IMAGE Safari Extension - Handover Notes

Context

Goal: Make the IMAGE Browser Extension work reliably on macOS & iOS Safari and get the extension on the App Store.

Right now, content scripts add “Interpret this graphic with IMAGE” controls to pages and send visual data (images, map snapshots, chart data) to background/service worker, which forwards requests to the IMAGE server, tracks responses, and relays audio/haptic renderings back to the page. Feedback page lets users rate interpretations and optionally consent to data use.

Key problem we were solving: Apple submission failed about 2 and a half years ago, and the reason was: “Apple does not allow background scripts to run persistently in iOS devices due to memory and power constraints.”

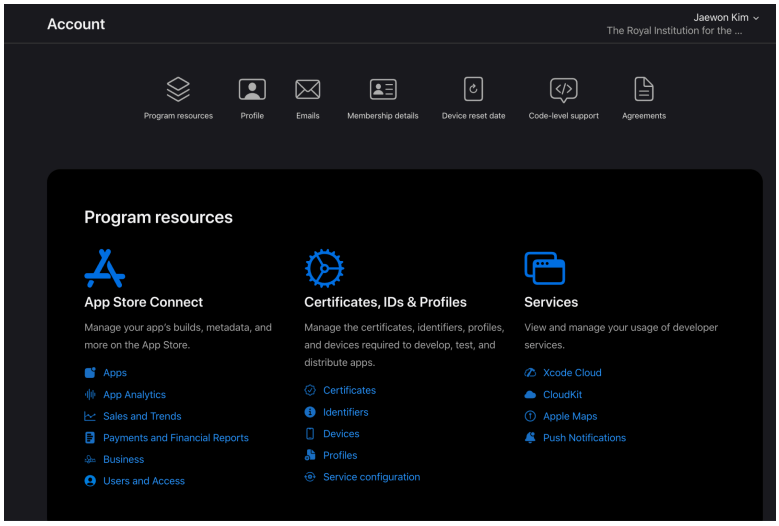
Environment

Setting up the project and running it on your phone. (Required: macOS)

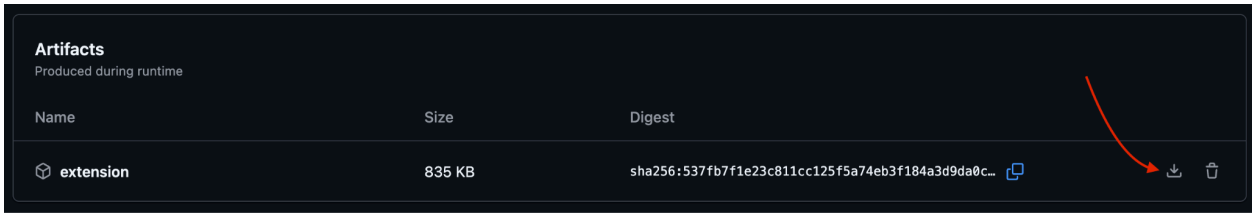
1. MAKE SURE TO GET THE INVITE TO THE APPLE DEVELOPER TEAM WITH THE CORRECT PERMISSIONS FROM THE GET-GO!

   ✅With Access to Certificates, Identifiers & Profiles.
   ✅Access to Cloud Managed Distribution Certificate

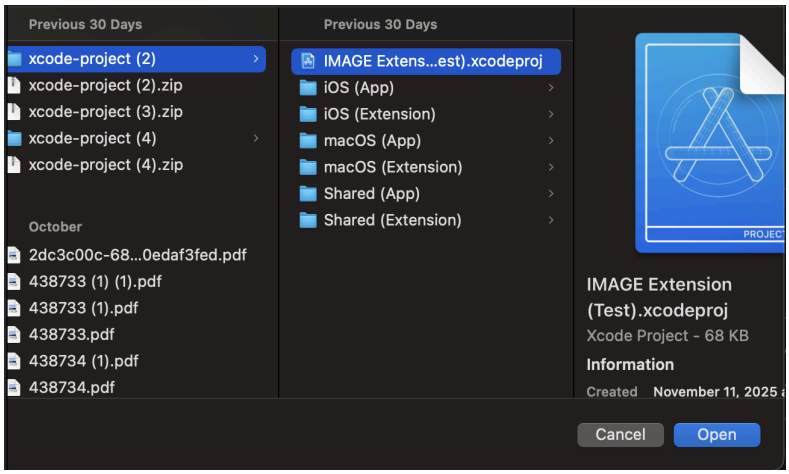
2. This is how your Apple Developer console should look like:



