注意：我试过用[CentOS](https://www.linuxidc.com/topicnews.aspx?tid=14) 6.5玩Selenium，但是很惨，CentOS 6.5安装配置非常麻烦，而CentOS 7安装过程很顺畅，原因可能是CentOS 6.5对于chrome和chromedriver的支持非常之不好，准确的说不支持chrome而必须采用chromium，chromium的安装很不顺畅，所以，建议玩CentOS 6.5的，可以升级或者重新上一台CentOS 7服务器吧！在无界面的CentOS7上安装Selenium+Chrome，并使用facebook的php-webdriver

**测试系统环境**

**CentOS Linux 7 (Core)**

Operating System: CentOS Linux 7 (Core)

Kernel: Linux 3.10.0-693.17.1.el7.x86\_64

Architecture: x86-64

**安装 chrome**

使用下面的命令，在root用户下就可以安装最新的 Google Chrome:

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

**安装 selenium**

在[seleniump](http://selenium-release.storage.googleapis.com/index.html)官网 找到最新的版本，下载selenium-server-standalone-X.XX.X.jar文件

注意：最新的版本3.11，不是3.9（2018年3月）

selenium-server-standalone-3.11.0.jar

**selenium服务初始化**

将上述selenium放入一个文件夹中，输入如下命令初始化

java -jar selenium-server-standalone-3.11.0.jar

注意，需要 java8 环境，可以参考如下

**--------------------------------------分割线  开始--------------------------------------**

 **CentOS7安装Java运行环境jdk**

*检测java jdk是否已经安装*

*1、查看Linux自带的JDK是否已安装*

*java -version*

*2、查看JDK信息*

*rpm -qa | grep java*

*显示：*

*java-x.x.x-gcj-compat-x.x.x.x-xxjpp.xxx*

*java-x.x.x-openjdk-x.x.x.x-x.x.bxx.exx*

*卸载（有几个卸载几个）*

*rpm -e –nodeps java-x.x.x-gcj-compat-x.x.x.x-xxjpp.xxx*

*rpm -e –nodeps java-x.x.x-openjdk-x.x.x.x-x.x.bxx.exx*

*还有一些其他的命令*

*rpm -qa | grep gcj*

*rpm -qa | grep jdk*

*如果出现找不到openjdk source的话，那么还可以这样卸载*

*yum -y remove java java-x.x.x-gcj-compat-x.x.x.x-xxjpp.xxx*

*yum -y remove java java-x.x.x-openjdk-x.x.x.x-x.x.bxx.exx*

*我当前安装并没有java版本，故我这就没有卸载这个流程。*

*下载安装*

*首先到[Oracle](https://www.linuxidc.com/topicnews.aspx?tid=12)官网下载jdk*

*然后解压jdk*

*tar -zxvf jdk-8u162-linux-x64.tar.gz*

*将解压后的文件夹移到自己安装第三方软件的常用文件夹中：*

*mv jdk1.8.0\_162 /usr/local/*

*配置JDK环境变量*

*安装完成后需要配置一下环境变量，编辑/etc/profile文件：*

*vim /etc/profile*

*在文件尾部添加如下配置：*

*#JDK1.8*

*export JAVA\_HOME=/opt/jdk1.8.0\_162*

*export JAVA\_BIN=/opt/jdk1.8.0\_162/bin*

*export PATH=$PATH:$JAVA\_HOME/bin*

*export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar*

*export JAVA\_HOME JAVA\_BIN PATH CLASSPATH*

*重新加载/etc/profile文件*

*source命令重新加载/etc/profile文件，使得修改后的内容在当前shell窗口有效：*

*source /etc/profile*

*验证是否安装成功*

*java -version*

*如果出现以下信息则说明安装成功。*

*java version "1.8.0\_162"*

*Java(TM) SE Runtime Environment (build 1.8.0\_162-b12)*

*Java HotSpot(TM) 64-Bit Server VM (build 25.162-b12, mixed mode)*

--------------------------------------分割线 结束 --------------------------------------

**安装 chromerriver**

在 [chromerriver](https://sites.google.com/a/chromium.org/chromedriver/downloads)官网下载最新的ChromeDriver压缩包，解压得到chromedriver.exe文件

chromedriver\_linux64.zip 2018-03-20 15:22:39

将下载的文件解压，放在如下位置(据说不放在这个目录下有坑)

