1. **需求和目的**

请求12306模拟浏览器下单，审核和获取支付链接地址。

1. **环境和部署**

语言：python3.6+

框架：抓取selenium，展示flask

服务器：centos7+ 1核1G1M 美团云

数据库：mongodb 2.x+

Git地址：http://gitlab.hangtian123.com.cn/train\_beijing/headless/tree/dev4.0

服务器地址：见服务器excel.

**1.先更新系统到最新yum update -y**

**2.再安装其他需要的系统安装包**

yum install libffi-devel liberation-fonts pax\* redhat-lsb\* libXss\* libappindicator\* zlib\* openssl-devel bzip2-devel expat-devel gdbm-devel readline-devel sqlite-devel gcc gcc-c++ openssl-devel zbar mongodb mongodb-server dos2unix –y yum -y install lrzsz

**3.安装python3环境**

wget https://www.python.org/ftp/python/3.7.2/Python-3.7.2.tgz

tar xvf Python-3.7.2.tgz

cd Python-3.7.2.tgz

./configure

make & make install

ln -s /usr/local/python3/bin/python3 /usr/bin/python3

pip3 install --upgrade pip

**4.安装python需要的第三方环境包**

pip3 install selenium requests pillow flask psutil mitmproxy ipython gunicorn gevent flask\_bootstrap pymongo jsbeautifier python-levenshtein redis pyzbar

**5.如果是用无头浏览器抓取的机器还需要在机器上安装谷歌浏览器**

wget https://dl.google.com/linux/direct/google-chrome-stable\_current\_x86\_64.rpm

