**ДОМАШНЯЯ РАБОТА №3**

*На уроке мы разбирали основные принципы запуска и структуру автоматизированных тестов на базе Python + SeleniumWebDriver + Behave. В качестве базы мы использовали тестовый сценарий на базе собраного по сайту https://habr.com/ru/post/262929/.*

*В качестве домашнего задания нужно обновить тестовый сценарий и сделать так, чтобы он проверял результаты запроса на поиск в поисковик. Необходимо проверить количество выданных поиском результатов на странице, общее количество найденных страниц (текст в результате поиска) и проверить правильное размещение картинок в результатах поиска.*

*Напомню условия задачи:*

|  |
| --- |
| *Задача:*   1. *Исправить тестовый сценарий таким образом, чтобы он проверял результаты запроса на поиск в поисковик.* 2. *Необходимо проверить корректность поиска* 3. *Необходимо проверить количество выданных поиском результатов на странице*      1. *Необходимо проверить общее количество найденных страниц (текст в результате поиска)*      1. *Необходимо проверить правильное размещение картинок в результатах поиска.* |
| *Ограничения:*   1. *Тест должен содержать один файл \*.feature.* 2. *Тест должен содержать 2 сценария (проверка в 2х браузерах)..* |
| *Подсказки, найденные на паре:*   1. *Смотрите скриншоты в чат* |
| *Что будет оцениваться:*   1. *Работа тестового сценария* 2. *Именование переменных* 3. *Наличие функций и их именование.* 4. *Структура проекта (насколько легко разобраться в том что и как работает)* |

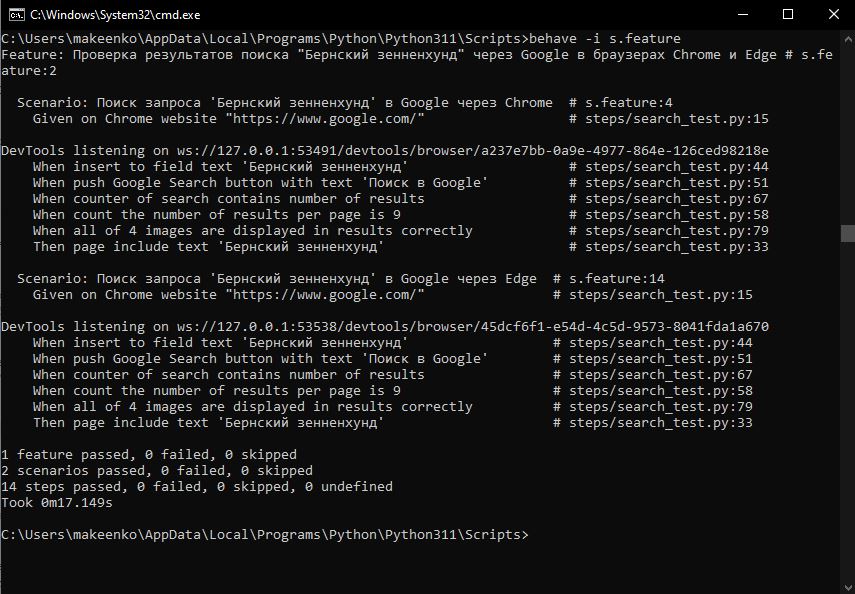
Скрипт открывает Google в двух браузерах (Chrome, Edge), находит поле ввода запроса, вводит запрос, кликает кнопку поиска, далее проверяет отображение счётчика результатов и сравнивает полученное количество и с искомым, после этого считает количество результатов на странице и сравнивает с ожидаемым значением, а также проверяет корректность отображения всех 4 изображений в результатах, согласно стилю, и наконец проверяет что открылась страница, содержащая текст запроса и выводит отчёт.

