对于采用异步加载技术的网页，可以使用Selenium模块模拟浏览器来进行爬虫，不用像之前那样通过网页的响应来反向构造爬虫。

Selenium是一个用于Web应用程序测试的工具，同时也是一个网络数据采集工具，直接运行在浏览器中，它可以让浏览器自动加载页面，这样就可以获取用异步加载技术的网页数据。

Selenium需要与浏览器一起使用，常用的浏览器是FireFox、Chrome。

Selenium与浏览器结合，可以使用带有GUI的浏览器，也可以是无界面（Headless）浏览器。

带有GUI的浏览器：

例：程序crawler\_test/crawler\_test15.py

from selenium import webdriver  
from selenium.webdriver.chrome.options import Options  
import time  
import pymongo  
from mongodb\_test import mongo\_host  
from mongodb\_test import mongo\_port  
  
  
client = pymongo.MongoClient(mongo\_host, mongo\_port)  
mydb = client['mydb']  
qq\_shuo = mydb['qq\_shuo']  
  
# 带有GUI  
driver = webdriver.Chrome()  
driver.maximize\_window()  
  
  
def get\_info(qq):  
 driver.get('https://user.qzone.qq.com/{0}/311'.format(qq))  
 driver.implicitly\_wait(10)  
  
 # 是否需要登陆  
 try:  
 driver.find\_element\_by\_id('login\_div')  
 flag = True  
 except:  
 flag = False  
  
 # 需要登录  
 if flag:  
 driver.switch\_to.frame('login\_frame')  
 driver.find\_element\_by\_id('switcher\_plogin').click()  
 driver.find\_element\_by\_id('u').clear()  
 driver.find\_element\_by\_id('u').send\_keys('357455175') # 账号  
 driver.find\_element\_by\_id('p').clear()  
 driver.find\_element\_by\_id('p').send\_keys('lss19880124') # 密码  
 driver.find\_element\_by\_id('login\_button').click()  
 time.sleep(3)  
 driver.implicitly\_wait(3)  
  
 # 是否可以访问QQ好友空间  
 try:  
 driver.find\_element\_by\_id('QM\_OwnerInfo\_Icon')  
 flag1 = True  
 except:  
 flag1 = False  
  
 if flag1:  
 driver.switch\_to.frame('app\_canvas\_frame')  
 contents = driver.find\_elements\_by\_css\_selector('.content')  
 times = driver.find\_elements\_by\_css\_selector('.c\_tx.c\_tx3.goDetail')  
 for content, tim in zip(contents, times):  
 data = {  
 'time': tim.text,  
 'content': content.text  
 }  
 qq\_shuo.insert\_one(data)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 get\_info('393466863')

无界面浏览器

例：程序crawler\_test/crawler\_test16.py

from selenium import webdriver  
from selenium.webdriver.chrome.options import Options  
from lxml import etree  
import time  
import pymongo  
from mongodb\_test import mongo\_host  
from mongodb\_test import mongo\_port  
  
client = pymongo.MongoClient(mongo\_host, mongo\_port)  
mydb = client['mydb']  
taobao = mydb['taobao']  
  
# 无GUI  
chrome\_options = Options()  
chrome\_options.add\_argument('--headless')  
chrome\_options.add\_argument('--disable-gpu')  
driver = webdriver.Chrome(chrome\_options=chrome\_options)  
  
  
def get\_info(url, page):  
 page += 1  
 driver.get(url)  
 driver.implicitly\_wait(10)  
 selector = etree.HTML(driver.page\_source)  
 infos = selector.xpath('//div[@class="item J\_MouserOnverReq "]')  
 for info in infos:  
 data = info.xpath('div/div/a')[0]  
 goods = data.xpath('string(.)').strip()  
 price = info.xpath('div/div/div/strong/text()')[0]  
 sell = info.xpath('div/div/div[@class="deal-cnt"]/text()')[0]  
 shop = info.xpath('div[2]/div[3]/div[1]/a/span[2]/text()')[0]  
 address = info.xpath('div[2]/div[3]/div[2]/text()')[0]  
 commodity = {  
 'good': goods,  
 'price': price,  
 'sell': sell,  
 'shop': shop,  
 'address': address  
 }  
 taobao.insert\_one(commodity)  
 if page <= 5:  
 next\_page(url, page) # 进入下一页  
 else:  
 pass  
  
  
def next\_page(url, page):  
 driver.get(url)  
 driver.implicitly\_wait(10)  
 driver.find\_element\_by\_xpath('//a[@trace="srp\_bottom\_pagedown"]').click()  
 time.sleep(4)  
 driver.get(driver.current\_url)  
 driver.implicitly\_wait(10)  
 get\_info(driver.current\_url, page)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 page = 1  
 url = 'https://www.taobao.com/'  
 driver.get(url)  
 driver.implicitly\_wait(10)  
  
 # 淘宝网页面搜索框输入的<input>元素id为q  
 driver.find\_element\_by\_id('q').clear()  
 driver.find\_element\_by\_id('q').send\_keys('男士短袖')  
  
 # 搜索按钮的class='btn-search'  
 driver.find\_element\_by\_class\_name('btn-search').click()  
 get\_info(driver.current\_url, page)

采用嵌入js脚本的方式来执行web页面的事件（如点击事件）

例：程序crawler\_test/crawler\_test17.py

from selenium import webdriver  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 driver = webdriver.Chrome()  
  
 page = 1  
 url = 'https://www.taobao.com/'  
 driver.get(url)  
 driver.implicitly\_wait(10)  
  
 # 淘宝网页面搜索框输入的<input>元素id为q  
 driver.find\_element\_by\_id('q').clear()  
 driver.find\_element\_by\_id('q').send\_keys('男士短袖')  
  
 # 搜索按钮的class='btn-search'  
 # 操作事件时采用对应的click()方法  
 # driver.find\_element\_by\_class\_name('btn-search').click()  
  
 # 采用嵌入js脚本操作事件  
 btn\_search = driver.find\_element\_by\_class\_name('btn-search')  
 driver.execute\_script('arguments[0].click();', btn\_search)

写爬虫脚本时发现，如果使用selenium操作时，貌似无法使用多进程爬虫，需要额外的配置。

爬取图片：

例：程序crawler\_test/crawler\_test18.py