Toro Scrape

`toro_scrape.py` is a web scraping script designed to automate the process of logging into https://identity.toro.com/as/authorization.oauth2?response_type=code&client_id=InsiteCommerceClient&redirect_uri=https%3A%2F%2Fshop.thetorocompany.com%2Fidentity%2Fexternalcallbackextension&scope=openid%20profile%20email%20address, extracting product details, and saving the information into a CSV file. Additionally, the script has the option to upload the output CSV file to an FTP server once scraping is complete.

Features

- Automates the login process using provided credentials.
- Scrapes product information such as brand, item code, description, retail price, quantity on hand, and more.
- Allows FTP upload of the output file, with options to overwrite existing files or append a timestamp to the filename.
- Supports a headless browser mode for silent operation without opening a window.
- Handles product inventory and price data extraction.

Requirements

Minimum System Requirements

- **Operating System**: Windows 10/11 or modern Linux distributions (Ubuntu 20.04+, Fedora 32+, CentOS 8+, etc)
- **Python Version**: v3.10 or 3.11 recommended (3.8+ minimum)

Dependencies

You can install the necessary dependencies via the provided `requirements.txt`. Run the following command to install them:

`pip install -r requirements.txt`

`python -m playwright install chromium`

On Linux: `python -m playwright install-deps`

The `requirements.txt` includes the following libraries:

- `pandas`: For creating and formatting CSV
- `playwright`: For automating web browser interactions.
- `requests`: For making HTTP requests.

Running the packaged EXE (no Python required)

Zip file includes the following

- ToroScraper.exe

- browses/(Playwright browsers; e.g., chromium_headless_shell-XXXX/...)
- config.txt
- SolidCommerceProducts.csv

Configuration (`config.txt`)

The configuration file (`config.txt`) contains the necessary parameters for the script. Here is a breakdown of the configuration options:

```
{
    "login_url": "https://identity.toro.com/as/authorization.oauth2?response_type=code&client_id=InsiteCommerceClient&redirect_uri=https%3A%2F%2Fshop.thetorocompany.com%2Fidentity%2Fexternalcallbackextension&scope=openid%20profile%20email%20address",
    "username": "your_username",
    "password": "your_password",
    "headless_mode": false,
    "max_rows": "all",
    "input_file": "input.csv",
    "output_file": "output.csv",
    "ftp_host": "None",
    "ftp_port": 21,
    "ftp_username": "ftp_user",
    "ftp_password": "ftp_password",
    "ftp_directory": "/path/to/directory",
    "overwrite_existing": true,
    "rsv_qty": 1
}
```

Parameters:

- `login_url`: The URL for logging into the website.
- `username`: Your username for logging into the website.
- `password`: Your password for logging into the website.
- `headless_mode`: Boolean (true or false), if true, the browser will operate in headless mode (no GUI).
- `max_rows`: Set to "all" for scraping all rows or an integer to limit the number of rows to scrape.
- `input_file`: The filename of the input CSV file containing the products to be scraped.
- `output_file`: The filename where the output file will be saved (default is output.csv)
- `ftp_host`: The FTP server hostname or IP address. If set to empty string "", the file is saved locally.
- `ftp_port`: The FTP server port (default is 21).
- `ftp_username`: Your FTP username.
- `ftp_password`: Your FTP password.
- `ftp_directory`: The directory on the FTP server where the file will be uploaded.
- `overwrite_existing`: Boolean (true or false), determines whether to overwrite an existing file on the FTP server. If false and output_file exists, a timestamp is appended.
- `rsv_qty`: The default reserved quantity, set to 1 by default.

Running the Script

From source

- To run the script, execute the following command in the terminal:

```
python -m venv .venv
.venv\Scripts\activate
pip install -r requirements.txt
python -m playwright install chromium
# Linux only:
# python -m playwright install-deps
python toro_scrape.py --config config.txt
```

From EXE

- Double-click ToroScraper.exe
- If running from command line: `ToroScraper.exe` or `Toroscraper.exe --config "C:\path\config.txt"`

Ensure that:

- The configuration file (`config.txt`) is set up with the correct parameters.
- The input CSV file exists, and the FTP server information is correctly configured if you intend to upload the file to an FTP server.
- If login fails or Playwright errors mention missing executable, ensure browsers/ folder exists and matches the expected name (don't rename)
- config.txt path should remain in same directory as ToroScraper.exe file.

FTP Upload:

After scraping is complete, the script will upload the output CSV file to the specified FTP server. If the `overwrite_existing` flag is set to `false`, a timestamp will be appended to the file name before uploading.

Input CSV

The script looks for SolidCommerceProducts.csv by default which has the fields: `LDSKU,Product Name,Date,Weight,UPC,Manufacturer,SKU,Model Number,ReleaseDate,CreateDate,MSRP,Qty Alternate Images,Product Image,KitInfo,HSCode,ScheduleBCode,HSDescription,CaptureSerialNumber,PackageInsuranceRequired,SignatureRequired,PackagingPreferences,SpecialProduct`

- The script requires a column named “SKU” and extracts product numbers from values that start with “TOR~” (it takes the text after ‘TOR~’ up to the next ‘~’ or end of string). -- Sample: SKU: TOR41-6820SOMETHING → Product Number: 41-6820
- Non-Toro SKUs are ignored, empty or invalid SKUs are filtered out.

Output Fields

The output file will contain the following fields, each representing a specific piece of product information:

```
product_number,product_id,material_id,item_status,unit_list_price,unit_regular_price,unit_net_price,actual_price,is_on_sale,unit_of_measure,distribution_centre,division,category_group,order_group,qty_on_hand,availability_message,available_date,short_description,erp_number,erp_description,large_image_url,shipping_length,shipping_width,shipping_height,shipping_weight,unit_of_measure_description,is_active,is_discontinued,can_back_order,track_inventory,minimum_order_qty,multiple_sale_qty,sku,upc_code,model_number,brand,product_line,tax_code1,tax_code2,tax_category,product_detail_url,is_special_order,is_gift_card,is_subscription,can_add_to_cart,can_add_to_wishlist,can_show_price,can_show_unit_of_measure,can_enter_quantity,requires_real_time_inventory,availability_message_type,meta_description,meta_keywords,page_title
```

- Note that some values may be empty depending on the product/API response.

Troubleshooting

Common Issues:

- FTP connection errors: Check that the FTP credentials and server path in `config.txt` are correct. Ensure that the FTP server is accessible.
- Config file option headless_mode true runs without showing a window; set false to debug login flows.

License

This script is provided as-is. Use it at your own risk. The author is not responsible for any damage or issues caused by this script.

Contact Information

For any questions, issues, or feedback regarding this script, please reach out:

- **Author**: Jim Tyranski
- **Email**: jim@tyranski.com

Please ensure to provide detailed information about the issue you're experiencing, including any relevant error messages and the configuration details used when running the script.

Changelog

0.1.0 - Initial Release

0.2.0 - Added configuration options

- Added FTP function
- Added Playwright browser location info for script