

Chrome Extension Basics

王福强

KEEVOL

王福强

杭州福强科技有限公司创始人
20+年互联网与金融技术研发与管理经验,
原阿里巴巴大数据中间件canal的产品与技术奠基人,
腾讯云最具价值专家(TVP),
《Spring揭秘》、《SpringBoot揭秘》以及多本自出版电子书作者。
- 原多家创业公司与成长型公司CTO
- 原挖财技术VP及首席架构师(Chief Architect)
- 原天猫产品技术部资深架构师
- 原阿里巴巴平台技术部海量数据部门高级技术专家

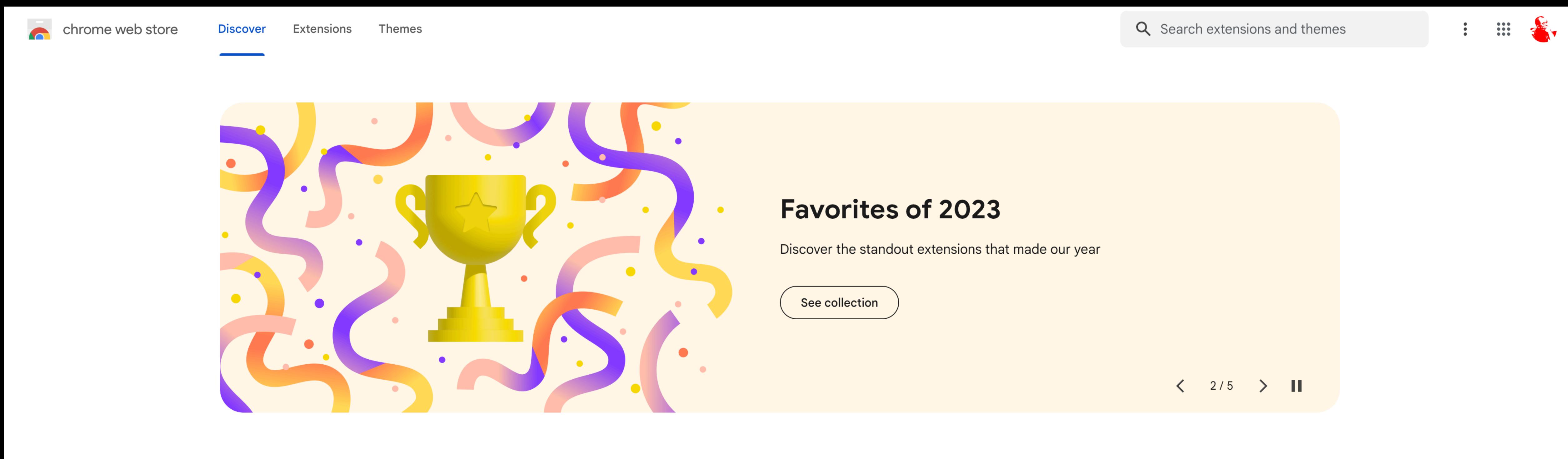
杭州福强科技有限公司专注于技术、人才与组织成长,
主要为企业用户提供技术战略、组织管理、数字/数智化转型
与互联网产品和技术顾问/咨询服务。
行业上，互联网和金融领域涉猎居多，
主导过包括信贷，理财，债券，外汇(保证金)交易等
多种金融系统的设计与研发。
除了金融行业客户，亦服务过房地产、人力资源、医疗服务、
平台SaaS等多个行业的企业客户。

