Blasting Browser Security

with Extensions

AGENDA

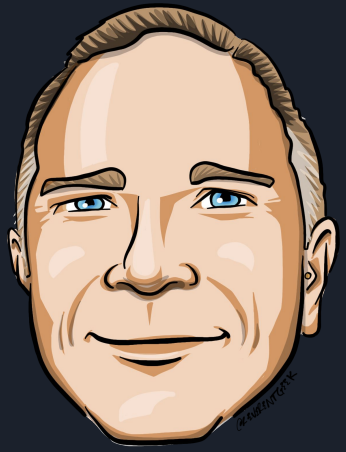
- What are Browser Extensions
- Anatomy of an Extension
- Cross-platform extensions with web-ext
- Introducing Pretty Kitty
  - Background
  - Popup
  - Content script
- Packaging
- Deploying
  - Google Chrome Web Store
  - Mozilla Add-ons Store
- Review Process?

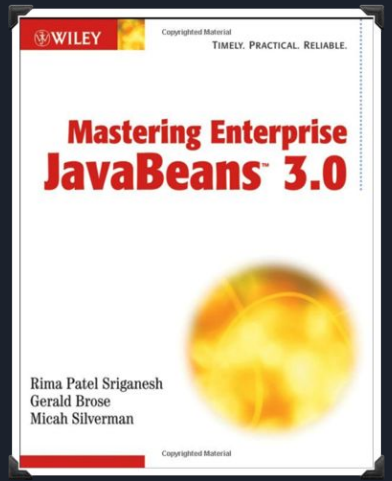
MICAH SILVERMAN

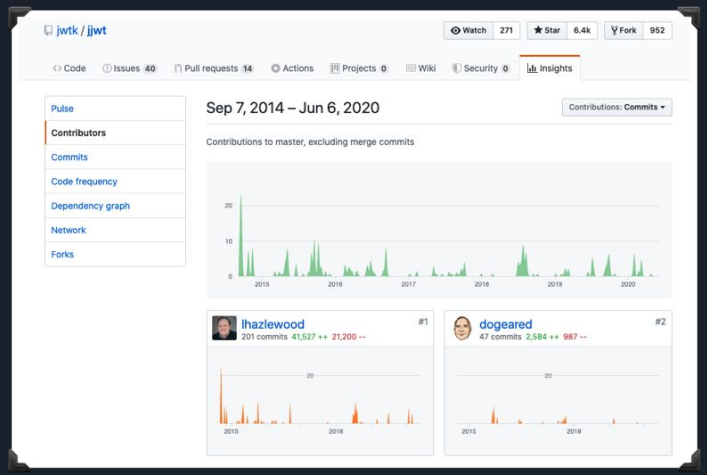
Lead Developer Advocate

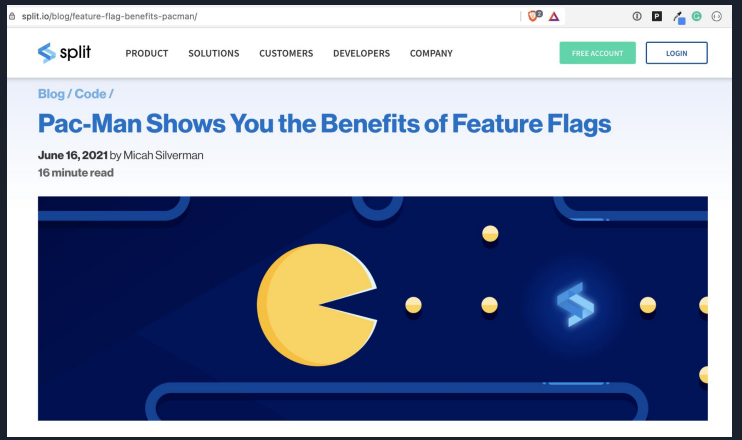
@AFITNERD

@OKTADEV









https://github.com/jwtk/jjwt

https://www.split.io/blog

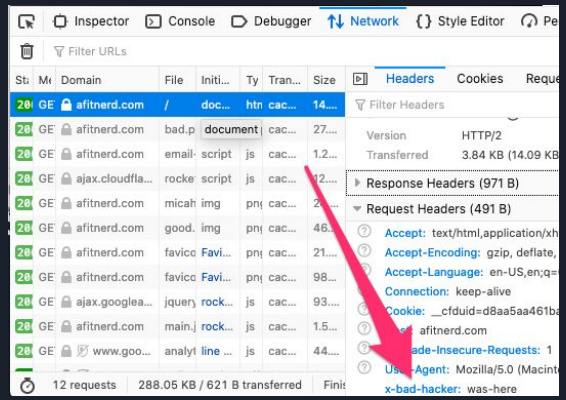
What is a Browser Extension?

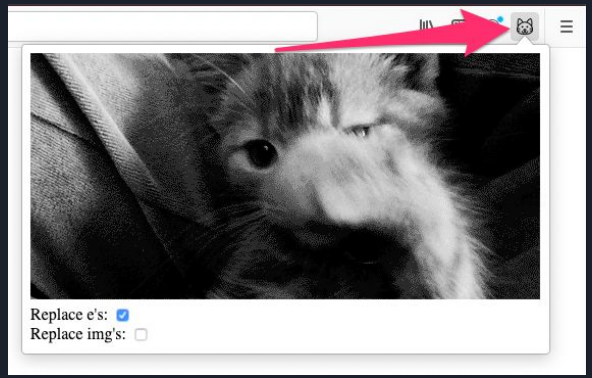
https://en.wikipedia.org/wiki/Browser_extension

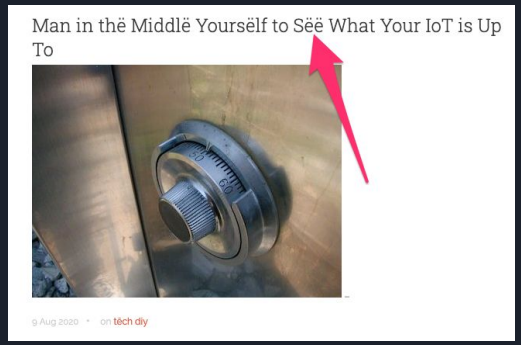
A browser extension is a small software module for customizing a web browser.

Anatomy of a Browser Extension

A **`manifest.json`** file describes:





