TaxyAI / **browser-extension** Public

Automate your browser with GPT-4

MIT license

420 stars **11** forks

☆ Star ▾ | Notifications

Code | Issues 4 | Pull requests | Actions | Projects | Security | Insights

main ▾ | Go to file

corbt centralize action definitions ... ✓ 1 hour ago **179**

View code

README.md

Taxy AI: Full Browser Automation

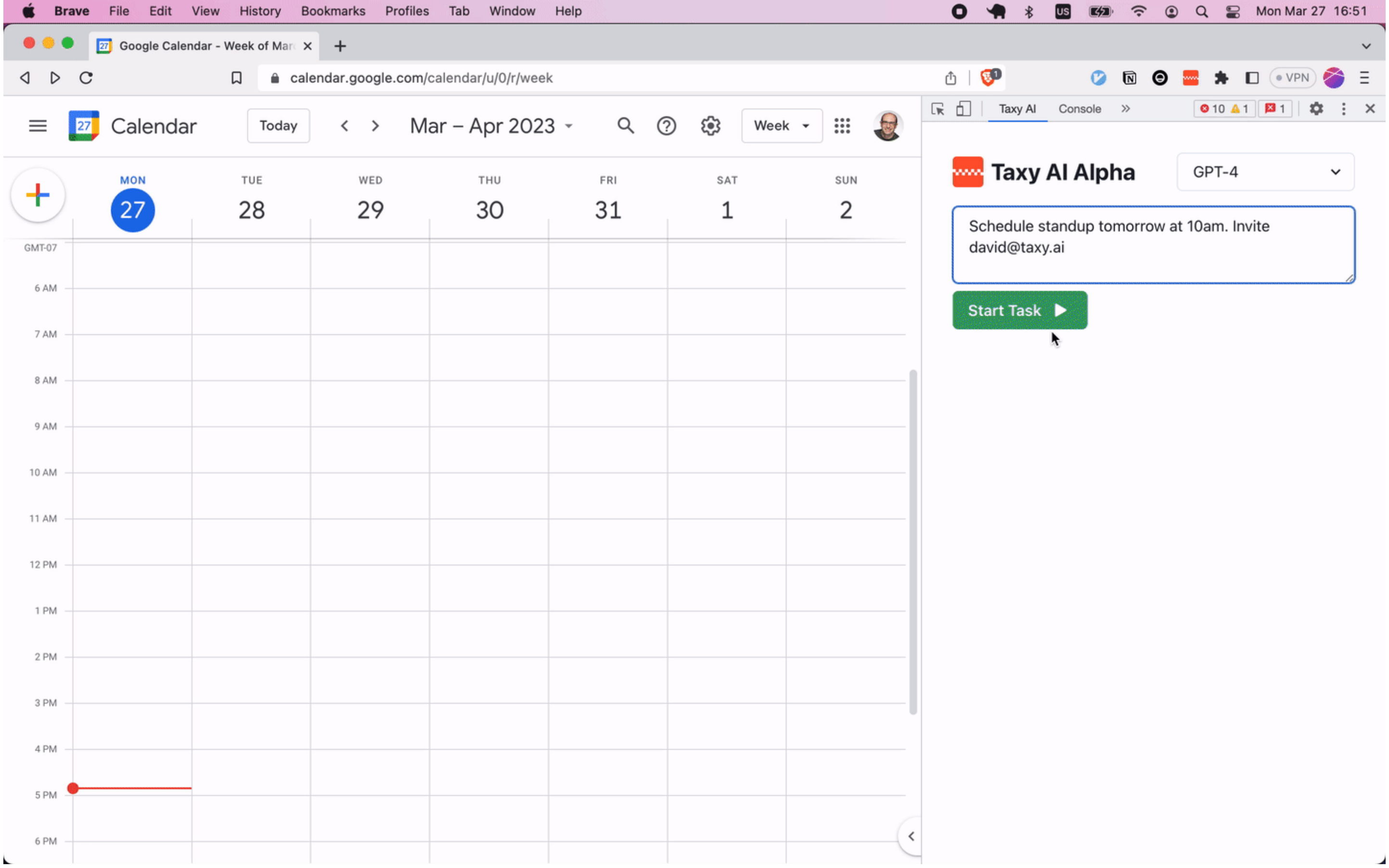
Waitlist | Discord

Taxy uses GPT-4 to control your browser and perform repetitive actions on your behalf. Currently it allows you to define ad-hoc instructions. In the future it will also support saved and scheduled workflows.