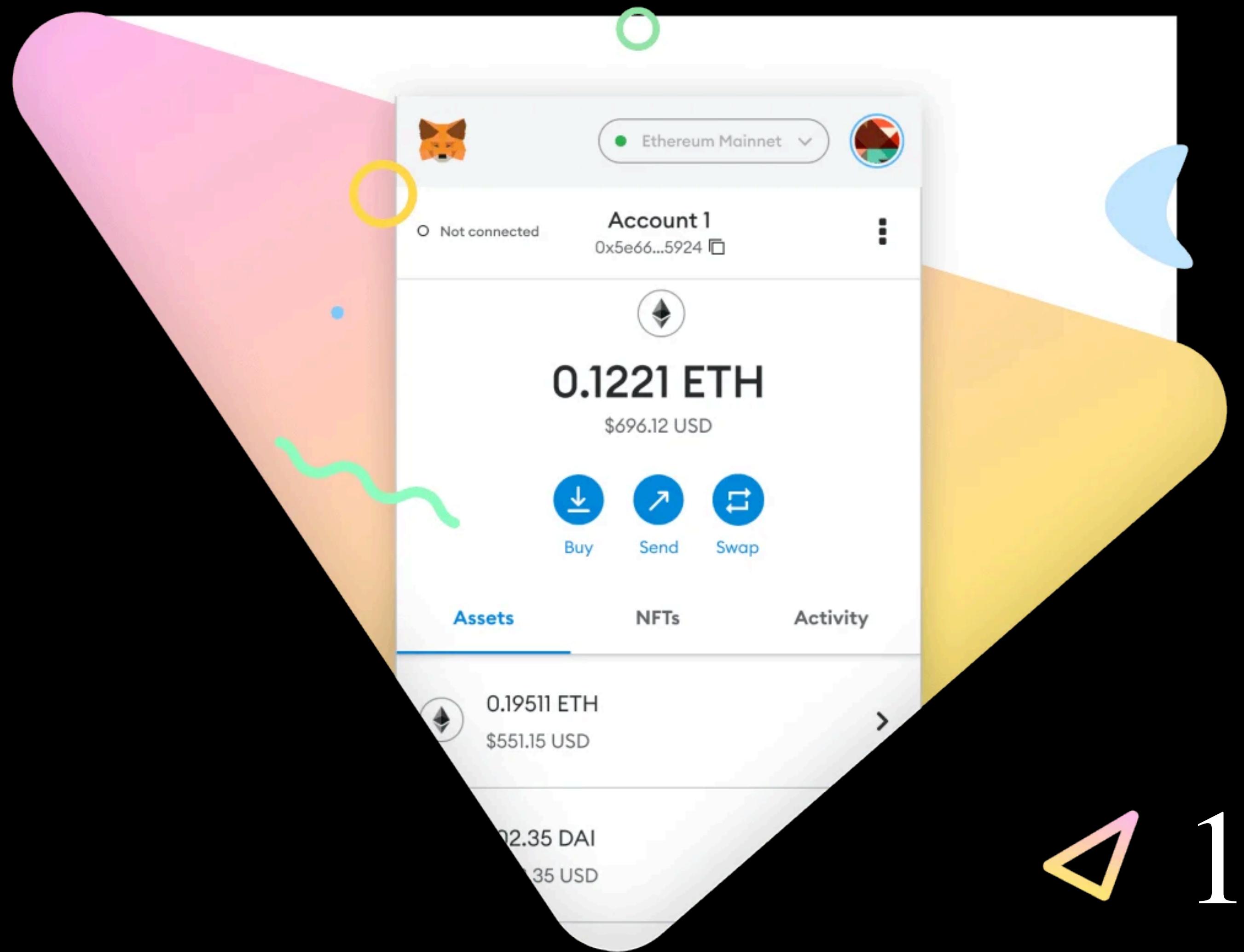


Famous Extensions

Where to get

Chrome Web Store - <https://chromewebstore.google.com/>

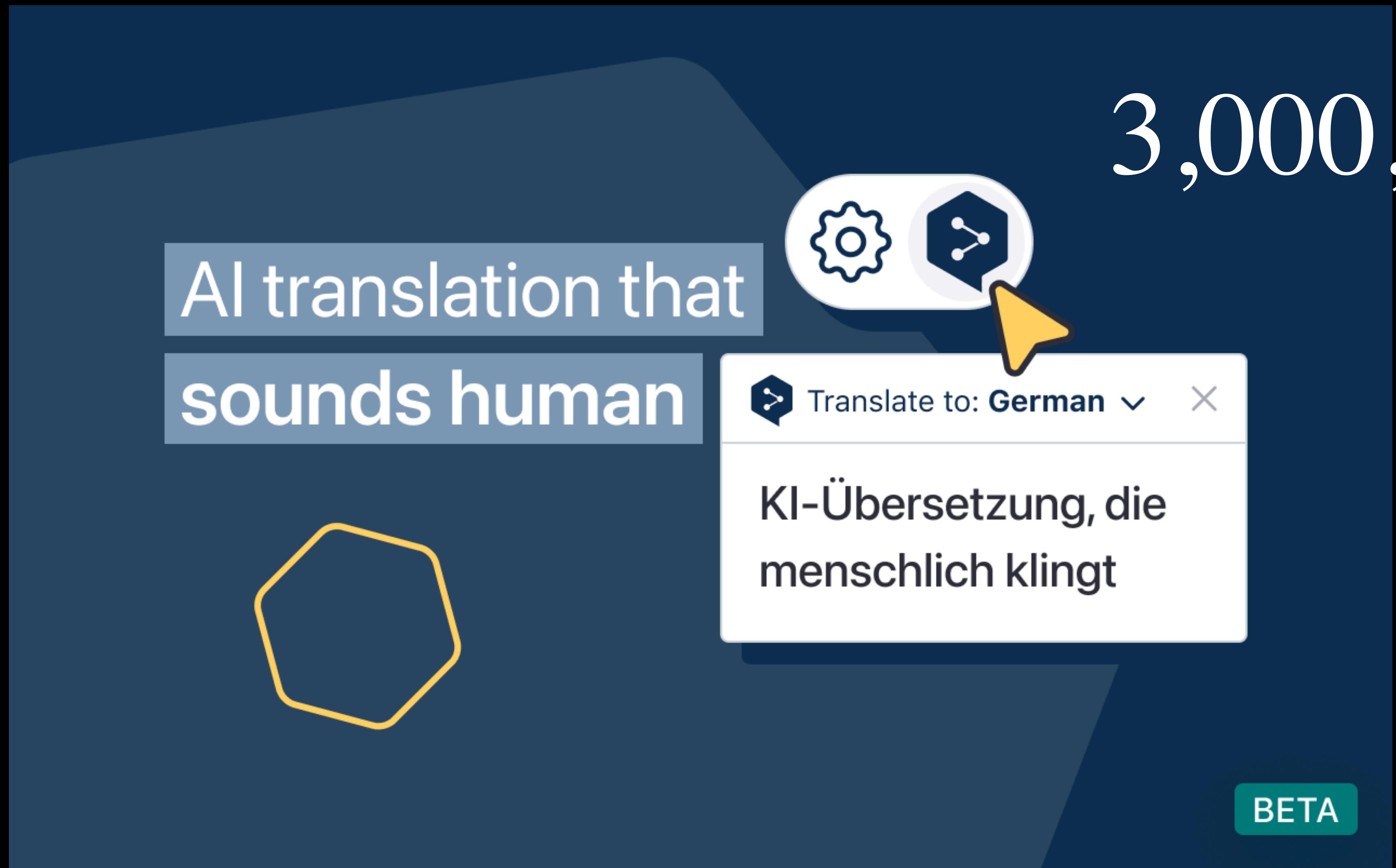




MetaMask

▷ 15,000,000 users

DeepL Translate: Reading & writing translator



The image shows a screenshot of the DeepL Translate mobile application. At the top left, there is a blue banner with white text that reads "AI translation that sounds human". Above the banner, there is a large yellow arrow pointing upwards towards the app's interface. The main part of the screen displays a white card with a gear icon and a neural network icon. A yellow arrow points from the banner towards this icon. Below the icons, there is a dropdown menu with the text "Translate to: German" and a close button. Underneath the dropdown, the German translation of the English phrase is shown: "KI-Übersetzung, die menschlich klingt". In the bottom right corner of the card, there is a small green button labeled "BETA".

