动态页面内容的获取

用户使用浏览器时会触发许多JavaScript动作，这是程序行为与用户行为的主要区别之一。

JavaScript

JavaScript 是网络上最常用也是支持者最多的客户端脚本语言。它可以收集 用户的跟踪数据,不需要重载页面直接提交表单，在页面嵌入多媒体文件，甚至运行网页游戏。

我们可以在网页源代码的标签里看到，比如:

```
<script type="text/javascript" src="https://statics.huxiu.com/w/mini/static_2015/js/sea.js?v=201601150944"></script>
```

jQuery

jQuery 是一个十分常见的javascript库,70% 最流行的网站(约 200 万)和约 30% 的其他网站(约 2 亿)都在使用。一个网站使用 jQuery 的特征,就是源代码里包含了 jQuery 入口,比如:

```
<script type="text/javascript" src="https://statics.huxiu.com/w/mini/static_2015/js/jquery-1.11.1.min.js?v=201512181512"></script>
```

如果你在一个网站上看到了 jQuery，那么采集这个网站数据的时候要格外小心。jQuery 可 以动态地创建 HTML 内容,只有在 JavaScript 代码执行之后才会显示。如果你用传统的方 法采集页面内容,就只能获得 JavaScript 代码执行之前页面上的内容。

Ajax

我们与网站服务器通信的唯一方式，就是发出 HTTP 请求获取新页面。如果提交表单之后，或从服务器获取信息之后，网站的页面不需要重新刷新，那么你访问的网站就在用Ajax 技术。

Ajax 其实并不是一门语言,而是用来完成网络任务(可以认为 它与网络数据采集差不多)的一系列技术。Ajax 全称是 Asynchronous JavaScript and XML(异步 JavaScript 和 XML)，网站不需要使用单独的页面请求就可以和网络服务器进行交互 (收发信息)。

DHTML

与Ajax 一样，动态 HTML(Dynamic HTML, DHTML)也是一系列用于解决网络问题的 技术集合。DHTML 是用客户端语言改变页面的 HTML 元素(HTML、CSS，或者二者皆 被改变)。比如页面上的按钮只有当用户移动鼠标之后才出现,背景色可能每次点击都会改变，或者用一个 Ajax 请求触发页面加载一段新内容，网页是否属于DHTML，关键要看有没有用 JavaScript 控制 HTML 和 CSS 元素。

如何获取动态页面中的内容呢？早期手工找ajax加载页面的方法过于原始，下面我们学习使用slelenium与phantomjs来获取这些动态内容。

Selenium

Selenium是一个Web的自动化测试工具，最初是为网站自动化测试而开发的，类型像我们玩游戏用的按键精灵，可以按指定的命令自动操作，不同是Selenium 可以直接运行在浏览器上，它支持所有主流的浏览器（包括PhantomJS这些无界面的浏览器）。

Selenium 可以根据我们的指令，让浏览器自动加载页面，获取需要的数据，甚至页面截屏，或者判断网站上某些动作是否发生。

Selenium 自己不带浏览器，不支持浏览器的功能，它需要与第三方浏览器结合在一起才能使用。但是我们有时候需要让它内嵌在代码中运行，所以我们可以用一个叫 PhantomJS 的工具代替真实的浏览器。

> 可以从 PyPI 网站下载 Selenium库https://pypi.python.org/simple/selenium ，也可以用 第三方管理器 pip用命令安装：pip install selenium

Selenium 官方参考文档：http://selenium-python.readthedocs.io/index.html

In [1]:
```
import selenium
```

PhantomJS

PhantomJS 是一个基于Webkit的“无界面”(headless)浏览器，它会把网站加载到内存并执行页面上的 JavaScript，因为不会展示图形界面，所以运行起来比完整的浏览器要高效。

如果我们把 Selenium 和 PhantomJS 结合在一起，就可以运行一个非常强大的网络爬虫了，这个爬虫可以处理 JavaScrip、Cookie、headers，以及任何我们真实用户需要做的事情。

> 注意：PhantomJS 只能从它的官方网站http://phantomjs.org/download.html) 下载。 因为 PhantomJS 是一个功能完善(虽然无界面)的浏览器而非一个 Python 库，所以它不需要像 Python 的其他库一样安装，但我们可以通过Selenium调用PhantomJS来直接使用。

