infocusp
Innovations

Chrome Extension Development

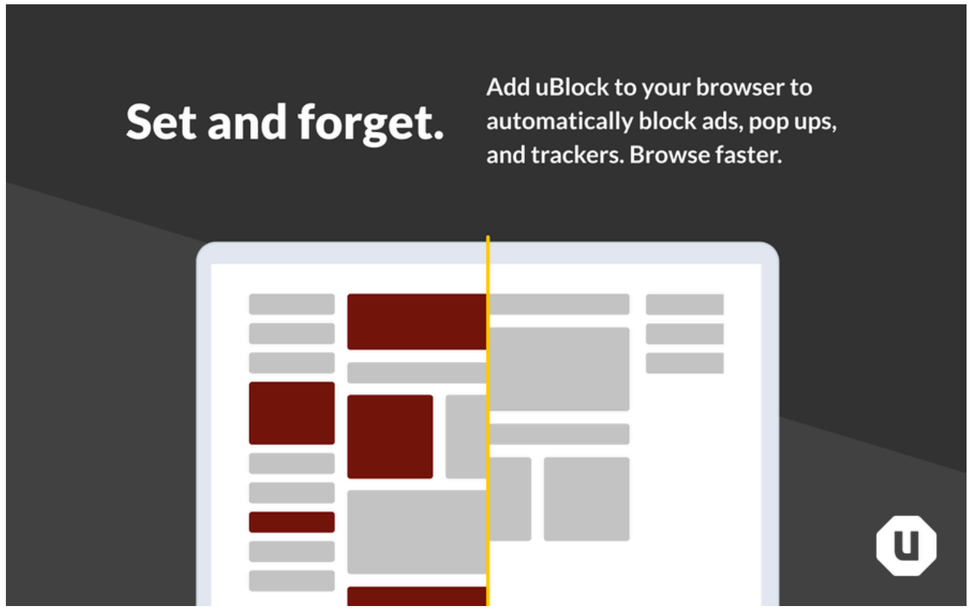
19 June 2024

Dharansh Patel

Vinay Kakkad

Why Build Chrome Extensions?

