Standard Operating Procedure (SOP) for Running Stefan Automation

**Table of Contents**

1. Introduction
2. Prerequisites
3. Environment Setup
4. Modules and Functions Description
5. Step-by-Step Execution

### 1. Introduction

This SOP outlines the steps to run the Midway Automation script. The script automates various web interactions such as login, cookie acceptance, page navigation, and file handling for the Midway platform.

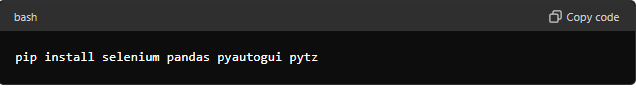
### 2. Prerequisites

Before running the script, ensure you have the following:

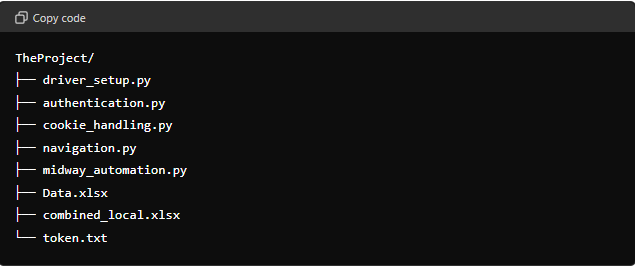
* Python 3.x installed on your system.
* Required Python packages installed.
* Appropriate WebDriver for your browser installed.
* Input data in the specified Excel file (Data.xlsx).
* Correct directory structure.

### 3. Environment Setup

1. **Install Python Packages:** Ensure you have the necessary Python packages installed. You can install them using pip:



1. **Download WebDriver:** Download the appropriate WebDriver for your browser (e.g., ChromeDriver for Chrome) and ensure it is accessible from your PATH.
2. **Directory Structure:** Ensure your project directory is structured as follows:



### 4. Modules and Functions Description

#### 4.1. midway\_automation.py

Contains the main class MidwayAutomation that orchestrates the automation.

* **Class: MidwayAutomation**
  + \_\_init\_\_(self, download\_dir, username, password): Initializes the WebDriver and user credentials.
  + login(self): Logs into the Midway portal.
  + accept\_cookies(self): Accepts cookies on the Midway portal.
  + navigate\_to\_page(self, url, forecasting\_group, email): Navigates to a specific page and executes custom functions.
  + close\_driver(self): Closes the WebDriver.
  + run\_python\_script(self, script\_path): Runs an external Python script.

#### 4.2. authentication.py

Handles user login.

* **Function: login\_to\_midway(driver, username, password)**: Automates the login process.

#### 4.3. cookie\_handling.py

Handles cookie acceptance.

* **Function: accept\_cookies(driver)**: Handles cookie consent if prompted.

#### 4.4. navigation.py

Handles page navigation and custom interactions.

* **Function: navigate\_to\_page(driver, url, custom\_function=None)**: Navigates to a given URL and executes a custom function.
* **Function: custom\_function(driver, forecasting\_group, email)**: Performs custom actions on the page.
* **Function: rename\_downloaded\_file(download\_dir, new\_file\_prefix="Forecasting")**: Renames the most recently downloaded file.
* **Function: run\_python\_script(script\_path)**: Runs a Python script.

### 5. Step-by-Step Execution

#### Step 1: Initialize and Setup

1. **Prepare Input Data:**
   * Ensure Data.xlsx contains the necessary input data (Username, Password, Forecasting Group, Email).
2. **Prepare the token.txt**
   * Ensure that the token is new because have a 15 minutes valability.
3. **Prepare the ASIN list** 
   * Copy-paste ASIN in Data.txt

### Troubleshooting and Tips

1. **Common Issues:**
   * Ensure the WebDriver is compatible with the browser version.
   * Verify that element locators (e.g., XPaths) are up-to-date.
2. **Debugging:**
   * Print statements are added to log progress and errors.
   * Check console output for any exceptions or errors.