PhantomJS 官方参考文档：http://phantomjs.org/documentation

Selenium入门

Selenium 库里有个叫 WebDriver 的 API。WebDriver 有点儿像可以加载网站的浏览器，但是它也可以像 BeautifulSoup 或者其他 Selector 对象一样用来查找页面元素，与页面上的元素进行交互(发送文本、点击等)，以及执行其他动作来运行网络爬虫。

In [2]:
```
"""selenium 入门示例"""
import time
from selenium import webdriver

# 要想调用键盘按键操作需要引入keys包
from selenium.webdriver.common.keys import Keys

# 调用环境变量指定的PhantomJS浏览器创建浏览器对象
#driver = webdriver.PhantomJS(executable_path='chromedriver.exe')
#driver = webdriver.PhantomJS(executable_path='./phantomjs/bin/phantomjs
.exe')
driver = webdriver.Chrome()
# get方法会一直等到页面被完全加载，然后才会继续程序，通常测试会在这里选择 time.slee
p(2)
driver.get("https://www.baidu.com/")
time.sleep(3)

# 获取页面名为 wrapper的id标签的文本内容
```

```
data = driver.find_element_by_id("wrapper").text

# 打印数据内容
print(data)

# 打印页面标题 "百度一下，你就知道"
print(driver.title)
```

```
新闻
hao123
地图
视频
贴吧
学术
登录
设置
更多产品
百度
把百度设为主页关于百度About  Baidu百度推广
©2018 Baidu 使用百度前必读 意见反馈 京ICP证030173号  京公网安备11000002000001号

百度一下，你就知道
```

