

DOI: 10.12731/2227-930X-2023-13-3-26-35

УДК 004.041



Научная статья | Эксплуатация автомобильного транспорта

УЛУЧШЕННОЕ ЛОГИРОВАНИЕ ОШИБОК ПРИ РАБОТЕ С МОДУЛЕМ SELENIUM НА ЯЗЫКЕ PYTHON

Р.Р. Крапивин, М.Р. Хамидулин

В статье рассматривается способ логирования (сохранения) ошибок, возникающие в результате работы модуля selenium, в случае возникновения непредвиденной или наоборот, предполагаемой ошибки, сохранить все данные о ней для будущего решения этой ошибки.

Цель: разработка метода для сохранения непредвиденных и предполагаемых ошибок возникающие в работе модуля selenium.

Метод или методология проведения работы: в статье ведется разработка модуля, способный сохранять текст ошибки, в каком месте кода она возникла, её время и изображение страницы браузера в момент возникновения ошибки.

Результат: разработан модуль по логированию ошибок selenium.

Ключевые слова: ошибка; selenium; Python; chrome driver; лог; сохранение

Для цитирования. Крапивин Р.Р., Хамидулин М.Р. Улучшенное логирование ошибок при работе с модулем selenium на языке Python // International Journal of Advanced Studies. 2023. Т. 13, № 3. С. 26-35. DOI: 10.12731/2227-930X-2023-13-3-26-35

Original article | Operation of Road Transport

IMPROVED ERROR LOGGING WHEN WORKING WITH THE SELENIUM MODULE IN PYTHON

R.R. Krapivin, M.R. Khamidulin

The article discusses the method of logging (saving) errors resulting from the operation of the selenium module, in case of an unexpected

or, conversely, an assumed error, save all data about it for future solutions to this error.

Purpose: development of a method for saving unexpected and expected errors that occur during the operation of the selenium module.

Method or methodology of work: the article is developing a module capable of saving the error text, where in the code it occurred, its time and the image of the browser page at the time of the error.

Result: the selenium error planning module has been developed.

Keywords: error; selenium; Python; chrome driver; log; save

For citation. Krapivin R.R., Khamidulin M.R. Improved Error Logging when Working with the Selenium Module in Python. *International Journal of Advanced Studies*, 2023, vol. 13, no. 3, pp. 26-35. DOI: 10.12731/2227-930X-2023-13-3-26-35

Введение

Обнаружение и устранение ошибок в работе модуля selenium весьма повседневная и рутинная задача каждого разработчика программного обеспечения. Из-за того, что динамичные сайты имеют непредвиденные сценарии работы, возникают ошибки, которые не позволяют выполнять работу программ парсинга корректно. Для решения подобных непредвиденных ошибок используют логирование.

Логирование – это фиксация негативных сценариев работы веб-ресурса, помогающая разработчику их обрабатывать, вовремя исправлять баги системы, предотвращать появление ошибок или сбоев в будущем.

В python так и в самом модуле selenium есть множество способов фиксации таких ошибок, но порой не понятно, как ошибка могла возникнуть, даже имея на руках зафиксированное событие, не всегда получается его обработать и предотвратить появление в будущих запусках программы. Для этого требуется сохранять изображение страницы окна браузера на момент возникновения ошибки, запись в логах, в какое время и на какой строчке возникла ошибка.

Цель работы: разработать метод, сохранения ошибки с изображением окна браузера на момент возникновения ошибки

Для реализации такого метода, сначала необходима какая-то структура, где будут выполняться тесты парсинга, например авторизация на популярном сайте разработчиков: ru.stackoverflow.com (рис. 1).

```

log error.py > ...
1
2 from selenium import webdriver
3
4 def open_driver():
5     global driver
6     driver = webdriver.Chrome()# Открываем окно браузера chrome driver
7     driver.maximize_window()# делает окно браузера на весь экран
8
9 def auf_site():
10    url = 'https://ru.stackoverflow.com/users/login?src=head&returnurl=https%3a%2f%2fru.stackoverflow.com%2f'
11    driver.get(url)# переход по ссылке авторизации
12    driver.find_element_by_xpath('//*[@id="email"]').send_keys('')# ввод почты
13    driver.find_element_by_xpath('//*[@id="password"]').send_keys('')# ввод пароля
14    driver.find_element_by_xpath('//*[@id="submit-button"]').click()# нажимаем войти
15
16 def main():
17     open_driver()
18     auf_site()
19
20 if __name__ == "__main__":
21     main()
  
```

Рис. 1. Код python для авторизации на сайте ru.stackoverflow.com.

