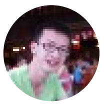
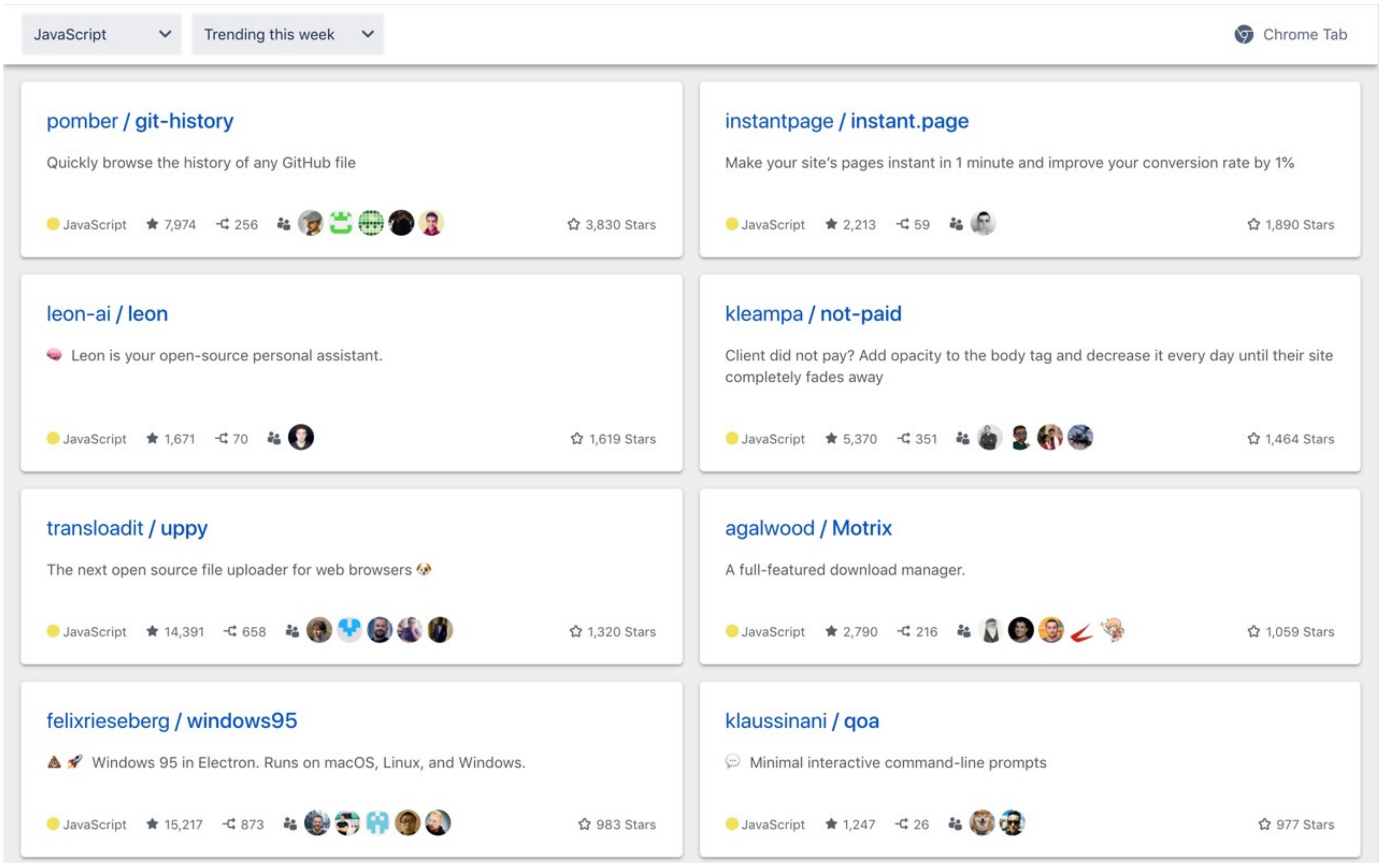
# How to use React.js to create a cross-browser extension in 5 minutes



Hu Chen
Feb 20 · 6 min read



JavaScript | Trending this week | Chrome Tab

**pomber / git-history**
Quickly browse the history of any GitHub file
JavaScript ★ 7,974 ⑂ 256 — ☆ 3,830 Stars

**instantpage / instant.page**
Make your site's pages instant in 1 minute and improve your conversion rate by 1%
JavaScript ★ 2,213 ⑂ 59 — ☆ 1,890 Stars