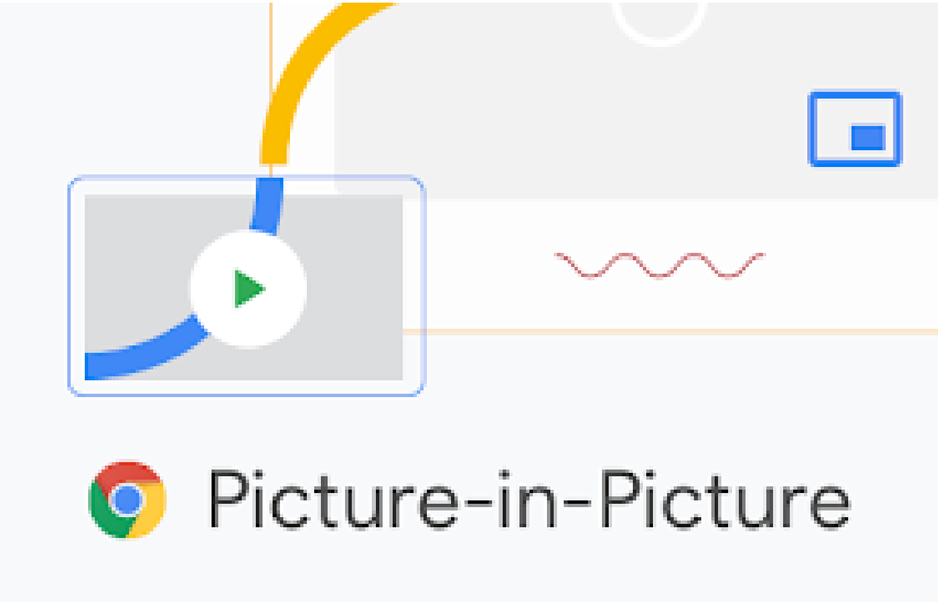
Ad Blocking and Privacy Protection



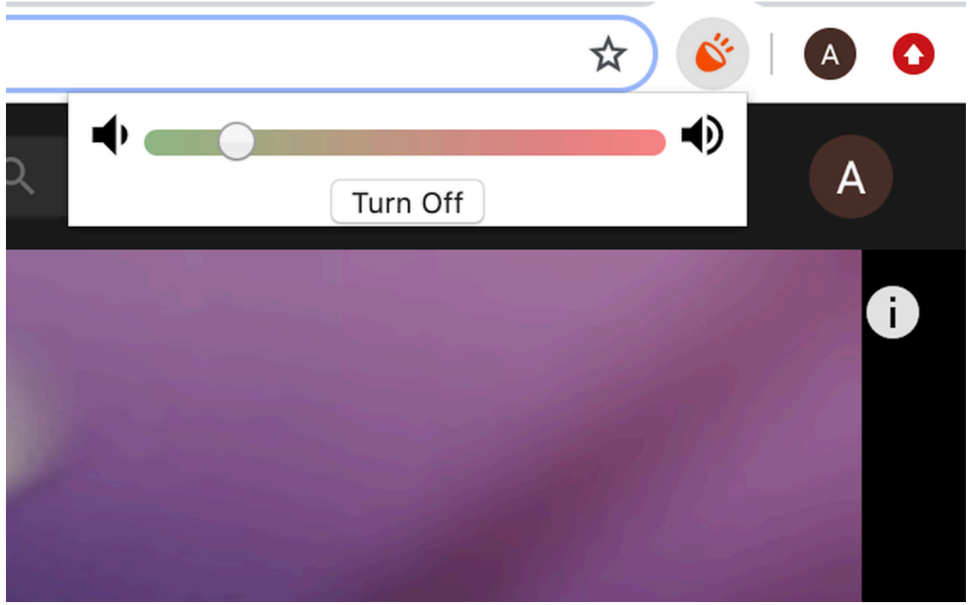
uBlock

HTTPS Everywhere

Accessibility Improvements

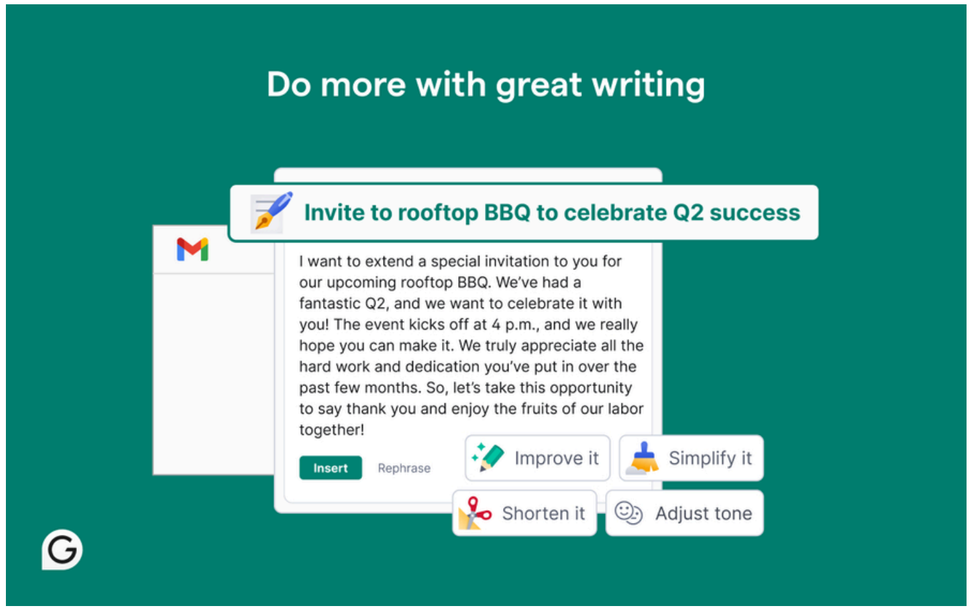


Picture-in-Picture

