问题背景

在selenium驱动的web自动化中，经常会碰Chrome和chromedriver版本不匹配的状况；本篇讲解如何自动下载与本地Chrome版本相匹配的chromedriver。

解题思路（windows）

1. **获取本地Chrome版本号**：从注册表HKEY\_CURRENT\_USER\Software\Google\Chrome\BLBeacon\version中得到
2. **获取本地chromedriver版本号**：找到就返回版本号，未找到就返回 0；
3. **比对版本号**：Chrome Version 长这样"*79.0.3945.88*"，取前三位版本号比对，即“*79.0.3945*”
4. **若不一致**：去[淘宝镜像](http://npm.taobao.org/mirrors/chromedriver/)下载匹配的chromedriver。
5. **最终实现请直接移步到本篇末尾**

细节如下（python3实现，最后会打包成 *.exe*程序，供双击运行）

* 获取Chrome版本号
* import winreg # 和注册表交互
* import re # 正则模块
* version\_re = re.compile(r'^[1-9]\d\*\.\d\*.\d\*') # 匹配前3位版本号的正则表达式
* def getChromeVersion():
* try:
* # 从注册表中获得版本号
* key = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER, r'Software\Google\Chrome\BLBeacon')
* \_v, type = winreg.QueryValueEx(key, 'version')
* print('Current Chrome Version: {}'.format(\_v)) # 这步打印会在命令行窗口显示
* return version\_re.findall(\_v)[0] # 返回前3位版本号
* except WindowsError as e:
* print('check Chrome failed:{}'.format(e))
* 获取chromedriver版本号
  + 方式(在chromedriver.exe的目录下执行cmd命令)
* G:\py\prac\chromedriver.exe --version
* prac 79.0.3945.36 (3582db32b33893869b8c1339e8f4d9ed1816f143-refs/branch-heads/3945@{#614})
  + 代码实现如下
* import subprocess # 用于执行cmd命令
* def getDriverVersion(absPath):
* """
* :param absPath: chromedriver.exe的绝对路径
* """
* cmd = r'{} --version'.format(absPath) # 拼接成cmd命令
* try:
* # 执行cmd命令并接收命令回显
* out, err = subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE).communicate()
* out = out.decode('utf-8')
* \_v = out.split(' ')[1] # 拆分回显字符串，获取版本号
* print('Current chromedriver Version:{}'.format(\_v))
* return version\_re.findall(\_v)[0]
* except IndexError as e:
* print('check chromedriver failed:{}'.format(e))
* return 0
* 建个配置文件吧
  + 存放路径：与该程序同级
  + 内容如下：
  + [driver]
  + ; absolute path of chromedriver
  + absPath=G:\py\prac\chromedriver.exe
  + ; DO NOT modify url
  + url=http://npm.taobao.org/mirrors/chromedriver/
  + 定义个读取.ini文件的方法吧
* from configparser import RawConfigParser # 用于解析.ini格式的配置文件
* def get\_ini(file):
* \_ini = RawConfigParser()
* \_ini.read(file, encoding='utf-8')
* return \_ini
* 对比版本号
* import os
* def checkVersionMatch():
* # 读取conf.ini中的配置项
* conf = get\_ini('conf.ini')
* absPath = conf.get('driver', 'absPath')
* print('Chrome version compares with chromedriver version')
* c\_v = getChromeVersion()
* d\_v = getDriverVersion(absPath)
* if c\_v == d\_v:
* # 若匹配，在命令行窗口提示下面的信息
* input('Chrome and chromedriver are matched. Press Enter to exit.')
* else:
* # 若不匹配，走下面的流程去下载chromedriver
* \_v = c\_v
* url = conf.get('driver', 'url') # 从conf.ini中获取淘宝镜像url
* save\_d = os.path.dirname(absPath) # 下载文件的保存路径，与chromedriver同级
* downLoadDriver(\_v, url, save\_d) # call下载文件的方法
* 下载文件并解压
* import urllib.request # 发送http请求
* import urllib.parse # 拼接url
* import zipfile # 操作.zip文件
* def downLoadDriver(\_\_v, url, save\_d):
* # 访问淘宝镜像首页
* rep = urllib.request.urlopen(url).read().decode('utf-8')
* # '<a href="/mirrors/chromedriver/84.0.4147.30/">84.0.4147.30/</a>'
* directory = re.compile(r'>(\d.\*?/)</a>').findall(rep) # 匹配文件夹（版本号）
* # 获取期望的文件夹（版本号）
* match\_list = []
* for i in directory:
* v = version\_re.findall(i)[0]
* if \_\_v == v:
* match\_list.append(i)
* # http://npm.taobao.org/mirrors/chromedriver/83.0.4103.39/chromedriver\_win32.zip
* dirUrl = urllib.parse.urljoin(url, match\_list[-1])
* downUrl = urllib.parse.urljoin(dirUrl, 'chromedriver\_win32.zip') # 拼接出下载路径
* print('will download {}'.format(downUrl))
* # 指定下载的文件名和保存位置
* file = os.path.join(save\_d, os.path.basename(downUrl))
* print('will saved in {}'.format(file))
* # 开始下载，并显示下载进度(progressFunc)
* urllib.request.urlretrieve(downUrl, file, progressFunc)
* # 下载完成后解压
* zFile = zipfile.ZipFile(file, 'r')
* for fileM in zFile.namelist():
* zFile.extract(fileM, os.path.dirname(file))
* zFile.close()
* input('Complete, press Enter to exit')
* 显示下载进度的方法
* import sys
* def progressFunc(blocknum, blocksize, totalsize):
* '''回调函数
* :blocknum: 已经下载的数据块
* :blocksize: 数据块的大小
* :totalsize: 远程文件的大小
* '''
* percent = 100.0 \* blocknum \* blocksize / totalsize
* if percent > 100:
* percent = 100
* downsize = blocknum \* blocksize
* if downsize >= totalsize:
* downsize = totalsize
* s = "%.2f%%" % (percent) + "====>" + "%.2f" % (downsize / 1024 / 1024) + "M/" + "%.2f" % (totalsize / 1024 / 1024) + "M \r"
* sys.stdout.write(s)
* sys.stdout.flush()
* if percent == 100:
* print('')
* 以线程的方式运行
* import threading
* if \_\_name\_\_ == '\_\_main\_\_':
* threading.Thread(target=checkVersionMatch, args=('')).start()

