# TGA Product Data Scraping Project

## Overview

This project is designed to scrape and extract product data from the Therapeutic Goods Administration (TGA) website. It focuses on gathering information about medicines, particularly those in short supply or unavailable, along with their approval status and related details.

## Project Structure

```

├── data\_extractor\_test.py # Main scraping script with enhanced error handling and screenshots

├── data\_extracter.py # Original data extraction script

├── link\_fetcher.py # Script for fetching product URLs

├── medicine\_links.txt # Storage for medicine URLs

├── tga\_product\_urls.csv # Input file containing URLs to scrape

├── tga\_product\_data.csv # Output file containing scraped product data

├── new\_tga\_product\_data.csv # Additional output file for new data```

## Features

### 1. Robust Data Extraction

- Extracts multiple fields from each product page:

- Section 19A approved medicine

- Section 19A approval holder

- Phone

- Approved until

- Status

- Medicines in short supply/unavailable

- Indication(s)

### 2. Error Handling & Debugging

- Comprehensive error handling for various scenarios

- Automatic screenshot capture on errors

- Detailed logging with emojis for better visibility

- Timeouts and retries for reliability

### 3. Data Processing

- CSV input/output handling

- Clean data formatting

- Proper UTF-8 encoding support

- Flexible URL processing (all or subset)

## Dependencies

- Selenium WebDriver

- Chrome Browser

- Python libraries:

- selenium

- webdriver\_manager

- csv

- time

- os

## Setup & Installation

1. Install Python dependencies:

```bash

pip install selenium webdriver-manager

```

1. Ensure Chrome browser is installed on your system

1. Verify the input file structure:

- Create/prepare `tga\_product\_urls.csv` with URLs to scrape via link\_fetcher.py

## Usage

### Running the Scraper

The script will:

1. Read URLs from `tga\_product\_urls.csv`

2. Process each URL and extract product data

3. Save results to `tga\_product\_data.csv`

### Configuration Options

- `INPUT\_CSV\_FILENAME`: Input file with URLs (default: "tga\_product\_urls.csv")

- `OUTPUT\_CSV\_FILENAME`: Output file for scraped data (default: "tga\_product\_data.csv")

- `FIELDS\_TO\_EXTRACT`: List of fields to extract from each page

### Browser Options

The script uses Chrome in visible mode by default. For headless operation, uncomment:

```python

options.add\_argument("--headless")

```

## Error Handling

The script handles various error scenarios:

- Page load timeouts

- Missing elements

- Network issues

- Invalid URLs

- CSV read/write errors

All errors are:

1. Logged to console

2. Captured in screenshots (when possible)

3. Recorded in the output CSV with appropriate error messages

## Output Format

The output CSV contains:

- Product Page URL

- All extracted fields

- Error indicators where applicable

## Debugging

- Review console output for detailed error messages

- Examine CSV output for specific field failures

## Best Practices

1. Start with a small subset of URLs for testing

2. Monitor the error screenshots directory

3. Adjust timeouts if needed for slow pages

4. Regular backups of output data

## Troubleshooting

Common issues and solutions:

1. WebDriver errors

- Update Chrome browser

- Reinstall webdriver-manager

2. Timeout errors

- Increase wait times

- Check network connection

3. CSV errors

- Verify file permissions

- Check file encoding