**leon-ai / leon**
Leon is your open-source personal assistant.
JavaScript ★ 1,671 ⑂ 70 — ☆ 1,619 Stars

**kleampa / not-paid**
Client did not pay? Add opacity to the body tag and decrease it every day until their site completely fades away
JavaScript ★ 5,370 ⑂ 351 — ☆ 1,464 Stars

**transloadit / uppy**
The next open source file uploader for web browsers
JavaScript ★ 14,391 ⑂ 658 — ☆ 1,320 Stars

**agalwood / Motrix**
A full-featured download manager.
JavaScript ★ 2,790 ⑂ 216 — ☆ 1,059 Stars

**felixrieseberg / windows95**
Windows 95 in Electron. Runs on macOS, Linux, and Windows.
JavaScript ★ 15,217 ⑂ 873 — ☆ 983 Stars

**klaussinani / qoa**
Minimal interactive command-line prompts
JavaScript ★ 1,247 ⑂ 26 — ☆ 977 Stars

Hacker Tab: Chrome extension to view GitHub trending projects on new tab 📈

> ***Update**: this blog post was originally targeted for Chrome extension, I have recently ported to Firefox and would like to share what I have learned here as well* ✨.

## Also Read:

Part 2: How to use background script to fetch data in Chrome extension

Recently I have tried to create a Chrome extension that **replaces new tab** screen with GitHub trending projects, and it got trending in Product Hunt. I built it using create-react-app and I would like to share how I did it, in 5 minutes.