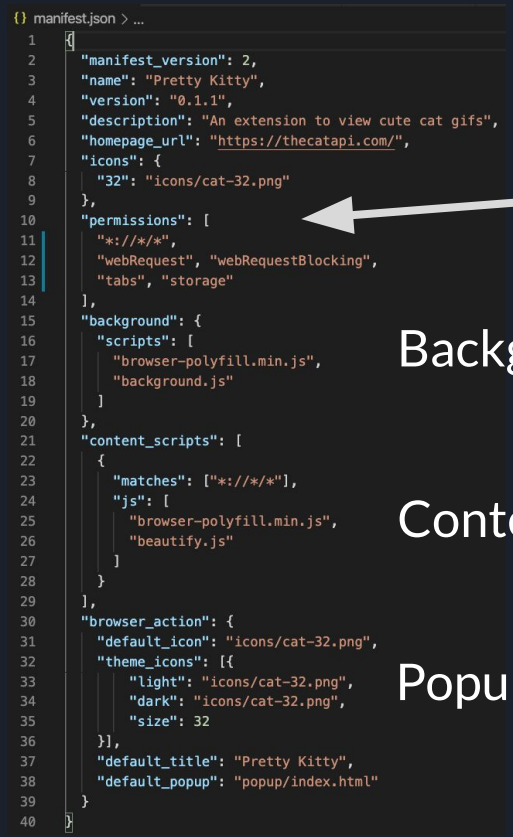


Background Scripts

Popup scripts

Content altering scripts

Anatomy of a Browser Extension



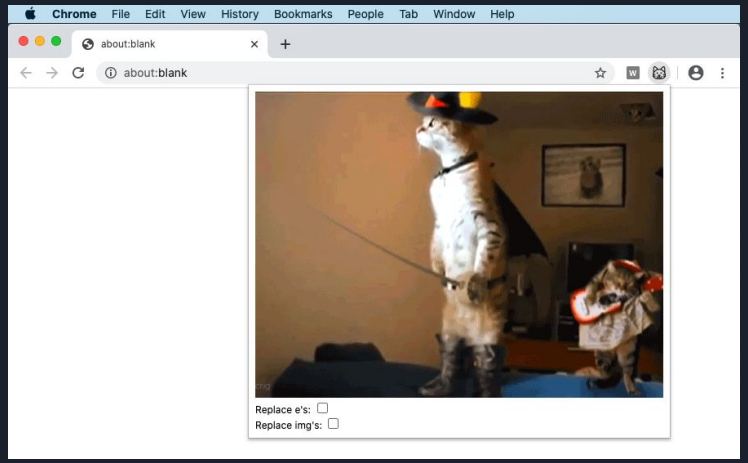
Background Scripts

Content altering scripts