打包成 .exe程序（打包教程略过，因为运行.py文件也行）

* 双击运行cdCheck.exe 即可（注意conf.ini中absPath的值）
* .exe[下载地址](https://download.csdn.net/download/qq_40957663/12562451)
* 运行效果图
* 欢迎指正交流！！！

////以下 这个版本很不错可以直接运行跑通

解决Chrome与ChromeDriver版本不一致问题的Python方案

但老师 2020-10-24 20:07:56 1611 收藏 9

分类专栏： Python 文章标签： 1024程序员节 python 正则表达式 chrome

版权

Python

专栏收录该内容

21 篇文章0 订阅

订阅专栏

文章目录

需求背景

干活

获取Chrome版本

获取ChromeDriver版本

核对版本信息

下载文件

解压文件

替换文件

总结

封装

需求背景

由于Chrome是自动更新的,在浏览器里面也没办法关闭这一选项,而自动升级总是很容易带来问题.例如ChromeDriver不兼容的问题

selenium.common.exceptions.SessionNotCreatedException: Message: session not created: This version of ChromeDriver only supports Chrome version 83

1

在使用Python的Selenium模块的时候,就会因为这个问题报错而导致任务不会如期运行

所以我琢磨着写一个自动更新ChromeDriver的程序,伪代码是这样的

try:

selenium程序()

except:

自动更新ChromeDriver程序()

1

2

3

4

干活

主体架构如下:

- 获取Chrome版本()

- 获取ChromeDriver版本列表()

- 核对版本信息,获取最接近的ChromeDriver版本号()

- 拼接下载链接,下载文件()

- 解压文件()

- 替换文件()

1

2

3

4

5

6

获取Chrome版本

这一方法需要通过注册表来获取

def getChromeVersion():

import winreg

key = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,r'Software\Google\Chrome\BLBeacon')

chrome\_version = winreg.QueryValueEx(key,'version')[0]

return chrome\_version

chrome = getChromeVersion()

1

2

3

4

5

6

7

我电脑中运行的结果,获取到的版本是这样的

>>> chrome

'80.0.3987.162'

1

2

获取ChromeDriver版本

这里需要抓取网页源码,利用正则从网页源码提取版本信息用以核对

URL = 'http://chromedriver.storage.googleapis.com/?delimiter=/&prefix='

REG = r'<Prefix>(\d.\*?)</Prefix>'

def getChromeDriverVersion(link,regexp):

# 从网页源代码获取目前提供的版本号

# 返回列表

from re import findall

from requests import get

resp = get(link)

driver\_version = findall(regexp,resp.text) # 返回正则括号中结果的列表

return driver\_version

driver = getChromeDriverVersion(URL,REG)