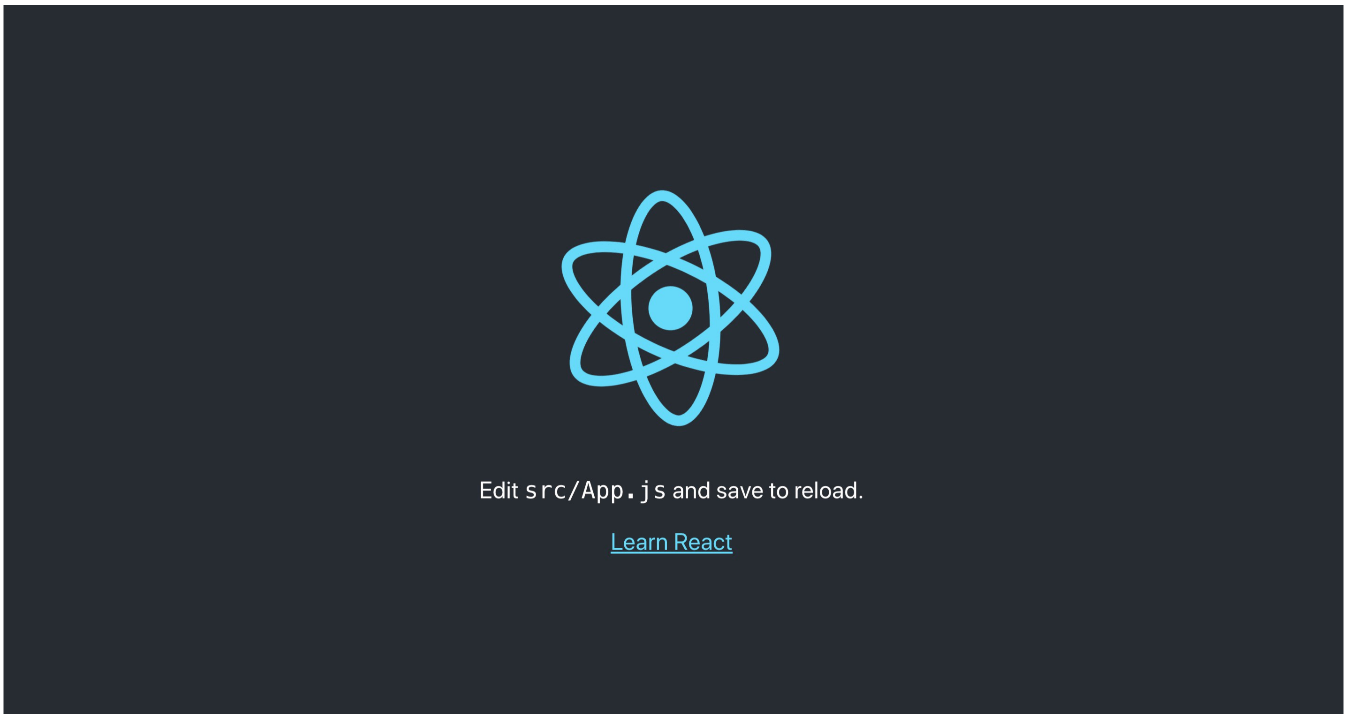
## 1. Create the application

The Chrome extension I created replaces new tab screen to a custom page which I built with **create-react-app**. It does not matter which framework you are using — React.js, Vue.js, or vanilla HTML with CSS and JS, as long as you could serve the HTML file in the browser, it will work as a Chrome extension.

In my project, I used create-react-app to create, run, and bundle application. I also used Redux for state management and styled-components for styling.

```
$ npx create-react-app my-awesome-extension
$ cd my-awesome-extension
$ npm start
```

You should be able to see the default `creact-react-app` launching page at http://localhost:3000


Edit `src/App.js` and save to reload.

Launching page of create-react-app

This tutorial won't cover how to develop a webpage using React.js, there are tons of great resources out there. I will focus on how to turn the app into a Chrome extension.

Now heads down and code your application.

## 2. Prepare for Chrome extension

Once you have a working version locally, there are a few other things you will need to config when preparing your extension.

### Create `manifest.json`

There was a `manifest.json` file in the `public` folder which is for PWA purposes or configuring our extension. You will need to replace it with content below, which will be copied over to the `build` folder during `build` process, and Chrome will use it as the configuration file of the extension.

```json
{
  "manifest_version": 2,
  "name": "Hacker Tab",
  "author": "Hu Chen",
  "version": "1.0.1",
  "description": "Replace new tab screen with GitHub trending projects.",
  "icons": {
    "16": "16.png",
    "48": "48.png",
    "128": "128.png"
  },
  "chrome_url_overrides": {
    "newtab": "index.html"
  },
  "permissions": ["storage"]
}
```

manifest.json hosted with ♡ by GitHub view raw

The above snippet is what I have used in my extension. You can find other config options as well.

- `version`: the version of your extension, you should update this when you plan to release a new version.
- `icons`: you need different sizes of Icons to be used in different cases (e.g. Chrome Web Store), normally you will only need `16x16`, `48x48` and `128x128`.
- `chrome_url_overrides`: as we are overriding chrome new tab to a custom HTML page, we need to replace `newtab` to `index.html`

- `permissions` : include the permissions you need in your extension. I am using `chrome.storage` API to store user preferred programming languages so I have added `storage` permission. In general practice, you should only add permissions required in the application.

## INLINE_RUNTIME_CHUNK=false

By default, Create React App embeds a small runtime script into `index.html` during the production build, this is to reduce the number of HTTP requests. But unfortunately, you will see console errors related to CSP. You could turn off the embedding behavior by setting the `INLINE_RUNTIME_CHUNK` flag to `false` .

**change in `package.json` :**

```
"build": "INLINE_RUNTIME_CHUNK=false react-scripts build"
```

## Switch to Chrome default tab

Sometimes people would like to switch back to the default Chrome tab to access top sites. We could use `chrome.tabs` API.

```
<Button
  onClick={() => {
    chrome.tabs.getCurrent(tab => {
      chrome.tabs.update(tab.id, {
        url: 'chrome-search://local-ntp/local-ntp.html',
      });
    });
  }}>
  Chrome Tab
</Button>
```

> *The URL is not working for Firefox, I have used `react-useragent` to only show this button for Chrome. Check code here.*

## Delete favicon

Create React App comes with a default `favicon.ico` in the public folder. For Chrome New Tab, we do not want to have any favicons for the new tab, so we remove it from public folder and also remove the link in `index.html` .

## Adding launching placeholder

Because React.js initially loads an empty page before the application is rendered in Javascript, from my experience there is a noticeable flash of blank page before everything gets rendered. This is not a great experience for the user.

There are a few ways to resolve this, one way is to pre-render into static HTML files using react-snapshot or react-snap.

This method does not suit the use case of my extension because it also pre-renders the data from the server, causing initial HTML to always contain outdated repositories before new data comes in.