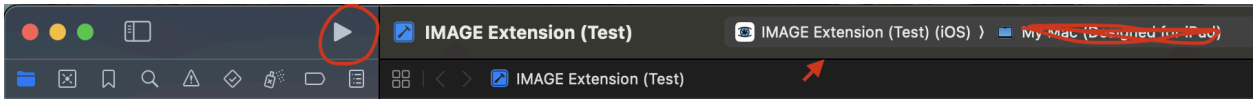
3. To test, download the latest extension from GitHub https://github.com/Shared-Reality-Lab/IMAGE-browser/actions/runs/19288969422



4. Open the .xcodeproj from XCode



5. Plug in your phone to your mac
6. Run the extension using the instructions below



   a. Make sure the IMAGE Extension (Test) (iOS) scheme is chosen
   b. Make sure it says "{your name}'s iPhone" where its scribbled
   c. Run the extension using the play button

If you want to build your own project after making changes, follow the instructions here:

https://github.com/Shared-Reality-Lab/IMAGE-browser/blob/main/Safari.md#convert-the-chrome-extension-to-obtain-an-xcode-project-common-to-ios-and-macos-safari

Where to look in the codebase first to understand IMAGE

When you open the repo, prioritize:

background.ts: Main message router and state holder (where the App Store rejection problem lies).

background-utils.ts: Keep-alive patterns, port management, timers/alarms that may need redesign.

offscreen.js: Any offscreen document logic intended to keep work going; may be violating iOS policies.

Content scripts (map/chart/image handlers): How requests are assembled, sent, and UI is updated in response.

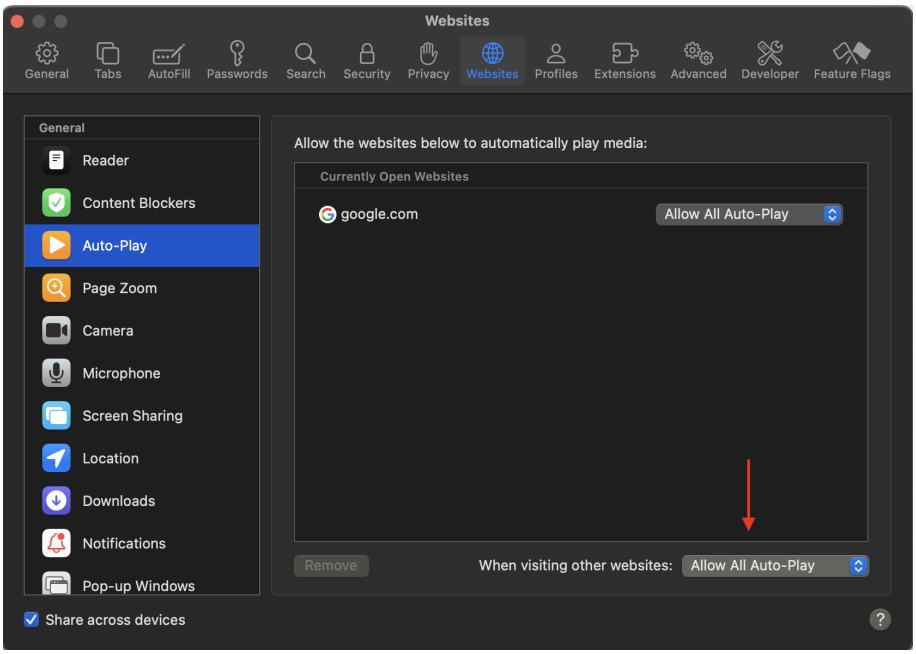
Feedback page code: Now uses window.location.href; keep this pattern intact unless there's a strong reason to change.

If you start by understanding these pieces and then work through following steps, you'll be directly advancing the core remaining challenges of the Safari IMAGE extension rather than re-solving already-answered questions.

Bugs remaining

Bugs list from: https://github.com/Shared-Reality-Lab/IMAGE-browser/blob/main/Safari.md#known-issues-for-safari

- The extension does not produce any sound when an IMAGE request is sent.
    - For this bug, a workaround was found but no real final solution. Here's the workaround:



The real solution will require some deeper tech and research.

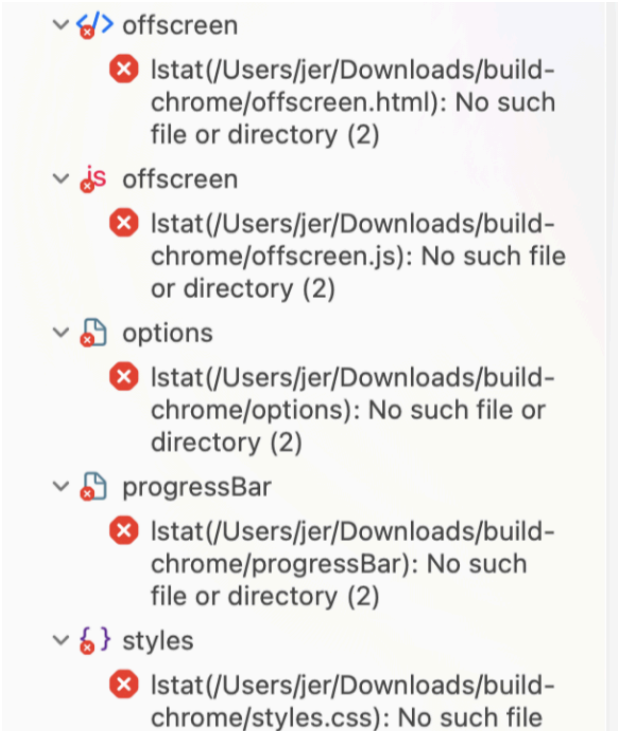
- The feedback link is missing when an IMAGE result is received for maps and charts.
    - This bug wasn't really found to be problematic.

