

## Лабораторная работа № 7. Работа с формами в Selenium WebDriver. Тестирование авторизации в веб-приложениях.

**Цель работы.** Получение базовых навыков работы с формами в Selenium WebDriver.

### Теоретическая часть.

В работе с формами на веб-странице как при ручном, так и при автоматизированном тестировании выделяют следующие шаги для их обработки:

1. Ввод пользовательских данных в поля формы.
2. Отправка формы на сервер для обработки.

Рассмотрим подробно каждый шаг с позиции автоматизации инструментом Selenium WebDriver:

1. Ввод пользовательских данных в поля формы.

В теоретической части лабораторной работы, посвященной первоначальному знакомству с Selenium WebDriver, были приведены функции для поиска элементов на веб-странице. При работе с формами, для поиска и задания значений полям формы используются те же функции.

2. Отправка формы на сервер.

Существует несколько способов отправки заполненной формы на сервер с помощью Selenium WebDriver. Рассмотрим основные из них:

а. Программное нажатие на кнопку отправки формы. Как правило, такие кнопки располагаются после всех полей для ввода пользовательской информации (рисунок 1).

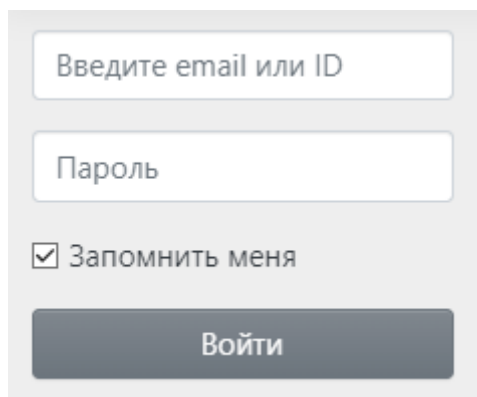
The image shows a login form with a light gray background. It contains two input fields: the first is labeled 'Введите email или ID' and the second is labeled 'Пароль'. Below these fields is a checkbox with the label 'Запомнить меня'. At the bottom of the form is a dark gray button with the text 'Войти'.

Рисунок 1 – Пример кнопки отправки формы на обработку серверу  
Для того, чтобы программно «нажать» на кнопку отправки данных необходимо:

1) Найти ее на форме функцией `find_element()` и нажать на нее, используя функцию `click()`.

Например, найдем кнопку с ID `button` и программно «нажмем» на нее:

```
driver.find_element(By.ID, "button").click()
```

Единственным и основным минусом данного подхода является отсутствие уникальных идентификаторов у кнопки отправки формы на сервер.

Зачастую эти кнопки не имеют ID, уникальных стилей, по которым можно было бы найти данную кнопку на всей веб-странице.

В таком случае следует использовать 2-ой способ.

b. Программная отправка формы на сервер. В данном способе необходимо произвести следующую последовательность действий:

- 1) Найти форму на странице по ее ID командой `find_element()`.
- 2) Отправить форму на сервер, используя функцию `submit()`.

Например, найдем форму с ID `login_form` и отправим ее на сервер:

```
driver.find_element(By.ID, "login_form").submit()
```

### **Тестирование функциональности авторизации в веб-приложении.**

Тестирование работы формы авторизации реализуется выполнением следующих последовательных шагов:

1. Заполнение формы пользовательскими данными
2. Отправка формы на сервер
3. Проверка корректности работы алгоритма авторизации

Первые шаги тестирования алгоритма авторизации в веб-приложении описаны в предыдущей главе данной ЛР. Рассмотрим последний шаг, описанный выше - проверка корректности работы алгоритма авторизации.

После отправки формы на сервер со страницы авторизации может возникнуть 2 ситуации:

1. Авторизация прошла успешно.
2. Авторизация завершилась ошибкой. Причиной может стать неверно введенная информация на форме, отсутствие пользователя с введенным логином/email/телефоном и т.п.

Как правило, успешная авторизация сопровождается перенаправлением на страницу профиля пользователя; авторизация, завершившаяся ошибкой – перенаправление на ту же страницу авторизации с текстом возникшей ошибки.

При ручном тестировании алгоритма авторизации данные сообщения легко увидеть невооруженным глазом. Проблема появляется, когда данный процесс необходимо автоматизировать.

Отследить успех и ошибку авторизации в Selenium WebDriver можно следующим образом:

1. Как было отмечено выше, успешная авторизация завершается перенаправлением на другую страницу (чаще всего – страницу профиля), что с легкостью позволяет проверить факт авторизации – достаточно всего лишь

найти какой-либо элемент на странице профиля, который виден только у авторизованных пользователей (например, кнопка «Выйти» и т.п.).

Если такой элемент присутствует на странице, значит авторизация прошла успешно.

2. Если же такого элемента на странице нет, значит авторизация завершилась ошибкой и перенаправления на другую страницу не произошло.

В Selenium WebDriver отследить наличие кнопки можно с помощью следующей конструкции:

```
# Отправим форму login_form на сервер
driver.find_element(By.ID, "login_form").submit()

try:
    # Пытаемся найти кнопку выхода по ID
    exit_button = driver.find_element(By.ID, "exit_button")

    # Если не сгенерировалось исключение
    NoSuchElementException, значит авторизация прошла
    успешно

    # В данном блоке необходимо писать логику,
    связанную с успешной авторизацией (например,
    нажатие на кнопку выхода для дальнейшего тестирования
    авторизации)
```

```
except NoSuchElementException:
    # Если данная кнопка не находится на странице,
    возникает исключение NoSuchElementException
    # В данном блоке необходимо писать логику,
    связанную с возникновении ошибки при авторизации
```

### Практическая часть

Необходимо разработать скрипт для тестирования алгоритма авторизации сайта <https://litlife.club>.

Входные данные: массив объектов (рисунок 2), каждый элемент которого представлен в виде:

- 0 – логин;
- 1 – пароль;

- 2 – флаг, соответствующий тому, должна ли выполняться авторизации с данными логином и паролем.

```
[
    "nikita.developer97@gmail.com",
    "Test12345",
    True
],
[
    "1234@gmail.com",
    "1234",
    False
],
[
    "testing@gmail.com",
    "test1278",
    True
]
```

Рисунок 2 – Набор входных данных

Выходные данные: сведения о выполненных тестах в следующем формате:

Done: <кол-во тестов, выполненных успешно>

Wrong: <кол-во тестов, завершенных ошибкой>

Шаблон для написания скрипта тестирования находится в приложении к данной лабораторной работе – файл main.py.

### **Содержание отчёта.**

По результатам выполнения работы оформляется отчет в соответствии с требованиями ГОСТ 7.32-2017 «Отчет о научно-исследовательской работе. Структура и правила оформления», включающий:

- титульный лист;
- цель работы;
- описание хода выполнения работы;
- скрипт, написанный на языке Python;
- выводы.

### **Контрольные вопросы.**

1. Опишите основные шаги при работе с формами с позиции инструмента Selenium WebDriver и дайте краткую характеристику каждого шага.
2. Перечислите основные способы отправки заполненной формы на сервер с помощью Selenium WebDriver.

3. Перечислите основные шаги, необходимые для тестирования работы формы авторизации.

4. Какие могут возникнуть ситуации после отправки формы авторизации на сервер. В каких случаях они могут возникнуть?

5. Как можно отследить успех и ошибку авторизации в Selenium WebDriver?

6. Напишите скрипт на языке Python для занесения данных в поля формы и отправки формы на сервер. Веб-приложение/сайт выбрать на свое усмотрение.