Перед тем как авторизоваться, сайт будет иметь форму показанную на рисунке 2:

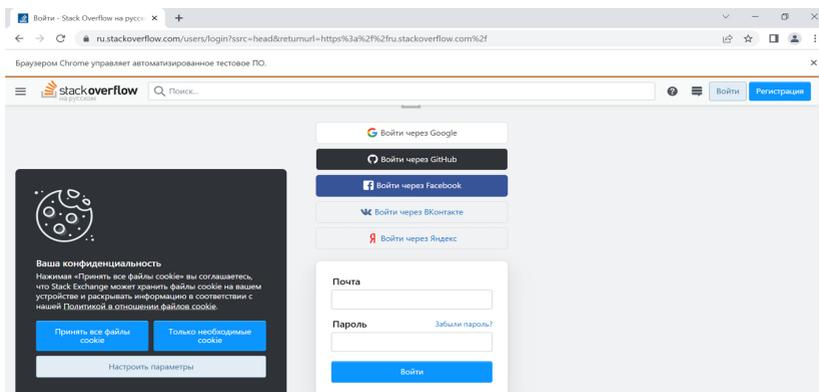


Рис. 2. Страница авторизации.

После успешной авторизации сайт будет выглядеть как показано на рисунке 3.

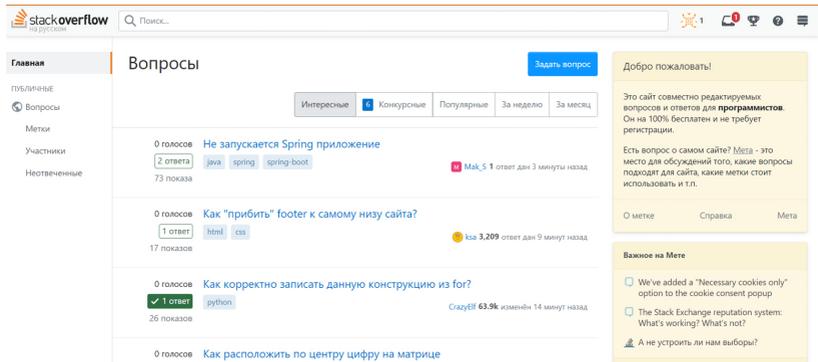


Рис. 3. Вид сайта после успешной авторизации.

Имя программу, которая способная проходить простейшую авторизацию, вводя почту и пароль, даже тут могут возникнуть ошибки, например:

1. Сайт не доступен.
2. Поменялась структура старлица авторизации.
3. Слишком долгое ожидание от сайта.

В реальных проектах, происходят вещи гораздо сложнее и количество ошибок в таких проектах растет в арифметической (иногда в геометрической) прогрессии. Для успешного логирования ошибок создадим функцию, которая будет срабатывать при возникновении ошибок (рис. 4).

```
def save_error(error):
    from datetime import datetime
    import sys
    import os

    datetime_error = datetime.now().strftime("%d-%m-%Y %H:%M")
    exc_type, exc_obj, exc_tb = sys.exc_info() # получаем номер строки кода, где возникла ошибка
    file_name = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
    error_string = "ERROR : Date: {}, File Name : {}, Line no : {}, Error Msg: {}".format(datetime_error, file_name, exc_tb.tb_lineno, error)# запись

    file_log = open("error_log.log", "a", encoding="utf-8")
    file_log.write(error_string)
    file_log.close()

    driver.save_screenshot(f'Error auf modul {datetime_error}.jpg')# сохраням изображение страницы на момент ошибки
    driver._exit()# закрываем экземпляр selenium chrome driver
```

Рис. 4. Функция логирования ошибок

Что в ней происходит:

1. На вход функции поступает текст ошибки
2. Импортируются встроенные библиотеки
3. Программа фиксирует время ошибки и помещает её в переменную `datetime_error`
4. Переменная `exc_td` храни номер линии кода, где возникла ошибка
5. `File_name` хранит имя исполняющего файла, т.е. `log_error.py`
6. `Error_string` формирует строку, содержащую информацию об ошибке, во сколько она была сохранена, на какой линии кода возникла, текст ошибки
7. Далее эта строка записывается в текстовый файл с расширением `log`, где она будет хранится
8. После всех операций, сохраняется изображение окна страницы на месте, где возникла ошибка
9. Закрывается образ `chrome drive`

Чтобы ошибка была сохранена, обернем все в блок `try` `except` (рис. 5).

```
def main():
    try:
        open_driver()
        auf_site()
    except Exception as error:
        save_error(error)
```

Рис. 5. Установка блока `try except`.