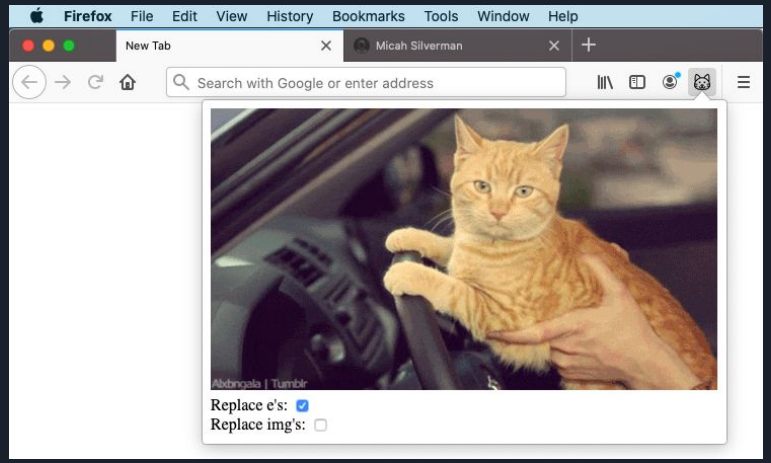
Popup scripts

History of Browser Extensions

chrome.webRequest.onBeforeRequest



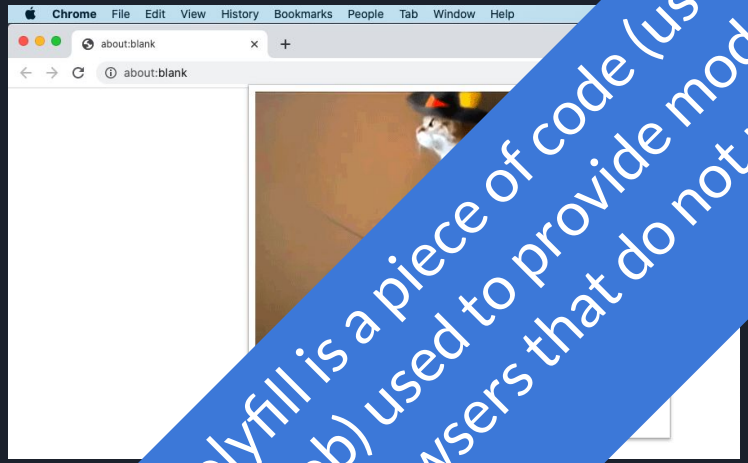
browser.webRequest.onBeforeRequest

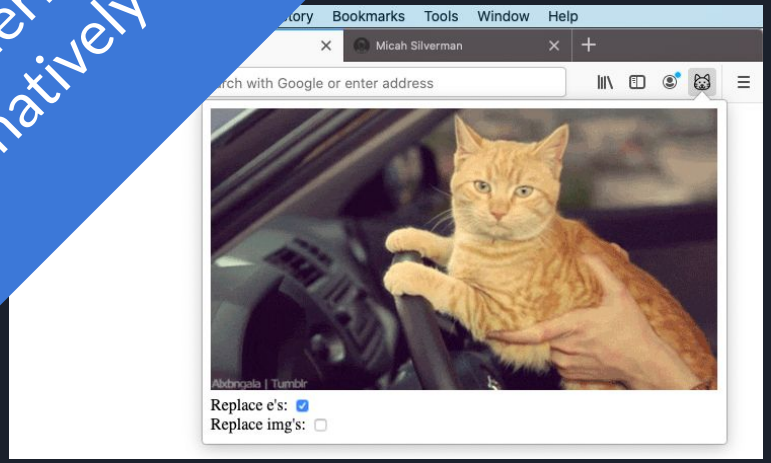


