

1.) Python run the Run.py at backend

```
1 from flask import Flask, request, jsonify
2 import time
3 from botasaurus_driver import Driver
4 from botasaurus.browser import Wait
5 from bs4 import BeautifulSoup
6 from deep_translator import GoogleTranslator, MyMemoryTranslator
7
8 app = Flask(__name__)
9
10
11 def close_intro_tooltip(driver):
12     try:
13         tooltips = driver.select_all("div.introjs-tooltip", wait=Wait.SHORT)
14         if tooltips:
15             driver.click("div.introjs-tooltip a.introjs-skipbutton", wait=Wait.SHORT)
16             time.sleep(1)
17     except Exception as e:
18         print("Tooltip close not found or an error occurred:", e)
19
20
21 def translate_chinese_to_english(text):
22     try:
23         return GoogleTranslator(source='auto', target='en').translate(text)
24     except Exception as e:
25         print(f"GoogleTranslator failed: {e}, trying MyMemoryTranslator...")
26         try:
27             return MyMemoryTranslator(source='auto', target='en').translate(text)
28         except Exception as e2:
29             print(f"MyMemoryTranslator also failed: {e2}")
30             # Either return the original text or raise an exception
31             return text # fallback is just returning the original text
32
33 @app.route('/search', methods=['POST'])
```

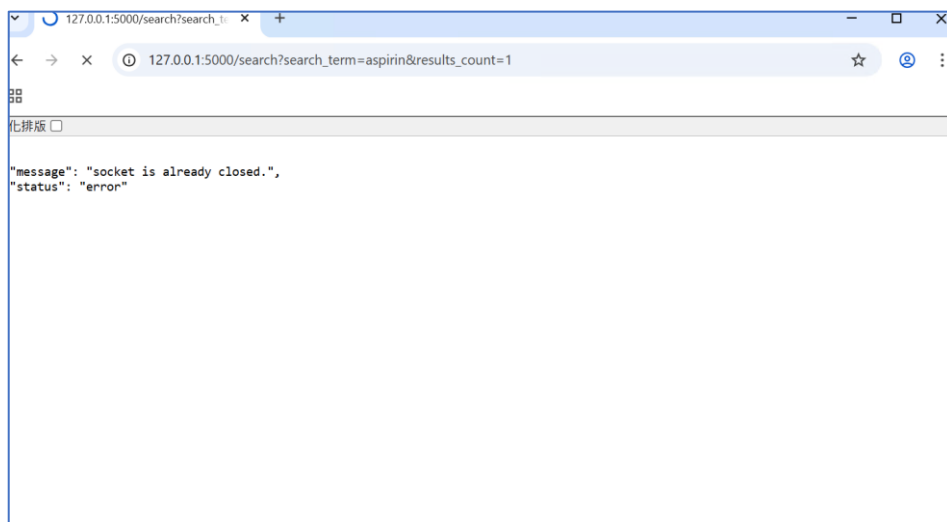
```
122 detail_tab.close()
123 driver.switch_to_tab(results_tab)
124 time.sleep(1)
125
126 # If the desired number of results is reached, break out of the loop
127 if results_count is not None and len(all_detail_data) >= results_count:
128     done = True
129     break
130
131 if done:
132     break
133
134 next_button = driver.select("button.btn-next", wait=Wait.SHORT)
135 if next_button and next_button.get_attribute("disabled") is None:
136     next_button.click(wait=Wait.LONG)
137     time.sleep(3)
138 else:
139     break
140
141 return jsonify({"status": "success", "detail_data": all_detail_data})
142
143 except Exception as e:
144     return jsonify({"status": "error", "message": str(e)})
145 finally:
146     try:
147         driver.close()
148     except Exception as close_err:
149         print("Error closing driver:", close_err)
150
151 if __name__ == '__main__':
152     app.run(host="0.0.0.0", port=5000, debug=True, use_reloader=False)
153
```

```
Command Prompt - python.exe
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd C:\Users\user\AppData\Local\Programs\Python\Python39

C:\Users\user\AppData\Local\Programs\Python\Python39>python.exe run.py
* Serving Flask app 'run'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://112.118.153.225:5000
Press CTRL+C to quit
```

2.) Open URL by 'http://127.0.0.1:5000/search?search_term=aspirin&results_count=1'



3.) A list of website carrying the data comes out and loop repeatedly.

国家药品监督管理局数据查询

mpa.gov.cn/datasearch/home-index.html

Google Lens

首页

政务服务

药品
Drugs

医疗器械
Medical Devices

化妆品
Cosmetics

境内生产药品

请输入批准文号 / 产品名称 / 上市许可持有人 / 生产单位

选择数据分类

关闭

上一步

下一步

境内生产药品

境外生产药品

境内生产药品备案信息公示

境外生产药品备案信息公示

药品生产企业

药品经营企业

国家药品监督管理局数据查询

国家药品监督管理局数据查询

mpa.gov.cn/datasearch/search-result.html

Chinese (Simplified)

English

Google Translate

首页

政务服务

境内生产药品

aspirin

高级搜索

搜索结果:

境内生产药品

序号	批准文号	产品名称	生产单位	药品本位码
1	国药准字H14022082	复方乙酰水杨酸片	长治市三宝生化药业有限公司	86902836000417
2	国药准字H20067545	阿司匹林泡腾片	吉林敖东延边药业股份有限公司	86903357000030
3	国药准字H14022718	复方乙酰水杨酸片	山西亨瑞达制药有限公司	86902886000245
4	国药准字H45021663	小儿氨酚匹林片	广西大力神制药股份有限公司	86905076000605
5	国药准字H61023178	阿苯片	西安利君制药有限责任公司	86902503000047
6	国药准字H20073550	小儿氨酚匹林咖啡因片	甘肃兰药药业有限公司	86905863002669