rpm -ivh google-chrome-stable\_current\_x86\_64.rpm

**6.安装浏览器驱动**

wget http://npm.taobao.org/mirrors/chromedriver/2.45/chromedriver\_linux64.zip

unzip chromedriver\_linux64.zip

mv chromedriver /bin/

**7.如果是存储服务器需要安装mongodb**

yum install mongodb mongodb-server –y

systemctl enable mongod

systemctl restart mongod

vim /etc/mongod.conf

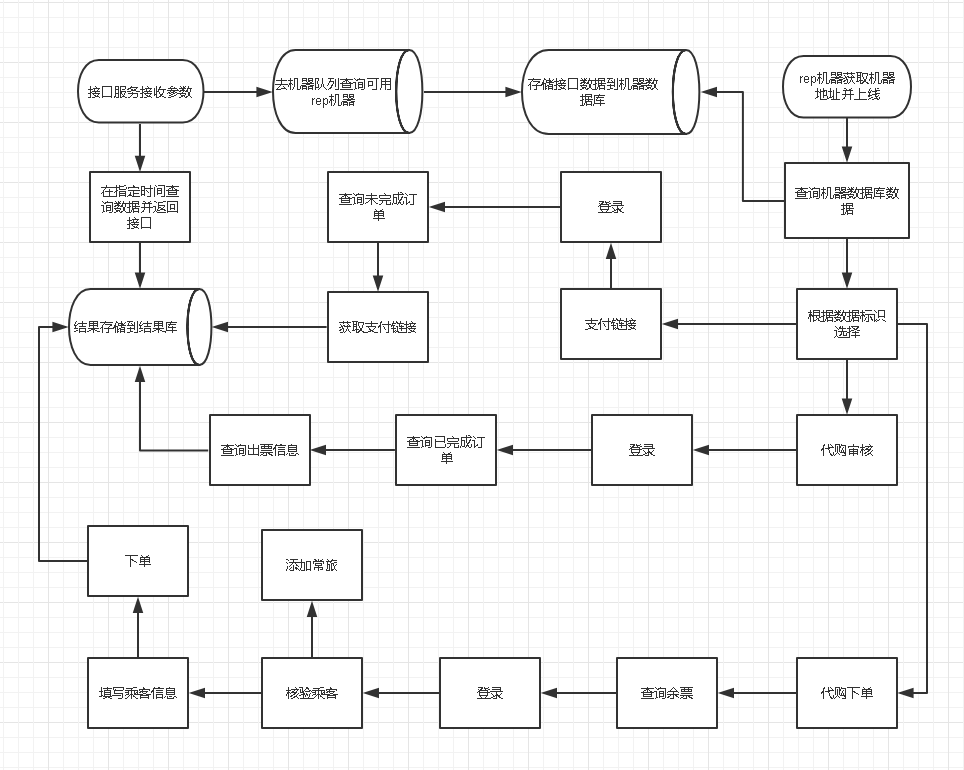
把bind\_ip 改成0.0.0.0

**9.建立数据库statistics，一个集合表monitor, 无索引。**

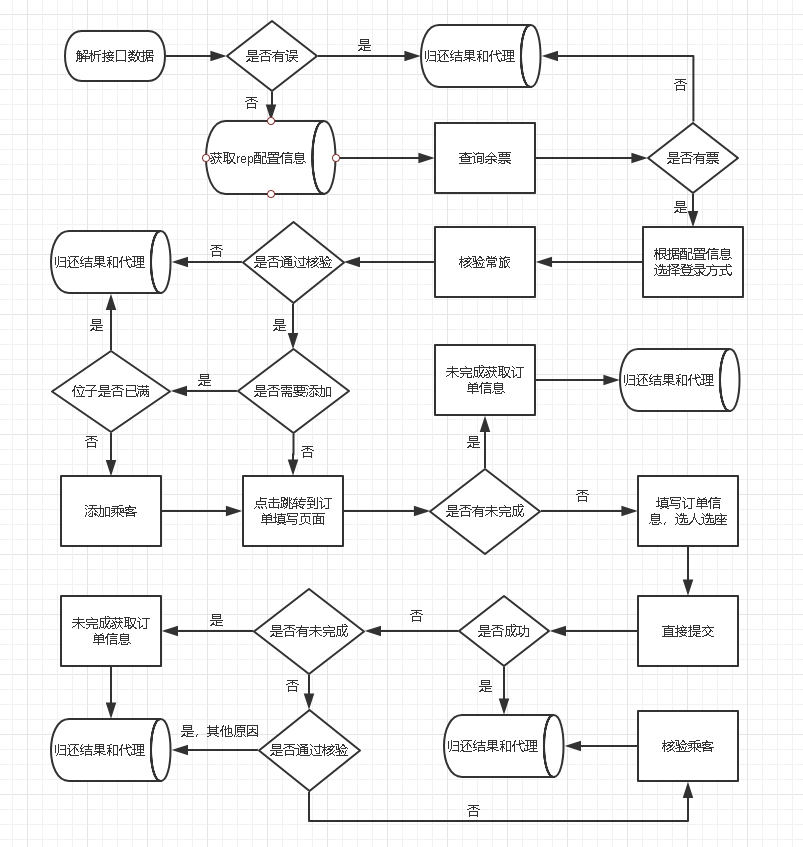
**Monitor表用来监控每次请求成功率和耗时的。** 原地址101.236.53.76

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 约束 | 说明 |
| Status | String | 无 | 监控类型，下单order/审核audit/支付pay |
| key | String / int |  | 接口传来的订单号 |
| Start\_time | Int |  | 接口进来的时间戳 |
| Run\_time | Float |  | 此订单此次运行时间 |
| ip | String |  | 运行在哪台机器 |
| proxy | String |  | 使用的哪个代理服务器 |
| Real\_msg | String |  | 真正的失败原因 |
| Return\_msg | string |  | 修饰返回接口的失败原因 |

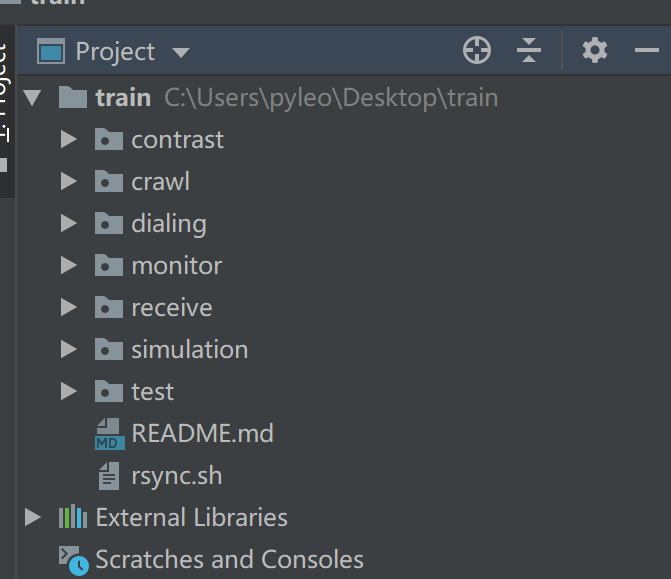
1. **架构和流程**
2. 服务架构图，rep返回数据的时候，接口查询到数据顺便存储一份到监控服务器上。