History of Browse

browser-

browser.web





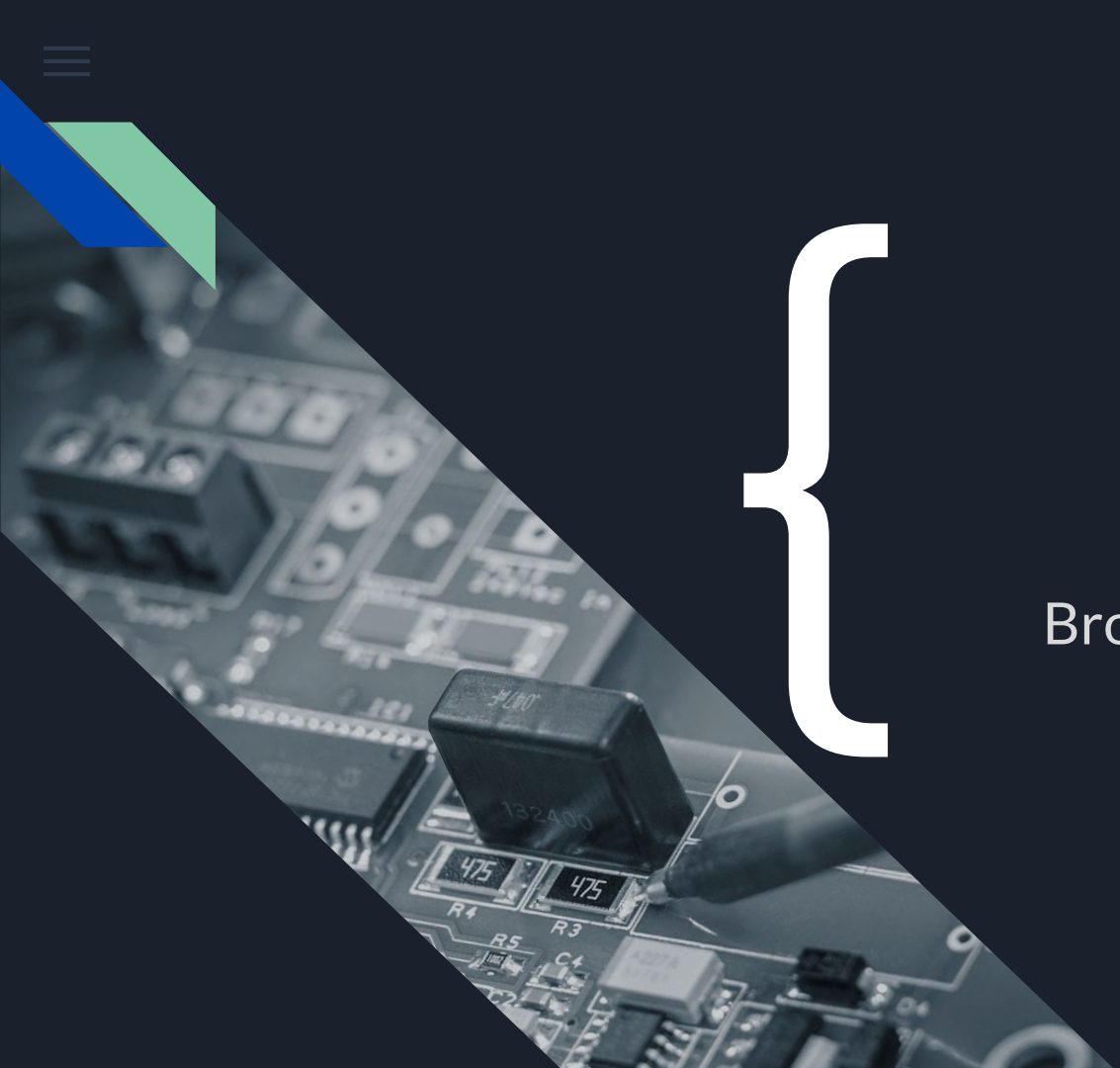
A polyfill is a piece of code (usually JavaScript on the Web) used to provide modern functionality on older browsers that do not natively support it.

