End-User Manual for SeeAct Chrome Web Agent

Table of Contents

Introduction

The SeeAct Chrome Web Agent is a powerful browser extension that allows you to instruct an AI 'agent' to perform tasks on your behalf through the browser. This agent can navigate websites, find information, and even fill out forms to prepare transactions, all based on your instructions.

Key features:

- AI-powered web navigation and interaction
- Ability to perform complex, multi-step tasks
- Monitor mode for supervised operation
- Task history and logging for transparency

Whether you're looking to automate a tedious web task (while you do something else away from the computer) or simply to explore the capabilities of AI-assisted browsing, SeeAct is here to help.

Please be aware that using this extension with a given AI cloud provider incurs modest costs (e.g. several cents per task) for the AI model's usage. The extension will not perform tasks without your explicit instruction, so you can control the costs by limiting the number of tasks you ask the agent to perform. There are also configuration options that can limit costs in cases where the agent gets stuck or confused.

Initial Setup

Once the extension is loaded into Chrome, please follow steps 3-5 of the README's installation instructions to prepare the extension for use (steps 6-12 of the exhaustive walkthrough).

Extension tips

Many UI elements (in side panel and options menu) have more explanation of their purpose and behavior in the form of tooltips that appear when hovering over the elements' labels.

Also, text blocks in the side panel (under "Action History" or "Pending Action" headings, plus some of the temporary status messages that appear just above the "Pending Action" section) will have tooltips with more detail when you hover over them.

Using the Extension's Web Agent Functionality

1. Navigate to the webpage where you want to start your task.
2. Click the SeeAct icon to open the side panel if it's not already open.
3. In the "Agent Task Specification" field, enter a description of the task you want the agent to perform. Be as specific as possible. For example:
    - "Find the cheapest flight from New York to London departing next month, but stop when asked for payment details"
    - "Locate the contact email for the HR department on this company website"
4. Click the "Start Agent" button to begin the task.
5. The agent will start performing actions on the page. You can follow its progress in the "Actions History" section of the side panel.
    - You can also get more information by hovering over the status update field (just below the "Actions History" section).
6. If you have Monitor Mode enabled, you'll be asked to approve or reject each action before it's taken.
    - If the agent seems confused or stuck, you can write a message in the 'Feedback' field before clicking the Reject button to give the AI agent a hint, clarification, prohibition, etc.
7. Additionally, if the built-in "automatic safety monitor" module determines that the agent's next action poses an elevated risk of difficult-to-reverse harm for the user, it will pause until you approve or reject the action.
    - You can configure the sensitivity of this module in the Options page.
8. Once the task is complete, the agent will usually terminate the task automatically. If it starts doing something unnecessary after achieving the goal, or if you want to stop the agent early for any reason, click the "Terminate Task" button.

Example Tasks

- Find a particular form or informational document on a large and complex website
- Find several items and add them to your cart on an eCommerce site
- Prepare the booking of an appointment or reservation on a service website

Keyboard Shortcuts

- In monitor mode, press `Alt + Shift + J` to approve the agent's proposed next action

- In monitor mode, press `Alt + Shift + U` to reject the agent's proposed next action
  - The feedback field will only work when the Reject button is clicked, not when the reject keyboard shortcut is used.

Using the Extension's 'Unsafe Action Annotation' Feature

Please see this separate document.

Troubleshooting

Common issues and possible solutions:

1. **The agent always quits just after a task starts**:
   - Ensure you've entered a valid API key in the Options page
   - Check that you're on a regular web page, not a chrome:// URL
   - Make sure you don't click outside the browser window after the task has started. That may break the agent's ability to interact with the browser.
2. **The agent seems stuck**:
   - Try terminating the task and starting again with a tweaked task description
3. **The agent quit early for no clear reason**:
   - Mouse over the "Task Ended" entry in the 'Actions History' section for a brief explanation from the agent about why it stopped the task.
   - Open the zip file which was downloaded at the end of the task and look near the end of the "agent.log" file inside it for error messages.
   - If the logs don't have a clear error message, you can open Chrome's Extensions management page (paste `chrome://extensions/` into the browser's URL bar), find the SeeAct extension, and click on its "Errors" button (if present)
   - You may very well need to report the issue (with as much detail as you can safely provide) to the extension's developers.
4. **The Hover action isn't working**
   - Make sure you keep your actual mouse cursor either inside the side panel or entirely outside of the browser window during the task.
5. **Weird Chrome extension error when agent tries to press Enter**
   - i.e. "Cannot access a chrome-extension:// URL of different extension"
   - This seems to involve a conflict between the SeeAct extension and another extension that's also trying to do something with the current website. Please try again in a new Chrome 'profile' which only has SeeAct enabled.

