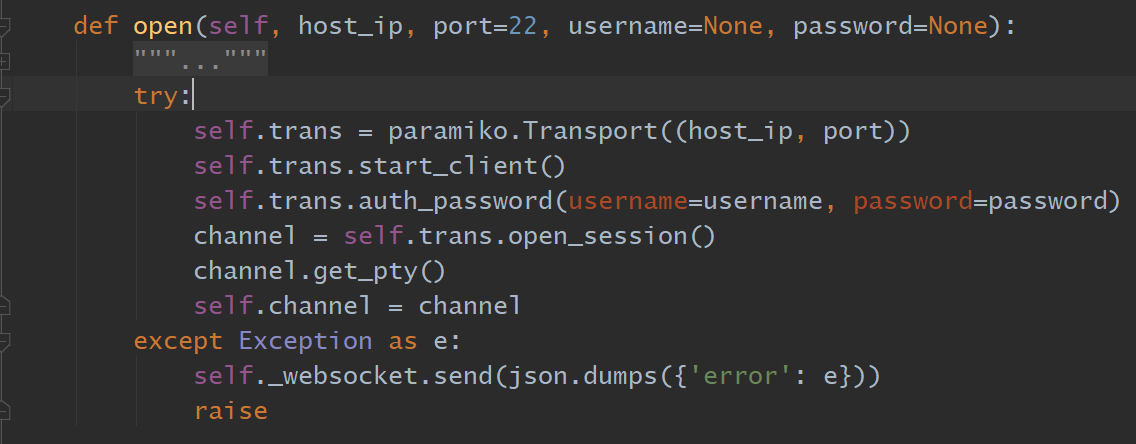


**2. 编写桥接类初始化方法**



**3. 编写连接ssh的open()方法**

打开通道，启动客户端

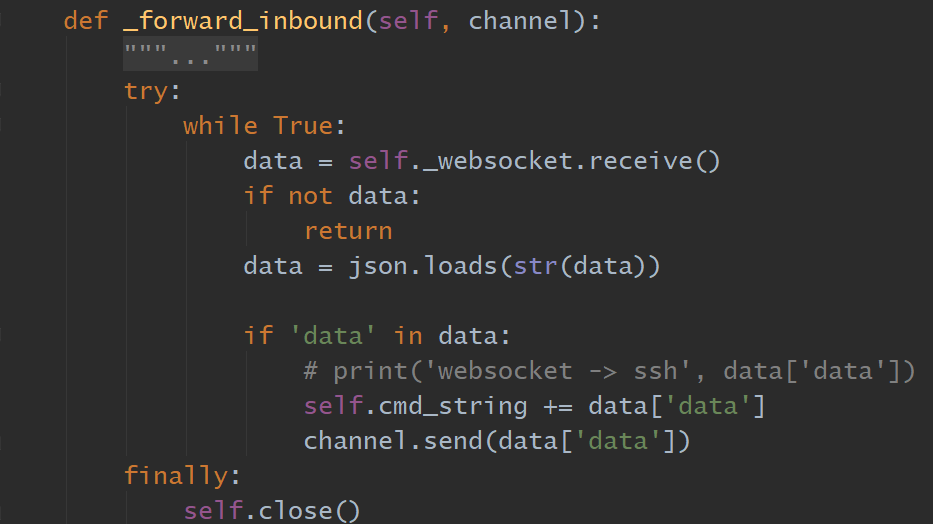


**4.正向发送数据 从用户通过websocket向ssh发送数据**

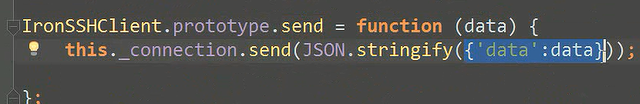
正向：web网页(websocket)——>堡垒机(ssh)——>远程主机

反向：远程主机——>堡垒机(ssh)——>web网页(websocket)

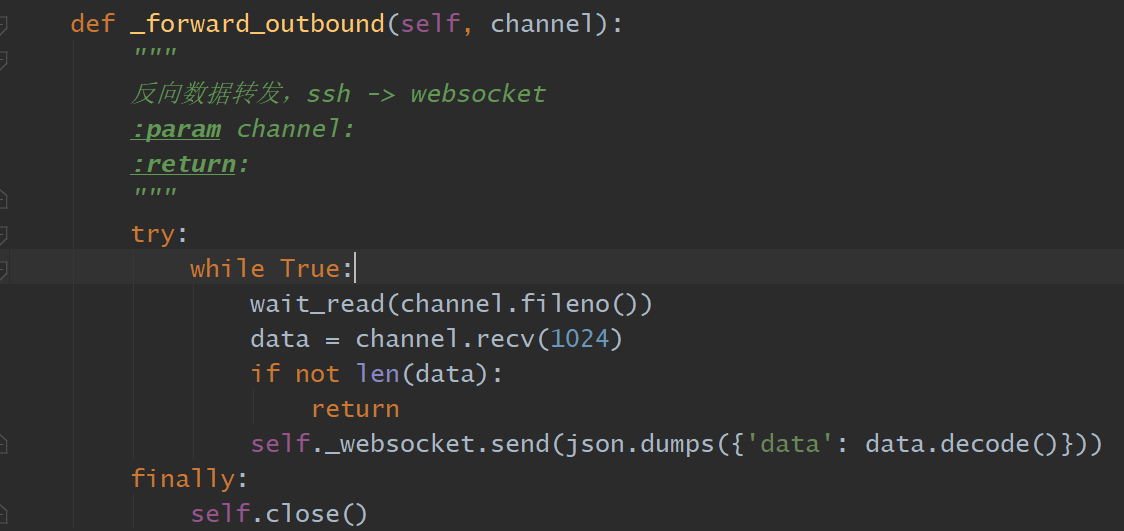
普通用户通过send方法把websocket的数据发送给Django服务器或者是gevent服务器