Taxy's current status is **research preview**. Many workflows fail or confuse the agent. If you'd like to hack on Taxy to make it better or test it on your own workflows, follow the instructions below to run it locally. If you'd like to know once it's available for wider usage, you can sign up for our waitlist.

Taxy is fully open-source, and we don't send any page contents or instructions to our servers.

Here's Taxy using Google Calendar with the prompt "Schedule standup tomorrow at 10am. Invite david@taxy.ai"



Schedule standup tomorrow at 10am. Invite david@taxy.ai

Table of Contents

Installing and Running

Currently this extension is only available through this GitHub repo. We'll release it on the Chrome Web Store after adding features to increase its usability for a non-technical audience. To build and install the extension locally on your machine, follow the instructions below.

Installing the extension

1. Ensure you have Node.js >= **16**.
2. Clone this repository
3. Run `yarn` to install the dependencies
4. Run `yarn start` to build the package
5. Load your extension on Chrome by doing the following:
    i. Navigate to `chrome://extensions/`
    ii. Toggle `Developer mode`
    iii. Click on `Load unpacked extension`
    iv. Select the `build` folder that `yarn start` generated

Running in your browser

1. Once installed, the browser plugin will be available in two forms:
    i. As a Popup. Activate by pressing `cmd+shift+y` on mac or `ctrl+shift+y` on windows/linux, or by clicking the extension logo in your browser.
    ii. As a devtools panel. Activate by first opening the browser's developer tools, then navigating to the `Taxy AI` panel.
2. The next thing you need to do is create or access an existing OpenAI API Key and paste it in the provided box. This key will be stored securely in your browser, and will not be uploaded to a third party.
3. Finally, navigate to a webpage you want Taxy to act upon (for instance the OpenAI playground) and start experimenting!