3,000,000 users

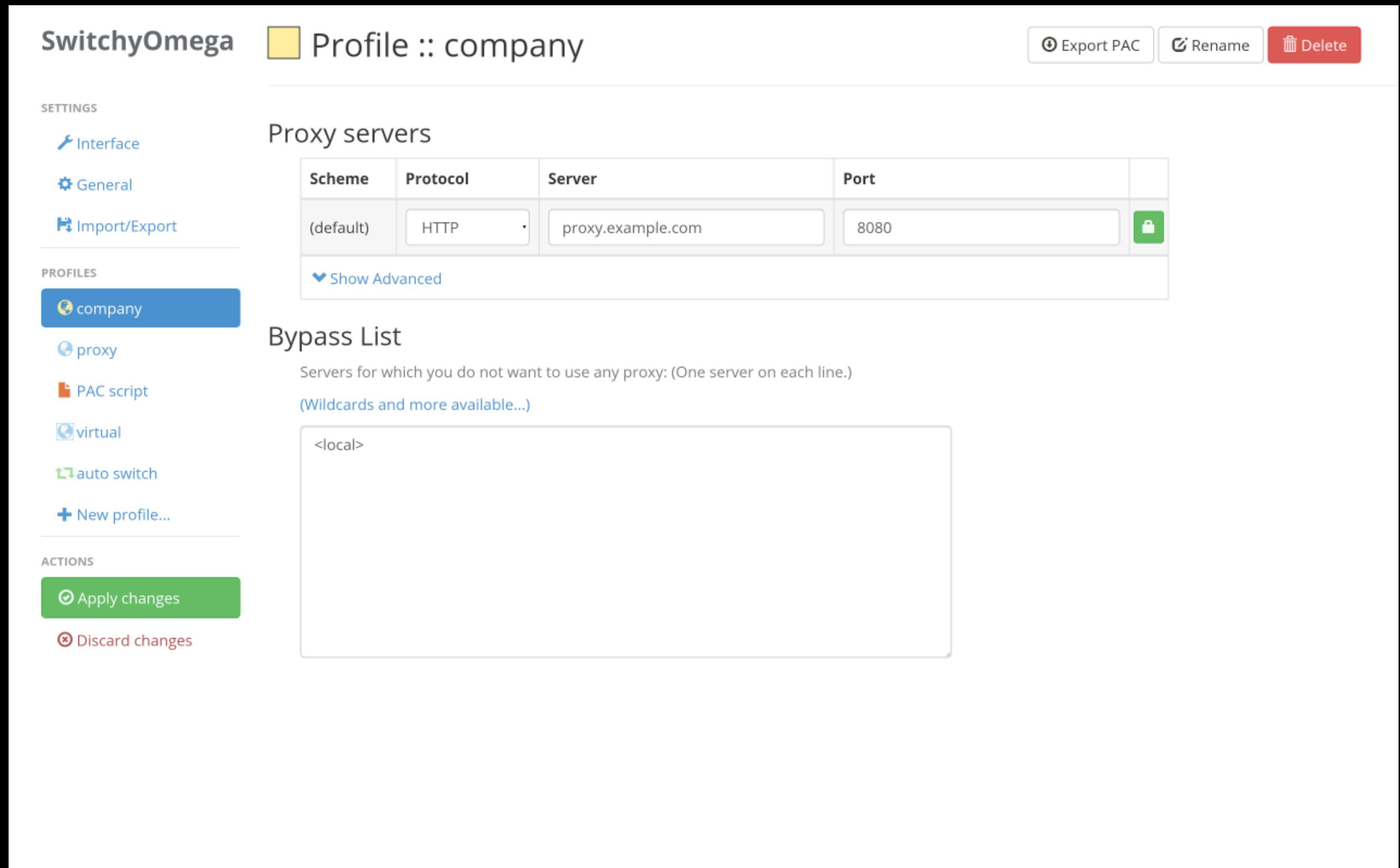
AI translation that sounds human

Translate to: German

KI-Übersetzung, die menschlich klingt

BETA

Proxy SwitchyOmega



The screenshot shows the SwitchyOmega proxy configuration interface. The main title is "SwitchyOmega" and the profile is "company". The interface includes a sidebar with "SETTINGS" (Interface, General, Import/Export), "PROFILES" (company, proxy, PAC script, virtual, auto switch, New profile...), and "ACTIONS" (Apply changes, Discard changes). The main content area shows the "Proxy servers" section with a table:

Schema	Protocol	Server	Port
(default)	HTTP	proxy.example.com	8080

Below the table is a "Bypass List" section with the placeholder text "<local>".

2,000,000 users

**What Chrome Extensions
can contribute...**

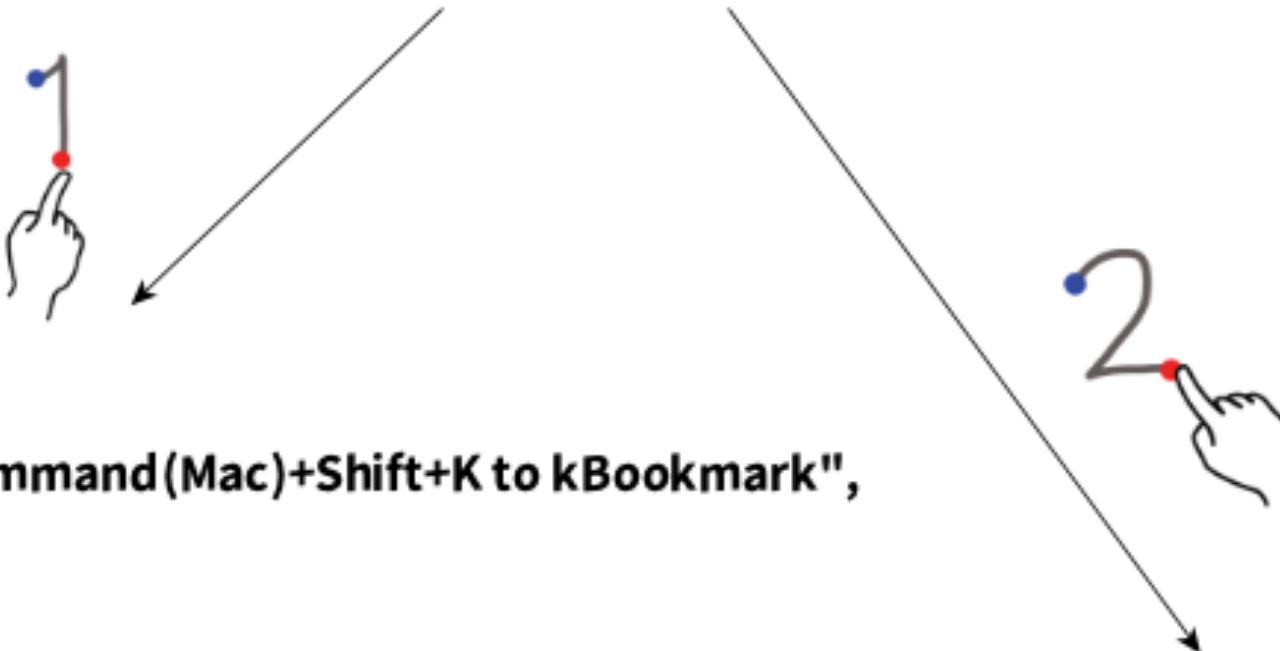
Action Button & Action Popups

处理chrome extension的action有两种方式:

一种是指定popup.html, 这样, 每次用户点击action按钮的时候就会显示一个popup;
另一种就是不配置popup.html, 而是在background.js里通过自定义action listener来处理action事件;



```
"action": {  
  "default_title": "Press Ctrl(Win)/Command(Mac)+Shift+K to kBookmark",  
  "default_icon": "icon.png"  
},
```



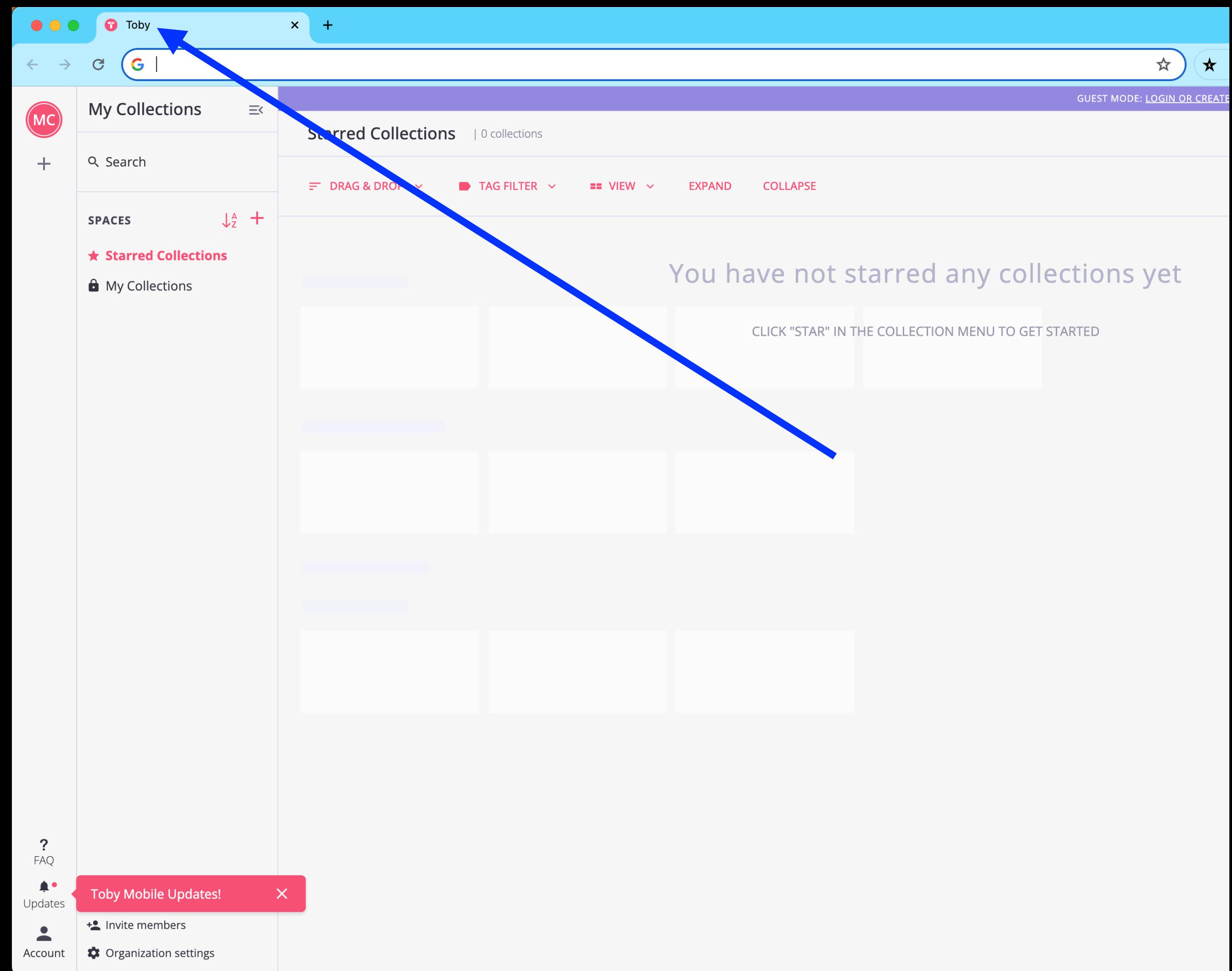
1. in manifest.json

```
"action": {  
  "default_title": "Press Ctrl(Win)/Command(Mac)+Shift+K to kBookmark",  
  "default_icon": "icon.png",  
  "default_popup": "popup.html"  
},
```

