

Evernote Test Automation Detailed Documentation

Project Overview

This project is an automation framework developed using Selenium and pytest for testing Evernote's note-taking features. It automates user login, note creation, and validation of various formatting features such as headers, bullet points, bold, and italic text. The framework includes a set of Python scripts to perform these automated tests, focusing on web-based interactions using an undetected Chromedriver setup to avoid detection.

Folder Structure

- **pages/**
This folder contains the Page Object Model (POM) implementations:
 - **base_page.py**: A base class that contains common methods such as element interactions and navigation between pages (base_page).
 - **login_page.py**: Handles login-related actions such as entering credentials and verifying login success (login_page).
 - **notes_page.py**: Manages interactions related to note creation and viewing (notes_page).
 - **editor_page.py**: Controls note editing actions such as formatting and writing (editor_page).
 - **tests/**
Contains the test scripts written with pytest:
 - **test_login_create_note.py**: Automates the login process, note creation and correct formatting(test_login_create_note).
 - **test_login_find_note.py**: Automates the login process, verifies that a previously created note with the correct formatting is saved (test_login_find_note).
 - **Other Files**
 - **requirements.txt**: Lists all Python package dependencies for the project(requirements).
 - **.gitignore**: Specifies files and directories that should not be tracked by Git.
-

Setup Instructions

Follow the steps on the readme.md file in the GitHub repository.

Configuration

The config.yaml (on the root) file contains important settings for the tests, including:

- Evernote URLs
- User Credentials (with email & password)
- User manual open mode

If `use_manual_open` is set to True, the credentials should be inputted each time; if set to False, they can be stored in config.yaml.

Automation Flow

1. **Login Flow:**
 - The user is logged in via `LoginPage` by entering their email and password(login_page).
 2. **Note Creation:**
 - The `NotesPage` handles the creation of a new note(notes_page).
 - The note is formatted with headers, bullet points, bold, and italic text via the `EditorPage`(editor_page).
 3. **Note Validation:**
 - The tests validate that the note persists and the format is as expected(test_login_find_note).
-