In [3]:

```
"""selenium 入门示例"""
import time
from selenium import webdriver

# 要想调用键盘按键操作需要引入keys包
from selenium.webdriver.common.keys import Keys

# 调用环境变量指定的PhantomJS浏览器创建浏览器对象
#chromeOptions = webdriver.ChromeOptions()
#chromeOptions.add_argument("headless")
#driver = webdriver.Chrome(chrome_options=chromeOptions)
driver = webdriver.Chrome()
# 下面方法被废止了，单还可以用
# driver = webdriver.PhantomJS(executable_path='./phantomjs/bin/phantomjs.exe')

# get方法会一直等到页面被完全加载，然后才会继续程序，通常测试会在这里选择 time.sleep(2)
driver.get("https://www.baidu.com/")
time.sleep(3)

# 生成当前页面快照并保存
driver.save_screenshot("baidu.png")

# id="kw"是百度搜索输入框，输入字符串"长城"
driver.find_element_by_id("kw").send_keys(u"长城")

# id="su"是百度搜索按钮，click() 是模拟点击
driver.find_element_by_id("su").click()
time.sleep(2)
# 获取新的页面快照
driver.save_screenshot("长城.png")

# 打印网页渲染后的源代码
print(driver.page_source[:1000])
```

```
# 获取当前页面Cookie
print(driver.get_cookies())

# ctrl+a 全选输入框内容
driver.find_element_by_id("kw").send_keys(Keys.CONTROL,'a')

# ctrl+x 剪切输入框内容
driver.find_element_by_id("kw").send_keys(Keys.CONTROL,'x')

# 输入框重新输入内容
driver.find_element_by_id("kw").send_keys("itcast")

# 模拟Enter回车键
driver.find_element_by_id("su").send_keys(Keys.RETURN)

# 清除输入框内容
driver.find_element_by_id("kw").clear()

# 生成新的页面快照
driver.save_screenshot("itcast.png")

# 获取当前url
print(driver.current_url)

# 关闭当前页面，如果只有一个页面，会关闭浏览器
# driver.close()

# 关闭浏览器
time.sleep(20)
driver.quit()
```

```
<!DOCTYPE html><!--STATUS OK--><html xmlns="http://www.w3.org/1999/xhtml
"><head><script type="text/javascript" charset="gb2312" src="//www.baidu
.com/cache/aladdin/ui/tabs5/tabs5.js?v=20170208" data-for="A.ui"></scrip
t><script charset="utf-8" async="" src="https://ss0.bdstatic.com/-0U0bnS
m1A5BphGlnYG/tam-ogel/a8707a09-4a93-403c-a661-53146c99a08e.js"></script>

    <meta http-equiv="content-type" content="text/html;charset=utf-8" />
<style data-for="result" id="css_result" type="text/css">body{color:#333
;background:#fff;padding:6px 0 0;margin:0;position:relative;min-width:90
0px}body,th,td,.p1,.p2{font-family:arial}p,form,ol,ul,li,dl,dt,dd,h3{mar
gin:0;padding:0;list-style:none}input{padding-top:0;padding-bottom:0;-mo
z-box-sizing:border-box;-webkit-box-sizing:border-box;box-sizing:border
-box}table,img{border:0}td{font-size:9pt;line-height:18px}em{font-style:
normal;color:#c00}a em{text-decoration:underline}cite{font-style:normal;
color:green}.m,a.m{color:#666}a.m:visited{color:#606}.g,a.g
[{'domain': '.baidu.com', 'httpOnly': False, 'name': 'H_PS_PSSID', 'path
': '/', 'secure': False, 'value': '1462_21090_26350'}, {'domain': '.baid
u.com', 'httpOnly': False, 'name': 'delPer', 'path': '/', 'secure': Fals
e, 'value': '0'}, {'domain': '.baidu.com', 'expiry': 3687782540.136964,
'httpOnly': False, 'name': 'BAIDUID', 'path': '/', 'secure': False, 'val
ue': '8CD0B232AEA7E5D1CCBEEDCFCD5D84F8:FG=1'}, {'domain': '.baidu.com',
'expiry': 3687782540.137061, 'httpOnly': False, 'name': 'PSTM', 'path':
'/', 'secure': False, 'value': '1540298899'}, {'domain': '.baidu.com', '
expiry': 3687782540.137026, 'httpOnly': False, 'name': 'BIDUPSID', 'path
': '/', 'secure': False, 'value': '8CD0B232AEA7E5D1CCBEEDCFCD5D84F8'}, {
'domain': 'www.baidu.com', 'httpOnly': False, 'name': 'BD_HOME', 'path':
'/', 'secure': False, 'value': '0'}, {'domain': '.baidu.com', 'expiry':
1540385299.383934, 'httpOnly': False, 'name': 'BDORZ', 'path': '/', 'sec
ure': False, 'value': 'B490B5EBF6F3CD402E515D22BCDA1598'}, {'domain': 'w
ww.baidu.com', 'expiry': 1541162893, 'httpOnly': False, 'name': 'BD_UPN'
```

```
, 'path': '/', 'secure': False, 'value': '12314753'}, {'domain': 'www.ba
idu.com', 'httpOnly': False, 'name': 'BD_CK_SAM', 'path': '/', 'secure':
False, 'value': '1'}, {'domain': '.baidu.com', 'httpOnly': False, 'name'
: 'PSINO', 'path': '/', 'secure': False, 'value': '1'}, {'domain': 'www.
baidu.com', 'expiry': 1540301491, 'httpOnly': False, 'name': 'H_PS_645EC
', 'path': '/', 'secure': False, 'value': '26caCOZZsBBXrUvxKEIfLpxL3INvz
InNGj4TUC1AbTDWuo9pRadwgTPXmGw'}, {'domain': 'www.baidu.com', 'httpOnly'
: False, 'name': 'BDSVRTM', 'path': '/', 'secure': False, 'value': '169'
}]
https://www.baidu.com/s?ie=utf-8&f=8&rsv_bp=1&rsv_idx=1&tn=baidu&wd=itca
st&oq=%25E9%2595%25BF%25E5%259F%258E&rsv_pq=dac9f5580007ec01&rsv_t=26caC
OZZsBBXrUvxKEIfLpxL3INvzInNGj4TUC1AbTDWuo9pRadwgTPXmGw&rqlang=cn&rsv_ent
er=0&inputT=181&rsv_sug3=9&rsv_sug4=181
```

页面操作

Selenium 的 WebDriver提供了各种方法来寻找元素，假设下面有一个表单输入框：

```
<input type="text" name="user-name" id="passwd-id" />
```

那么，可以有如下定位UI元素 (WebElements)的方法：

关于元素的选取，有如下的API 单个元素选取：

- find_element_by_id
- find_elements_by_name
- find_elements_by_xpath
- find_elements_by_link_text
- find_elements_by_partial_link_text
- find_elements_by_tag_name
- find_elements_by_class_name
- find_elements_by_css_selector