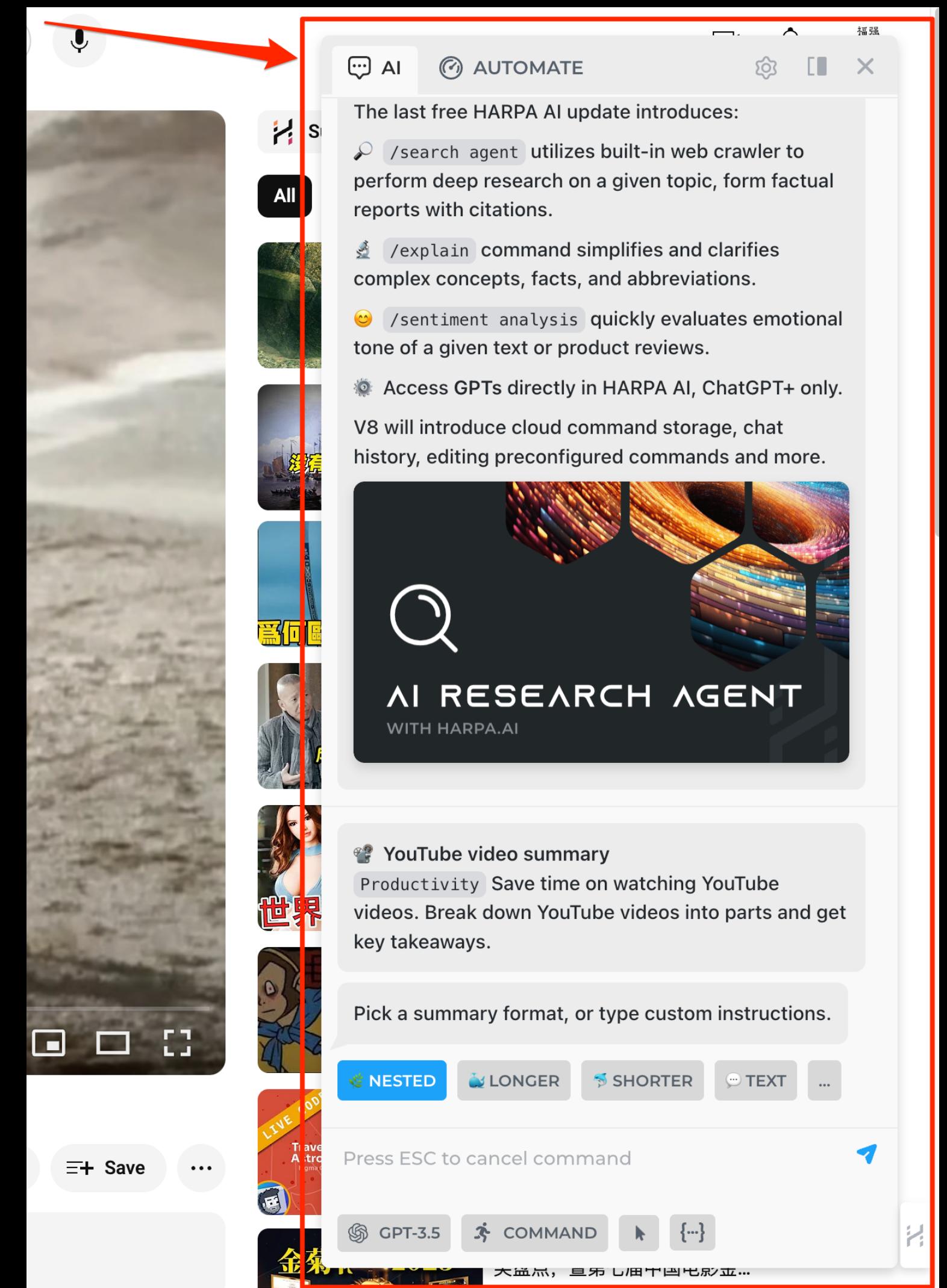
2. in background.js

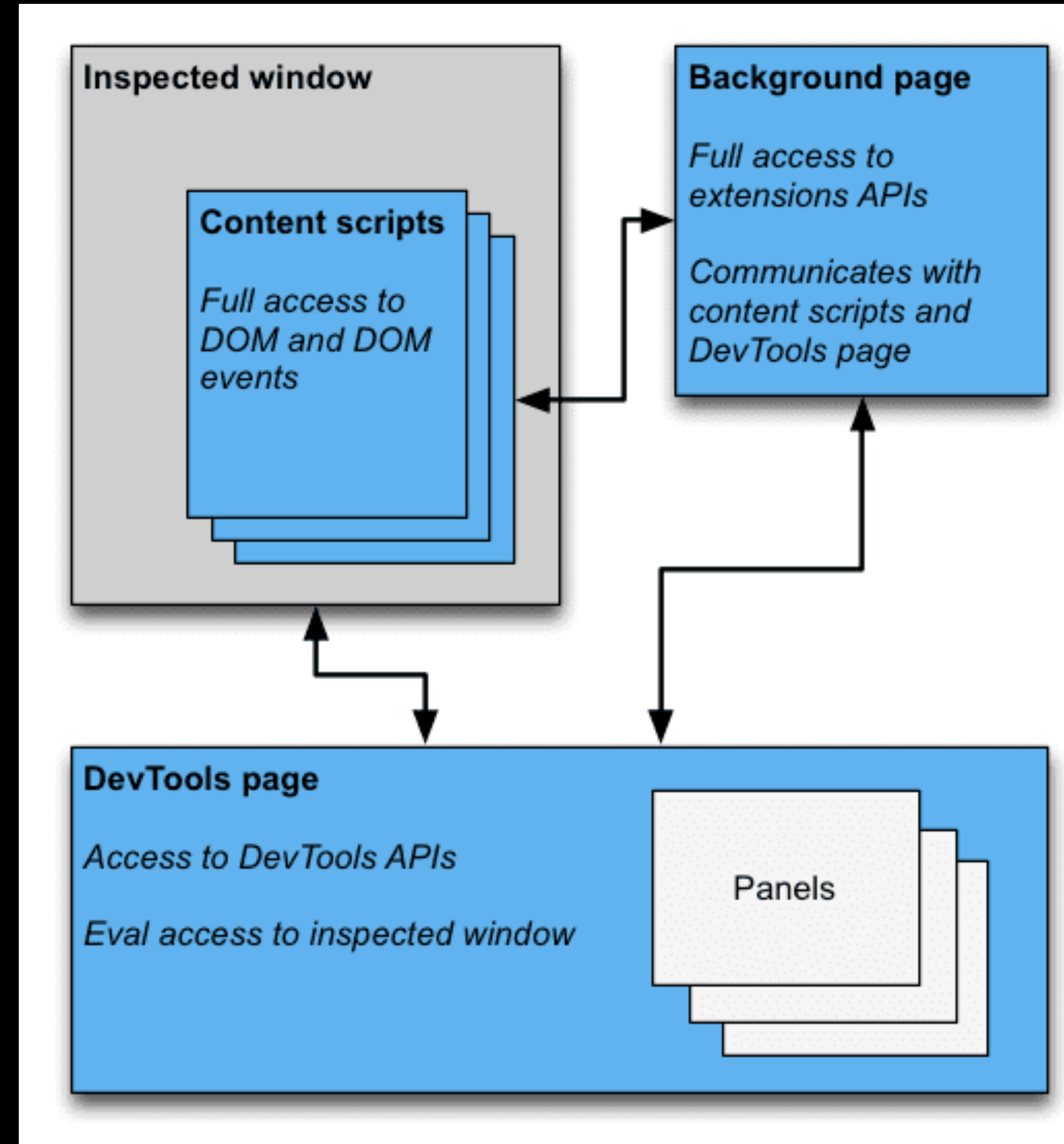
```
chrome.action.onClicked.addListener(function(tab) {  
  chrome.action.setTitle({tabId: tab.id, title: "You are on tab:" + tab.id});  
  // or open a new window: https://stackoverflow.com/questions/5345435/pop-up-window-center-screen  
  var w = 440;  
  var h = 220;  
  var left = (screen.width/2)-(w/2);  
  var top = (screen.height/2)-(h/2);  
  
  chrome.windows.create({url: 'redirect.html', type: 'popup', width: w, height: h, left: left, top: top}, function(window) {  
    //  
  });  
});
```