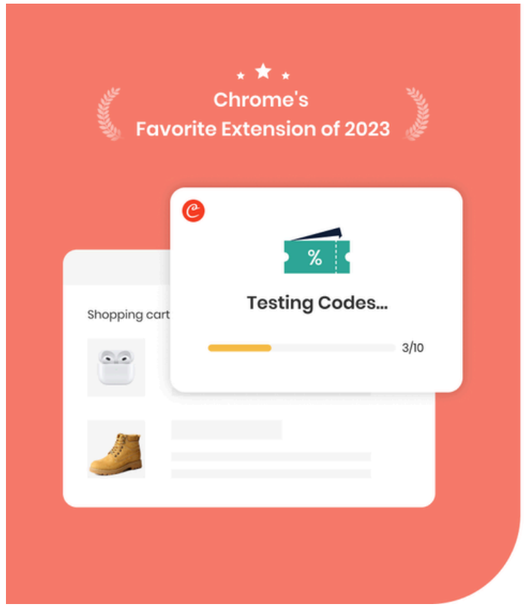


Volume Booster

Enhancing Websites



Grammarly



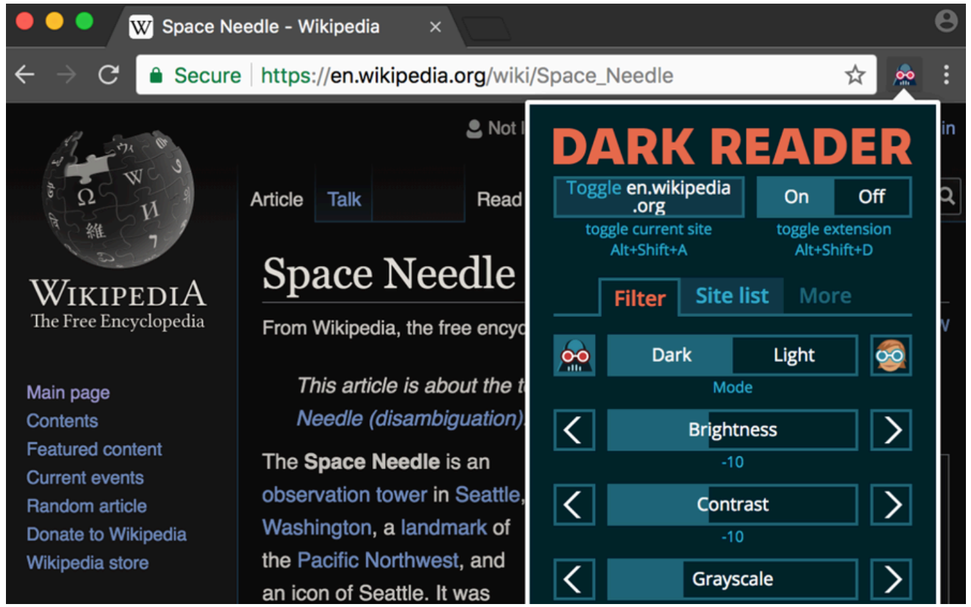
6+ million smart shoppers selected Coupert

Coupert searches for the internet's best coupons and tests them all in seconds.