At the time of writing, I have not figured out how to resolve this issue if using snapshot libraries, so I come with a simpler solution.

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta
      name="viewport"
      content="width=device-width, initial-scale=1, shrink-to-fit=no"
    />
    <title>New Tab</title>
    <style>
      body {
        background-color: rgb(238, 238, 238);
      }

      .logo-container {
        position: fixed;
        top: 0;
        left: 0;
        bottom: 0;
        right: 0;
        display: flex;
        justify-content: center;
        align-items: center;
      }

      .logo {
        height: 250px;
        width: 250px;
        background-image: url('512.png');
```

```
          background-repeat: no-repeat;
          background-size: 100%;
        }
      </style>
    </head>

    <body>
      <noscript> You need to enable JavaScript to run this app. </noscript>
      <div id="root">
        <div class="logo-container">
          <div class="logo"></div>
        </div>
      </div>
    </body>
  </html>
```

index.html hosted with ♡ by GitHub view raw

I have styled `index.html` so that it displays a grey logo at the center of the page before the application gets rendered, and it works quite nicely.

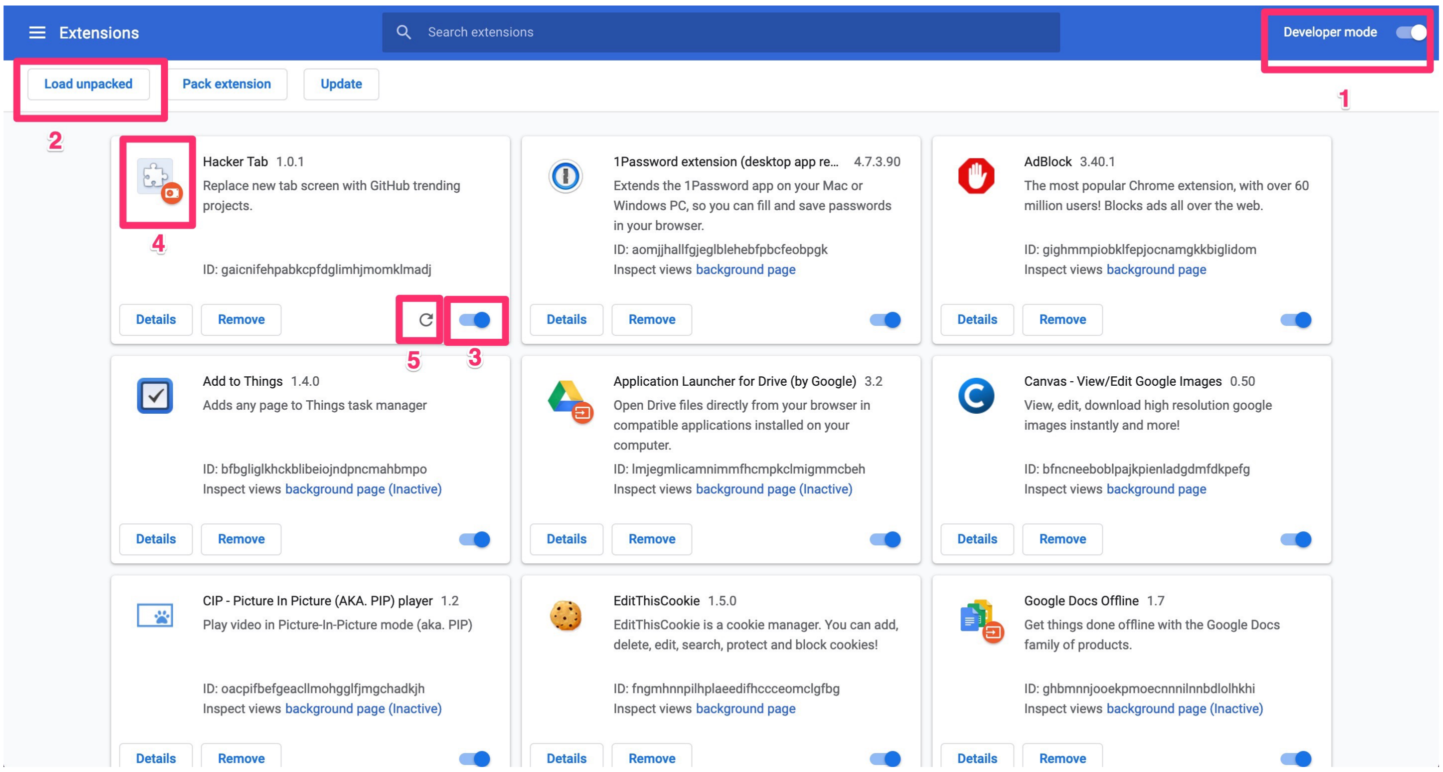
Launching Page

Once Javascript finished rendering, `root` will be replaced with actual content.

## 3. Testing locally for Chrome

Before publishing to Chrome Web Store, it is always good to test locally.