/usr/bin/chromedriver

给予执行权限

chmod +x /usr/bin/chromedriver

**安装 XVFB**

**（xvfe用来提供无界面打开chrom）**

输入如下命令

# yum install Xvfb -y

# yum install xorg-x11-fonts\* -y

新建在/usr/bin/ 一个名叫 xvfb-chrom 的文件写入以下内容：

vi /usr/bin/xvfb-chrome

#!/bin/bash

\_kill\_procs() {

  kill -TERM $chrome

  wait $chrome

  kill -TERM $xvfb

}

# Setup a trap to catch SIGTERM and relay it to child processes

trap \_kill\_procs SIGTERM

XVFB\_WHD=${XVFB\_WHD:-1280x720x16}

# Start Xvfb

Xvfb :99 -ac -screen 0 $XVFB\_WHD -nolisten tcp &

xvfb=$!

export DISPLAY=:99

chrome --no-sandbox --disable-gpu$@ &

chrome=$!

wait $chrome

wait $xvfb

添加执行权限

chmod +x /usr/bin/xvfb-chrome

查看当前映射关系

ll /usr/bin/ | grep chrom

-rwxr-xr-x  1 root root  7874704 Mar 20 14:55 chromedriver

lrwxrwxrwx  1 root root        31 Mar 20 00:24 google-chrome -> /etc/alternatives/google-chrome

lrwxrwxrwx  1 root root        32 Mar 20 14:30 google-chrome-stable -> /opt/google/chrome/google-chrome

**注意：因为已经安装了xvfp这个虚拟桌面，打开浏览器的弹窗等问题会在虚拟桌面程序中打开，所以无需将将指向打开chrome程序的软连接指向 xvfp-chrome,所以以下修改软连接的小字体部分可以不执行，之所以保留，可以作为做软连接修改的参考：**

更改Chrome启动的软连接

ln -s /etc/alternatives/google-chrome /usr/bin/chrome

rm -rf /usr/bin/google-chrome

ln -s /usr/bin/xvfb-chrome /usr/bin/google-chrome

查看修改后的映射关系

ll /usr/bin/ | grep chrom

-rwxr-xr-x  1 root root  7874704 Mar 20 14:55 chromedriver

lrwxrwxrwx  1 root root        31 Mar 20 00:24 chrome -> /etc/alternatives/google-chrome

lrwxrwxrwx  1 root root        22 Mar 20 00:11 google-chrome -> /usr/bin/xvfb-chromium

lrwxrwxrwx  1 root root        32 Mar 20 14:30 google-chrome-stable -> /opt/google/chrome/google-chrome

-rwxr-xr-x  1 root root      432 Mar 20 00:09 xvfb-chrome

装完

备份修改软连接之前的设置：

#####################################################

#没有修改链接之前

[root@iz2ze4hh8q8vtj37112qwaz bin]# ll /usr/bin/ | grep chrome

lrwxrwxrwx 1 root root 29 May 17 00:12 chrome -> /usr/bin/google-chrome-stable

-rwxrwxrwx 1 root root 10664568 May 16 23:11 chromedriver

lrwxrwxrwx 1 root root 31 May 16 23:15 google-chrome -> /etc/alternatives/google-chrome

lrwxrwxrwx 1 root root 32 May 16 23:15 google-chrome-stable -> /opt/google/chrome/google-chrome

-rwxr-xr-x 1 root root 422 May 17 09:40 xvfb-chrome

修改之后：

[root@iz2ze4hh8q8vtj37112qwaz ~]# ls -al /usr/bin/ | grep chrome

lrwxrwxrwx 1 root root 29 May 17 00:12 chrome -> /usr/bin/google-chrome-stable

-rwxrwxrwx 1 root root 10751616 May 17 10:57 chromedriver

lrwxrwxrwx 1 root root 20 May 17 10:20 google-chrome -> /usr/bin/xvfb-chrome

lrwxrwxrwx 1 root root 32 May 16 23:15 google-chrome-stable -> /opt/google/chrome/google-chrome

-rwxr-xr-x 1 root root 422 May 17 09:40 xvfb-chrome

---------------------------------------------------------------------------------------------