Итак, нужно проверить функцию, для этого искусственно вызовем ошибку в методе, где устанавливается пароль заменив строку

```
driver.find_element_by_xpath('//*[@id="password"]').send_keys('тут пароль')
```

на

```
driver.find_element_by_xpath('//ищем не существующее').send_keys('тут пароль')
```

Запускаем тест и имеем:

1. Запись в файле логов (рис. 6).
2. Изображение окна браузера на момент ошибки (рис. 7).

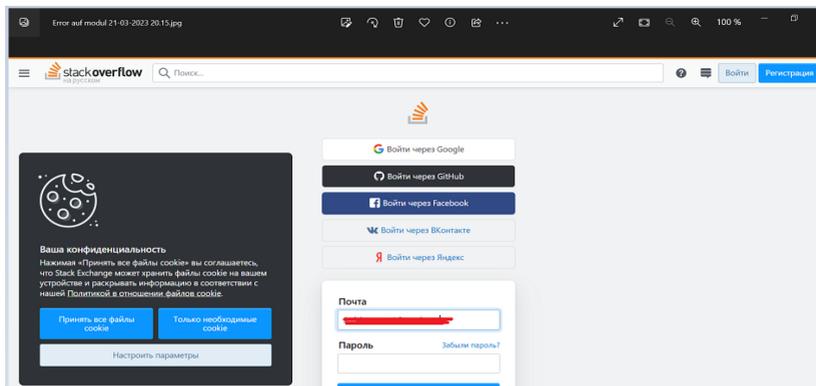


Рис. 6. Запись в логах ошибок

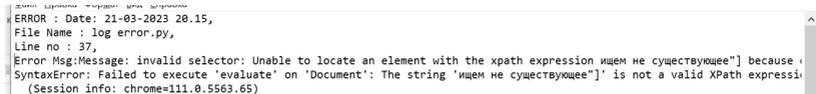


Рис. 7. Изображение окна браузера, где возникла ошибка

Результаты работы

Разработано программный метод для улучшенного логирования ошибок для модуля selenium, позволяющий сохранять не только текст ошибки, но и изображение места, где эта ошибка произошла для лучшей обработки непредвиденных ошибок. Такой метод можно использовать в любом участке кода, проекта и систем. В перспективе, развивая дальше такой метод, можно заполнять данные об ошибке в базу данных, не оставляя локальных файлов.

Список литературы

1. Логинова Е.В. Необходимость изучения информационных потоков предприятия / Е.В.Логинова, Т.А. Сарыева // Проблемы современной науки и образования, 2017. – № 2. С. 45-48.

2. Методы и модели исследования сложных систем и обработки больших данных: Монография / И. Ю. Парамонов, В. А. Смагин, Н. Е. Косых, А. Д. Хомоненко; под редакцией В. А. Смагина и А. Д. Хомоненко. – Санкт-Петербург: Лань, 2020. – 236 с.
3. Бенгфорт, Б. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка / Б. Бенгфорт. – СПб.: Питер, 2019. – 368 с.
4. Пономарева Л.А., Чискидов С.В., Ронжина И.А., Голосов П.Е. Проектирование компьютерных обучающих систем: Монография. М-во образования и науки РФ, РАНХиГС, МГПУ ИЦО. Тамбов: Консалтинговая компания Юком, 2018. 120 с.
5. Прокофьева Е.Н. Оценка качества управления информационными потоками в организациях / Е.Н. Прокофьева, А.В. Вострикова // Вестник РМАТ, 2017. – 330 с.
6. Прохоренок Н.А. Python 3 и PyQt. Разработка приложений. – СПб.: БХВ-Петербург, 2012. – 704 с.
7. Самойлова И. А. Технологии обработки больших данных / Молодой ученый. - 2017. - № 49 (183). - С. 26-28.
8. Модели и методы исследования информационных систем: монография / А.Д. Хомоненко, А.Г. Басыров, В.П. Бубнов [и др.]. - Санкт-Петербург: Лань, 2019. - 204 с.
9. Канаев К.А., Фалеева Е.В., Пономарчук Ю.В. Сравнительный анализ форматов обмена данными, используемых в приложениях с клиент-серверной архитектурой // Фундаментальные исследования. – 2015. – № 2-25. – С. 5569-5572.
10. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.
11. Виноградова Е.Ю. Интеллектуальные информационные технологии – теория и методология построения информационных систем: монография / М-во образования и науки РФ, Урал. гос. экон. ун-т. – Екатеринбург: Изд-во Урал. гос. экон. ун-та, 2011. – 263 с.
12. Белкова А. Л. Осваиваем работу с реляционными базами в MS Excel 2013 / А.Л. Белкова, С.Н. Леора // Теория и практика об-

- разования в современном мире: материалы VI Междунар. науч. конф. – Санкт-Петербург: Заневская площадь, 2014. – С. 349-356.
13. Уорсли, Дж. PostgreSQL. Для профессионалов / Дж. Уорсли, Дж. Дрейк. - М.: СПб: Питер, 2002. - 496 с.
 14. Hans-Jürgen Schönig Mastering PostgreSQL 13 - Fourth Edition: Build, administer, and maintain database applications efficiently with PostgreSQL 13. - Packt Publishing, – 2020. - 476 p.
 15. Baji Shaik, Avinash Vallarapu Beginning PostgreSQL on the Cloud: Simplifying Database as a Service on Cloud Platforms. – Apress, - 2018. - 381 p.