In [4]:

```
# 获取id标签值
import time
from selenium import webdriver

# 要想调用键盘按键操作需要引入keys包
from selenium.webdriver.common.keys import Keys

# 调用环境变量指定的PhantomJS浏览器创建浏览器对象
driver = webdriver.Chrome()
# 下面方法被废止了，单还可以用
# driver = webdriver.PhantomJS(executable_path='./phantomjs/bin/phantomj
s.exe')

# get方法会一直等到页面被完全加载，然后才会继续程序，通常测试会在这里选择 time.slee
p(2)
driver.get("http://mail.chinasafety.gov.cn/")
time.sleep(5)

######################
element = driver.find_element_by_id("F_password")
# 获取name标签值
element = driver.find_element_by_name("F_email")
# 获取标签名值
element = driver.find_elements_by_tag_name("input")
# 也可以通过XPath来匹配
```

```
element = driver.find_element_by_xpath("//input[@id='F_password']")
```

鼠标动作链

有些时候，我们需要再页面上模拟一些鼠标操作，比如双击、右击、拖拽甚至按住不动等，我们可以通过导入 ActionChains 类来做到：

In [ ]:
```
#导入 ActionChains 类
from selenium.webdriver import ActionChains

# 鼠标移动到 ac 位置
ac = driver.find_element_by_xpath('element')
ActionChains(driver).move_to_element(ac).perform()


# 在 ac 位置单击
ac = driver.find_element_by_xpath("elementA")
ActionChains(driver).move_to_element(ac).click(ac).perform()

# 在 ac 位置双击
ac = driver.find_element_by_xpath("elementB")
ActionChains(driver).move_to_element(ac).double_click(ac).perform()

# 在 ac 位置右击
ac = driver.find_element_by_xpath("elementC")
ActionChains(driver).move_to_element(ac).context_click(ac).perform()

# 在 ac 位置左键单击hold住
ac = driver.find_element_by_xpath('elementF')
ActionChains(driver).move_to_element(ac).click_and_hold(ac).perform()

# 将 ac1 拖拽到 ac2 位置
ac1 = driver.find_element_by_xpath('elementD')
ac2 = driver.find_element_by_xpath('elementE')
ActionChains(driver).drag_and_drop(ac1, ac2).perform()
```

填充表单

我们已经知道了怎样向文本框中输入文字，但是有时候我们会碰到 标签的下拉框。直接点击下拉框中的选项不一定可行。

```
<select id="status" class="form-control valid" onchange="" name="
status">
    <option value=""></option>
    <option value="0">未审核</option>
    <option value="1">初审通过</option>
    <option value="2">复审通过</option>
    <option value="3">审核不通过</option>
</select>
```

Selenium专门提供了Select类来处理下拉框。 其实 WebDriver 中提供了一个叫 Select 的方法，可以帮助我们完成这些事情：

In [ ]:
```
# 导入 Select 类
from selenium.webdriver.support.ui import Select
```

```
# 找到 name 的选项卡
select = Select(driver.find_element_by_name('status'))

#
select.select_by_index(1)
select.select_by_value("0")
select.select_by_visible_text(u"未审核")
```

以上是三种选择下拉框的方式，它可以根据索引来选择，可以根据值来选择，可以根据文字来选择。注意：

- index 索引从 0 开始
- value是option标签的一个属性值，并不是显示在下拉框中的值
- visible_text是在option标签文本的值，是显示在下拉框的值
- 全部取消选择怎么办呢？很简单：

> select.deselect_all()

弹窗处理

当你触发了某个事件之后，页面出现了弹窗提示，处理这个提示或者获取提示信息方法如下：

> alert = driver.switch_to_alert()

页面切换

一个浏览器肯定会有很多窗口，所以我们肯定要有方法来实现窗口的切换。切换窗口的方法如下：

> driver.switch_to.window("this is window name")

也可以使用 window_handles 方法来获取每个窗口的操作对象。例如：

```
for handle in driver.window_handles:
    driver.switch_to_window(handle)
```

页面前进和后退

操作页面的前进和后退功能：

```
driver.forward()      #前进
driver.back()         # 后退
```

Cookies

获取页面每个Cookies值，用法如下

```
for cookie in driver.get_cookies():
    print "%s -> %s" % (cookie['name'], cookie['value'])
```

删除Cookies，用法如下：

In [ ]:
```
# By name
driver.delete_cookie("CookieName")

# all
driver.delete_all_cookies()
```

页面等待

注意：这是非常重要的一部分！！

现在的网页越来越多采用了 Ajax 技术，这样程序便不能确定何时某个元素完全加载出来了。如果实际页面等待时间过长导致某个dom元素还没出来，但是你的代码直接使用了这个WebElement，那么就会抛出NullPointer的异常。

为了避免这种元素定位困难而且会提高产生 ElementNotVisibleException 的概率。所以 Selenium 提供了两种等待方式，一种是隐式等待，一种是显式等待。

