Hausaufgabe 2

Aufgabenstellung

Die *Webtesting AG* hat die UI der Websites ihrer Kunden bisher durch Werkstudenten manuell testen lassen. Leider stehen diese nicht mehr zur Verfügung. Es wurde daher beschlossen, dass UI-Testing zu automatisieren. Im Team wurde sich für Selenium entschieden und Sie wurden mit der Aufgabe betraut, einen ersten Testcase für die Website https://sqlify.io/ zu erstellen.
Die Website https://sqlify.io/ ermöglicht es Usern, kleinere CSV oder JSON Files zu konvertieren. Es soll folgende Journey getestet:

1. Ein User fügt ein Dokument in das dafür vorgesehene Feld ein. Folgendes muss nach diesen Aktionen stimmen:
    a. Text der Datei wird in dem Feld angezeigt.
2. Ein User klickt auf den „Upload“ Button. Folgendes muss nach dieser Aktion passieren:
    a. Weiterleitung auf die nächste Seite (User kann Felder der csv-Datei auswählen) ohne Fehlermeldungen.
3. Ein User nennt die Spaltennamen von “abc_<n>“ in “column_<n>“ um. Anschließend klickt der User auf den „Export data“ Button. Folgendes muss nach dieser Aktion passieren:
    a. Popupfenster ist sichtbar
4. User klickt in dem Popup-Fenster auf „Export file“ (JSON Format bleibt ausgewählt). Folgendes muss nach dieser Aktion passieren:
    a. Popup-Fenster enthält den Schriftzug „Export completed“.
    b. Download Button ist klickbar und die Datei befindet sich im Ordner Downloads

Eine Mitarbeiterin in Ihrem Team merkt an, dass Selenium auch im Rahmen von Integration Testing von Webanwendungen genutzt werden kann. Überprüfen Sie daher nach der Journey mittels eines Skripts, ob die Konvertierung auch funktioniert hat:

- Ist der Inhalt der Datei gleichgeblieben?
- Wurden die Namen der Spalten erfolgreich geändert und die letzte Spalte gelöscht?

Schreiben Sie mit dem Selenium-Framework für diese Journey einen UI-Test. Verwenden Sie dabei das Design Pattern *Page Object*, welches bei dem Testen vom UI sehr häufig zur Anwendung kommt. Führen sie die Journey mit beiden Dokumenten separat aus!
Die Dokumente, welche hochgeladen werden sollen, finden Sie im Ordner mit dem Python Framework („selenium_framework/website_testing/selenium/assets/“) und in der übergeordneten Struktur sie heißen document_1 und document_2.

Setup

1. Entscheiden Sie sich für eine Programmiersprache, in der Sie die Selenium Tests schreiben wollen (Selenium Language Bindings)
2. Laden Sie sich den Driver für den Google Chrome - Browser herunter. Dieser führt dann gemäß der Selenium-Tests die Interaktion mit dem Broser automatisiert aus.
3. Schauen Sie sich die Dokumentation zu Selenium an:
    a. Python: https://selenium-python.readthedocs.io/

   b. Python: https://selenium.dev/documentation/en/
   c. Java: https://selenium.dev/documentation/en/
4. Setzen Sie sich mit dem Design Pattern *Page Object* auseinander
   a. Python: https://selenium-python.readthedocs.io/page-objects.html
   b. Java: https://selenium.dev/documentation/en/guidelines_and_recommendations/page_object_models/

Hinweise

- Wie inspiziert man eine Website bezüglich deren Elemente?
  - Rechts-klick auf eine Element (Button, Symbol, Text) und Option „Inspect“ auswählen
  - Es erscheint der HTML DOM, durch den navigiert werden kann. Dabei wird das jeweilige Element auf der Website visuell hervorgehoben
- Wie wählt man in Selenium Elemente mittels CSS Klasse aus?
  - https://www.seleniumeasy.com/selenium-tutorials/css-selectors-tutorial-for-selenium-with-examples
- Falls Sie die Python Vorlage verwenden:
  - Fügen Sie die neuen Page Objects der Klasse *SeleniumTestCase* als Attribute hinzu und lassen Sie das Object in der Methode *setUpClass*() initialisieren. Dann können Sie das Objekt mit seinen Methoden in der Testklasse verwenden (siehe example.py)
- Wie man Dokumente mittels Selenium hochläd:
  - http://allselenium.info/file-upload-using-python-selenium-webdriver/ (Achtung, hier wird nicht das Page Object Pattern verwendet)