Coupert

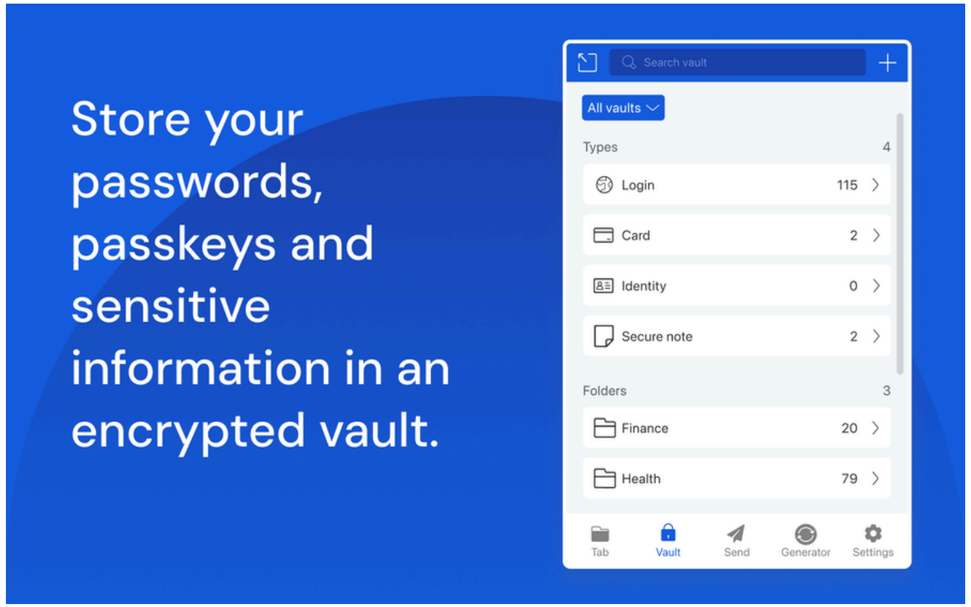
Personalise Browser & Sites

Daily.dev

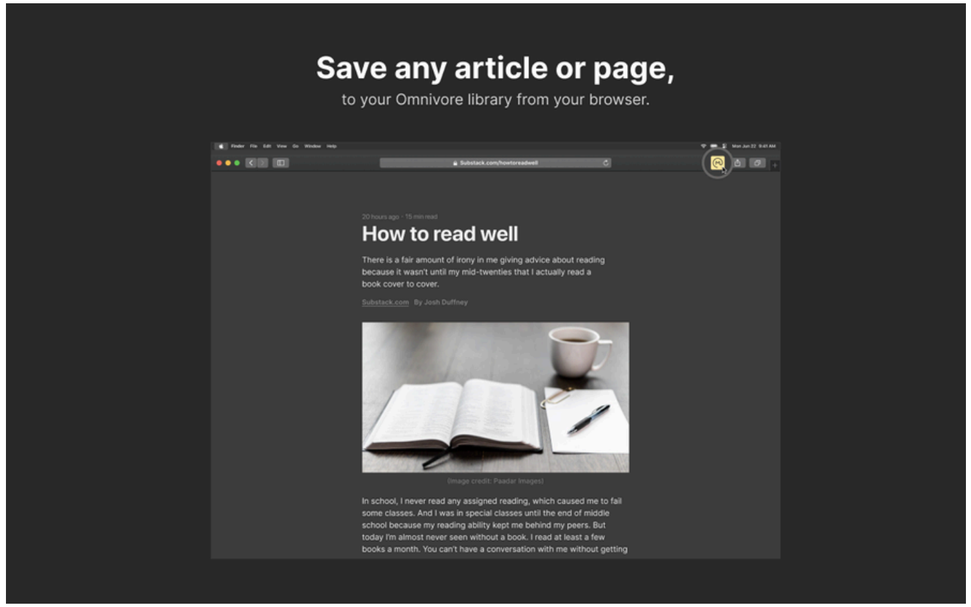


DarkReader

Extension Products



Bitwarden

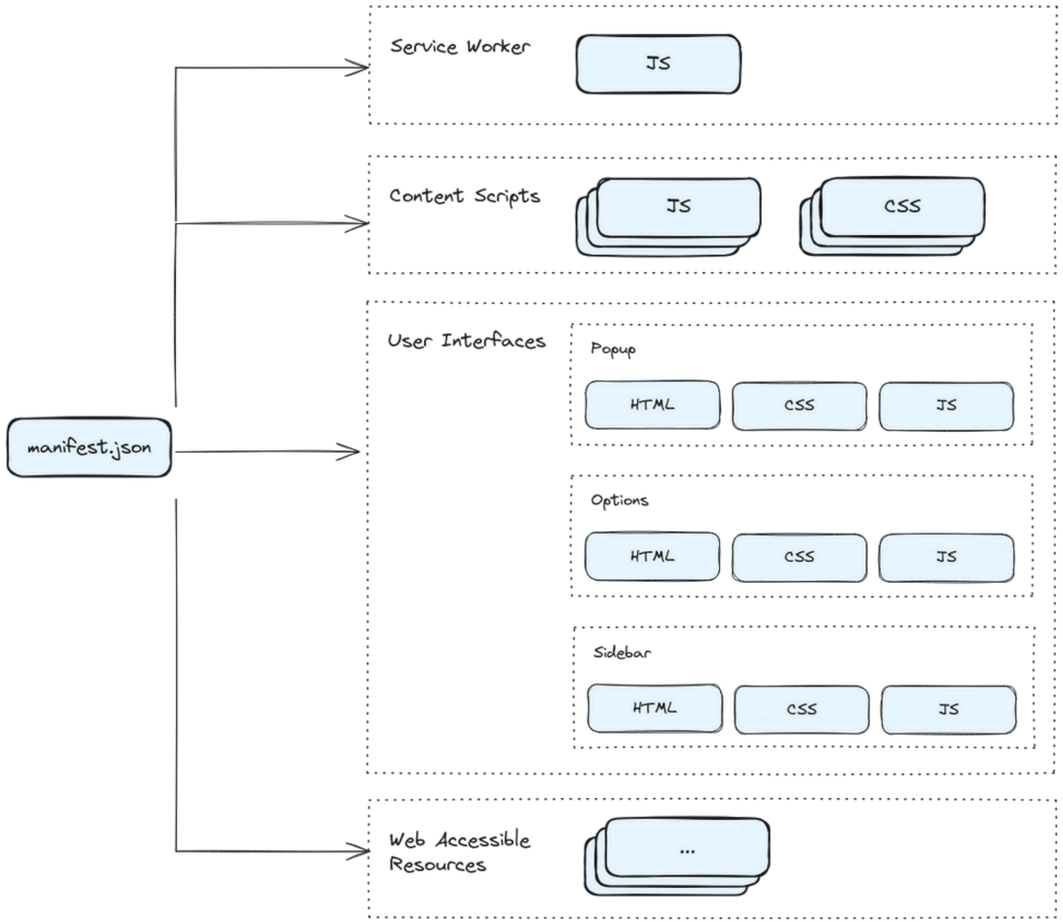


Omnivore

Range of Extension

- Processing complexity
  - From volume booster to grammarly
- Changes in web page styling
  - From bitwarden to dark reader
- Applicability of extensions
  - From codeforces to uBlock
- User Interaction
  - From uBlock to grammarly

Anatomy of Extension




```
{
  "name": "TF.js mobilenet in a Chrome extension",
  "version": "0.0.0",
  "description": "Classifies Images",
  "permissions": [...],
  "action": {"default_popup": "src/popup.html"},
  "options_ui": {"page": "src/options.html" },
  "background": {"service_worker": "src/service_worker.js"},
  "content_scripts": [{
      "matches": ["http://*/*", "https://*/*"],
      "js": ["src/content.js"],
      "css": ["src/content.css"],
  }],
  "manifest_version": 3,
  "icons": {
    "16": "images/get_started16.png",
    "32": "images/get_started32.png",
  }
}
```

Content Scripts

- Used for interacting with and modifying existing webpages
- Run in context of webpages but in an isolated world (separate from the webpage's JavaScript)
- Capabilities
  - Has access to the DOM of a webpage
  - Has access to *subset of extenesion APIs* (web-accessible resources, storage, messaging)
  - Has access to the fetch API
- Can be attached
  - Statically: Declared in the manifest file
  - Dynamically: Injected using chrome.tabs.executeScript
  - Programmatically: Using chrome.scripting.executeScript (requires "scripting" permission)