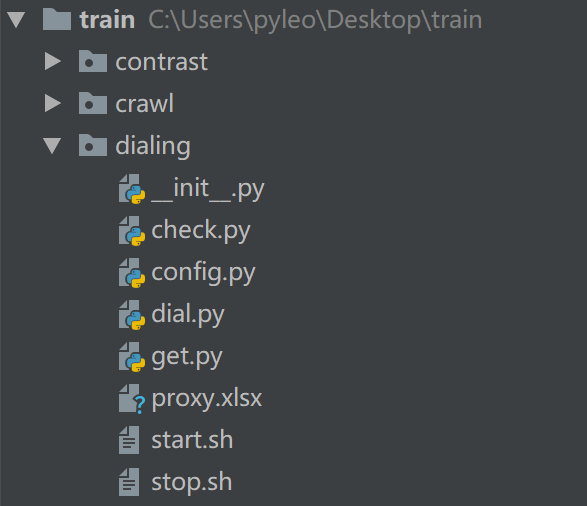
1. 下单rep代码流程图



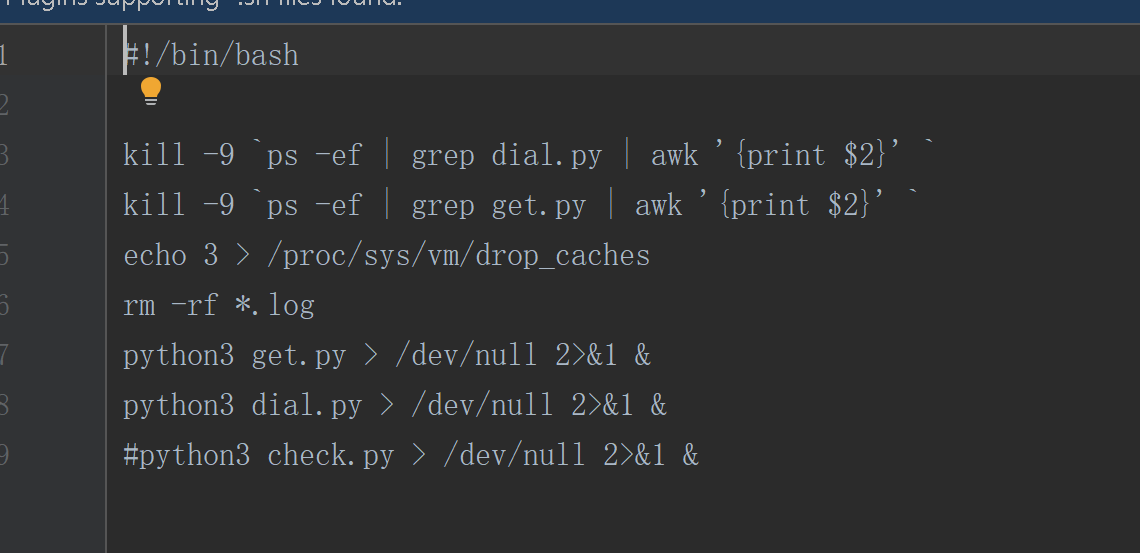
1. **代码部署**
2. 首先部署接口接收数据服务，复制receive目录到服务器上，运行start.sh脚本即可。



