IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 5)
Available online at: www.ijariit.com

Automate data extraction

Keerthivas B. H.
spl.gamer2212@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

Rishi K.
rishi.ak007@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

Jeya Ganesh B.
jeyaganesh17@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

Jeya Srinivasan S.
sjsrini2@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

Veeramani
spl.gamer2212@gmail.com
SRM Institute of Science and Technology, Chennai, Tamil Nadu

ABSTRACT

Data extraction is considered as one of the most import steps in Software development life cycle (SDLC). Collecting data manually will extend the delivery time of a product. So data extraction part is automated in order to deliver things in time and also with nil errors.

Keywords— *Selenium, Bulk data, Python, Advantages and disadvantages, Existing system, Proposed system*

1. TO AUTOMATE BULK DATA

As we all know that working manually is much more difficult especially when there is a number of DATA. Our aim is to automate bulk numbers of data which reduce Manpower and it also saves time.

Eg. The user does not need to manually type the name of the content to get its detail and collect it one by one instead, all the content can be automated and the detail can be collected automatically. This saves time and makes it very easy for the user. This can be achieved by using Selenium.

2. TOOL USED

SELENIUM

3. LANGUAGE

PYTHON

4. EXISTING SYSTEM

a) Extraction of data for any circumstances in Software/Analytics area is done manually.
b) It also needs lots of Manpower to extract data manually.
c) Also, timely support needs to be done for dynamic data extraction from websites such as news, sports etc.

4.1 Disadvantages of existing system

a) Human error
b) Crossing delivery deadlines
c) Un-cleaned data
d) Need for support and maintenance on a timely basis.
e) Time taken for the quality check will be more when the data is improper.

5. PROPOSED SYSTEM

a) Selenium web driver is an automation testing tool which is used to automate the data extraction and these process will be implemented in Python.
b) "Which" data needs to be automatically extracted is feed into the scripts. The portion and the ways to extract the data from the website are scripted in python and configure it with selenium.
c) After the data extraction, script to clean data and the type of data delivery (CSV, Excel, database etc.) is written.

d) A random quality check is done and delivered.

5.1 Advantages of proposed system

a) No human error.
b) Less time consumption for data extraction.
c) Cleaned data.
d) Can run batch wise if the data is huge and also scripts can be called in a timely basis, which reduces support and maintenance?
e) The quality check will be easier since the data is in a structured manner and cleaned.

6. MODULES

6.1 Testing

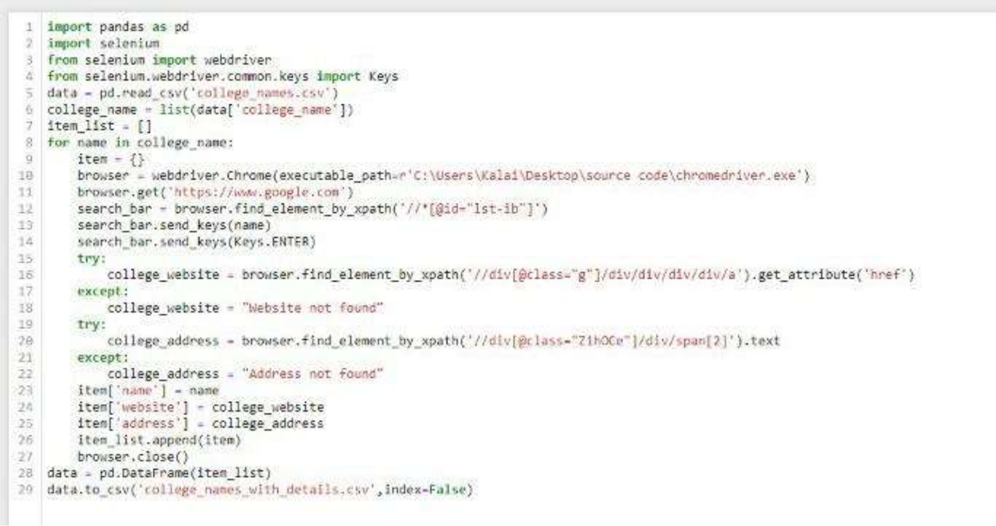
- To understand which "data" to be automated.
- Write test scripts and check for the output and compare with the sample output given by the end user.

6.2 Development

- After a successful test, the scripts are optimized and made to run completely.
- Handle Exceptions if any.

7. DELIVERY AND ITERATION

- Deliver the results in the desired manner.
- Check for any corrections from the end user.
- If any implement it in Testing stage and proceed for development.



```python
import pandas as pd
import selenium
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
data = pd.read_csv('college_names.csv')
college_name = list(data['college_name'])
item_list = []
for name in college_name:
    item = {}
    browser = webdriver.Chrome(executable_path=r'C:\Users\Kalai\Desktop\source code\chromedriver.exe')
    browser.get('https://www.google.com')
    search_bar = browser.find_element_by_xpath('//*[@id="lst-ib"]')
    search_bar.send_keys(name)
    search_bar.send_keys(Keys.ENTER)
    try:
        college_website = browser.find_element_by_xpath('//div[@class="g"]/div/div/div/a').get_attribute('href')
    except:
        college_website = "Website not found"
    try:
        college_address = browser.find_element_by_xpath('//div[@class="Z1hOCe"]/div/span[2]').text
    except:
        college_address = "Address not found"
    item['name'] = name
    item['website'] = college_website
    item['address'] = college_address
    item_list.append(item)
    browser.close()
data = pd.DataFrame(item_list)
data.to_csv('college_names_with_details.csv',index=False)
```

Fig. 1: Source code

8. CONCLUSION

Therefore this web-crawler type process enhances a better and swift way to extract bulk data from the preferred search engine (eg .Google), thus proving that automation is an upcoming self-replicating innovation in the near future.

9. REFERENCES

[1] https://www.seleniumhq.org/
[2] https://en.wikipedia.org/wiki/Web_crawler
[3] https://en.wikipedia.org/wiki/Data_extraction