1

2

3

4

5

6

7

8

9

10

11

12

13

这一段需要注意,ChromeDriver的原本地址是不会告诉你上面的地址的.地址见这篇

ChromeDriver下载地址

get到的网页源代码都是js,是看不到任何版本信息的.但是仔细观察可以发现是有一段链接是有完整版本号的.就是上面这段url

至于正则,这是一段很简单的正则,因为网页源代码的结构很简单,就直接取数字和小数点了

这一段我的运行结果如下

>>> driver

['2.0/','2.1/','2.10/','2.11/','2.12/','2.13/','2.14/','2.15/','2.16/','2.17/','2.18/','2.19/','2.2/','2.20/','2.21/','2.22/','2.23/','2.24/','2.25/','2.26/','2.27/','2.28/','2.29/','2.3/','2.30/','2.31/','2.32/','2.33/','2.34/','2.35/','2.36/','2.37/','2.38/','2.39/','2.4/','2.40/','2.41/','2.42/','2.43/','2.44/','2.45/','2.46/','2.5/','2.6/','2.7/','2.8/','2.9/',

'70.0.3538.16/','70.0.3538.67/','70.0.3538.97/','71.0.3578.137/','71.0.3578.30/','71.0.3578.33/','71.0.3578.80/','72.0.3626.69/','72.0.3626.7/','73.0.3683.20/','73.0.3683.68/','74.0.3729.6/','75.0.3770.140/','75.0.3770.8/','75.0.3770.90/','76.0.3809.12/','76.0.3809.126/','76.0.3809.25/','76.0.3809.68/','77.0.3865.10/','77.0.3865.40/','78.0.3904.105/','78.0.3904.11/','78.0.3904.70/','79.0.3945.16/','79.0.3945.36/',

'80.0.3987.106/','80.0.3987.16/','81.0.4044.138/','81.0.4044.20/','81.0.4044.69/','83.0.4103.14/','83.0.4103.39/','84.0.4147.30/','85.0.4183.38/','85.0.4183.83/','85.0.4183.87/','86.0.4240.22/','87.0.4280.20/']

1

2

3

4

核对版本信息

文本与列表核逐一对,需要遍历.由于版本号往往不是完全一致,并且实际也不需要完全一致,大概对的上就可以运行,所以定下规则

版本的前两处信息对得上就可以

目前是根据版本的那个.拆分为列表,核对列表的前两个元素

def compareVersion(chrome,driver):

# 版本号的前两部分相等即为匹配

# 返回核对后正确的完整版本号

right\_version = ''

chrome\_list = self.chrome.split('.')

for ver in driver:

\_tmp = ver.split('.')

if chrome\_list[0] == \_tmp[0] and chrome\_list[1] == \_tmp[1]:

right\_version = ver

break

return right\_version

ver = compareVersion(chrome,driver)

1

2

3

4

5

6

7

8

9

10

11

12

13

结果如下

>>> ver

'80.0.3987.106/'

1

2

版本号的前两处对得上就可以

下载文件

用requests的get方法似乎是成功了,但是文件都不知道去哪了,还是得用一个能指定下载文件夹的方法

url\_down = 'http://chromedriver.storage.googleapis.com/%schromedriver\_win32.zip'

folder = r'c:\selenium\_download'

FILE\_NAME = 'chromedriver\_win32.zip'

def downloadChromeDriver(version):

# 下载,为了指定路径必须用urllib库

# 返回下载文件路径

from urllib.request import urlretrieve as urlrt

url\_download = url\_down % version

path\_file = r'%s\%s' % (folder,FILE\_NAME)

urlrt(url\_download,path\_file) # 已测试会自动覆盖

return self.path\_file

file = downloadChromeDriver(ver)

1

2

3

4

5

6

7

8

9

10

11

12

13

14

>>> file

'c:\\selenium\_download\\\chromedriver\_win32.zip'

1

2

解压文件

FILE\_EXE = 'chromedriver.exe'

def unzipFile(zip,folder):

# 解压

# 返回解压后的文件路径

from zipfile import ZipFile as unzip

cont = unzip(zip,'r')

for file in cont.namelist():

cont.extract(file,folder)

cont.close()

driver\_file = r'%s\%s' % (folder,FILE\_EXE)

return self.driver\_file

exe = unzipFile(file,folder)

1

2

3

4

5

6

7

8

9

10

11

12

13

14

>>> exe

'c:\\selenium\_download\\chromedriver.exe'