All you need to do is to run `npm run build` to put everything required into `build` folder, then launch `chrome://extensions/` in Google Chrome.



Hacker Tab 1.0.1
Replace new tab screen with GitHub trending projects.
ID: gaicnifehpabkcpfdglimhjmomklmadj

1Password extension (desktop app re... 4.7.3.90
Extends the 1Password app on your Mac or Windows PC, so you can fill and save passwords in your browser.
ID: aomjjhallfgjeglblehebfpbcfeobpgk
Inspect views background page

AdBlock 3.40.1
The most popular Chrome extension, with over 60 million users! Blocks ads all over the web.
ID: gighmmpiobklfepjocnamgkkbiglidom
Inspect views background page

Add to Things 1.4.0
Adds any page to Things task manager
ID: bfbgliglkhckblibeiojndpncmahbmpo
Inspect views background page (Inactive)

Application Launcher for Drive (by Google) 3.2
Open Drive files directly from your browser in compatible applications installed on your computer.
ID: lmjegmlicamnimmfhcmpkclmigmmcbeh
Inspect views background page (Inactive)

Canvas - View/Edit Google Images 0.50
View, edit, download high resolution google images instantly and more!
ID: bfncneeboblpajkpienladgdmfdkpefg
Inspect views background page

CIP - Picture In Picture (AKA. PIP) player 1.2
Play video in Picture-In-Picture mode (aka. PIP)
ID: oacpifbefgeacllmohgglfjmgchadkjh
Inspect views background page (Inactive)

EditThisCookie 1.5.0