接受数据，通过loads转化为python可以操作的对象



**5. 接受用户发送的数据并保存带data**

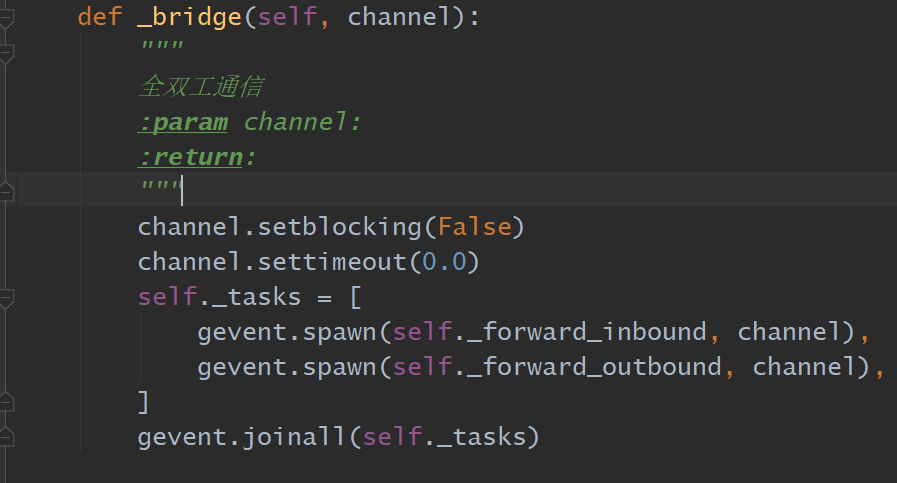


**6. 通过\_bridge类把发送数据和接受数据有序地进行**

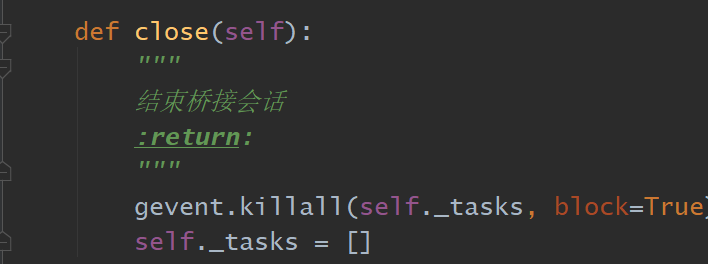
gevent.spawn()绑定任务，能在内部生成两个协程

channel.setblocking(False)和channel.settimeout(0.0)