1

2

替换文件

将文件从Python的文件夹里面替换

首先要获取Python安装路径

删除原文件(如果已存在),移入新文件

def replaceDriverInPython(driver):

import os

import sys

isdone = False

# 替换Python Script文件夹里面的chromedriver

# 在里面寻找Python路径,要求下级必须有Scripts文件夹,且路径末尾是Python关键字

pypath = ''

for path in sys.path:

if path.split('\\')[-1] == 'Python' and os.path.isdir(path):

for d in os.listdir(path):

if os.path.isdir(os.path.join(path,d)) and d == 'Scripts':

pypath = '%s\\%s' % (path,d)

break

if pypath != '':

break

if pypath != '':

dst\_path = '%s\\%s' % (pypath,driver)

if os.path.isfile(dst\_path):

os.remove(dst\_path)

shutil.move(self.driver\_file,pypath)

isdone = True

else:

isdone = False

return isdone

done = replaceDriverInPython(exe)

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

>>> done

True

1

2

总结

需要用到的内部模块

import re # 正则

import urllib # 网络

import zipfile # 压缩解压

import winreg # windows注册表

import os # 操作系统信息

import sys # 设备信息

import shutil # 文件操作

1

2

3

4

5

6

7

外部模块

import requests # 用来获取网页源码

1

相关知识点

正则

爬虫

由于关乎到注册表,目前这段代码仅支持在Windows上运行

封装

'''

only for windows

'''

from requests import get

from re import findall

from urllib.request import urlretrieve as urlrt

from zipfile import ZipFile as unzip

import winreg

import os

import sys

import shutil

URL = 'http://chromedriver.storage.googleapis.com/?delimiter=/&prefix='

REG = r'<Prefix>(\d.\*?)</Prefix>'

FILE\_NAME = 'chromedriver\_win32.zip'

FILE\_EXE = 'chromedriver.exe'

url\_down = 'http://chromedriver.storage.googleapis.com/%schromedriver\_win32.zip'

class sd\_chrome():

def \_\_init\_\_(self,\_folder=r'c:\selenium\_download'):

# 防止文件夹路径不存在

# 指定下载路径文件夹

if not os.path.exists(\_folder):

os.makedirs(\_folder)

self.down\_folder = \_folder

def getChromeVersion(self):

# 从注册表获取chrome版本

key = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,r'Software\Google\Chrome\BLBeacon')

self.chrome\_version = winreg.QueryValueEx(key,'version')[0]

return self.chrome\_version

def getChromeDriverVersionList(self):

# 从网页源代码获取目前提供的版本号

resp = get(URL)

self.driver\_version = findall(REG,resp.text) # 返回正则括号中结果的列表

return self.driver\_version

def compareVersion(self):

# 版本号的前两部分相等即为匹配

self.right\_version = ''

chrome\_version\_list = self.chrome\_version.split('.')

for ver in self.driver\_version:

\_tmp = ver.split('.')

if chrome\_version\_list[0] == \_tmp[0] and chrome\_version\_list[1] == \_tmp[1]:

self.right\_version = ver

break

return self.right\_version

def downloadChromeDriver(self):

# 下载,为了指定路径必须用urllib库

if self.right\_version != '':

url\_download = url\_down % self.right\_version

self.path\_file = r'%s\%s' % (self.down\_folder,FILE\_NAME)

urlrt(url\_download,self.path\_file) # 已测试会自动覆盖

print('已下载到:%s' % self.path\_file)

return self.path\_file

else:

print('未获取到版本号,没有下载')

def unzipFile(self):

# 解压

zip = unzip(self.path\_file,'r')

for file in zip.namelist():

zip.extract(file,self.down\_folder)

zip.close()

print('解压成功')

self.driver\_file = r'%s\%s' % (self.down\_folder,FILE\_EXE)

return self.driver\_file

def replaceDriverInPython(self):

isdone = False

# 替换Python Script文件夹里面的chromedriver

# 在里面寻找Python路径,要求下级必须有Scripts文件夹,且路径末尾是Python关键字

pypath = ''

for path in sys.path:

if path.split('\\')[-1] == 'Python' and os.path.isdir(path):

for d in os.listdir(path):

if os.path.isdir(os.path.join(path,d)) and d == 'Scripts':

pypath = '%s\\%s' % (path,d)

break

if pypath != '':

break

if pypath != '':

dst\_path = '%s\\%s' % (pypath,FILE\_EXE)

if os.path.isfile(dst\_path):

os.remove(dst\_path)

shutil.move(self.driver\_file,pypath)

isdone = True

else:

isdone = False

return isdone

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

down = r'c:\selenium\_download'

goo = sd\_chrome(down)

a = goo.getChromeVersion()

b = goo.getChromeDriverVersionList()

c = goo.compareVersion()

d = goo.downloadChromeDriver()

e = goo.unzipFile()

f = goo.replaceDriverInPython()