EditThisCookie is a cookie manager. You can add, delete, edit, search, protect and block cookies!
ID: fngmhnnpilhplaeedifhccceomclgfbg
Inspect views background page

Google Docs Offline 1.7
Get things done offline with the Google Docs family of products.
ID: ghbmnnjooekpmoecnnnilnnbdlolhkhi
Inspect views background page (Inactive)

Test extension locally

1. Make sure developer mode is on as shown on the image above (1).
2. Click “Load unpacked” and target “build” folder in your project, it should contain `index.html` , `manifest.json`, different sizes of logos, and bundled javascript files (2).
3. Make sure other extensions like Momentum which also modify new tab is disabled (3).
4. If you have already published a version in production and you are testing a newer version, you should also disable the production version. (The local version will have a red icon at bottom right corner as in the picture (4))

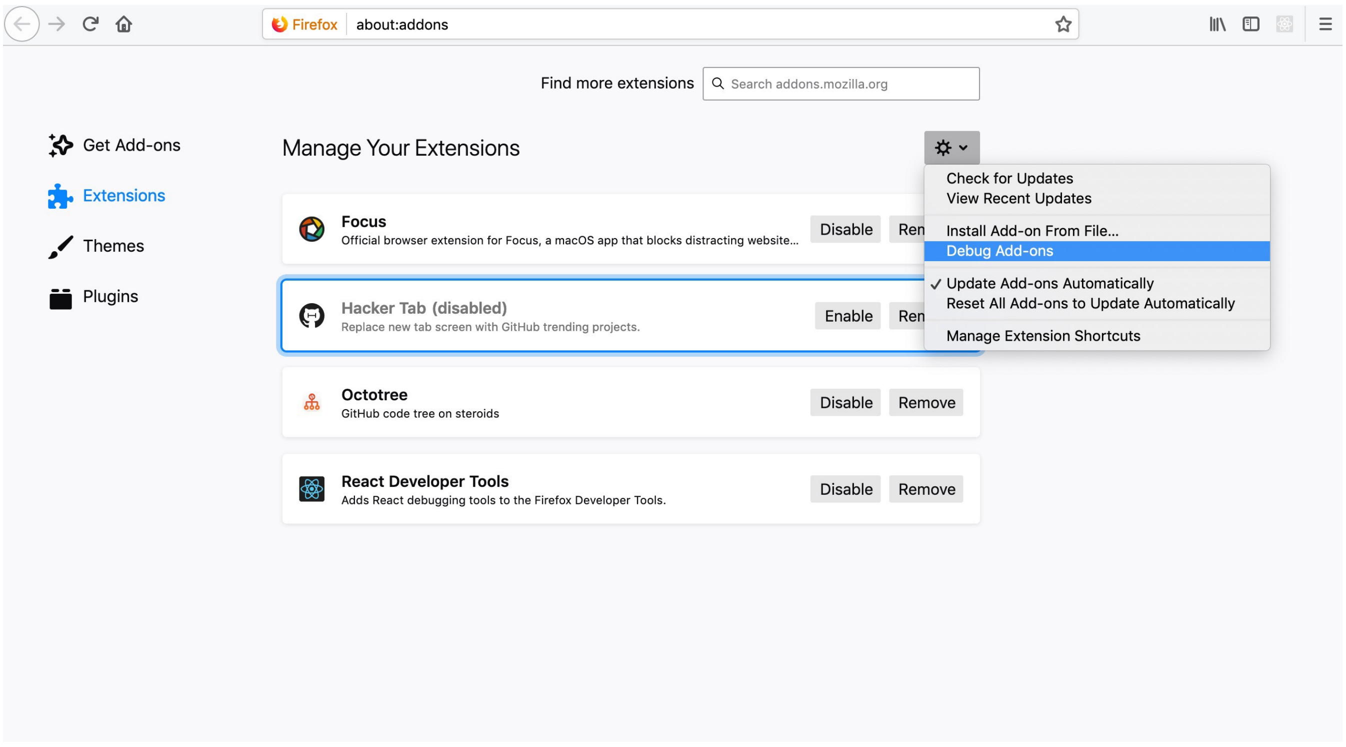
Opening a new tab should allow you to test your shiny new extensions!

To update the extension, you will need to change the code, run `npm run build` again, and click the “Refresh” button (5).

## 4. Testing locally on Firefox

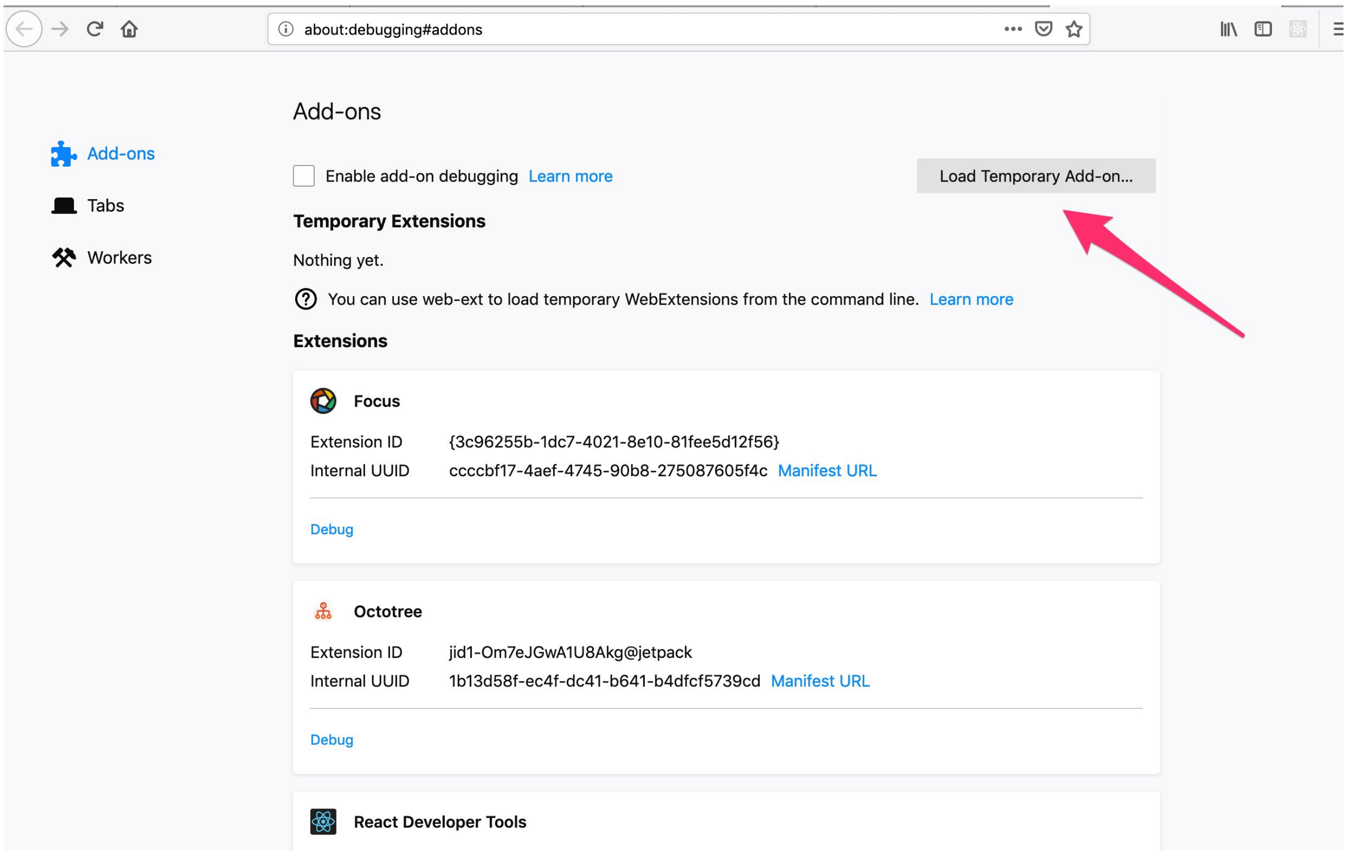
Luckily the code we have just done works out of the box on Firefox, testing on Firefox is similar as well.

Launch `about:addons` in Firefox browser and turn on “debug addons”.


Firefox about:addons
Find more extensions
Search addons.mozilla.org
Get Add-ons