隐式等待是等待特定的时间，显式等待是指定某一条件直到这个条件成立时继续执行。

显式等待

显式等待指定某个条件，然后设置最长等待时间。如果在这个时间还没有找到元素，那么便会抛出异常了。

In [ ]:
```
from selenium import webdriver
from selenium.webdriver.common.by import By
# WebDriverWait 库，负责循环等待
from selenium.webdriver.support.ui import WebDriverWait
# expected_conditions 类，负责条件出发
from selenium.webdriver.support import expected_conditions as EC

driver = webdriver.Chrome()
driver.get("http://www.xxxxx.com/loading")
try:
    # 页面一直循环，直到 id="myDynamicElement" 出现
    element = WebDriverWait(driver, 10).until(
        EC.presence_of_element_located((By.ID, "myDynamicElement"))
    )
finally:
    driver.quit()
```

如果不写参数，程序默认会 0.5s 调用一次来查看元素是否已经生成，如果本来元素就是存在的，那么会立即返回。

下面是一些内置的等待条件，你可以直接调用这些条件，而不用自己写某些等待条件了。

> title_is title_contains presence_of_element_located visibility_of_element_located visibility_of presence_of_all_elements_located text_to_be_present_in_element text_to_be_present_in_element_value frame_to_be_available_and_switch_to_it invisibility_of_element_located element_to_be_clickable – it is Displayed and Enabled. staleness_of element_to_be_selected element_located_to_be_selected element_selection_state_to_be element_located_selection_state_to_be alert_is_present

隐式等待

隐式等待比较简单，就是简单地设置一个等待时间，单位为秒。

如果不设置，默认等待时间为0。

In [ ]:
```
from selenium import webdriver

driver = webdriver.Chrome()
driver.implicitly_wait(10) # seconds
driver.get("http://www.xxxxx.com/loading")
myDynamicElement = driver.find_element_by_id("myDynamicElement")
```

案例一：网站模拟登录

In [6]:
```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time

driver = webdriver.Chrome()
driver.get("http://www.douban.com")

# 输入账号密码
driver.find_element_by_name("form_email").send_keys("xxxxx@xxxx.com")
driver.find_element_by_name("form_password").send_keys("xxxxxxxx")

# 模拟点击登录
driver.find_element_by_xpath("//input[@class='bn-submit']").click()

# 等待3秒
time.sleep(3)

# 生成登陆后快照
driver.save_screenshot("douban.png")

with open("douban.html", "wb") as file:
    file.write(driver.page_source.encode('utf-8'))

driver.quit()
```

案例二：动态页面模拟点击

下面的例子中用到了python单元测试库unitest。Unitest is a Python language version of JUnit,封装了一些校验返回的结果方法和一些用例执行前的初始化操作。

In [9]:
```
# python的测试模块
import unittest
from selenium import webdriver
from bs4 import BeautifulSoup

# 定义测试类
class douyuSelenium(unittest.TestCase):
    # 初始化方法
    def setUp(self):
        self.driver = webdriver.Chrome()
```

```
    #具体的测试用例方法，一定要以test开头
    def testDouyu(self):
        self.driver.get('http://www.douyu.com/directory/all')
        while True:
            # 指定xml解析
            soup = BeautifulSoup(driver.page_source, 'xml')
            # 返回当前页面所有房间标题列表 和 观众人数列表
            titles = soup.find_all('h3', {'class': 'ellipsis'})
            nums = soup.find_all('span', {'class': 'dy-num fr'})

            # 使用zip()函数来可以把列表合并，并创建一个元组对的列表[(1,2), (3,4)]
            for title, num in zip(nums, titles):
                print("观众人数:" + num.get_text().strip(), u"\t房间标题: "
+ title.get_text().strip())
            # page_source.find()未找到内容则返回-1
            if driver.page_source.find('shark-pager-disable-next') != -1
:
                break
            # 模拟下一页点击
            self.driver.find_element_by_class_name('shark-pager-next').c
lick()

    # 退出时的清理方法
    def tearDown(self):
        print('加载完成。')
        self.driver.quit()

if __name__ == "__main__":
    unittest.main()
```

```
E
======================================================================
ERROR: C:\Users\leo\AppData\Roaming\jupyter\runtime\kernel-00b407ad-5f07
-4a20-9adb-0dce861c29cf (unittest.loader._FailedTest)
----------------------------------------------------------------------
--
AttributeError: module '__main__' has no attribute 'C:\Users\leo\AppData
\Roaming\jupyter\runtime\kernel-00b407ad-5f07-4a20-9adb-0dce861c29cf'

----------------------------------------------------------------------
--
Ran 1 test in 0.002s

FAILED (errors=1)
```

```
An exception has occurred, use %tb to see the full traceback.

SystemExit: True
```

```
D:\pythonspace\Anaconda3\lib\site-packages\IPython\core\interactiveshell
.py:2971: UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

案例三：执行 JavaScript 语句

In [11]:

```
"""隐藏百度图片"""
from selenium import webdriver

driver = webdriver.Chrome()
driver.get("https://www.baidu.com/")
```

```
# 给搜索输入框标红的javascript脚本
js = "var q=document.getElementById(\"kw\");q.style.border=\"2px solid r
ed\";"

# 调用给搜索输入框标红js脚本
driver.execute_script(js)

#查看页面快照
driver.save_screenshot("redbaidu.png")

#js隐藏元素，将获取的图片元素隐藏
img = driver.find_element_by_xpath("//*[@id='lg']/img")
driver.execute_script('$(arguments[0]).fadeOut()',img)

# 向下滚动到页面底部
driver.execute_script("$('.scroll_top').click(function(){$('html,body').
animate({scrollTop: '0px'}, 800);});")

#查看页面快照
driver.save_screenshot("nullbaidu.png")

#driver.quit()
```

Out[11]: True

In [17]:
```
"""模拟滚动条滚动到底部"""

from selenium import webdriver
import time

driver = webdriver.Chrome()
driver.get("https://movie.douban.com/typerank?type_name=剧情&type=11&int
erval_id=100:90&action=")

# 向下滚动10000像素
js = "document.body.scrollTop=10000"
#js="var q=document.documentElement.scrollTop=10000"
time.sleep(3)

#查看页面快照
driver.save_screenshot("douban.png")

# 执行JS语句
driver.execute_script(js)
time.sleep(10)


#查看页面快照
driver.save_screenshot("newdouban.png")

#driver.quit()
```

Out[17]: True

本周作业

作业**1** 淘宝商品比价

需求

- 获取淘宝搜索页面的信息，提取其中的商品名称和价格。
- 需要设置翻页

技术路线

综合使用前面所学的技术，例如requests、bs4、selenium等

作业**2** 股票信息获取

需求

- 获取 https://quote.eastmoney.com/stocklist.html 股票代码及其信息。
- 使用selenium完成数据获取。

技术路线

综合使用前面所学的技术，例如requests、bs4、selenium等

In [ ]:
```
import requests