How it Works - The Action Cycle

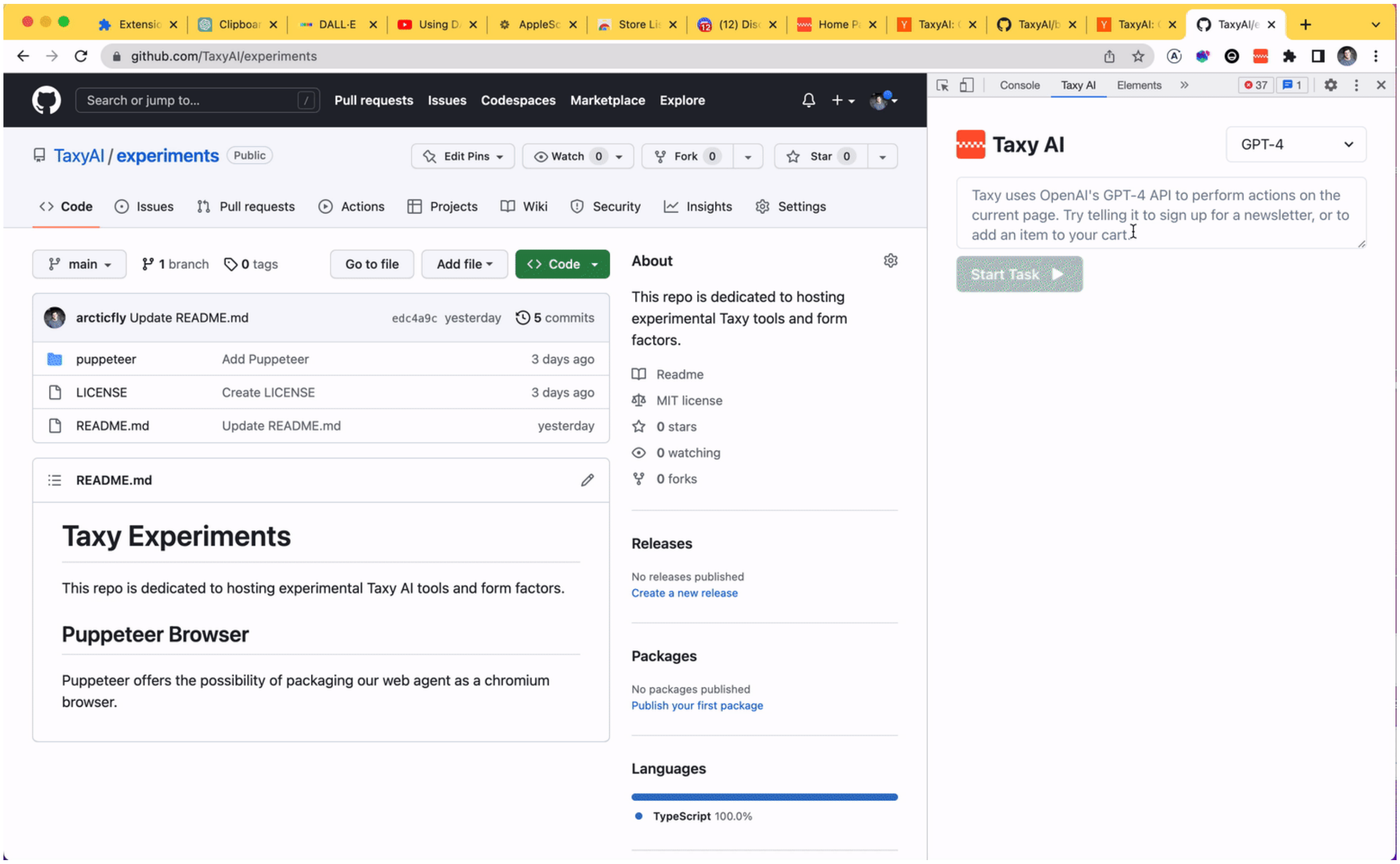
1. Taxy runs a content script on the webpage to pull the entire DOM. It simplifies the html it receives to only include interactive or semantically important elements, like buttons or text. It assigns an id to each interactive element. It then "templatizes" the DOM to reduce the token count even further.
2. Taxy sends the simplified DOM, along with the user's instructions, to a selected LLM (currently GPT-3.5 and GPT-4 are supported). Taxy informs the LLM of two methods to interact with the webpage:
    i. `click(id)` - click on the interactive element associated with that id

    ii. `setValue(id, text)` - focus on a text input, clear its existing text, and type the specified text into that input

3. When Taxy gets a completion from the LLM, it parses the response for an action. The action cycle will end at this stage if any of the following conditions are met:
    i. The LLM believes the task is complete. Instead of an action, the LLM can return an indication that it believes the user's task is complete based on the state of the DOM and the action history up to this point.
    ii. The user stopped the task's execution. The user can stop the LLM's execution at any time, without waiting for it to complete.
    iii. There was an error. Taxy's safety-first architecture causes it to automatically halt execution in the event of an unexpected response.
4. Taxy executes the action using the chrome.debugger API.
5. The action is added to the action history and Taxy cycles back to step 1 and parses the updated DOM. All prior actions are sent to the LLM as part of the prompt used to determine the next action. Taxy can currently complete a maximum of 50 actions for a single task, though in practice most tasks require fewer than 10 actions.

Simple Demos

Protecting the main branch in GitHub



arcticfly Update README.md	edc4a9c yesterday	5 commits
puppeteer	Add Puppeteer	3 days ago
LICENSE	Create LICENSE	3 days ago
README.md	Update README.md	yesterday

README.md

Taxy Experiments

This repo is dedicated to hosting experimental Taxy AI tools and form factors.

Puppeteer Browser

Puppeteer offers the possibility of packaging our web agent as a chromium browser.

About

This repo is dedicated to hosting experimental Taxy tools and form factors.

