**ДОМАШНЯЯ РАБОТА №4**

***Задание***

*На уроке мы разворачивали и собирали систему для автоматизированного тестирования на python, а также смотрели тестовый вариант сценария для тестов (собран по сайту https://habr.com/ru/post/262929/). Тестовый сценарий не проходил.*

*В качестве домашнего задания нужно обновить тестовый сценарий и сделать так, чтобы он проходил.*

*Напомню условия задачи:*

|  |
| --- |
| *Задача:*   1. *Исправить тестовый сценарий таким образом, чтобы он проходил.* |
| *Ограничения:*   1. *Изменения в сценарии должны быть минимальными.* |
| *Подсказки, найденные на паре:*   1. *Сценарий не проходит, потому как на странице нет кнопки “Найти” на которую ему нужно нажать.* 2. *В качестве решения задачи можно выбрать другой поисковик для теста.* 3. *В качестве решения задачи, можно попытаться что-нибудь ввести в строку поиска, чтобы кнопка поиск появилась.* |
| *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)* |

Мне удалось добиться успешного прохождения всех трёх шагов теста. Для этого пришлось редактировать аргументы вызываемых функций, поскольку в примере был код для старой версии selenium [(ссылка).](https://pythoninoffice.com/fixing-attributeerror-webdriver-object-has-no-attribute-find_element_by_xpath/) Пример на скриншоте ниже.



Скрипт открывает Яндекс в Firefox, находит поле ввода запроса, вводит «пробел», имитирует нажатие «ENTER», затем проверяет что открылась страница «Задан пустой поисковый запрос» и выводит отчёт.

|  |
| --- |
| 1. **from** behave **import** **\*** 2. **from** selenium **import** webdriver # необходим в этом скрипте, отсутствовал в примере 3. **from** selenium.webdriver.common.keys **import** Keys 4. **from** selenium.webdriver.common.by **import** By # необходим для новой версии Selenium 5. **from** selenium.webdriver.support.ui **import** WebDriverWait 6. **from** selenium.webdriver.support **import** expected\_conditions as EC 7. **import** time 9. #Откроем главную страницу. Передадим в качестве аргумента адрес страницы. 10. @given('website "{url}"') 11. **def** step(context, url): 12. #Измените строку, для выполнения теста в другом браузере 13. context.browser **=** webdriver.Firefox() 14. context.browser.maximize\_window() 15. context.browser.get("[http://ya.ru](http://ya.ru/)") 17. #Теперь нажмем на кнопку "Найти" 18. @then("push button with text '{text}'") 19. **def** step(context, text): 20. WebDriverWait(context.browser, 120).until( 21. EC.element\_to\_be\_clickable((By.XPATH, '//button')) 22. ) 23. context.browser.find\_element(By.NAME, 'text').send\_keys(' ') 24. context.browser.find\_element(By.NAME, 'text').send\_keys(Keys.ENTER)  27. #Проверим, что мы на странице с результатами поиска, есть некоторый искомый текст 28. @then("page include text '{text}'") 29. **def** step(context, text): 30. WebDriverWait(context.browser, 120).until( 31. EC.presence\_of\_element\_located((By.XPATH, '//\*[contains(text(), "%s")]' **%** text)) 32. ) 33. **assert** context.browser.find\_element(By.XPATH,'//\*[contains(text(), "%s")]' **%** text) 35. context.browser.quit() |



