УДК 004

Использование Selenium для парсинга отчетов в формате HTML

Киселева Елизавета Александровна
Приамурский государственный университет им. Шолом-Алейхема
Студент

Аннотация
Во многих компаниях, особенно крупных возникают моменты, когда требуется узнать характеристики всех компьютеров компании для дальнейшего апгрейда либо замены. Для данных задач отлично подходит программа AIDA 64. Но, к сожалению, каждый компьютер сохраняется в отдельном файле и для объединения всей информации в одну таблицу уйдет не мало времени. В статье описана разработка программы, которая сканирует отчеты программы AIDA 64 и сохранять всю информацию в отдельный файл.
Ключевые слова: Python, Selenium, Парсинг.

Using Selenium to parse HTML reports

Kiseleva Elizaveta Alexandrovna
Sholom-Aleichem Priamursky State University
Student

Abstract
In many companies where a further upgrade or replacement is required. AIDA 64 is perfect for these tasks. But, unfortunately, each computer uses all the necessary data for a single table. The article describes the development of a program that should contain reports on AIDA 64 programs and save all information in a separate file.
Keywords: Python, Selenium, Parser.

Во многих компаниях, особенно крупных возникают моменты, когда требуется узнать характеристики всех компьютеров компании для дальнейшего апгрейда либо замены. Для этого требуется зайти на каждый компьютер и вручную просмотреть требуемую информацию, такую, как оперативная память, процессор, видеокарта и т.д. К сожалению, это очень трудоемкий труд, и если вручную все просматривать, то может уйти и не один месяц. Для данных задач отлично подходит программа AIDA 64. Но, к сожалению, каждый компьютер сохраняется в отдельном файле и для объединения всей информации в одну таблицу уйдет не мало времени. Таким образом, в ходе работы планируется разработать простую программу, которая будет сканировать отчеты программы AIDA 64 и сохранять всю информацию в отдельный файл.

Цель исследования: разработка программы для сканирования отчетов программы AIDA 64 и сохранения результата в отдельном файле.

Многие ученые сталкивались с проблемой сбора и анализа данных. Т.А. Абрамова [1] описал разработку парсинг-системы для получения скрытых ссылок со страниц социальных сетей. М.Е. Кочитов [2] раскрыл возможность написание парсеров сайтов с помощью cURL и phpQuery. Т.С. Неволина и Р.А. Алешко [3] описали возможность парсинга больших объёмов данных. Р.П. Игнатьев [4] разработал программу для парсинга и вычисления арифметического выражения на языке php.

Программа AIDA 64 в приоритете сохранят файлы в HTML виде. Каждые снятые характеристики компьютера хранятся в отдельном файле. К сожалению, нет стандартных средств для объединения в отдельную таблицу. Таким образом, было решено использовать язык программирования Python для реализации данной цели. На рисунке 1 изображен пример отчета программы AIDA 64.

AIDA64 Engineer

Версия	AIDA64 v4.70.3237 Beta/ru
Тестовый модуль	4.1.627-x64
Домашняя страница	http://www.aida64.com/
Тип отчёта	Мастер отчётов
Компьютер	[redacted]
Генератор	
Операционная система	Microsoft Windows 7 Professional 6.1.7601.23796 (Win7 RTM)
Дата	2019-02-20
Время	15:50

Суммарная информация

Компьютер:	
Тип компьютера	ACPI x64-based PC
Операционная система	Microsoft Windows 7 Professional
Пакет обновления ОС	Service Pack 1
Internet Explorer	11.0.9600.18665
DirectX	DirectX 11.1
Имя компьютера	[redacted]
Имя пользователя	
Вход в домен	
Дата / Время	2019-02-20 / 15:50
Системная плата:	
Тип ЦП	QuadCore Intel Core i5-4590T, 2600 MHz (26 x 100)
Системная плата	Hewlett-Packard HP ProDesk 600 G1 DM
Чипсет системной платы	Intel Lynx Point Q85, Intel Haswell
Системная память	4001 МБ (DDR3-1600 DDR3 SDRAM)
DIMM1: Samsung M471B5173QH0-YK0	4 ГБ DDR3-1600 DDR3 SDRAM (11-11-11-28 @ 800 МГц) (10-10-10-27 @ 761 МГц)
Тип BIOS	Compaq (10/22/2014)
Коммуникационный порт	Intel(R) Active Management Technology - SOL (COM3)
Отображение:	
Видеоадаптер	Intel(R) HD Graphics 4600 (1819168 КБ)

Рисунок 1 – Пример отчета программы AIDA 64

Сканирование информации, можно заменить, словом, парсинг. Parser - это программное обеспечение для сбора данных и преобразования их в структурированный формат, чаще всего работа с текстовым типом информации.