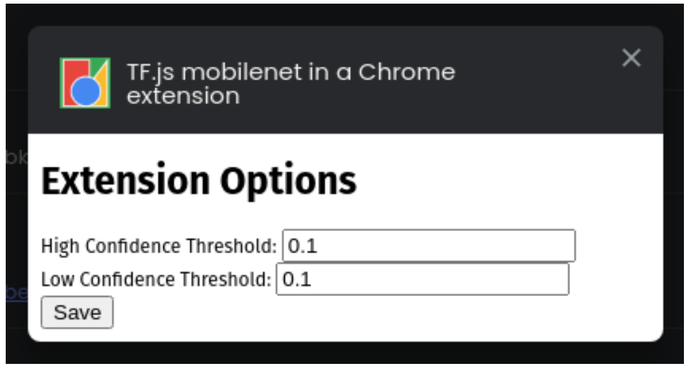
Demo 1

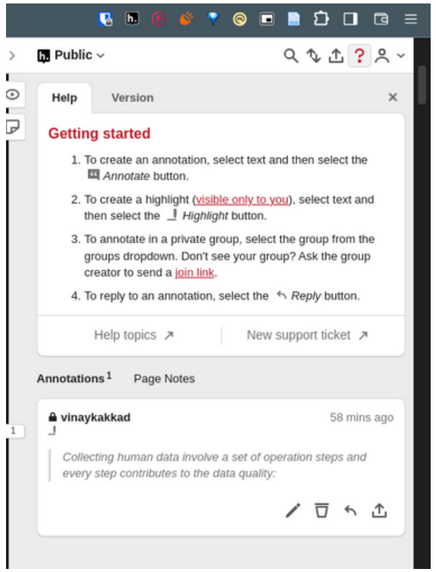
Background Script / Service Worker

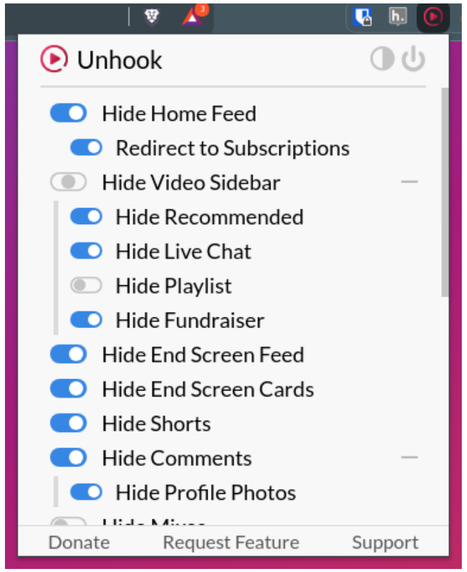
- Runs in the context of a special page called a *background page*.
- Capabilities:
  - Has access to all the *extension APIs*
  - Has access to the fetch API for making network requests
- Used for
  - listening to different browser and other events (extension event handler)
  - data synchronization and background processing
- Lifecycle:
  - Installed when the extension is installed or updated
  - Can be terminated under certain conditions:
    - Inactivity timeout: 30 seconds (reset by events or API calls)
    - Single request processing limit: 5 minutes
    - fetch() response timeout: 30 seconds

Sidebar, Popup, and Options Page

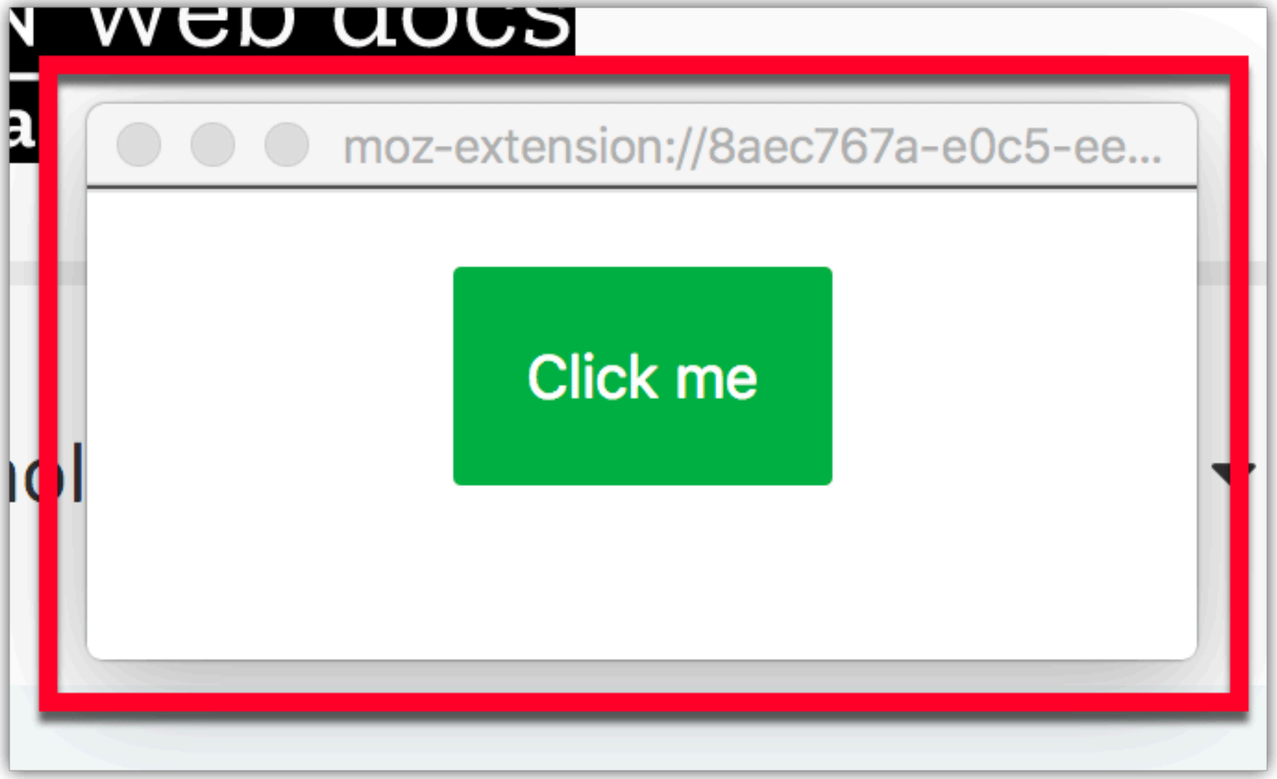
- Different UI elements supported by extensions
- For each element, a HTML file is linked through manifest
  - browser_action - popup
  - sidebar_action - sidebar
  - options_page or options_ui - options page
- The JS files for each of these has priviliges as the service worker



