Introduction

This document builds upon the **T1 2025 Coles Scraper Documentation**, which covers the core implementation of the scraper, including API patterns, proxy rotation, stealth techniques, and the overall data extraction workflow.

Overview

This trimester, we extended the Coles.com.au scraper by adding two features:

1. **Timestamp Capture** – each product record now stores an ISO-formatted timestamp (datetime.now().isoformat()). This enables more reliable temporal analysis of price changes and product availability across runs.
2. **Automated hCaptcha Interaction with PyAutoGUI** – an experimental approach was introduced using PyAutoGUI to simulate human mouse movements and clicks on the hCaptcha "I am not a robot" checkbox. This reduces manual effort in bypassing Incapsula's CAPTCHA challenges.

The scraper continues to use Coles' JSON API endpoints for structured data retrieval, while retaining anti-bot measures such as rotating SmartProxy IPs, user-agent spoofing, Selenium Stealth, and randomized delays. Data is stored in MongoDB as before, with filenames and collections tagged by supermarket, location, and timestamp for traceability.

New Additions in T2 2025

1. Timestamp Capture

Every product record now includes a timestamp field:

"timestamp": datetime.now().isoformat()

This ensures that analysts can track when exactly a dataset was scraped. For example:

{

"product_code": "12345",

"item_name": "Coles Milk 2L",

"best_price": 2.50,

"timestamp": "2025-09-24T12:35:22.120394"

}

The addition of timestamp metadata is important for monitoring dynamic pricing strategies and validating freshness of scraped data.

---

2. Automated hCaptcha Checkbox Clicking

Previously, CAPTCHA solving was entirely manual. We added a function using **PyAutoGUI** to click the hCaptcha checkbox at fixed screen coordinates:

```
def click_hcaptcha_checkbox_locked():
    pyautogui.moveTo(785, 505, duration=0.4)
    pyautogui.click()
    print("Clicked hCaptcha checkbox")
    time.sleep(10)
```

We launch fixed size screen to support the function above:

```
driver.set_window_rect(x=0, y=0, width=1536, height=816)
```

This is integrated into the wait_for_captcha function. The workflow is:

- Detect Incapsula hCaptcha challenge.
- Move mouse pointer and click checkbox automatically.
- Wait for iframe to disappear.
- If unsuccessful after retries, rotate proxy/IP and reattempt.

Note: This is an experimental feature. The click coordinates are hardcoded, which may vary by system resolution and browser window. It reduces manual workload but is not yet a fully reliable CAPTCHA bypass. You can build on it to implement AI/ML libraries for CAPTCHA solving.

Workflow Summary

- **Initialization:** A fresh Chrome driver is launched with stealth settings and rotating SmartProxy credentials.
- **Build ID Extraction:** Build ID is parsed dynamically from __NEXT_DATA__.
- **Category Discovery:** The Browse API is used to extract valid categories for the given store.
- **Product Data Extraction:** Each category and paginated results are fetched via Coles' API endpoints.
- **Data Parsing:** Product metadata (ID, name, prices, promos, links, and timestamp) is extracted.
- **Storage:** Data is written both as JSON files and into MongoDB collections named after the run timestamp.

Example of Enhanced Product Schema


```
{
  "product_code": "98765",
  "category": "Bakery",
  "item_name": "White Sandwich Bread",
  "best_price": 1.80,
  "item_price": 2.00,
  "unit_price": 0.90,
  "special_text": "On Sale",
  "promo_text": "Save 20%",
  "link": "https://www.coles.com.au/product/98765",
  "timestamp": "2025-09-24T12:35:22.120394",
  "category_slug": "bakery"
```

}

Current Limitations

- The **PyAutoGUI hCaptcha interaction** works only on systems with a graphical environment and fixed window dimensions. If browser size or resolution changes, the click coordinates must be adjusted.
- CAPTCHA challenges still appear frequently due to Coles' anti-bot system and low-reputation datacenter proxies. Auto-clicking only partially addresses this.
- Long runs may require 3–4 CAPTCHA challenges to be solved manually or semi-automatically.

Note: If we use high quality residential proxy IPs instead of cheap datacenter proxies to rotate IPs, then no CAPTCHAs are encountered. This was tested in T1. However, this incurs higher costs.

Future Work

- Enhance CAPTCHA handling with dynamic checkbox detection instead of fixed coordinates.
- Integrate AI/ML-based CAPTCHA solvers (if available to hCAPTCHA support) for better automation.
- Improve error handling and logging for CAPTCHA retries.
- Explore residential proxy usage (if budget permits) to reduce CAPTCHA frequency.
- Continue modularizing the scraper for easier testing and debugging.
- Explore how we can integrate this into our automation pipeline.