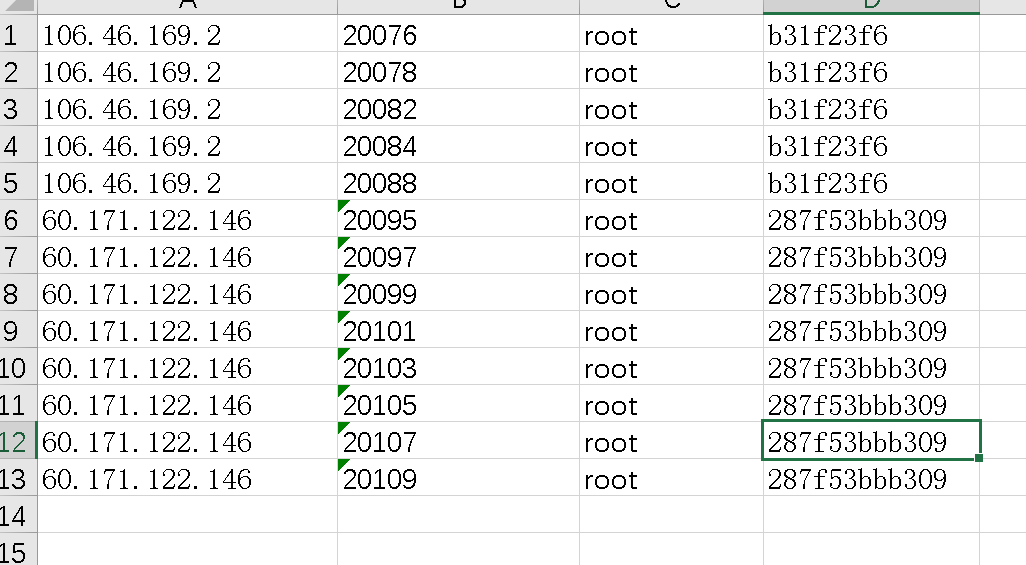
1. 其次部署拨号服务和检查代理服务，复制dialing目录分别到拨号服务器上和检查服务器上。



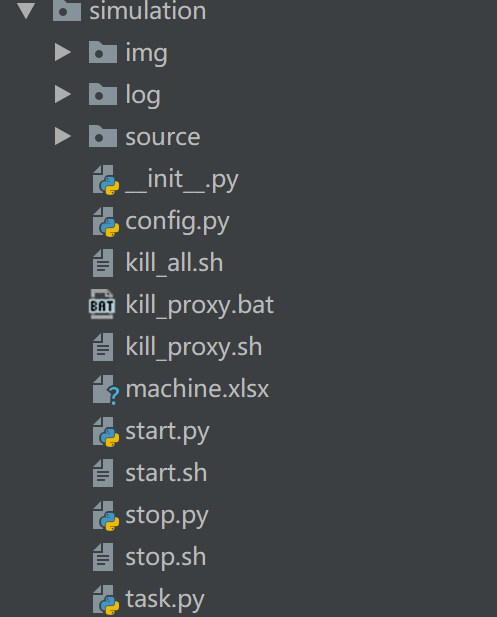
拨号修改脚本start.sh,运行get.py和dial.py，检查修改脚本start.sh，运行check.py。



Get.py用于部署代理队列上线，用excel复制好代理服务器列表，运行get.py自动上线代理服务机器，格式一定按照要求读取。



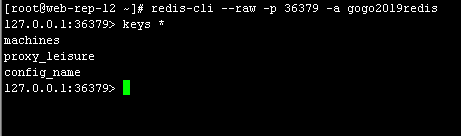
1. 部署监控服务，复制目录monitor到服务器下运行start.sh脚本即可。
2. 部署rep机器服务，复制目录simulation到模板机器上，让运维同步或者自己同步其他机器，同步脚本在模板机上，然后读取excel里的rep机器列表，运行start.py即可同步上线rep机器。



1. **Redis各种服务说明**
2. redis所有服务端口36379，密码gogo2019redis
3. redis服务有队列库，存放机器队列，代理队列，和配置

代理队列分为proxy\_dialing拨号队列，proxy\_check检测代理是否可用，proxy\_leisure最终可用的队列

机器队列machines是rep可用队列



1. redis配置config\_name包括以下配置信息：

order\_proxy下单是否开启使用代理模式，0不开启，1开启

audit\_proxy 审核是否开启使用代理模式，0不开启，1开启

pay\_proxy 支付是否开启使用代理模式，0不开启，1开启

cookie\_way 是否启用cookie登录模式 0 不开启 1 开启

login\_way 登录模式，0 老版登录 1 新版自适应登录 2 打码登录 3扫码登录

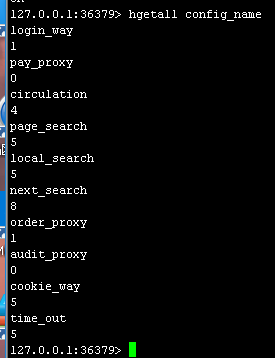
circulation 如果错误循环几次下单，默认4， 0不循环只下一次

time\_out 网页超时时间

page\_search 打开网页后标签查找超时时间

local\_search 点击按钮后本页加载数据超时时间

next\_search 点击按钮后跳转下一页超时时间



4．测试脚本在test目录下