url = 'https://s.taobao.com/search?q=%E5%8C%85&imgfile=&js=1&stats_click
=search_radio_all%3A1&initiative_id=staobaoz_20181023&ie=utf8&cps=yes&ca
t=50072721&style=list'
url = 'https://s.taobao.com/search?q=%E5%8C%85&imgfile=&js=1&stats_click
=search_radio_all%3A1&initiative_id=staobaoz_20181023&ie=utf8&cps=yes&ca
t=50072721&style=list&bcoffset=3&ntoffset=3&p4ppushleft=1%2C48&s=44'
url = 'https://s.taobao.com/search?q=%E5%8C%85&imgfile=&js=1&stats_click
=search_radio_all%3A1&initiative_id=staobaoz_20181023&ie=utf8&cps=yes&ca
t=50072721&style=list&bcoffset=-3&ntoffset=-3&p4ppushleft=1%2C48&s=132'
url = 'https://s.taobao.com/search?q=%E5%8C%85&imgfile=&js=1&stats_click
=search_radio_all%3A1&initiative_id=staobaoz_20181023&ie=utf8&cps=yes&ca
t=50072721&style=list&bcoffset=-6&ntoffset=-6&p4ppushleft=1%2C48&s=176'
url = 'https://s.taobao.com/search?q=%E5%8C%85&imgfile=&js=1&stats_click
=search_radio_all%3A1&initiative_id=staobaoz_20181023&ie=utf8&cps=yes&ca
t=50072721&style=list&bcoffset=-9&ntoffset=-9&p4ppushleft=1%2C48&s=220'
headers = {
    'accept':'text/html,application/xhtml+xml,application/xml',
    'User-agent':'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.3
6 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36',
    'cookie':'t=d016fa94babd2250eaff8a9498b118a2; cna=3dQxFOF5o3MCAS1N80
+OQ9RM; v=0; cookie2=37762b60f6829a4dc0193c1fa56744fa; _tb_token_=0e7758
fb7565; _m_h5_tk=eb49defdc350a31c834e7aa220b959b8_1540292680835; _m_h5_t
k_enc=ac49058915beb661f460ed8f1c9f4202; hng=CN%7Czh-CN%7CCNY%7C156; thw=
cn; unb=75965512; sg=m23; _l_g_=Ug%3D%3D; skt=928616aa5fa2bbb7; cookie1=
BqJk9D7342ZkMXFfMykDvEYqmkiwE8r43rljzNeky5E%3D; csg=e56a1450; uc3=vt3=F8
dByRmn8Tvmr6%2B3AeA%3D&id2=VASjULlFuZE%3D&nk2=DeSe0Q8ONA%3D%3D&lg2=UIHiL
t3xD8xYTw%3D%3D; existShop=MTU0MDI4Mzk5OQ%3D%3D; tracknick=nchycom; lgc=
nchycom; _cc_=UtASsssmfA%3D%3D; dnk=nchycom; _nk_=nchycom; cookie17=VASj
ULlFuZE%3D; tg=0; enc=z9qg0OeNSubO2nGWaKEA%2FuasjejLkVGYxRh1EkLqX6XSLoiW
EBSMAAE0eX7T45F6fSfkmQb09WdC3gLEnn3IxQ%3D%3D; uc1=cookie16=VT5L2FSpNgq6f
DudInPRgavC%2BQ%3D%3D&cookie21=U%2BGCWk%2F7p4mBoUyS4E9C&cookie15=Vq8l%2B
KCLz3%2F65A%3D%3D&existShop=false&pas=0&cookie14=UoTYNkUdebPe%2BQ%3D%3D&
tag=8&lng=zh_CN; mt=ci=120_1; JSESSIONID=32B49DB7295E05FF2833188C3051421
```

```
5; swfstore=226808; whl=-1%260%260%260; x=e%3D1%26p%3D*%26s%3D0%26c%3D0
%26f%3D0%26g%3D0%26t%3D0%26__ll%3D-1%26_ato%3D0; isg=BImJ6O9SNvaJCMpjqv_
nR8bimLUjfiievB5qRiv_KnCvcq2EVida2BngsBbhMRVA',
}
payload
r = requests.get(url,headers = headers)
r.status_code
```

In [5]:

```
#CrawBaiduStocksA.py
import requests
from bs4 import BeautifulSoup
import traceback
import re

def getHTMLText(url):
    try:
        r = requests.get(url)
        r.raise_for_status()
        r.encoding = r.apparent_encoding
        return r.text
    except Exception as e:
        print(e)
        return ""
def getStockList(lst, stockURL):
    html = getHTMLText(stockURL)
    soup = BeautifulSoup(html, 'html.parser')
    a = soup.find_all('a')
    for i in a:
        try:
            href = i.attrs['href']
            lst.append(re.findall(r"[s][hz]\d{6}", href)[0])
        except Exception as e:
            print(e)
            continue
def getStockInfo(lst, stockURL, fpath):
    for stock in lst:
        url = stockURL + stock + ".html"
        html = getHTMLText(url)
        try:
            if html=="":
                continue
            infoDict = {}
            soup = BeautifulSoup(html, 'html.parser')
            stockInfo = soup.find('div',attrs={'class':'stock-bets'})

            name = stockInfo.find_all(attrs={'class':'bets-name'})[0]
            infoDict.update({'股票名称': name.text.split()[0]})

            keyList = stockInfo.find_all('dt')
            valueList = stockInfo.find_all('dd')
            for i in range(len(keyList)):
                key = keyList[i].text
                val = valueList[i].text
                infoDict[key] = val

            with open(fpath, 'a', encoding='utf-8') as f:
                f.write( str(infoDict) + '\n' )
        except Exception as e:
            print(e)
            traceback.print_exc()
```

```
            continue

def main():
    print('Begin to crawl...')
    stock_list_url = 'https://quote.eastmoney.com/stocklist.html'
    stock_info_url = 'https://gupiao.baidu.com/stock/'
    output_file = 'BaiduStockInfo.txt'
    slist=[]
    getStockList(slist, stock_list_url)
    getStockInfo(slist, stock_info_url, output_file)
main()
```