即设置为非阻塞通道，无等待的实时通信



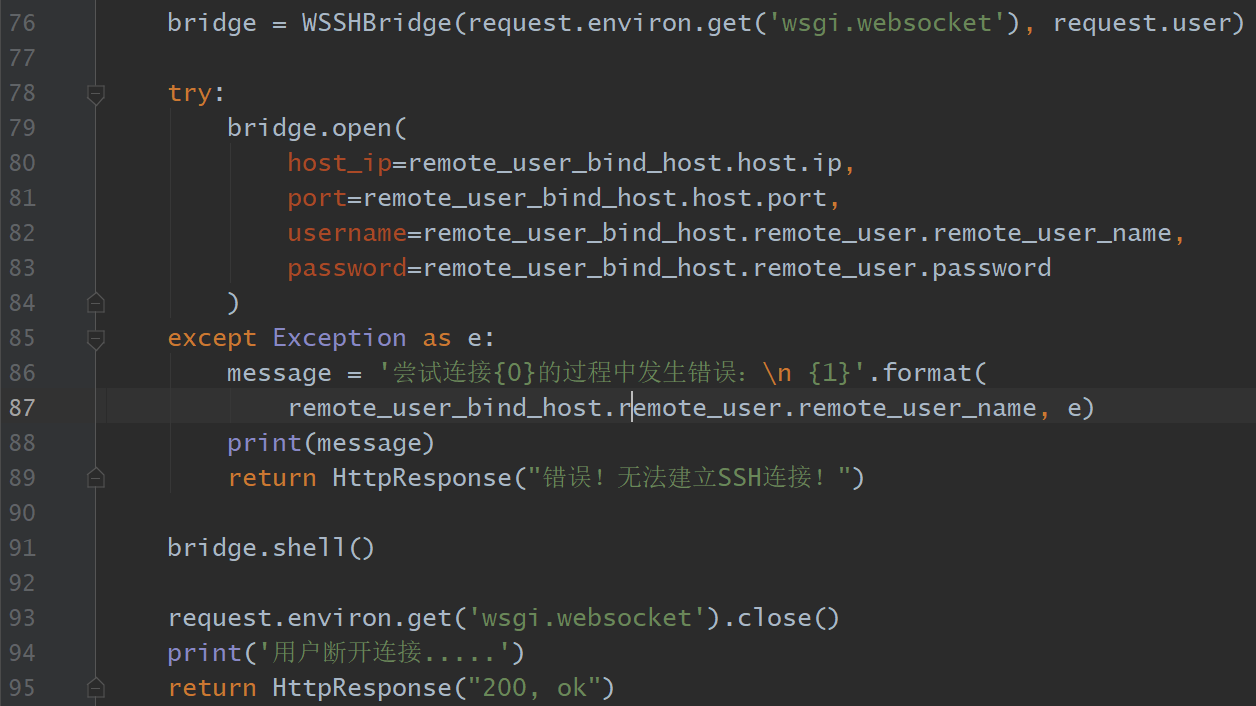
**7. 结束会话**



**8. 启动shell**



**9. 在views里面添加如下内容来调用server.py里面的内容：**



**十一：用户行为审计日志系统**

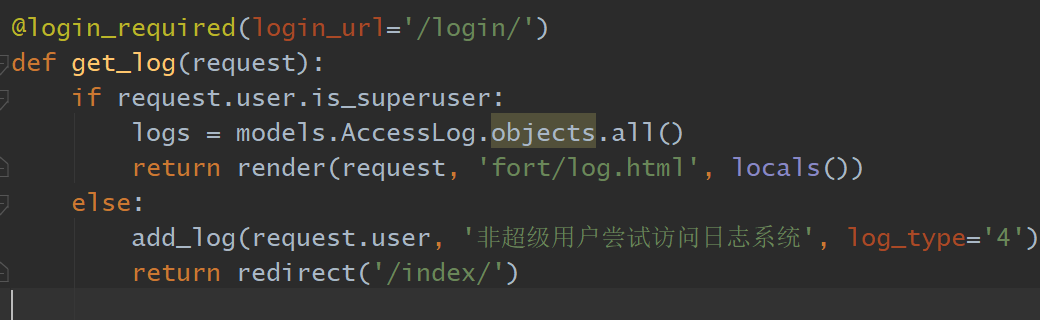
**1. 编写url**

path('log/', views.get\_log),

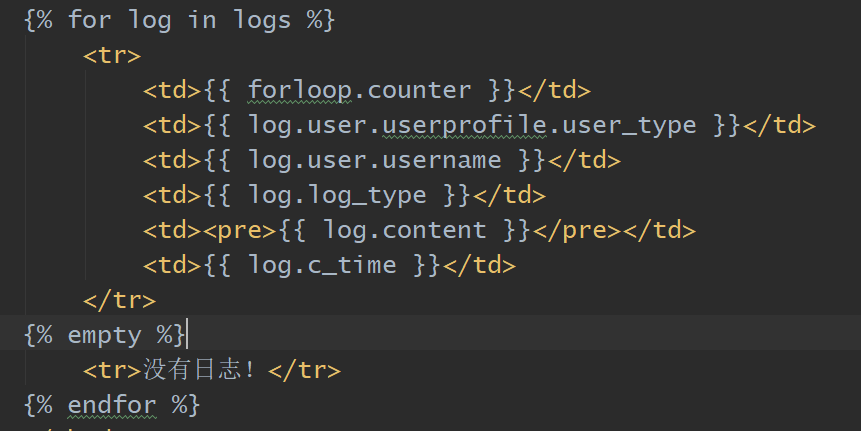
**2. 在server.py编写add\_log()函数(避免在views下形成循环导入)：**



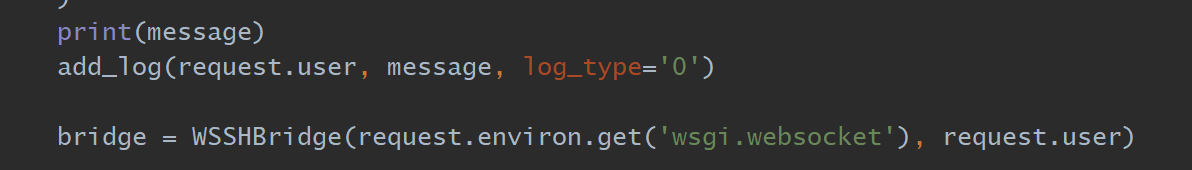
**3. 在views编写函数并且调用add\_log()**



**4. 编写log.html(如下部分)**



**5. 在views里面添加日志记录：**





**注：如果访问admin时出现如下情况，说明admin的静态文件导入有问题需要把admin静态文件手动添加到static(在虚拟环境下的python\_env/lib/python3.6/site-packages/django/contrib/admin/static/admin下)**