web-ext FTW!

https://github.com/mozilla/web-ext

“...it aims to support browser extensions in a standard, portable, cross-platform way.”

DEVELOP TEST BUILD DEPLOY



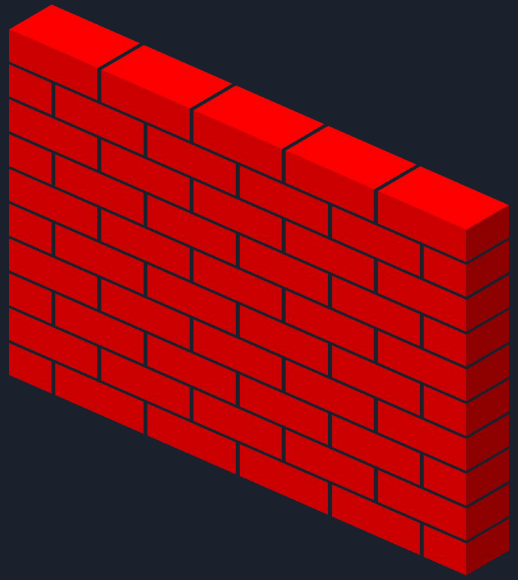
Presenting

Pretty Kitty

Browser Extension

The way it’s supposed to be

Background script
Popup script



Content script

The way it is

Background script

Content script

Popup script

Three ways to communicate

1. Programmatic Injection

popup.js:

```
document.getElementById('control').addEventListener(
    'click',
    function () {
        browser.tabs.query(
            { active: true, currentWindow: true}
        ).then(activeTabs => {
            browser.tabs.executeScript(
                activeTabs[0].id,
                { code: `// replace E’s`}
            );
        });
    });
```

Three ways to communicate

2. Message Passing

popup.js:

```
browser.tabs.sendMessage(
    activeTabs[0].id,
    { replaceEs: true, replaceImgs: false }
);
```

content.js:

```
browser.runtime.onMessage.addListener(request => {
    if(request.replaceEs) {
        // replace all e’s
    }
});
```