、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、

from requests import get

from re import findall

from urllib.request import urlretrieve as urlrt

from zipfile import ZipFile as unzip

import winreg

import os

import sys

import shutil

URL = 'http://chromedriver.storage.googleapis.com/?delimiter=/&prefix='

REG = r'<Prefix>(\d.\*?)</Prefix>'

FILE\_NAME = 'chromedriver\_win32.zip'

FILE\_EXE = 'chromedriver.exe'

url\_down = 'http://chromedriver.storage.googleapis.com/%schromedriver\_win32.zip'

class sd\_chrome():

def \_\_init\_\_(self,\_folder=r'f:\05.Python\chromedriver\_win32V88'):

# 防止文件夹路径不存在

# 指定下载路径文件夹

if not os.path.exists(\_folder):

os.makedirs(\_folder)

self.down\_folder = \_folder

def getChromeVersion(self):

# 从注册表获取chrome版本

key = winreg.OpenKey(winreg.HKEY\_CURRENT\_USER,r'Software\Google\Chrome\BLBeacon')

self.chrome\_version = winreg.QueryValueEx(key,'version')[0]

return self.chrome\_version

def getChromeDriverVersionList(self):

# 从网页源代码获取目前提供的版本号

resp = get(URL)

self.driver\_version = findall(REG,resp.text) # 返回正则括号中结果的列表

return self.driver\_version

def compareVersion(self):

# 版本号的前两部分相等即为匹配

self.right\_version = ''

chrome\_version\_list = self.chrome\_version.split('.')

for ver in self.driver\_version:

\_tmp = ver.split('.')

if chrome\_version\_list[0] == \_tmp[0] and chrome\_version\_list[1] == \_tmp[1]:

self.right\_version = ver

break

return self.right\_version

def downloadChromeDriver(self):

# 下载,为了指定路径必须用urllib库

if self.right\_version != '':

url\_download = url\_down % self.right\_version

self.path\_file = r'%s\%s' % (self.down\_folder,FILE\_NAME)

urlrt(url\_download,self.path\_file) # 已测试会自动覆盖

print('已下载到:%s' % self.path\_file)

return self.path\_file

else:

print('未获取到版本号,没有下载')

def unzipFile(self):

# 解压

zip = unzip(self.path\_file,'r')

for file in zip.namelist():

zip.extract(file,self.down\_folder)

zip.close()

print('解压成功')

self.driver\_file = r'%s\%s' % (self.down\_folder,FILE\_EXE)

return self.driver\_file

def replaceDriverInPython(self):

isdone = False

# 替换Python Script文件夹里面的chromedriver

# 在里面寻找Python路径,要求下级必须有Scripts文件夹,且路径末尾是Python关键字

pypath = ''

for path in sys.path:

if path.split('\\')[-1] == 'Python38-32' and os.path.isdir(path):

pypath = path + '\Lib\site-packages\selenium\webdriver\chrome'

if pypath != '':

break

"""

for path in sys.path:

if path.split('\\')[-1] == 'Python38-32' and os.path.isdir(path):

for d in os.listdir(path):

if os.path.isdir(os.path.join(path,d)) and d == 'Lib':

path = '%s\\%s' % (path,d)

print(path)

for d in os.listdir(path):

if os.path.isdir(os.path.join(path,d)) and d == 'site-packages':

pypath = '%s\\%s' % (path,d)

print(pypath)

break

if pypath != '':

break

"""

if pypath != '':

dst\_path = '%s\\%s' % (pypath,FILE\_EXE)

if os.path.isfile(dst\_path):

os.remove(dst\_path)

shutil.move(self.driver\_file,pypath)

isdone = True

else:

isdone = False

return isdone

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

down = r'f:\05.Python\chromedriver\_win32V88'

chromeClass = sd\_chrome(down)

ChromeVersion = chromeClass.getChromeVersion()

DriverVer = chromeClass.getChromeDriverVersionList()

compare = chromeClass.compareVersion()

download = chromeClass.downloadChromeDriver()

unzip = chromeClass.unzipFile()

replaceDriver = chromeClass.replaceDriverInPython()

print(replaceDriver)

————————————————

版权声明：本文为CSDN博主「但老师」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/sinat\_41870148/article/details/109263847