Extensions
Themes
Plugins
Manage Your Extensions
Check for Updates
View Recent Updates
Install Add-on From File...
Debug Add-ons
Update Add-ons Automatically
Reset All Add-ons to Update Automatically
Manage Extension Shortcuts
Focus
Official browser extension for Focus, a macOS app that blocks distracting website...
Hacker Tab (disabled)
Replace new tab screen with GitHub trending projects.
Octotree
GitHub code tree on steroids
React Developer Tools
Adds React debugging tools to the Firefox Developer Tools.
Disable
Enable
Remove

Turn on ”Debug Addons”

You could see interface has changed slightly, now click “Load Temporary Add-on”



Add-ons

Enable add-on debugging

**Temporary Extensions**

Nothing yet.

You can use web-ext to load temporary WebExtensions from the command line.

**Extensions**

**Focus**

| | |
|---|---|
| Extension ID | {3c96255b-1dc7-4021-8e10-81fee5d12f56} |
| Internal UUID | ccccbf17-4aef-4745-90b8-275087605f4c |

**Octotree**

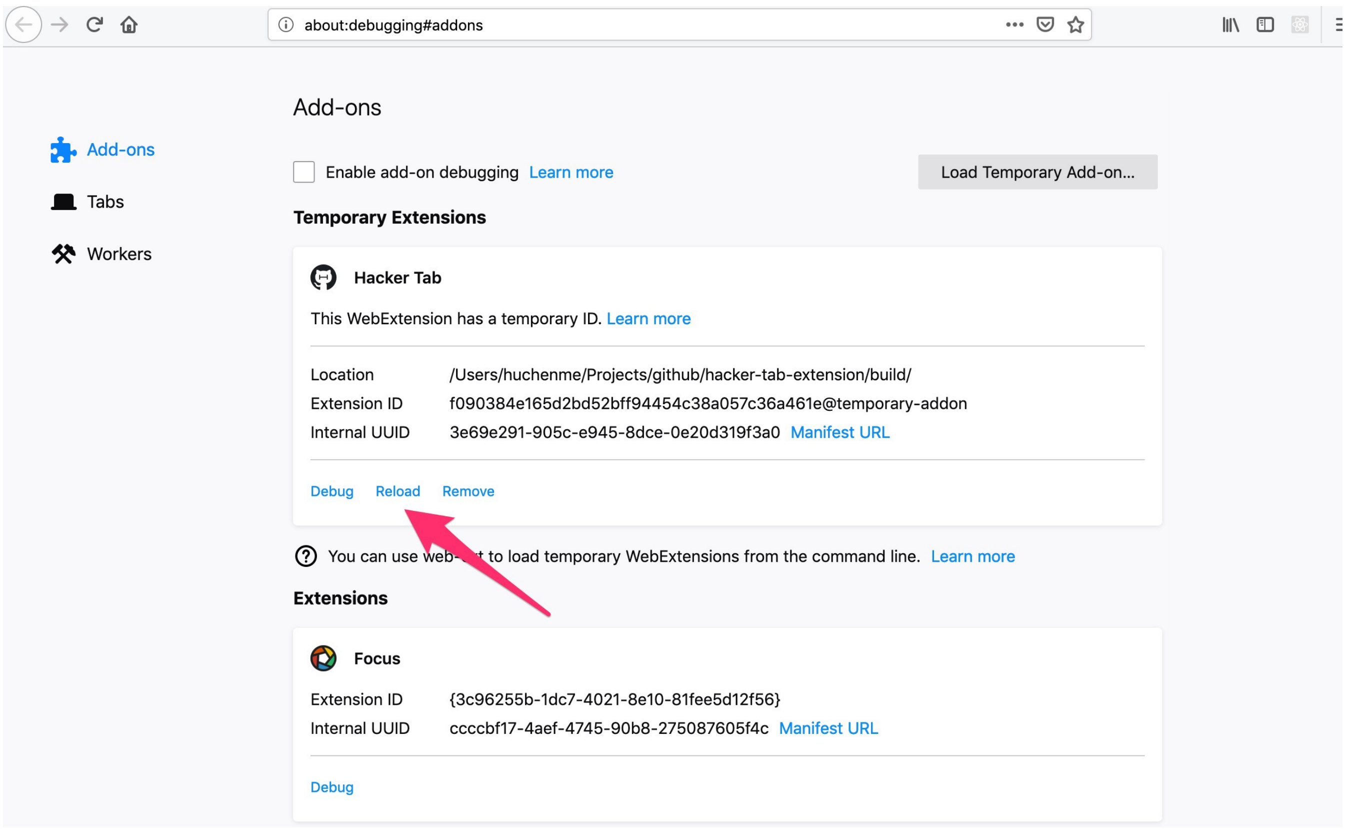
| | |
|---|---|
| Extension ID | jid1-Om7eJGwA1U8Akg@jetpack |
| Internal UUID | 1b13d58f-ec4f-dc41-b641-b4dfcf5739cd |

**React Developer Tools**

Reading manifest: Error processing background.persistent: Event pages are not currently supported. This will

Click any file in "build" folder, you could just choose "manifest.json".

Now you could see your extension in the list as well, open a new tab and test everything is ok, if you would like to make a change, run `npm run build` again and click "Reload" button.