New Tab Page Customize...



Side bar...





Enhance DevTools

Common Components in a Chrome Extension

当前页面

Popup页面

1. 点击action按钮
弹出action页面
也叫popup...

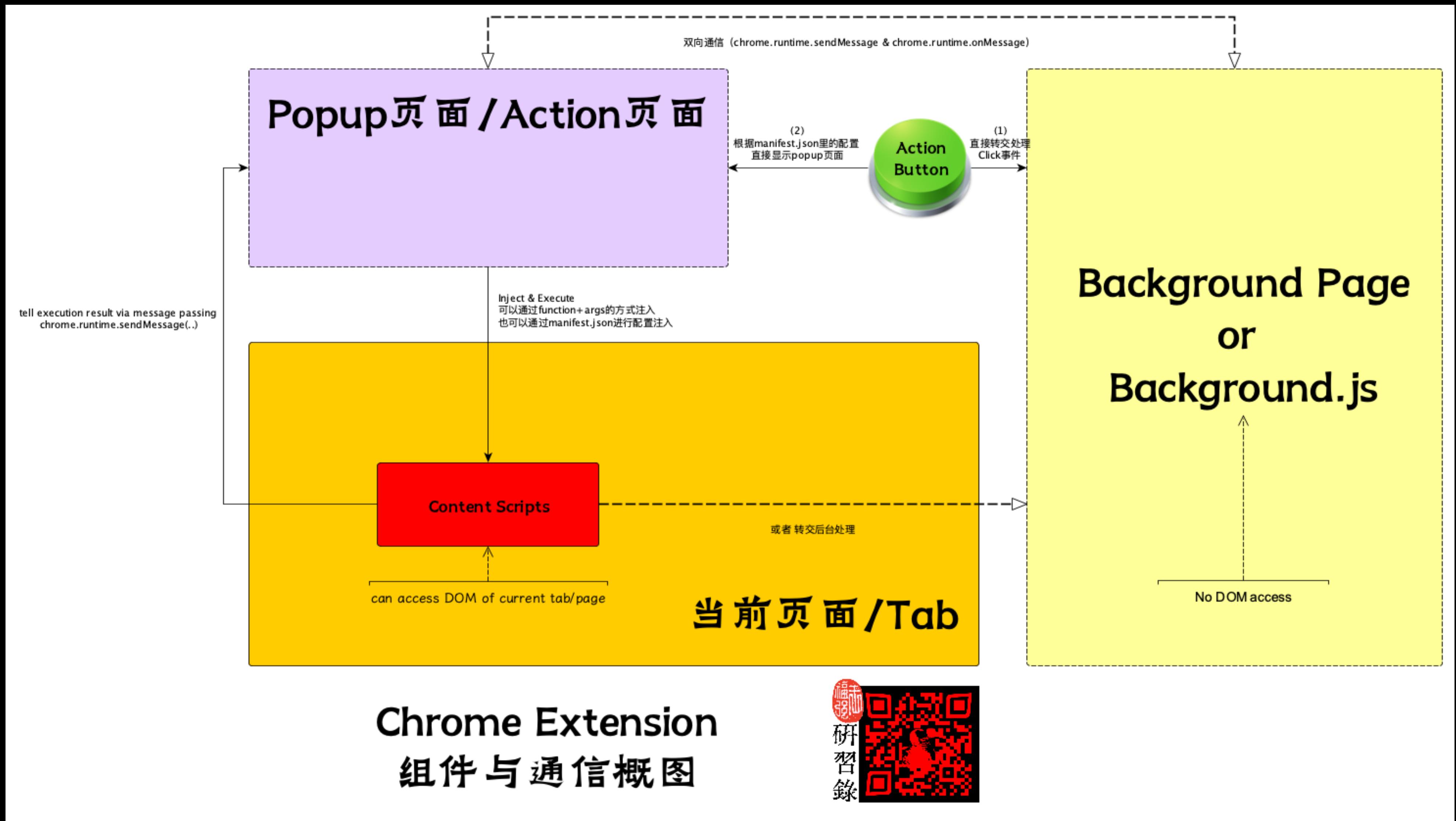
2. 点击popup页面里的按钮
触发相应动作，比如像当前
页面注入脚本执行（当前页
面是x/twitter页面）

3. 当前页面内执行完注入的
content scripts或者scripting function之后
如果还想要回传执行结果
可以通过chrome.runtime.sendMessage方法

4. 通过chrome.runtime.onMessage接收消息
然后将消息结果回填到想要回填到位置

Message Passing in Chrome Extension(s)

