



Browser Use

“Make websites accessible for AI agents. Automate tasks online with ease.”

What is it?

Star 59.4k

browser-use is Python library to connect

- AI API (I used [OpenAI](#))
- Your Python code
- Text instructions
- Browser (fresh or already open)
- playwright
 - “powerful and versatile end-to-end (E2E) testing framework for automating web browsers. It's developed and maintained by Microsoft and is designed to be a modern and robust alternative to tools like Selenium.”



What does it do?

- "Enable AI to control your browser" 
- "Make websites accessible for AI agents. Automate tasks online with ease."

<code>captcha.py</code>	Automates CAPTCHA solving on a demo website.
<code>check_appointment.py</code>	Checks for available visa appointment slots on the Greece MFA website.
<code>find_and_apply_to_jobs.py</code>	Searches for job listings, evaluates relevance based on a CV, and applies automatically.
<code>online_coding_agent.py</code>	Implements a multi-agent system for online code editors, with separate agents for coding and execution.
<code>post-twitter.py</code>	Provides a template for automated posting on X (Twitter), including new tweets, tagging, and replies.
<code>scrolling_page.py</code>	Automates webpage scrolling with various scrolling actions and text search functionality.
<code>twitter_post_using_cookies.py</code>	Automates posting on X (Twitter) using stored authentication cookies.
<code>web_voyager_agent.py</code>	A general-purpose web navigation agent for tasks like flight booking and course searching.

My use case: travel planning

It's tedious to re-enter the same filters over and over

- location, dates, 3 people, non-smoking, WiFi
- want to see what's available, multiple dates
- sites are annoying to use (popups, specials!, irrelevant results)

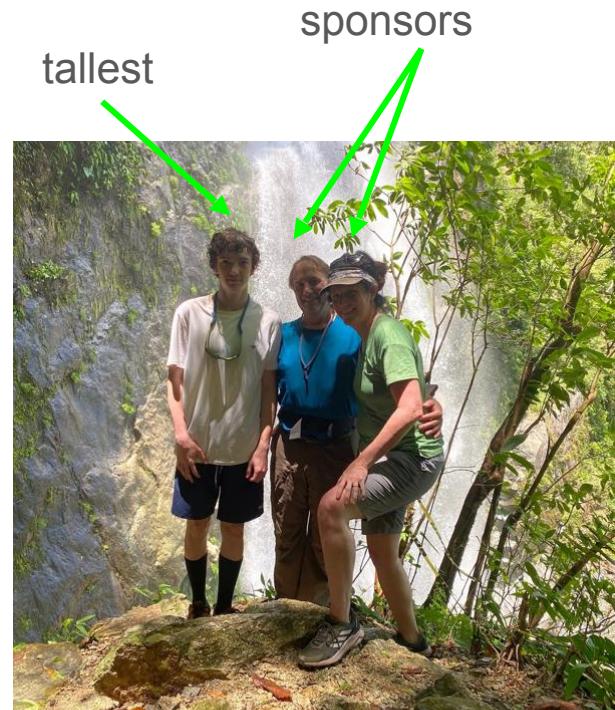
Sites are missing key (to me) filters

- double & sofa beds don't count!
- [booking.com](https://www.booking.com): 1 king bed is a "great choice" for 3 adults (no)

Cut through the noise, save time, focus on the fun parts

The dream (from the [web_voyager_agent.py example](#)):

- ""Find and book a hotel in Paris with suitable accommodations for a family of four (two adults and two children) offering free cancellation for the dates of February 14-21, 2025. on <https://www.booking.com/>""



Q: who wants the sofa bed?
A: nobody!

Setup

OpenAPI

- create account, add \$
- set OPENAI_API_KEY in environment

install browser-use

- pip install browser-use

install browsers so playwright can use them

- npx playwright install

Scribe to ChatGPT to English steps

copy example and start editing

- [web_voyager_agent.py](#)

Note: quick start if you understand this, huge lift if not

Demos

Airbnb

- set filters
- pause and ask a human to adjust map
- extract beds, screenshot calendar

[booking.com](#)

- set filters, extract data
- manipulate URL to start on results page, extract data

SCE

- scope down to extract details from a single page

Interesting features

- step hooks for custom actions: screenshot, html
- session recordings
- system instructions to close popups: great use case for vague actions
- output to pydantic models: convert unstructured info to structured
- connect to open logged in browser
 - but also scary: read how ChatGPT agent [spent \\$31 on a dozen eggs](#)

Conclusion: try again in a bit



Who is this for?

- requires Python, playwright, OpenAI, browser with remote debugging
- debugging is hard: iterating on prompts, matching logs with traces, iterations are slow
- use this to generate a “compiled” version? but Selenium browser does this, simpler

Is English the right language?

- great for popups; didn't work for date pickers
- maintenance unknown
- mixing English & DOM selectors?

Prototype / one off usage: \$2.50/1M tokens

- personal use: \$0.34/run is great (if it works)
- datafeeds: \$0.001/run (10k runs, \$10/day)