Full Python Program : ***How can it be run and test on Azure server ??***

```
from flask import Flask, request, jsonify
import time
from botasaurus_driver import Driver
from botasaurus.browser import Wait
from bs4 import BeautifulSoup
from deep_translator import GoogleTranslator, MyMemoryTranslator

app = Flask(__name__)

def close_intro_tooltip(driver):
    try:
        tooltips = driver.select_all("div.introjs-tooltip", wait=Wait.SHORT)
        if tooltips:
            driver.click("div.introjs-tooltip a.introjs-skipbutton", wait=Wait.SHORT)
            time.sleep(1)
    except Exception as e:
        print("Tooltip close not found or an error occurred:", e)

def translate_chinese_to_english(text):
    try:
        return GoogleTranslator(source='auto', target='en').translate(text)
    except Exception as e:
        print(f"GoogleTranslator failed: {e}, trying MyMemoryTranslator...")
    try:
```

```

        return MyMemoryTranslator(source='auto', target='en').translate(text)
except Exception as e2:
    print(f"MyMemoryTranslator also failed: {e2}")
    # Either return the original text or raise an exception
    return text # fallback is just returning the original text

@app.route('/search', methods=['POST'])
def search_product():
    data = request.get_json()
    category = data.get("category", "境内生产药品")
    search_term = data.get("search_term", "aspirin")

    # New parameter: number of results to process.
    # If results_count <= 0 or not provided, process all available links.
    try:
        results_count = int(data.get("results_count", 0))
    except ValueError:
        results_count = 0
    if results_count <= 0:
        results_count = None

    try:
        driver = Driver(headless=False, block_images=True)
        driver.get("https://www.nmpa.gov.cn/datasearch/home-index.html")
        time.sleep(10)
        driver.wait_for_element("div.pc-max.el-row", wait=Wait.LONG)
        category_selector = f"a[title='{category}']"
        driver.wait_for_element(category_selector, wait=Wait.LONG)

```

```
driver.click(category_selector, wait=Wait.LONG)
```

```
search_input_selector = "div.search-input input.el-input__inner"
```

```
driver.wait_for_element(search_input_selector, wait=Wait.LONG)
```

```
driver.type(search_input_selector, search_term, wait=Wait.LONG)
```

```
search_button_selector = "div.el-input-group__append button.el-button"
```

```
driver.wait_for_element(search_button_selector, wait=Wait.LONG)
```

```
driver.click(search_button_selector, wait=Wait.LONG)
```

```
time.sleep(5)
```

```
tabs = driver._browser.tabs
```

```
if len(tabs) < 2:
```

```
    return jsonify({"status": "error", "message": "Search results tab did not open."}), 500
```

```
results_tab = tabs[-1]
```

```
driver.switch_to_tab(results_tab)
```

```
close_intro_tooltip(driver)
```

```
pagination_size_selector = "span.el-pagination__sizes"
```

```
driver.wait_for_element(pagination_size_selector, wait=Wait.LONG)
```

```
driver.click(pagination_size_selector, wait=Wait.LONG)
```

```
time.sleep(1)
```

```
driver.run_js("""
```

```
Array.from(document.querySelectorAll('.el-select-dropdown__item'))
```

```
.find(item => item.innerText.trim() === '20 条/页')?.click();
```

```
""")
```

```
time.sleep(3)
```

```
results_table_selector = "div.search-result-table"
```

```
driver.wait_for_element(results_table_selector, wait=Wait.LONG)
```

```
all_detail_data = []
```

```
done = False
```

```
while True:
```

```
    close_intro_tooltip(driver)
```

```
    driver.wait_for_element(results_table_selector, wait=Wait.LONG)
```

```
    detail_buttons = driver.select_all("div.search-result-table button.el-button--primary.el-button--mini", wait=Wait.LONG)
```

```
    for button in detail_buttons:
```

```
        button.scroll_into_view()
```

```
        button.click(wait=Wait.LONG)
```

```
        time.sleep(3)
```

```
    detail_tab = next((tab for tab in driver._browser.tabs if tab != results_tab), None)
```

```
    if not detail_tab:
```

```
        continue
```

```
    driver.switch_to_tab(detail_tab)
```

```
    time.sleep(5)
```

```
    detail_html = driver.page_html
```

```
    detail_soup = BeautifulSoup(detail_html, "html.parser")
```

```
    table_div = detail_soup.find("div", id="dataTable")
```

```
    detail_data = {}
```

```

if table_div:
    tbody = table_div.find("tbody")
    if tbody:
        rows = tbody.find_all("tr")
        for row in rows:
            cols = row.find_all("td")
            if len(cols) >= 2:
                key = translate_chinese_to_english(cols[0].get_text(strip=True))
                value = translate_chinese_to_english(cols[1].get_text(strip=True))
                detail_data[key] = value

all_detail_data.append(detail_data)
detail_tab.close()
driver.switch_to_tab(results_tab)
time.sleep(1)

# If the desired number of results is reached, break out of the loop
if results_count is not None and len(all_detail_data) >= results_count:
    done = True
    break

if done:
    break

next_button = driver.select("button.btn-next", wait=Wait.SHORT)
if next_button and next_button.get_attribute("disabled") is None:
    next_button.click(wait=Wait.LONG)
    time.sleep(3)

```


else:

break

return jsonify({"status": "success", "detail_data": all_detail_data})

except Exception as e:

return jsonify({"status": "error", "message": str(e)})

finally:

try:

driver.close()

except Exception as close_err:

print("Error closing driver:", close_err)

if __name__ == '__main__':

app.run(host="0.0.0.0", port=5000, debug=True, use_reloader=False)