Readme
MIT license
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

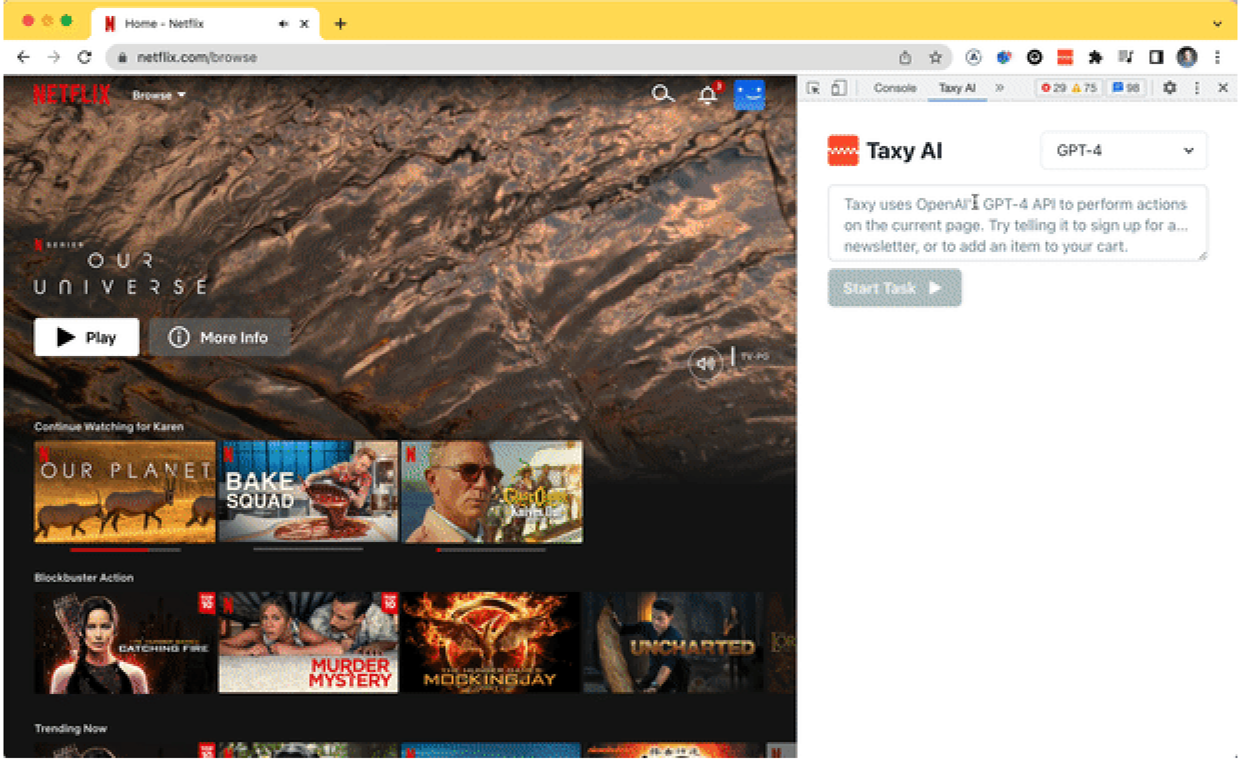
No packages published
Publish your first package

