Лабораторная работа № 6. Знакомство с Selenuim WebDriver. Установка и выполнение простейших операций в Selenium.

Цель работы. Получение базовых навыков работы с инструментом для автоматизации действий веб-браузера Selenium WebDriver на языке Python.

Теоретическая часть

Selenium WebDriver – программная библиотека для управления браузерами. WebDriver представляет собой драйверы для различных браузеров и клиентские библиотеки на разных языках программирования, предназначенные для управления этими драйверами. Использование такого веб-драйвера сводится к созданию бота, выполняющего всю ручную работу с браузером автоматизированно.

Библиотеки WebDriver доступны на языках Java, C#, Python, Ruby, JavaScript. Драйверы реализованы для браузеров Firefox, InternetExplorer, Chrome и Opera.

Selenium WebDriver используется для тестирования функционала веб-сайтов/ веб-ориентированных приложений.

Автоматизированное тестирование удобно, потому что позволяет многократно запускать повторяющиеся тесты. Регрессионное тестирование, является типичным примером, когда необходима автоматизация.

Selenium WebDriver предоставляет все необходимые методы, обеспечивает высокую скорость теста и гарантирует корректность проверки (поскольку человеческий фактор исключен).

В официальной документации Selenium приводятся следующие плюсы автоматизированного тестирования веб-приложений:

- возможность проводить чаще регрессионное тестирование;
- быстрое предоставление разработчикам отчета о состоянии продукта;
- получение потенциально бесконечного числа прогонов тестов;
- обеспечение поддержки Agile и экстремальным методам разработки;
- сохранение строгой документации тестов;
- обнаружение ошибок, которые были пропущены на стадии ручного тестирования.

Функционал WebDriver позволяет использовать его не только для тестирования, но и для администрирования веб-сервисов, сократив до возможного предела количество действий, производимых вручную.

Еще один плюс Selenium заключен в том, что действия веб-драйвера видимы визуально и требуют минимального времени нахождения на странице, это позволяет с удобством демонстрировать функционал сайта, когда необходима презентация сервиса.

Некоторые недостатки Selenium WebDriver:

- поведение тестов может отличается в разных браузерах;
- возникновение сложностей с поиском элементов на страницах;
- необходимо четко продумывать архитектуру теста, часто использовать assert или ожидания, чтобы тест умел «думать», когда делать и когда нет.

1. Установка Selenuim

Привязка Selenium к Python предоставляет собой простой API для написания тестов функциональности с использованием веб-драйвера Selenium WebDriver.

Привязка Python-Selenium предоставляет удобный API для доступа к таким веб-драйверам Selenium как Firefox, IE, Chrome и других. На данный момент поддерживаются версии Python 2.7, 3.2, 3.3 и 3.4.

Для установки Selenium WebDriver необходимо создать проект в IDE с поддержкой языка программирования Python (например, PyCharm) и, используя систему управления пакетами pip, установить пакет selenium:

```
pip install selenium
```

Для использования Selenium WebDriver, необходимо его импортировать в проект:

```
from selenium import webdriver
```

Как отмечалось ранее, в Selenium реализованы драйверы для работы со многими браузерами. Для работы с конкретным браузером (например, с Google Chrome) необходимо воспользоваться следующей командой:

```
driver = webdriver.Chrome(),
```

где webdriver – компонент Selenium, включенный в проект на предыдущем шаге.

Таким образом, в переменной driver будет храниться драйвер того браузера, который был выбран в webdriver (в данном примере - Chrome).

На данном этапе установка и настройка Selenium WebDriver завершена.

2. Базовые функции в Selenuim

Подробная информация обо всех функциях, существующих в Selenium WebDriver на языке Python, представлена в официальной документации (www.selenium.dev/selenium/docs/api/py/index.html).

Рассмотрим некоторые базовые функции:

1. Функция get(url) – открытие страницы в выбранном браузере по ее пути url. Например, для открытия страницы по адресу «vk.com» в браузере Google Chrome используются следующие команды:

```
driver = webdriver.Chrome()
driver.get("https://www.vk.com")
```

2. Функция find_element(by, value) – поиск элемента на странице по HTML. Входными являются параметры: by – вид элемента, по которому будет происходить поиск элемента (ID, имя и т.п.), value - значение элемента. Для работы с этой функцией необходимо включить в скрипт следующий компонент:

```
from selenium.webdriver.common.by import By
```

Данный компонент необходим для задания параметра by в исходной функции.

Например, скрипт получения элемента со страницы vk.com с id, равным index_email выглядит следующим образом:

```
driver.get("https://www.vk.com")
element = driver.find_element(By.ID, 'index_email')
```

3. Функция send_keys(value) – занесение значений в элемент из HTML.

Например, получив текстовое поле с HTML и записав его в переменную element на прошлом шаге, занесем в него значение test@mail.ru следующей командой:

```
element.send_keys("test@mail.ru")
```

4. Функция save_screenshot(filename) – сохранение скриншота окна браузера с именем filename.

Например, для сохранения скриншота screen.png используется команда:

```
driver.save_screenshot("screen.png")
```

5. Функция close() – закрытие браузера.

Примечание. Рекомендуется использовать функцию, задерживающую выполнение скрипта sleep() из пакета time для корректных отображений выполняемых действий после каждой функции, вызываемой от driver.

Практическая часть.

I. Установка Selenium WebDriver.

II. Изучение базовых функций Selenium WebDriver.

1. Откройте страницу в браузере по пути https://github.com.
2. Сделайте скриншот страницы в браузере.
3. Найдите поле для ввода email на странице.
4. Занесите в него значение selenium@mail.ru.
5. Сделайте скриншот страницы в браузере с введенным email.

III. Завершение работы.

1. Завершите работу браузера.

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;
- скриншоты, сделанные в пункте II практической части на шагах 2 и 5, отражающий корректное занесение в поле ввода email;
- скрипт, написанный на языке Python;
- выводы.

Контрольные вопросы

1. Что такое Selenium WebDriver и для чего он используется?
2. Работу с какими браузерами поддерживает компонент webdriver?
3. По каким признакам можно получить элемент HTML-страницы в Selenium WebDriver?
4. Для чего стоит использовать функцию sleep() при тестировании? Приведите пример.
5. Приведите плюсы автоматизированного тестирования веб-приложений.
6. Опишите процесс установки (добавления в проект) Selenium WebDriver на языке Python.
7. Напишите скрипт для открытия страницы https://www.cyberforum.ru/auth.php и занесения своего адреса электронной почты в поле Email.