**Тема занятия №38: Парсинг данных. Библиотеки - beatifulsoup4 и selenium**

**1. Использование библиотеки selenium**

Поиск элемента по XPath

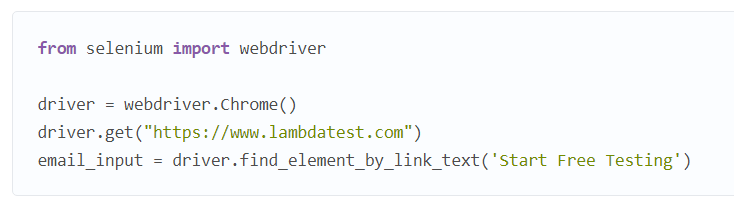
XPath использует путь для выбора узлов и определения местоположения необходимого веб-элемента. Метод find\_element\_by\_xpath() используется для поиска соответствующего элемента в документе с помощью XPath. Вы можете почитать руководство по XPath в Selenium, чтобы получить более глубокое представление об использовании XPath для поиска веб-элементов при автоматизации с Selenium.



Поиск элемента по тексту ссылки и частичному тексту ссылки

Этот метод ищет элемент по тексту ссылки (целому или части). При поиске частичного текста ссылки не ищется точное совпадение со строковым значением, поскольку ищется подмножество строк (в тексте ссылки).

Локатор текста ссылок в Selenium и частичного текста ссылок работает только со ссылками текущего веб-приложения.



Ниже показан поиск элемента на странице входа LambdaTest с помощью локатора частичного текста ссылок:



Различные методы поиска элементов

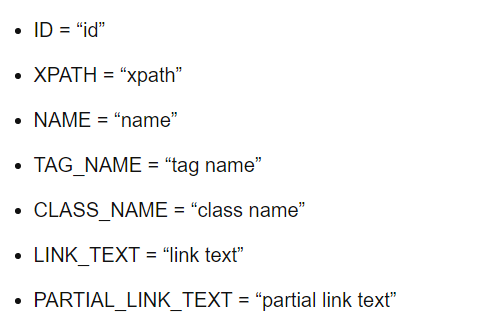
Есть два метода, которые могут быть полезны для поиска элементов страницы в сочетании с классом «By» для выбора атрибутов.

Обратите внимание, что нет никакой разницы между методами find\_element\_by\_tag и find\_element(By.tag). По умолчанию метод find\_element\_by\_tag вызывает find\_element(By.tag).

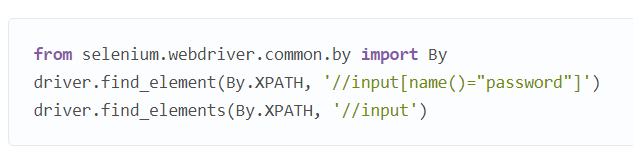
find\_element возвращает первый экземпляр из нескольких веб-элементов с определенным атрибутом в DOM. Метод вызывает исключение NoSuchElementException, если ни один элемент не соответствует требуемому локатору. Прочитайте статью о популярных исключениях Selenium, чтобы лучше понять, из-за чего они возникают.

find\_elements возвращает список всех экземпляров веб-элементов, соответствующих определенному атрибуту. Список будет пустым, если в DOM нет нужных элементов.

Ниже вы увидите список атрибутов, доступных для класса By:



Дальше показан пример использования метода find\_element, который использует локатор XPath для поиска нужного веб-элемента:

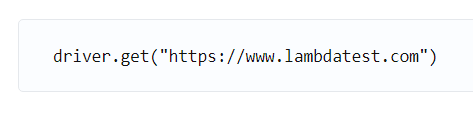


Открытие ссылки или документа

Перед выполнением любых операций с веб-элементами, присутствующими на странице, важно открыть целевой URL-адрес (или тестовый URL-адрес). Далее вы увидите несколько способов открыть URL-адрес в Selenium с Python:

driver.get(URL)

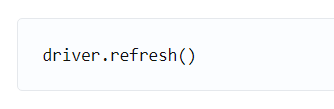
Метод driver.get() переходит на страницу, которая передается методу в параметре. Selenium WebDriver будет ждать, пока страница не загрузится полностью, после чего он запустит событие «onload» и вернет управление сценарию тестирования. В статье Selenium Waits in Python вы можете найти дополнительную информацию про обработку ожиданий в Selenium.



Обновление страницы

Бывают сценарии, в которых нужно обновлять содержимое страницы. Метод обновления страниц в Selenium WebDriver используется для обновления веб-страниц.

Метод driver.refresh() обновляет текущую веб-страницу. Он не принимает никаких аргументов и не возвращает никаких значений.



Ввод текста в веб-элемент

Метод send\_keys() в Python используется для ввода текста в текстовый элемент. Такой текст передается методу в качестве аргумента. Тот же метод можно использовать для имитации нажатия клавиш в любом поле (например, в полях ввода формы).

Пример использования send\_keys(), в котором адрес электронной почты передается текстовому элементу на странице регистрации LambdaTest:



Удаление текста в веб-элементе

Метод element.clear() в Selenium используется для удаления текста из полей, таких как поля ввода формы и т.д.

Пример использования метода очистки содержимого поля ввода электронной почты на домашней странице LambdaTest:



Нажатие на веб-элемент

Метод element.click() в Selenium используется для нажатия на элемент, такой как ссылка-якорь, кнопка и т.д.

Вот так с помощью метода click() можно нажать на кнопку на домашней странице LambdaTest:

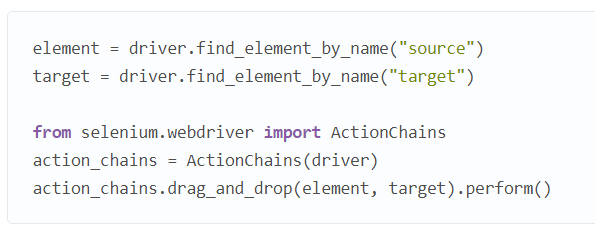


Перетаскивание веб-элемента

Перетаскивание объектов является одним из широко используемых сценариев в популярных приложениях (или программах), таких как Canvas, Google Drive, Trello, Asana и т.д. Метод drag\_and\_drop(элемент, цель) в Selenium помогает автоматизировать функционал перетаскивания веб-элементов из источника и переноса их в целевую область (или элемент).

У класса Actions в Selenium есть два метода, с помощью которых вы можете выполнять операции перетаскивания при тестировании кроссбраузерной совместимости. Обязательно ознакомьтесь с нашим подробным руководством, в котором есть информация о том, как выполнять перетаскивание в Selenium.

Вот простой пример использования метода drag\_and\_drop():



Получение кода страницы

Метод page\_source() в Selenium WebDriver используется для получения кода страницы. 