Файл s.feature

|  |
| --- |
| Feature: Проверка результатов поиска "Бернский зенненхунд" через Google в браузерах Chrome и Edge  Scenario: Поиск запроса 'Бернский зенненхунд' в Google через Chrome  Given on Chrome website "https://www.google.com/"  When insert to field text 'Бернский зенненхунд'  When push Google Search button with text 'Поиск в Google'  When counter of search contains number of results  When count the number of results per page is 9  When all of 4 images are displayed in results correctly  Then page include text 'Бернский зенненхунд'    Scenario: Поиск запроса 'Бернский зенненхунд' в Google через Edge  Given on Chrome website "https://www.google.com/"  When insert to field text 'Бернский зенненхунд'  When push Google Search button with text 'Поиск в Google'  When counter of search contains number of results  When count the number of results per page is 9  When all of 4 images are displayed in results correctly  Then page include text 'Бернский зенненхунд' |

Файл search\_test.py

|  |
| --- |
| 1. **from** behave **import** **\*** 2. **from** selenium **import** webdriver 3. **from** selenium.webdriver.common.keys **import** Keys 4. **from** selenium.webdriver.common.by **import** By 5. **from** selenium.webdriver.support.ui **import** WebDriverWait 6. **from** selenium.webdriver.support **import** expected\_conditions as EC 7. **from** selenium.webdriver.chrome.options **import** Options 8. **import** time 10. #================ GENERAL FUNCTIONS ==================#  13. #Откроем главную страницу в Chrome. Передадим в качестве аргумента адрес страницы. 14. @given('on Chrome website "{url}"') 15. **def** step(context, url): 16. #Измените строку, для выполнения теста в другом браузере 17. chrome\_options **=** Options() 19. context.browser **=** webdriver.Chrome() 20. context.browser.maximize\_window() 21. context.browser.get(url) 23. #Откроем главную страницу в Edge. Передадим в качестве аргумента адрес страницы. 24. @given('on Edge website "{url}"') 25. **def** step(context, url): 26. #Измените строку, для выполнения теста в другом браузере 27. context.browser **=** webdriver.Edge() 28. context.browser.maximize\_window() 29. context.browser.get(url) 31. #Проверим, что мы на странице с результатами поиска, есть некоторый искомый текст 32. @then("page include text '{text}'") 33. **def** step(context, text): 34. WebDriverWait(context.browser, 120).until( 35. EC.presence\_of\_element\_located((By.XPATH, '//\*[contains(text(), "%s")]' **%** text)) 36. ) 37. **assert** context.browser.find\_element(By.XPATH,'//\*[contains(text(), "%s")]' **%** text)  40. #====================== GOOGLE ===========================# 42. #Теперь введём запрос в Google 43. @when("insert to field text '{text}'") 44. **def** step(context, text): 45. context.browser.implicitly\_wait(4) 46. context.browser.find\_element(By.CSS\_SELECTOR, '[name="q"]').send\_keys(text)  49. #Теперь нажмем на кнопку "Поиск в Google" 50. @when("push Google Search button with text '{text}'") 51. **def** step(context, text): 52. context.browser.find\_element(By.CSS\_SELECTOR, '[name="btnK"]').click()  55. #Проверяем, что количество выданных поиском результатов на странице равно 9 56. @when("number of search results per page is 9") 57. **def** step(context): 58. CounterOfResultOnPage **=** context.browser.find\_elements(By.XPATH, '//\*[@id="rso"]/div') 60. **assert** int(len(CounterOfResultOnPage)) **==** 9  63. #Проверяем наличие отчёта о найденных результатах и его содержание 64. @when ("counter of search contains number of results") 65. **def** step(context, ): 66. TextOfResault **=** context.browser.find\_element(By.XPATH, '//\*[@id="result-stats"]').is\_displayed() 67. context.browser.find\_element(By.XPATH,'//\*[contains(text(), "Результатов: примерно 777")]')  70. #Проверяем правильность размещения картинок в результатах поиска 71. @when ("all of 4 images are displayed in results correctly") 72. **def** step(context): 73. context.browser.implicitly\_wait(20) 74. ImageInResult **=** context.browser.find\_element(By.XPATH,'//\*[@alt = "Бернский зенненхунд, источник: ru.wikipedia.org"]').is\_displayed() 75. StyleCorrect **=** context.browser.find\_elements(By.XPATH,"//\*[contains(@style,'border-radius:8px;height:92px;width:92px')]") 77. **assert** ImageInResult **==** True **and** int(len(StyleCorrect)) **==** 4 |



Судя по отчёту в консоли, оба сценария и 14 шагов прошли успешно. Однако должен заметить, что количество найденных результатов порой меняется с 777 тысяч на другие значения, а позже возвращается.