1. **需求和目的**

监测12306登录当中的的所有请求地址，请求地址后缀参数，headers参数和cookie参数的前后俩次变化情况。

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：

|  |  |
| --- | --- |
| 101.236.17.132 | mongo监测数据库 |
| 101.236.39.211 | python登录抓包 |
| 101.236.54.3 | python监测数据展示 |

**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

**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 paramiko xlrd

**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package，三个集合表time,parameter,content,建立俩个索引。**

**Time表用来存储抓取的俩个时间范围，一个最新一次抓取时间，一个上一次抓取时间**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 约束 | 说明 |
| type | string | 无 | 定义类型来查找时间 |
| New\_time | int | 无 | 最新时间 |
| Old\_time | int | 无 | 上一次时间 |

Parameter表用来存储抓取的参数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 约束 | 说明 |
| package | String |  | 抓取包名称 |
| package\_type | String |  | 抓取包的类型，xhr/img |
| package\_method | String |  | 抓取包的方法get/post |
| package\_url | String | 联合scrape\_time约束unique | 链接地址 |
| package\_args | list |  | 地址后面的参数 |
| package\_headers | list |  | 地址请求头参数 |
| package\_cookies | list |  | 地址请求cookie参数 |
| package\_sets | list |  | 地址响应回传cookie参数 |
| scrape\_time | int | 联合索引 | 抓取时间戳 |

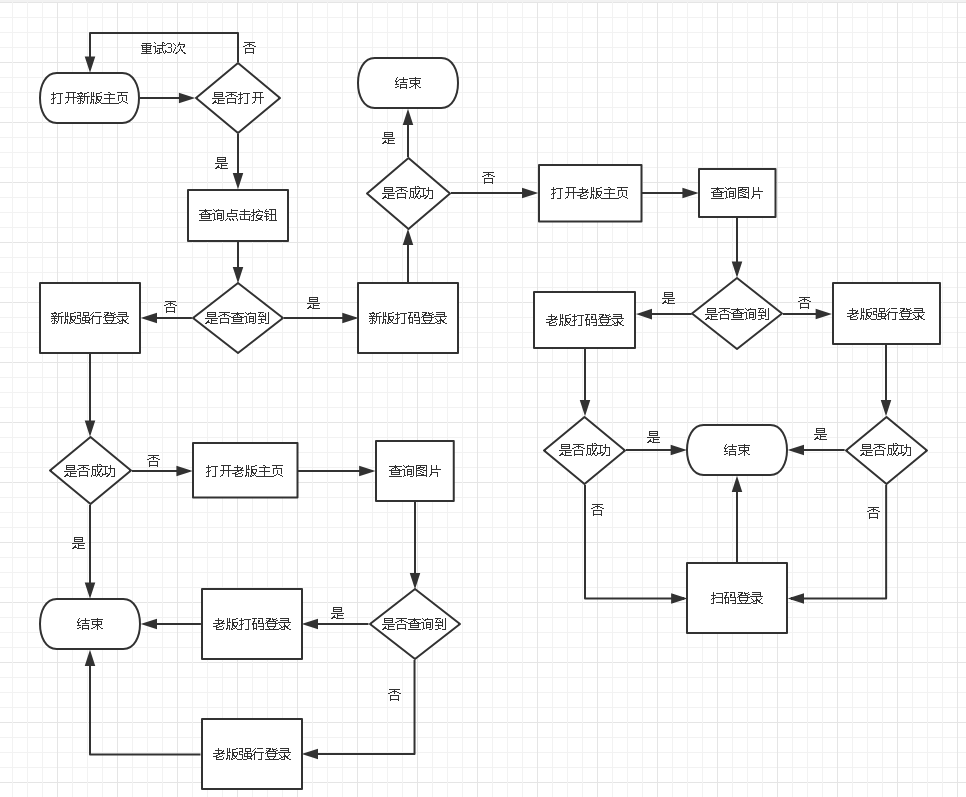
Content表用来存储抓取js的内容

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 约束 | 说明 |
| Package\_url | string | 联合scrape\_time约束unique | 链接地址 |
| Package\_text | String |  | 地址响应内容 |
| scrape\_time | int | 联合索引 | 抓取时间戳 |

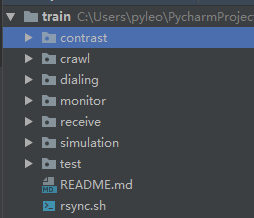
表建立索引语句示范如下：

db.parameter.createIndex({package\_url: 1, scrape\_time: 1}, {unique: true});

1. **架构和流程**
2. 服务器架构图
3. 抓取代码流程图



1. 代码部署
2. 抓取代码在train目录下crawl整个目录，复制crawl目录到服务器下任何目录即可，用dos2unix命令转化windows下乱码，如：cd crawl; dos2unix .\*, 然后运行start.sh即可，定时任务1分钟抓取一次，修改login.py最下行代码的时间选项修改抓取频率，抓取过程中会因为超时抓不到属正常，如果频繁的抓不到即会有异常，抓取出现异常会钉钉预警。



1. 展示代码复制contrast目录至服务器上任何目录下，用dos2unix转化乱码，运行start.sh即可。三个页面分别home展示基础连接对比，detail展示链接参数对比，content展示请求js内容对比，刷新频率是30秒。

