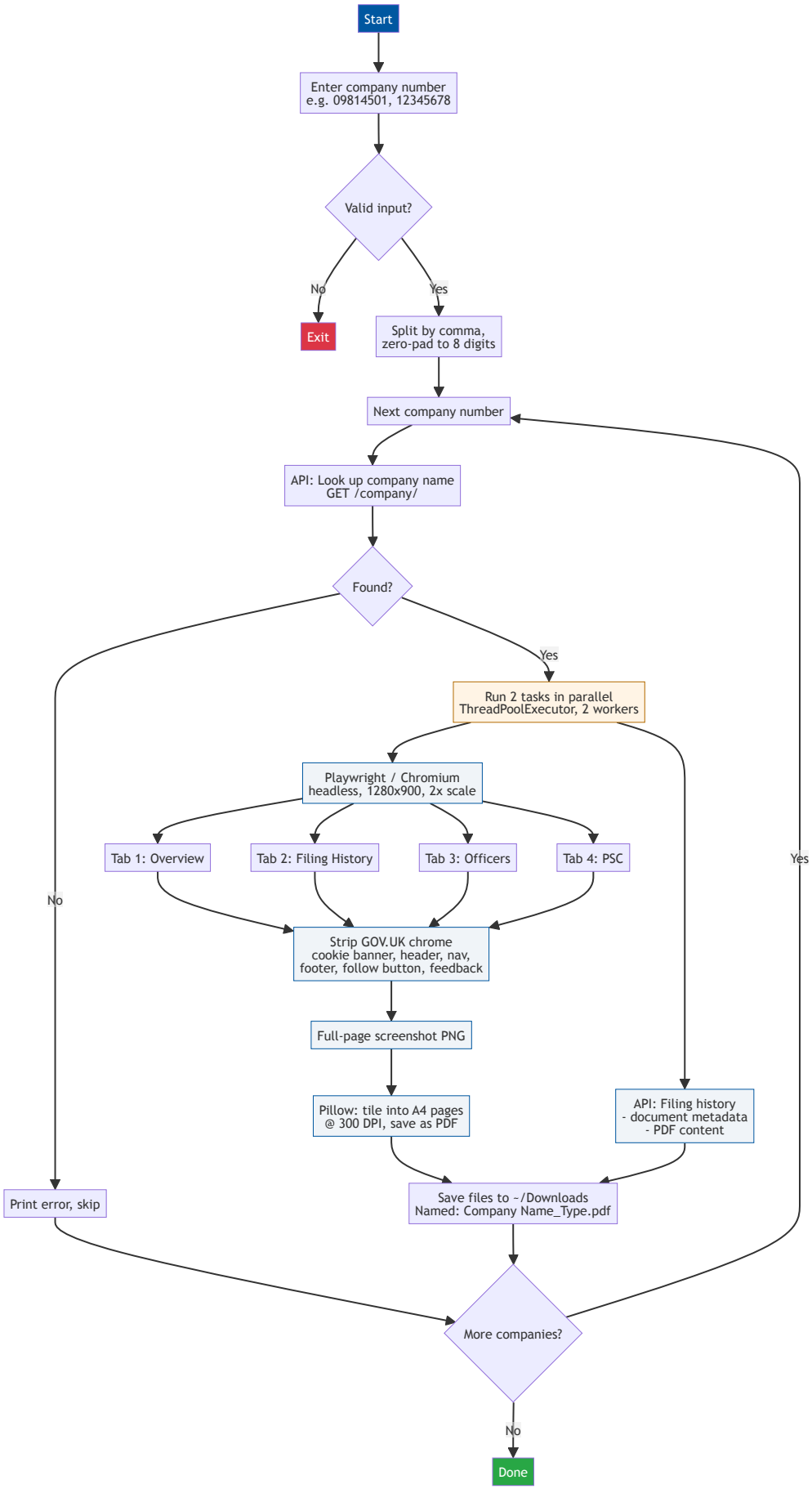
Companies House Export Tool

A command-line tool that takes one or more UK company numbers and produces a consistent pack of 5 PDFs per company, saved to `~/Downloads`.

How It Works



Timing: Tool vs Manual

Per company, end-to-end:

Step	Tool	Manual (via browser)
Search + open company		~10 s
Overview page		~10 s
Filing History page		~10 s
Officers page		~10 s
PSC page		~10 s
Latest Confirmation Statement		~20 s
Total	**~7–8 s** (measured)	**~1¼ min**

Tool time is an actual measured run (`time` on company `09814501` = **7.68 s real** end-to-end — lookup + all 4 pages in parallel tabs + Confirmation Statement download + file writes).

Roughly **9–10× faster** than doing it by hand, and you get identical, consistently-named files every time — no typos, no inconsistent separators, no "Confirmation Statement (1).pdf" duplicates.