```
Begin to crawl...
HTTPSConnectionPool(host='quote.eastmoney.com', port=443): Max retries e
xceeded with url: /stocklist.html (Caused by SSLError(CertificateError("
hostname 'quote.eastmoney.com' doesn't match either of 'webssl.chinanetc
enter.com', 'i.l.inmobicdn.net', '*.fn-mart.com', 'www.1zhe.com', 'dl.jp
hbpk.gxpan.cn', 'dl.givingtales.gxpan.cn', 'dl.toyblast.gxpan.cn', 'dl.s
ds.gxpan.cn', 'download.ctrip.com', 'mh.tiancity.com', 'cdn.hxjyios.iwan
4399.com', 'ios.hxjy.iwan4399.com', 'gjzx.gjzq.com.cn', 'f.3000test.com'
, 'tj.img4399.com', '*.zhe800.com', '*.qiyipic.com', '*.vxinyou.com', '*
.gdjh.vxinyou.com', '*.3000.com', 'pay.game2.cn', 'static1.j.cn', 'stati
c2.j.cn', 'static3.j.cn', 'static4.j.cn', 'video1.j.cn', 'video2.j.cn',
'video3.j.cn', 'online.j.cn', 'playback.live.j.cn', 'audio1.guang.j.cn',
'audio2.guang.j.cn', 'audio3.guang.j.cn', 'img1.guang.j.cn', 'img2.guang
.j.cn', 'img3.guang.j.cn', 'img4.guang.j.cn', 'img5.guang.j.cn', 'img6.g
uang.j.cn', '*.4399youpai.com', 'w.tancdn.com', '*.3000api.com', 'static
11.j.cn', '*.kuyinyun.com', '*.kuyin123.com', '*.diyring.cc', '3000test.
com', '*.3000test.com', 'www.3387.com', '*.cankaoxiaoxi.com', '*.service
.kugou.com', 'xiuxiu.huodong.meitu.com', '*.meitu.com', '*.meitudata.com
', '*.wheetalk.com', 'xiuxiu.web.meitu.com', 'api.account.meitu.com', 'o
pen.web.meitu.com', 'id.api.meitu.com', 'api.makeup.meitu.com', 'im.live
.meipai.com', '*.meipai.com', 'img1.homekoocdn.com', 'cdn.homekoocdn.com
', 'cdn1.homekoocdn.com', 'cdn2.homekoocdn.com', 'cdn3.homekoocdn.com',
'cdn4.homekoocdn.com', 'img.homekoocdn.com', 'img2.homekoocdn.com', 'img
3.homekoocdn.com', 'img4.homekoocdn.com', '*.macauslot.com', '*.samsunga
pps.com', 'auto.tancdn.com', '*.winbo.top', 'static.bst.meitu.com', 'api
.xiuxiu.meitu.com', 'api.photo.meituyun.com', 'h5.selfiecity.meitu.com',
'api.selfiecity.meitu.com', 'h5.beautymaster.meiyan.com', 'api.beautymas
ter.meiyan.com', 'cpg.meitubase.com', '*.4399sy.com', 'ms.awqsaged.cn',
'fanxing2.kugou.com', 'fanxing.kugou.com', 'sso.56.com', 'upload.qf.56.c
om', 'sso.qianfan.tv', 'cdn.danmu.56.com', 'www-ppd.hermes.cn', 'www-uat
.hermes.cn', 'www-ts2.hermes.cn', 'www-tst.hermes.cn', '*.syyx.com', '*.
zhuoquapp.com', '*.5054399.com', '*.aiwan4399.com', 'user.beevideo.bestv
.com.cn', '*.3839.com', '*.actdelivery.net', '*.4399.cn', '*.yx3.com', '
*.163.com', 'm.kf.cn', 'cmscn.bmwgroup.com', 'secure-int-web-tic-cn.bmwg
roup.com', 'pvmessage.cn.bmwgroup.com', 'secure-infonet3.bmwgroup.com',
'secure-infonet3-int.bmwgroup.com', 'secure-web-tic-mini-cn.bmwgroup.co
m', 'secure-int-web-tic-mini-cn.bmwgroup.com', 'secure-infonet2-int.bmw
group.com', 'secure-web-tic-cn.bmwgroup.com', 'yjzhres.bnngame.com', '*.
account.meitu.com', 'cdn.ssjj.iwan4399.com', '*.iwan4399.com', 'm.kyxnz.
cn', 'wheetalk.com', 'mfanxing.kugou.com', 'redirect.contdelivery.com',
'market.suanya.com', 'm.bbs.3839.com', '*.mysiteres.com'",),))
```