Possible fixes for rare/extreme issues:

1. **The extension isn't responding**:
   - Try closing and reopening the side panel
   - If problems persist, try disabling and re-enabling the extension

If you encounter persistent issues, please check the "Feedback" section for ways to report problems.

Configuration

Access the configuration menu by clicking the "Options" button in the side panel.

Key settings:

- **AI Model Provider**: Choose between OpenAI, Anthropic, or Google DeepMind. Additional options may be added in future.
- **API Key**: Enter your API key for the chosen provider
- **Annotator Mode**: This reveals the side-panel UI component for the "unsafe actions annotation" mode.
- **Monitor Mode**: When enabled, you must approve each action before it's taken
  - appropriate for enthusiasts trying to give the agent sensitive tasks where a mistake could cause monetary loss, or who want to see how the agent does with occasional hints/guidance
- **Max Operations**: Limit the number of actions the agent can take in a single task
- **Safety Monitor Threshold**: Adjust the sensitivity of the automatic safety monitor
  - choose a minimum predicted risk level that would trigger a pause for your approval
- **Log Level**: Control the detail level of logs (useful for troubleshooting)

If you aren't sure what a setting does, hover over the option's label for an explanation.

Feedback

We value your input! If you encounter issues, have suggestions, or want to share your experience, the best avenue is creating a GitHub Issue in the extension's repository.

This enables easier tracking of issues and makes them visible to other users who may have similar problems. It requires creating a (free) Github account but does not require any coding knowledge.

If you're unable to use GitHub, you can also email your feedback to the extension's developers at salisbury.100@buckeyemail.osu.edu . Please be aware that this method may take longer to receive a response.

In general, please provide as much detail as you can about the issue you're facing. This includes:

- The steps you took before encountering the issue
- A description of the problematic behavior
- A copy of the end-of-task zip file (if the issue occurred during a task), with any sensitive information redacted
  - If the task involved any pages containing sensitive personal information, this may require unzipping the file and redacting those parts of the screenshots and HTML files inside (e.g. using Paint or Notepad), then re-zipping the folder
  - We understand that this may not be feasible for people who are not fully comfortable with non-typical uses of computers.
- The log file produced by clicking the 'Download misc logs' button in the side panel shortly after the issue occurred
  - This may also contain sensitive data that needs to be redacted, but it's less likely to contain such data than the end-of-task zip file

In general, it may be safer to provide the zip file and log file to the developers via email after creating the Github issue (rather than posting them to the publicly visible Github issue). When doing so, please include the Github issue number

in the email.

Contributing

If you're interested in contributing to the development of the SeeAct Chrome Web Agent, please fork the extension's GitHub repository, commit your changes in the fork, and make a pull request back to the main repository.

If you contribute to the project, you agree to license your contributions under the Open RAIL-S license. You also agree to follow the SeeAct project's code of conduct

License

This extension is made available to users under an Open RAIL-S license.

Privacy Policy

Please see the full privacy policy. This can also be accessed from a link at the bottom of the extension's Options menu.

Acknowledgements

This extension is built on top of the SeeAct project, with guidance and supervision from Boyuan Zheng and Professor Yu Su of The Ohio State University's NLP Group.

The 'unsafe action annotation' feature is an implementation of ideas that were originated by Professor Yu Su & Ziyu (Maggie) Huan and then refined by Boyuan Zheng, Zeyuan Liu, and Scott Salisbury.
Additionally, the 'unsafe action annotation' feature's implementation was refined based on feedback from beta users Alvin Huang, Cloudy Zheng, Lee A. Davis, Michael Lin, Xiaolong Jin, Zeyuan Liu, and Zheng Du.

A majority of the logic for identifying interactive UI elements, collecting information about them, formatting that information in LLM prompts, and generally structuring the LLM prompts is ported from that project's Python code.
The 'monitor mode' feature's design was also based on that project (although the UI, keyboard shortcuts, and text-feedback-field are new).

This project of course benefits from a number of open-source libraries, from `idb` and `jszip` to `async-mutex` and `get-xpath` .

Finally, of course, the actual AI brains behind this extension's web agent functionality are provided by OpenAI, Anthropic, and Google DeepMind.