For a batch (e.g. 10 companies), the tool runs each company sequentially with the same ~8 s budget per company, so ~1½ minutes total vs ~12–13 minutes manually.

Output quality

It's not just about speed — the manual approach also produces worse-looking PDFs:

	Tool	Manual (browser Print → Save as PDF)
GOV.UK header, cookie banner, sign-in nav, footer	Stripped before capture	Still on every page
URL, date, page numbers in the margins	None	Added by the print dialog
Column/table layout	Preserved — single full-width screenshot tiled top-to-bottom	Often reflowed, squeezed, or cut mid-row
Long content (e.g. big filing history)	Continuous, like scrolling the page	Split at unpredictable points, extra whitespace at page breaks
Filename	`Company Name_Type.pdf`, consistent every time	Whatever the browser suggests — usually the URL slug

Because the tool takes a full-page screenshot at 2× device-scale, strips the GOV.UK chrome with a JS selector list, then tiles the image onto A4 at 300 DPI, the result reads like a clean printout of *just the company data*. The browser's Print dialog, by contrast, re-paginates the live web page and bakes in the navigation and print headers.

Output Files

For each company, 5 PDFs are saved to `~/Downloads`:

File	Source	Description
`Company Name_Overview.pdf`	Website screenshot	Company profile, status, address, SIC codes
`Company Name_Filing History.pdf`	Website screenshot	Full filing history page
`Company Name_Officers.pdf`	Website screenshot	Active and resigned officers
`Company Name_PSC.pdf`	Website screenshot	Persons with significant control
`Company Name_Confirmation Statement YYYY-MM-DD.pdf`	API download	The actual filed Confirmation Statement PDF

File-naming rules (`safe_filename` + `title_case_name`):

- Company name is title-cased, with abbreviations kept uppercase: `LTD`, `PLC`, `LLP`, `LLC`, `LP`, `UK`, `US`, `EU`, `NV`, `SA`, `AG`, and dotted forms like `B.V.`, `N.V.`
- Characters illegal in filenames (`< > : " / \ | ? *`) are replaced with `_`.

How Each Screenshot PDF Is Built

1. **Page load** — `page.goto(url, wait_until="domcontentloaded", timeout=30s)`.
2. **Best-effort settle** — `wait_for_load_state("networkidle", timeout=5s)`; failures ignored so a never-idle page doesn't stall.
3. **DOM cleanup** — a `page.evaluate()` script removes: cookie banner, GOV.UK header, service nav, disclaimer, sign-in nav, search bar, "Follow this company" button, pagination, feedback widget, skip-links, footer.
4. **Full-page screenshot** — PNG at 1280 × full-height, 2× device scale factor.
5. **Pillow → A4 PDF** — the PNG is resized to fit A4 width (2480 px @ 300 DPI, 80 px margins) and tiled top-to-bottom into however many A4 pages are needed. Saved as a multi-page PDF.

The Pillow step for each page runs in its own thread (`asyncio.to_thread`) so one page's image processing never blocks the others.

Concurrency Model

Two layers of parallelism run per company:

1. **Outer (`ThreadPoolExecutor`, 2 workers)**: the 4 website-page exports and the Confirmation-Statement API download run as two independent tasks at the same time. They share no state.
2. **Inner (`asyncio.gather` on 4 tabs)**: inside the page-export task, one Chromium browser with one shared context opens 4 tabs and loads/screenshots all 4 pages in parallel via the async Playwright API.

One browser, one context, four tabs — so viewport and device-scale-factor settings are defined once.

Confirmation Statement Download

Uses the Companies House public API (`api.company-information.service.gov.uk`) with Basic Auth:

1. `GET /company/{num}/filing-history?category=confirmation-statement&items_per_page=5`
2. For each filing, follow `links.document_metadata` → that JSON's `links.document` is the raw PDF URL.
3. First item that resolves to a downloadable PDF wins; the tool stops and saves it.

Requirements

- Python 3.10+
- Playwright with Chromium installed: `playwright install chromium`
- Pillow, `requests`, `python-dotenv`
- `.env` file next to the script containing: `COMPANIES_HOUSE_API_KEY=<your-key>`
- Free key from https://developer.company-information.service.gov.uk/

Usage

```
python ch_export.py
```

```
================================================
  Companies House Export Tool
================================================

Enter company number (or multiple separated by commas, e.g. 09814501, 12345678): 09814501
```

Input accepts:

- A single number: `09814501`
- Multiple numbers, comma-separated: `09814501, 12345678, 00445790`
- Short numbers — they're zero-padded to 8 digits automatically.

PyInstaller Notes

When frozen into a PyInstaller bundle (`sys.frozen`), the script points Playwright at the system browser cache:

- macOS: `~/Library/Caches/ms-playwright`
- Windows: `~/AppData/Local/ms-playwright`

This happens before `playwright` is imported, so the bundle doesn't need to ship a full Chromium binary — it reuses the one already on the machine.