Languages

TypeScript 100.0%

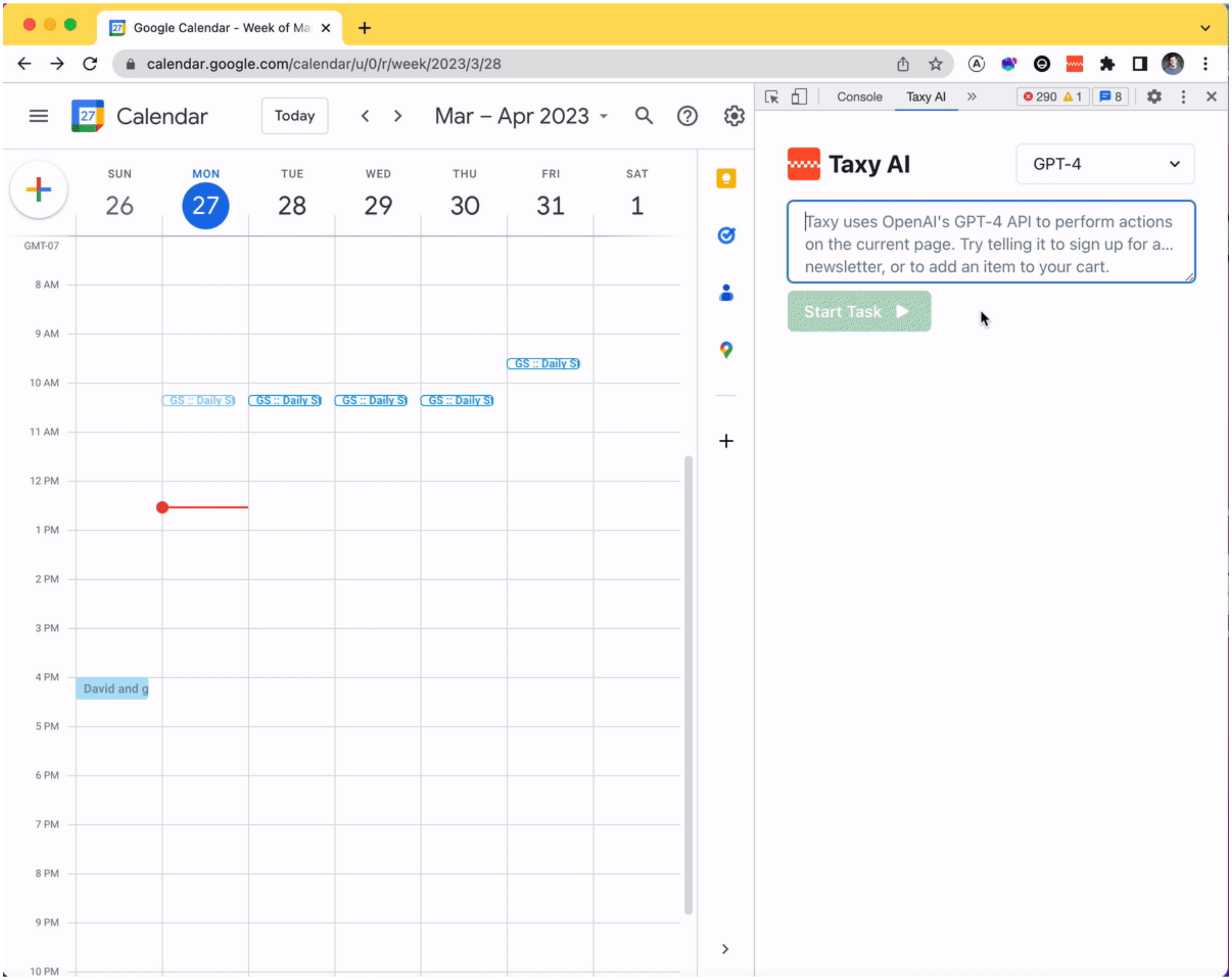
Taxy uses OpenAI's GPT-4 API to perform actions on the current page. Try telling it to sign up for a newsletter, or to add an item to your cart.