about:debugging#addons

Add-ons

Add-ons
Tabs
Workers

Enable add-on debugging Learn more

Load Temporary Add-on...

**Temporary Extensions**

Hacker Tab

This WebExtension has a temporary ID. Learn more

| | |
|---|---|
| Location | /Users/huchenme/Projects/github/hacker-tab-extension/build/ |
| Extension ID | f090384e165d2bd52bff94454c38a057c36a461e@temporary-addon |
| Internal UUID | 3e69e291-905c-e945-8dce-0e20d319f3a0 Manifest URL |

Debug Reload Remove

You can use web-ext to load temporary WebExtensions from the command line. Learn more

**Extensions**

Focus

| | |
|---|---|
| Extension ID | {3c96255b-1dc7-4021-8e10-81fee5d12f56} |
| Internal UUID | ccccbf17-4aef-4745-90b8-275087605f4c Manifest URL |

Debug

## That's it!

Now you have your extension ready, just follow this official documentation and publish your extension to Chrome Web Store or Firefox Add-ons Hub.

### Also Read:

Part 2: How to use background script to fetch data in Chrome extension

. . .

Thank you for reading this far. You could check the source code in Github or download the extension in Chrome Web Store and Firefox Add-ons Hub, happy hacking, and let me know what you have built!

. . .

**Learn React - Best React Tutorials (2019) | gitconnected**

The top 49 React tutorials - learn React for free. Courses are submitted and voted on by...

gitconnected.com

JavaScript Chrome Extension Github React Reactjs