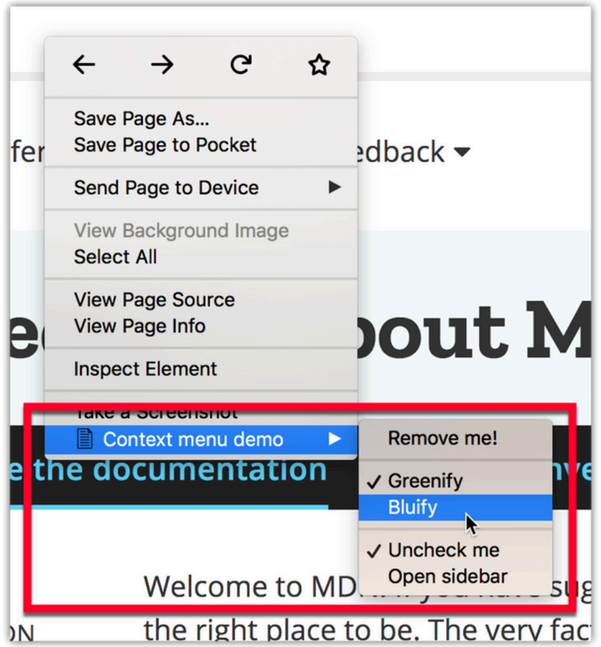




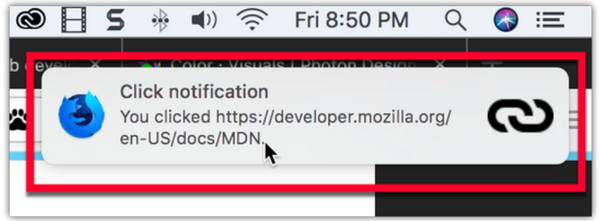
Other User Interfaces



Extension Page



Context Menu



Notifications

Message Passing

- Mechanism for communication between different parts of a Chrome extension
- Allows sending and receiving between all the discussed components
- There are different mechanisms for handling one-time communication and long-lived communication
  - One time messages
    - Content script to others -> chrome.runtime.sendMessage(message, ...)
    - From others to content script -> chrome.tabs.sendMessage(tabId, message, ...)
    - To listen for messages -> chrome.runtime.onMessage.addListener(handler)
  - Long lived messages
    - Connection: chrome.runtime.connect() & chrome.tabs.connect()
    - Messaging: port.onMessage.addListener(handler) & port.sendMessage()

Demo 2

Insights

- Separation of concern
- Keeping service worker alive
- Console messages and debugging
- Modularization, Bundling and Testing
  - https://github.com/guocaoyi/create-chrome-ext
  - https://parceljs.org/recipes/web-extension/
  - https://developer.chrome.com/docs/extensions/how-to/test/end-to-end-testing
- Model development

Resources

- Chrome Extension Docs
- Mozilla Extension Docs
- Chrome Extension Beginners Course
- Chrome Extension Development Playlist
- Github Repo

infocusp
Innovations

Thank You 🙏