

# 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](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. (Not implemented yet)
- 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:** 3.7+

### Dependencies

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

```
pip install -r requirements.txt
playwright install
```

The `requirements.txt` includes the following libraries:

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

### 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%2
0email%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, but can be .csv or .xlsx)
- **ftp\_host**: The FTP server hostname or IP address. If set to "None", 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.
- **rsv\_qty**: The default reserved quantity, set to 5 by default.

### Running the Script

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

```
python toro_scrape.py
```

or

```
python3 toro_scrape.py
```

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.

### FTP Upload (Not implemented yet):

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.

## Output Fields

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

-- TBD --

## 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.

## 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](mailto: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