Three ways to communicate

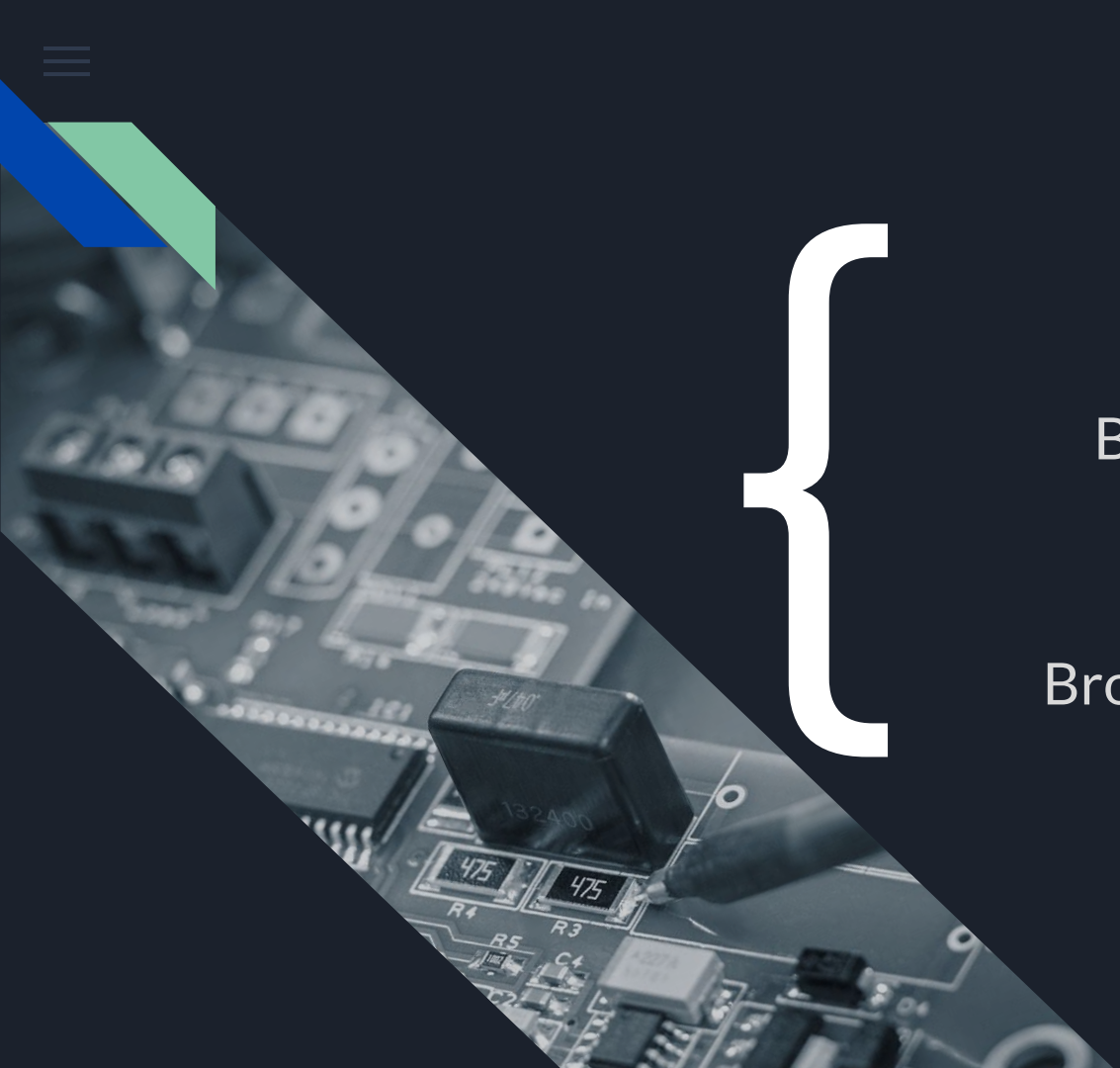
3. Local Storage

popup.js:

```
browser.storage.local.set(
    {'pretty.kitty.settings': { replaceEs: true }}
)
    .then(() => console.log("OK"));
```

content.js:

```
browser.storage.local.get('pretty.kitty.settings')
    .then(item => {
        if (item['pretty.kitty.settings'].replaceEs) {
          // replace E’s
        }
    })
```