Intra-extension message passing



one-time requests

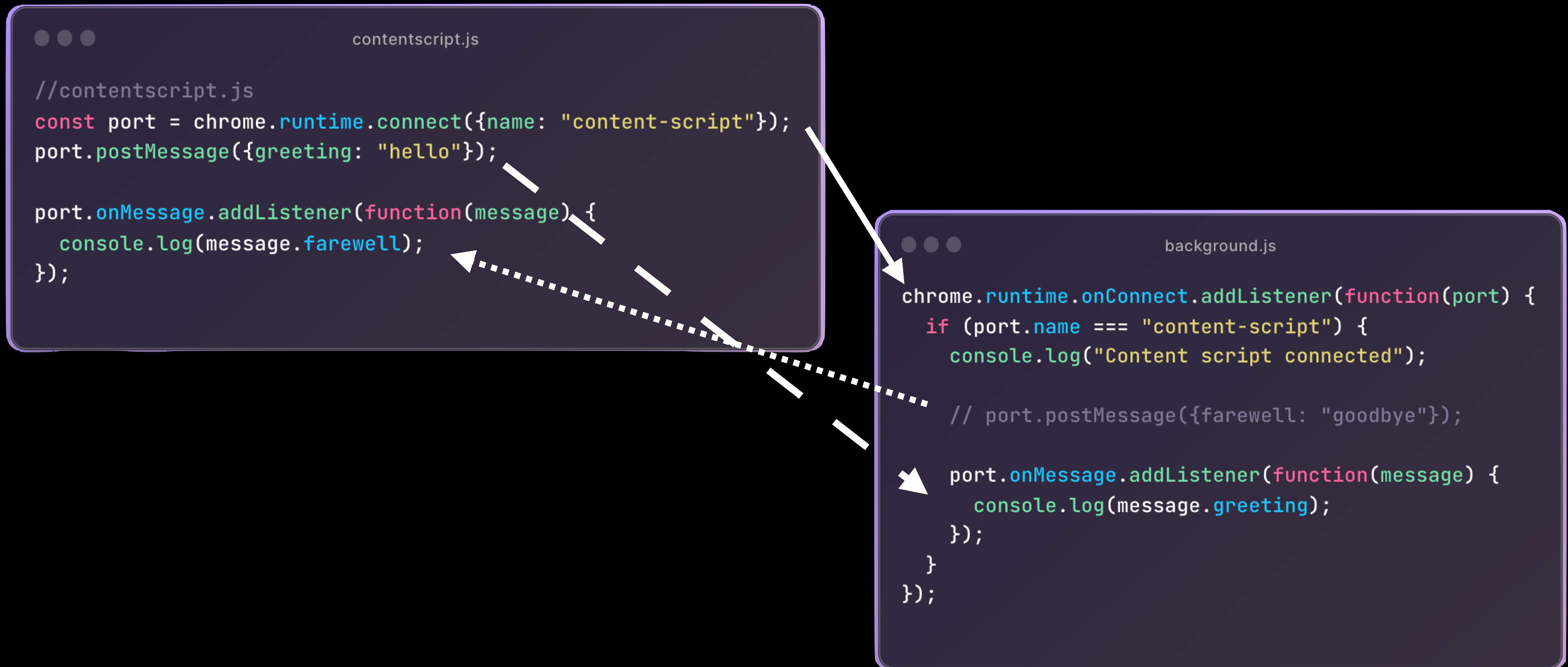
```
contentscript.js

chrome.runtime.sendMessage({greeting: "hello"}, function(response) {
    console.log(response.farewell);
});
```

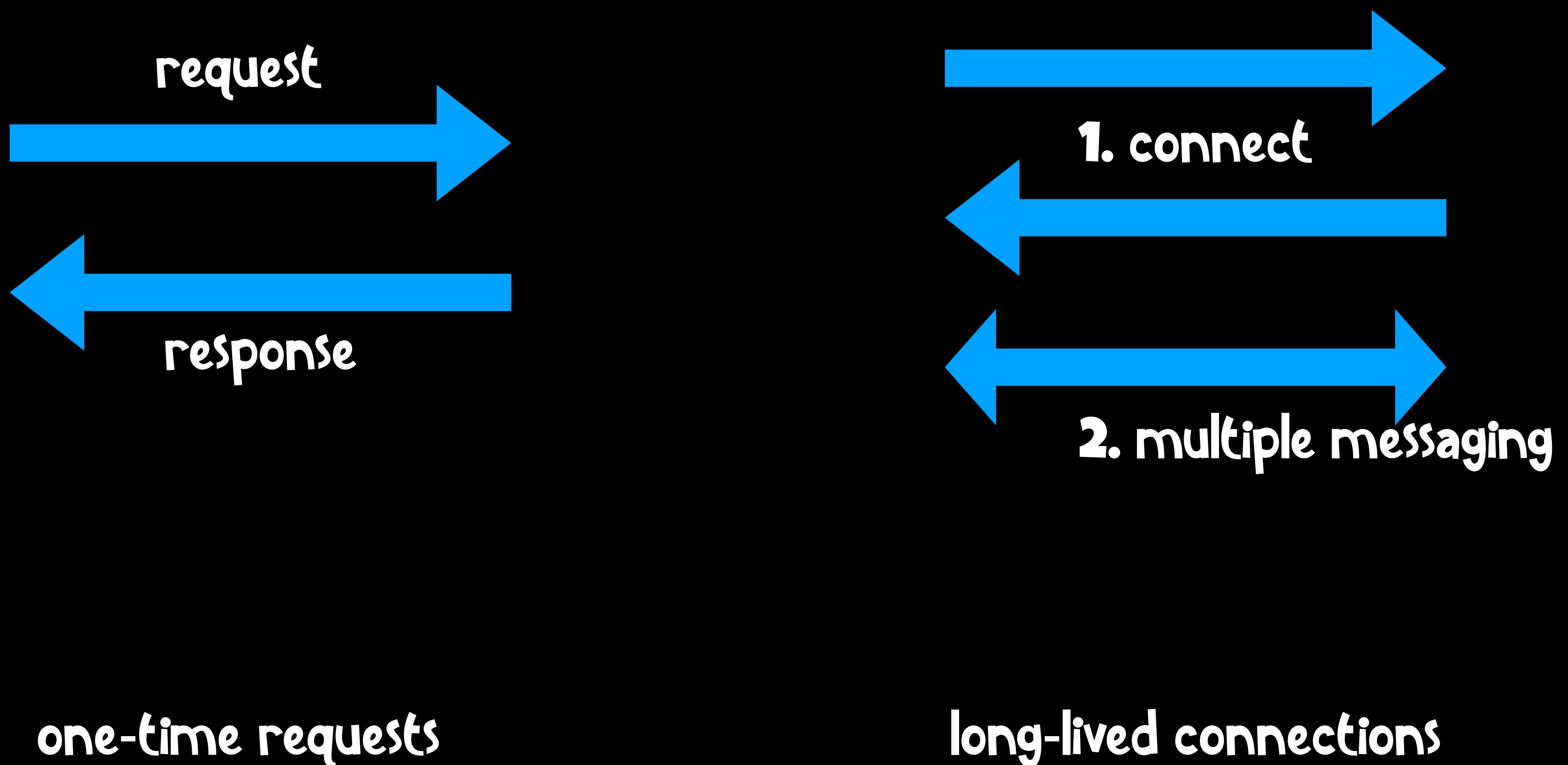
```
background.js

chrome.runtime.onMessage.addListener(function(request, sender, sendResponse) {
    if (request.greeting === "hello") {
        sendResponse({farewell: "goodbye"});
    }
});
```