- Connecting haptic devices is not supported.
    - True, one of the next steps is to allow haptic devices.

- iOS only: The extension will become unresponsive and require a manual restart every time Safari is restarted.
    - True, one of the more crucial bugs to fix.

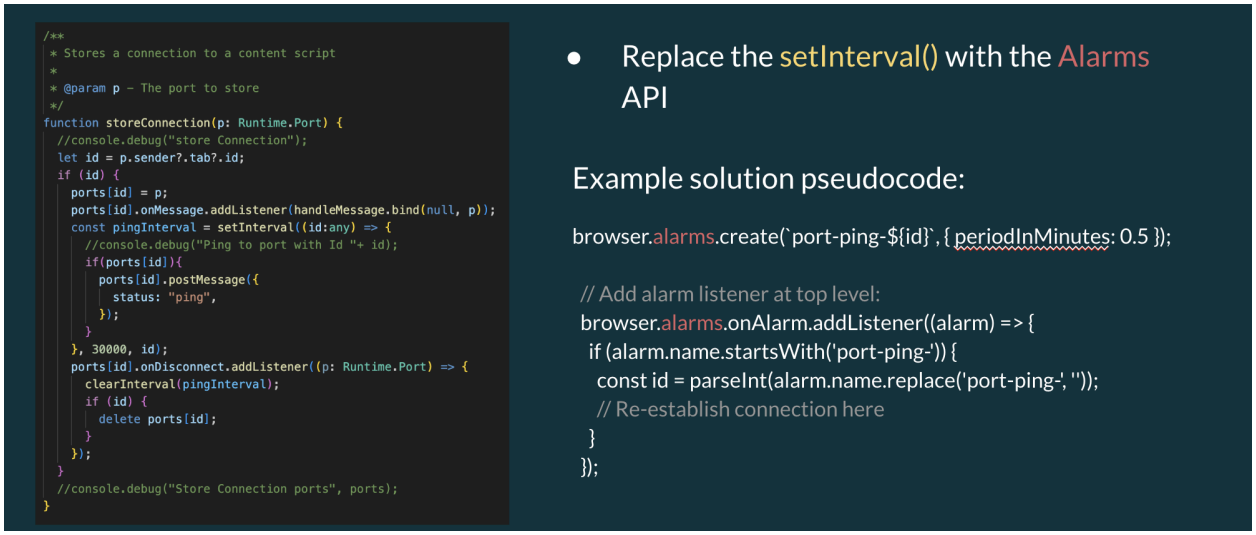
Recommended immediate next steps

1. Get to the App Store submission asap
    - Submit the current Safari extension to the App Store.
    - Why: We need concrete review feedback to know exactly which background patterns Apple does/doesn't accept, instead of guessing.
    - Current progress: Trying to get the latest extension to build on Professor Cooperstock's local mac in order to build the final version.
    - Currently blocked on the following build error when trying to build the extension



    - Steps to build the extension for App Store submission:
        - Xcode → open .xcodeproj → Product → Archive

2. Depending on the result of the App Store submission
    a. If the App Store rejects, look at the rejection message and analyze exactly what's wrong. If it's something with the background script and its persistence, try the following solution:



Then re-attempt submission ensuring that the functionality didn't break. If the second submission doesn't work, the entire background script might need a refactor into a more event-driven solution and a browser local state storage instead of the existing response map system and forced keep-alive mechanisms.

    b. If the App Store accepts, that's great we should publish the extension and then improve on the existing list of bugs listed out here: https://github.com/Shared-Reality-Lab/IMAGE-browser/blob/main/Safari.md#known-issues-for-safari

Recommended future work

1. Broader evaluation and user studies
    - After you have a stable build and App Store availability, plan structured tests with real users.
    - Measure end-to-end flow: triggering interpretation, audio clarity, navigation, feedback submission.
    - Conduct research on existing user experience and document any new bugs and usability issues, what's good and what's to change.
    - Use results to prioritize remaining technical work.
2. Adapting to haptic devices and devices that visually impaired individuals use in their daily lives
    - Adapting to existing devices like Braille or VoiceOver now come into scope once the extension is stable and released.