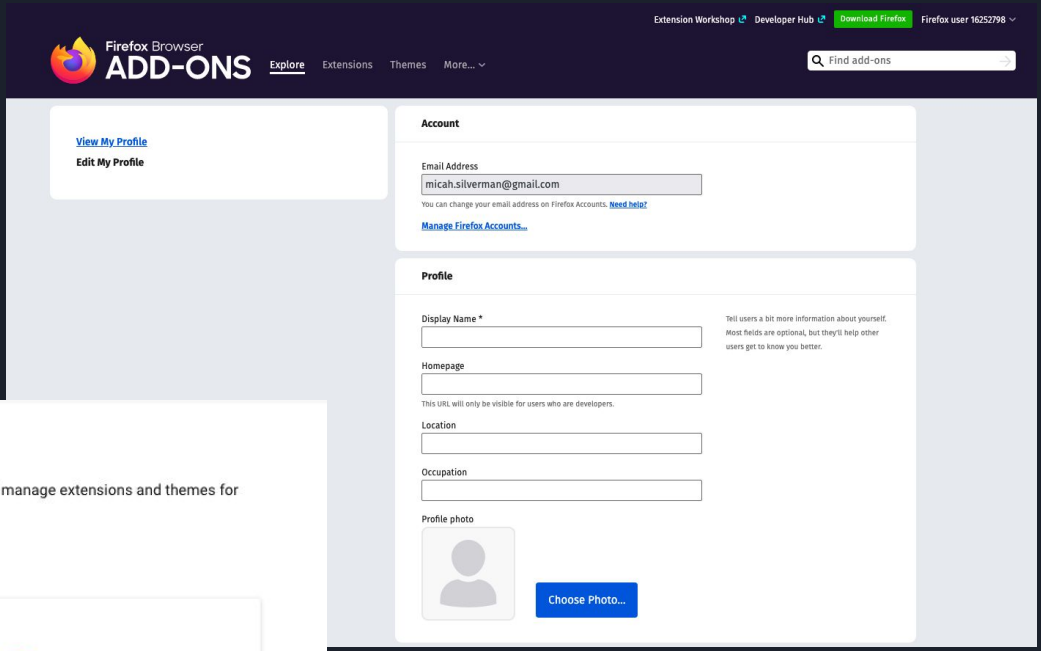


Presenting

Build & Deploy

Pretty Kitty

Browser Extension



Register as a Chrome Web Store Developer

Welcome to the Chrome Web Store Developer Dashboard. Once registered, you'll be able to distribute, monetize, and manage extensions and themes for the Google Chrome browser. Learn more

Get started:

Accept Developer Agreement and Privacy Policies

- Google Chrome Web Store Developer Agreement
- Google Privacy Policy
- Chrome Web Store Developer Program Policies

☑ **I have read and agree to the Developer Agreement and Policies**

Pay the $5 registration fee

A one-time registration fee is required in order to register your account.

Pay registration fee

Impact-driven development requires data

Only Split's unified platform closes The Impact Gap

Split Feature Delivery Platform

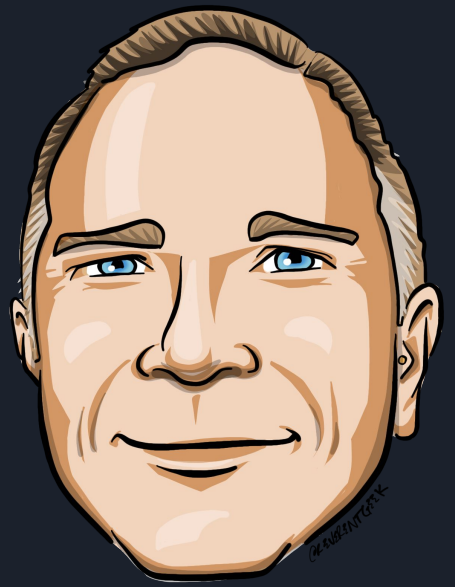
Manage
feature flags

Monitor
release performance

Experiment
for business impact

Feature flags
control user access and
record 'who sees what'

Performance & behavioral data
from engineering, product,
and business sources

Thank You!

MICAH SILVERMAN
Lead Developer Advocate
@AFITNERD
@SPLITSOFTWARE