Searching for and playing the movie Oblivion in Netflix



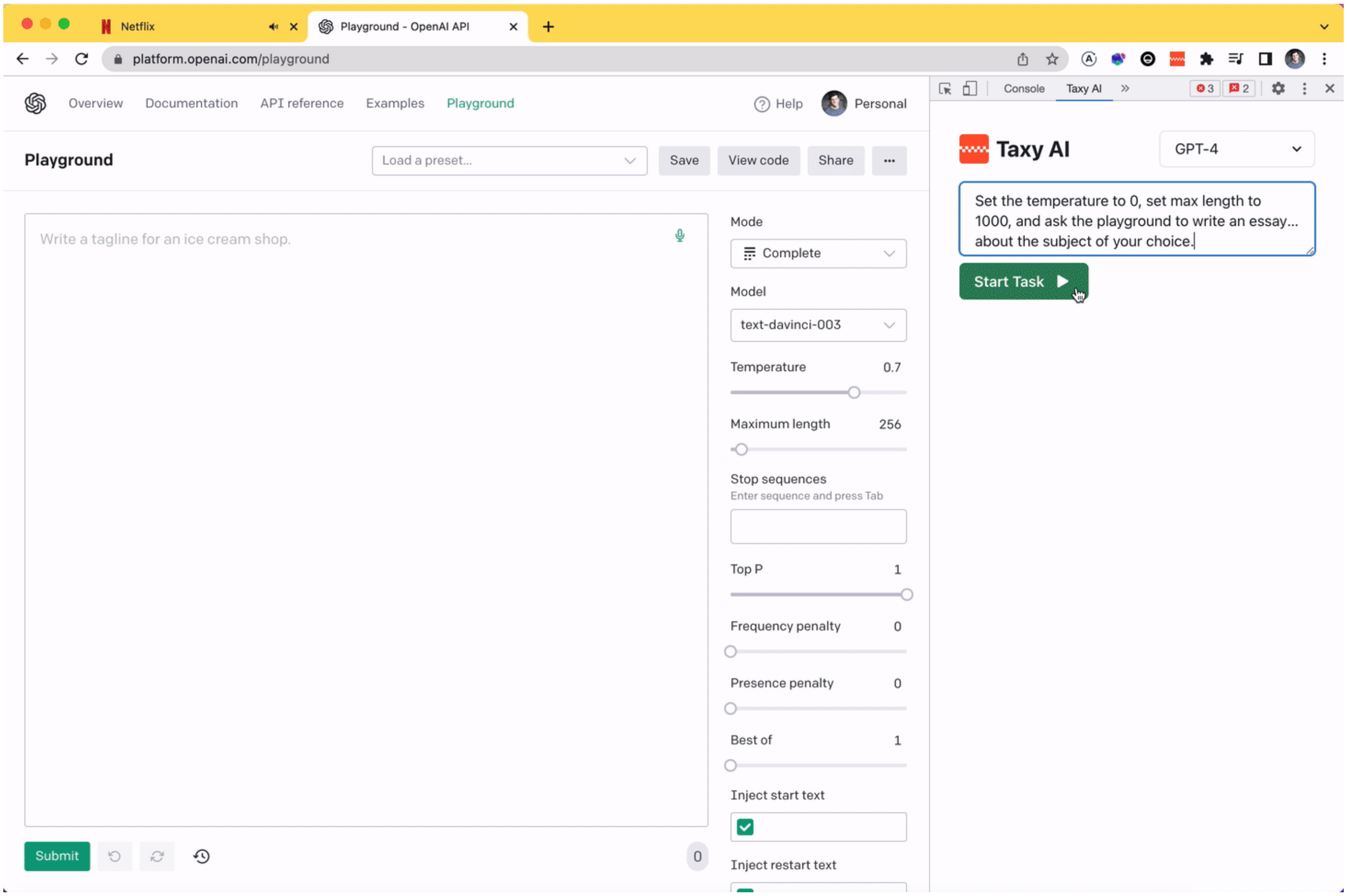
Taxy uses OpenAI's GPT-4 API to perform actions on the current page. Try telling it to sign up for a... newsletter, or to add an item to your cart.

Creating a calendar event



Taxy uses OpenAI's GPT-4 API to perform actions on the current page. Try telling it to sign up for a... newsletter, or to add an item to your cart.

Writing an essay in the OpenAI playground



Mode: Complete

Model: text-davinci-003

Temperature: 0.7

Maximum length: 256

Stop sequences: Enter sequence and press Tab

Top P: 1

Frequency penalty: 0

Presence penalty: 0

Best of: 1

Inject start text: ☑

Inject restart text

Set the temperature to 0, set max length to 1000, and ask the playground to write an essay... about the subject of your choice.

Add your own!

If you have an interesting demo you'd like to share, submit a PR to add your own!

Tech Stack

Technology currently used by this extension:

- Chrome Extension Manifest V3
- React 17
- Webpack 5
- Webpack Dev Server 4
- React Hot Loader
- eslint-config-react-app
- Prettier
- TypeScript

Resources

- Getting Started with Chrome Extensions

Releases

1 tags

Packages

No packages published

Contributors 2

arcticfly

corbt Kyle Corbitt

Languages

● **TypeScript** 84.7% ● **JavaScript** 11.9% ● **CSS** 2.2% ● **HTML** 1.1% ● **SCSS** 0.1%