Для разработки парсера данных имеется множество средств, в нашем случае, так как отчет хранится на веб-странице выбран Selenium в связке с Python. Selenium - это инструмент для автоматизации действий веб-браузера. В большинстве случаев используется для тестирования Web-приложений, но этим не ограничивается. В частности, он может быть использован для решения рутинных задач администрирования сайта или регулярного получения данных из различных источников.

На первом этапе требуется подключить веб драйвер строчкой «from selenium import webdriver» и создать переменую «driver», где прописан путь к файлу самого веб драйвера.

driver = webdriver.Chrome(executable_path='C:\\
chromedriver_win32\\chromedriver.exe')

Каждый файл по отдельности парсить не очень удобно. Поэтому они были помещены в отдельную папку. Именно это и облегчит получения имени файла для его дальнейшего открытия и сбора нужной информации. Для нахождения имен всех файлов в директории используется библиотека «os».

reports = [os.path.join(r, file) for r, d, f in os.walk("C:/test/") for file in f]

Далее для поиска нужных значений в HTML таблице используется встроенная функция в selenium «find_elements_by_xpath». На рисунке 2 изображен фрагмент кода для парсинга данных из таблицы.

```
for report in reports:
    report_counter += 1
    driver = self.driver
    driver.get(report)
    result[report_counter] = {}
    for parameter in parameters:
        elements = driver.find_elements_by_xpath(
            "//body/table[4]/tbody/tr/td[4][contains(text(),'" + parameter + "')]/following-sibling::td")

        values = []

        for element in elements:
            if (element.tag_name == 'a'):
                value = element.find_element_by_tag_name('a')
            elif (element.tag_name == 'td'):
                value = element.text
                value_element = value
            values.append(value_element)
        values = ' | '.join(values)
```

Рисунок 2 – Фрагмент года для парсинга данных из таблицы

Требуемые поля, из которых требуется информация берутся из переменной «parameters», где перечислены все поля, которые требуется получать из таблицы. На рисунке 3 изображены поля для получения значений.

```
def param_user(self):
    global value_element
    parameters = [
        'Операционная система',
        'Тип ЦП',
        'Системная плата',
        'Системная память',
        'Видеоадаптер',
        'Дисковый накопитель'
    ]
```

Рисунок 3 – Поля для получения значений

Далее создается файл формата CSV и данные через точку с запятой записываются по порядку с помощью цикла. На рисунке 4 изображен цикл для записи данных в таблицу.

```
file = open("C:/test/_result.csv", "w")

for parameterResult in parametersResult:
    file.write(parameterResult + ';')
file.write("\n")

for index, item in result.items():
    for parameterResult in parametersResult:
        if parameterResult in item:
            file.write(item[parameterResult])
        file.write(';')
    file.write("\n")
file.close()
```

Рисунок 4 – Цикл для записи данных в таблицу

По завершении сбора информации со всех файлов. CSV файл автоматически сохраняется и спокойно открывается в программе Excel. На рисунке 5 изображен результат программы.

A	B	C	D	
Операционная система	Тип ЦП	Системная плата	Системная память	Виде
Microsoft Windows 7 Prc	QuadCore Intel Core i5-4590T, 2600 MHz (26 x 100)	Hewlett-Packard HP ProDesk 600 G1 DM	4001 МБ (DDR3-1600 DDR3 S	Intel(
Microsoft Windows 7 Prc	DualCore Intel Core i3-4330, 3500 MHz (35 x 100)	Hewlett-Packard HP ProDesk 900 pm	3396 МБ (DDR3-1600 DDR3 S	VGA (

Рисунок 5 – Результат программы

Заключение. Таким образом, в ходе работы был написан парсер отчетов программы AIDA 64, которая способна собирать информацию о полных характеристиках компьютера и помещать в отдельную HTML таблицу. По результату работы, программа сохраняет требуемую информацию из всех

файлов, находящихся в отдельной директории в файл формата CSV. Удачно протестирован и используется в работе.

Библиографический список

1. Абрамова Т. А. Разработка парсинг-системы для получения скрытых ссылок со страниц социальных сетей //Вестник Пензенского государственного университета. 2016. №. 3 (15).
2. Кочитов М. Е. Парсинг сайтов с помощью cURL и phpQuery //Постулат. 2018. №. 8.
3. Неволина Т. С., Алешко Р. А. Парсинг больших объемов данных //Передовые инновационные разработки. Перспективы и опыт использования, проблемы внедрения в производство. 2019. С. 56-60.
4. Игнатьев Р. П. Парсинг и вычисление арифметического выражения на языке php //научное сообщество студентов xxi столетия. Технические науки. 2019. С. 44-48.