运行代码脚本验证：

seleniume\_driver.py:

# -\*- coding: utf-8 -\*-

from selenium import webdriver

from detect\_platform import get\_system\_platform

class SeleniumDriver:

def \_\_init\_\_(self):

self.ua = 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_13\_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'

def default\_driver(self):

chrome\_options = self.chrome\_profile()

driver = webdriver.Chrome(chrome\_options=chrome\_options)

return driver

def chrome\_profile(self):

from selenium.webdriver.chrome.options import Options

chrome\_options = Options()

if get\_system\_platform() == 'mac':

chrome\_options.add\_argument("--user-data-dir=/Users/yongleiqi/chromeData")

# chrome\_options.add\_argument("–no-sandbox")

# todo 如果使用无头模式，请将下面的注释去掉

# chrome\_options.add\_argument('--headless')

chrome\_options.add\_argument('user-agent=%s' % self.ua)

# chrome\_options.add\_argument('--disable-gpu')

# todo 下面的是设置，是否现实图片的，如果不现实，可以减少带宽，加快速度

# prefs = {"profile.managed\_default\_content\_settings.images": 2}

# chrome\_options.add\_experimental\_option("prefs", prefs)

return chrome\_options

else:

chrome\_options.add\_argument('--headless')

chrome\_options.add\_argument('--no-sandbox')

chrome\_options.add\_argument('--disable-gpu')

chrome\_options.add\_argument('--disable-dev-shm-usage')

return chrome\_options

selenium\_driver = SeleniumDriver()

if \_\_name\_\_ == '\_\_main\_\_':

driver = selenium\_driver.default\_driver()

driver.get('http://www.baidu.com')

print(driver.page\_source)

运行该脚本测试：报错：

self.start\_session(capabilities, browser\_profile)

File "/application/python/lib/python3.7/site-packages/selenium/webdriver/remote/webdriver.py", line 252, in start\_session

response = self.execute(Command.NEW\_SESSION, parameters)

File "/application/python/lib/python3.7/site-packages/selenium/webdriver/remote/webdriver.py", line 321, in execute

self.error\_handler.check\_response(response)

File "/application/python/lib/python3.7/site-packages/selenium/webdriver/remote/errorhandler.py", line 242, in check\_response

raise exception\_class(message, screen, stacktrace)

selenium.common.exceptions.SessionNotCreatedException: Message: session not created: Chrome version must be between 70 and 73

(Driver info: chromedriver=73.0.3683.68 (47787ec04b6e38e22703e856e101e840b65afe72),platform=Linux 3.10.0-693.2.2.el7.x86\_64 x86\_64)

查看本地的chrom版本是：

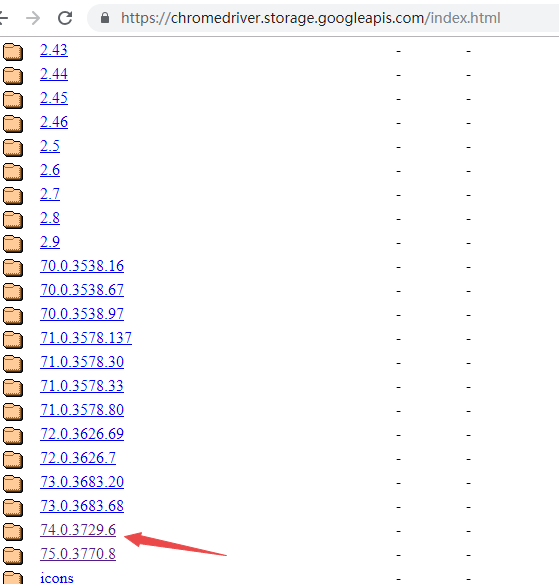
[root@iz2ze4hh8q8vtj37112qwaz sohuhao]# chrome -version

Google Chrome 74.0.3729.157

升级chromdriver：

到该网站下载对应的chromdriver版本：[https://chromedriver.storage.googleapis.com/index.html](https://chromedriver.storage.googleapis.com/index.html?path=74.0.3729.6/)

由于我的chrom版本是74.03，所以我选择下载了74.03的版本安装；



安装完之后，重新执行测试脚本，没有问题；