Toro Scrape

Version 0.3.0

`toro_scrape.py` automates logging into Toro Identity/Shop, extracts product details and pricing via authenticated API calls, and saves results to CSV. The script can optionally upload the resulting CSV to an FTP server. It now supports robust retries, configurable logging, graceful interruption with partial saves, and concurrent scraping for speed.

What's new in 0.3.0

- Graceful Ctrl+C handling with partial save and resume on next run.
- Config-driven logging level and optional log file.
- Centralized HTTP retry/backoff for 429/5xx and transient network errors.
- Incremental partial saving at a configured interval and automatic resuming.
- Threaded scraping with configurable concurrency.
- De-duplication of results by product_number.

Features

- Automates login using Playwright.
- Scrapes extensive product and pricing fields, including inventory and metadata.
- Robust HTTP calls with retry/backoff.
- Concurrent scraping with bounded thread pool.
- Incremental partial save, resume, and deduplication.
- Optional FTP upload of the final CSV.
- Headless mode support.

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

Install dependencies:

```
pip install -r requirements.txt
python -m playwright install chromium
# Linux only:
# python -m playwright install-deps
```

Included libraries:

- `pandas`: CSV creation and manipulation
- `playwright`: Browser automation
- `requests`: HTTP requests

Running the packaged EXE (no Python required)

Zip includes:

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

Configuration (`config.txt`)

Example config (JSON):

```
{
    "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",
    "overwrite_existing": true,
    "rsv_qty": 1,
    "ftp_host": "",
    "ftp_port": 21,
    "ftp_username": "",
    "ftp_password": "",
    "ftp_directory": "/path/to/directory",
    "log_level": "INFO",
    "log_file": "logs/toro_scrape.log",
    "save_interval": 0,
    "concurrency": 6
}
```

Parameters:

- `login_url`: The login URL for Toro Identity.
- `username`, `password`: Credentials for login.
- `headless_mode`: true for headless browser, false to show browser window.
- `max_rows`: "all" for all rows or an integer to limit the number processed.
- `input_file`: CSV containing products to scrape.
- `output_file`: Final CSV output filename.
- `overwrite_existing`: If false and output_file exists, a timestamp is appended on save.

- `rsv_qty`: Quantity used when requesting pricing.
- FTP settings (optional):
  - ftp_host, ftp_port, ftp_username, ftp_password, ftp_directory
  - Leave ftp_host empty to skip FTP upload.
- Logging:
  - log_level: DEBUG, INFO, WARNING, ERROR, CRITICAL (default INFO).
  - log_file: Optional path to write logs to a file in addition to console.
- Partial save and concurrency:
  - save_interval: 0 disables periodic saves; N > 0 saves every N processed products to output_file.partial (used for resuming).
  - concurrency: Number of worker threads for parallel product processing.

Running the Script

From source

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

```
python -m venv .venv
. .venv/Scripts/activate    # Windows: .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
- Or from command line:
  - ToroScraper.exe
  - ToroScraper.exe --config "C:\path\config.txt"

Ensure:

- config.txt is correctly set up and located alongside the EXE unless you pass an absolute path.
- Input CSV exists.
- If Playwright complains about missing executable, ensure the browsers/ folder exists, is not renamed, and uses the expected Playwright structure.

Input CSV

By default, the project uses SolidCommerceProducts.csv format with fields like: 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

- Requires a column named “SKU”.
- Extracts product numbers from SKUs that start with “TOR~”:
    - Takes text after “TOR~” up to next “~” or end of string.
    - Example: SKU: TOR41-6820SOMETHING → Product Number: 41-6820
- Non-Toro SKUs are ignored; empty/invalid SKUs are filtered out.

Output Fields

CSV columns include:
`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`

Notes: - Some fields may be empty depending on API responses. - Results are deduplicated by product_number before saving.

Partial Save and Resume

- If save_interval > 0, the script periodically writes a partial CSV at output_file.partial and logs a message indicating how many records have been saved.
- On the next run, if a partial file exists, the script:
    - Loads it to continue where it left off.
    - Skips product_numbers already present in the partial file.
- If an output_file already exists, the script also attempts to skip product_numbers previously completed to avoid duplicates.
- On normal completion, the final CSV is saved to output_file and the .partial file is deleted.

Concurrency

- The script uses a ThreadPoolExecutor to process multiple product_numbers concurrently.
- Configure via config.concurrency or override using --concurrency CLI flag.
- The script ensures thread-safe accumulation and skips already-scraped product_numbers.

Graceful Interruption

- Press Ctrl+C to stop gracefully.
- The script will:
    - Avoid starting new tasks.
    - Save partial results if save_interval > 0.
    - Exit with a message indicating interruption.

Logging

- Logging level and optional file are configured via log_level and log_file.
- Examples:
  - log_level: "DEBUG" for detailed troubleshooting.
  - log_file: "logs/toro_scrape.log" to write logs to disk (directory auto-created if needed).

FTP Upload

- If ftp_host, ftp_username, and ftp_password are provided, the script uploads the final CSV to the specified FTP directory.
- If overwrite_existing is false and output_file exists, a timestamp is appended to the filename prior to saving/uploading.
- FTP directory trees are created on the server if they don’t exist.

Troubleshooting

- Login or Playwright errors:
  - Ensure browsers/ exists and hasn’t been renamed.
  - Set headless_mode to false to watch the login flow.
- Rate limiting or server errors:
  - The script automatically retries with exponential backoff for 429 and 5xx responses.
- Partial resume issues:
  - Delete the .partial file if you want a clean run.
  - Ensure save_interval is a positive integer to enable partial saves.
- Duplicate rows:
  - The script deduplicates by product_number before both partial and final save.

License

This script is provided as-is. Use at your own risk.

Contact Information

- **Author**: Jim Tyranski
- **Email**: jim@tyranski.com Please include config details and error messages for faster support.

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 FTP function; Playwright browser location notes
- 0.3.0
  - Added graceful Ctrl+C handling and partial save on interrupt
  - Introduced config-based logging (log_level, log_file)
  - Implemented centralized HTTP retries/backoff
  - Added save_interval partial saves and automatic resume
  - Introduced configurable concurrency with threading and CLI override
  - Improved deduplication and robustness during final save