References

1. Loginova E.V. Necessity of studying information flows of an enterprise / E.V. Loginova, T.A. Sarieva // Problems of Modern Science and Education, 2017. - № 2. - pp. 45-48.
2. Methods and models of research of complex systems and big data processing: Monograph / I.Y. Paramonov, V.A. Smagin, N.E. Kosykh, A.D. Khomonenko; edited by V. A. Smagin and A. D. Khomonenko. - St. Petersburg: Lan', 2020. - 236 p.
3. Bengforth, B. Applied textual data analysis in Python. Machine learning and creating natural language processing applications / B. Bengforth. - St. Petersburg: Peter, 2019. - 368 p.
4. Ponomareva L.A., Chiskidov S.V., Ronzhina I.A., Golosov P.E. Designing computer learning systems: Monograph. Ministry of Education and Science of the Russian Federation, Russian Academy of National Economy and Public Administration, Moscow State Pedagogical University. Tambov: Consulting company Yukom, 2018. 120 p.
5. Prokofieva E.N. Assessment of the quality of information flow management in organizations / E.N. Prokof'eva, A.V. Vostrikova // Vestnik RMAT, 2017. - 330 p.
6. Prohorenok N.A. Python 3 and PyQt. Development of applications. - St. Petersburg: BHV-Peterburg, 2012. - 704 p.
7. Samoylova I. A. Technologies of big data processing / Young scientist. - 2017. - № 49 (183). - pp. 26-28.

8. Models and methods of research of information systems: monograph / A.D. Khomonenko, A.G. Basyrov, V.P. Bubnov [et al]. - Saint Petersburg: Lan^o, 2019. - 204 p.
9. Kanaev K.A., Faleeva E.V., Ponomarchuk Y.V. Comparative analysis of data exchange formats used in applications with client-server architecture // Fundamental Research. - 2015. - № 2-25. - pp. 5569-5572.
10. Zlatopolsky D.M. Fundamentals of programming in the Python language. - Moscow: DMK Press, 2017. - 284 p.
11. Vinogradova E. Yu. Intelligent information technology - theory and methodology of building information systems: monograph / Ministry of Education and Science of the Russian Federation, Ural State. Economics University. - Ekaterinburg: Publishing house of the Ural State University of Economics, 2011. - 263 p.
12. Belkova A.L. Mastering the work with relational databases in MS Excel 2013 / A.L. Belkova, S.N. Leora // Theory and practice of education in the modern world: proceedings of the VI International. scientific. conf. - St. Petersburg: Zanevskaya Square, 2014. - pp. 349-356.
13. Worsley, J. PostgreSQL. For professionals / J. Worsley, J. Drake. - M.: SPb: Peter, 2002. - 496 p.
14. Hans-Jürgen Schönig Mastering PostgreSQL 13 - Fourth Edition: Build, administer, and maintain database applications efficiently with PostgreSQL 13. - Packt Publishing, – 2020. - 476 p.
15. Baji Shaik, Avinash Vallarapu Beginning PostgreSQL on the Cloud: Simplifying Database as a Service on Cloud Platforms. – Apress, - 2018. - 381 p.

ДАнные ОБ АВТОРАХ

Крапивин Роман Русланович, студент

*Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ
ул. Академика Королева, 1, г. Набережные Челны, 423814,
Российская Федерация
Jerichotyran1@yandex.ru*

Хамидуллин Марат Раисович, доцент, кандидат экономических наук

Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ

ул. Академика Королева, 1, г. Набережные Челны, 423814,

Российская Федерация

nyaika_prom@mail.ru

DATA ABOUT THE AUTHOR

Roman R. Krapivin, Student

Kazan National Research Technical University named after A.N. Tupolev-KAI

1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation

Jerichotyran1@yandex.ru

Marat R. Khamidullin, PhD in Economics

Kazan National Research Technical University named after A.N. Tupolev-KAI

1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation

nyaika_prom@mail.ru

ORCID: <https://orcid.org/0000-0002-3326-0955>

Поступила 02.04.2023

После рецензирования 15.04.2023

Принята 20.04.2023

Received 02.04.2023

Revised 15.04.2023

Accepted 20.04.2023