long-lived connections



inter-extension message passing



one-time requests

inter-extension message passing

```
...  
background.js  
  
const targetExtensionId = "ExtensionID-1"  
  
chrome.runtime.sendMessage(targetExtensionId, {greeting: "hello"},  
function(response) {  
  
    console.log(response.farewell);  
  
});;
```

Extension 1

```
...  
background.js  
  
chrome.runtime.onMessageExternal.addListener(function(request, sender, sendResponse)  
{  
  
    if (sender.id === "ExtensionID-2") {  
        console.log(request.greeting);  
        sendResponse({farewell: "goodbye"});  
    }  
});
```

Extension 2

long-lived connections

inter-extensions message passing



跨Extension长连接通信

```
// 连接与发送消息
var port = chrome.runtime.connect(laserExtensionId);
port.postMessage(...);

// 连接与接收消息
chrome.runtime.onConnectExternal.addListener(function(port) {
  port.onMessage.addListener(function(msg) {
    // See other examples for sample onMessage handlers.
  });
});
```

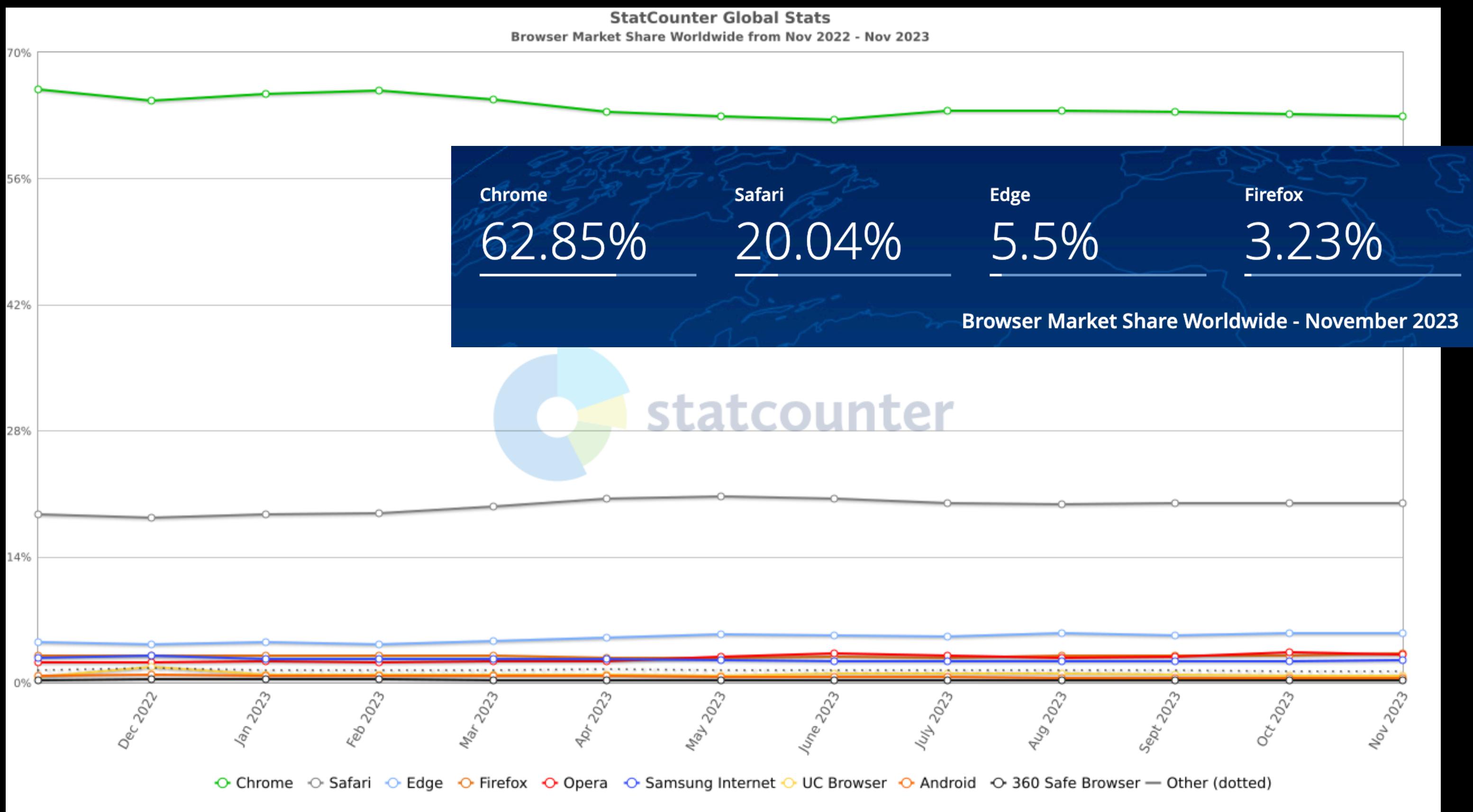
send message from webpage (to extension)



用Extension间的通信方式发送和接收消息就可以了，但只能单向的从web页面发送消息到Extension，而不能从Extension到Web页面。通过Extension到manifest.json配置限定Extension可以接收哪些页面到消息。

Pros & Cons

- from ChaosMonkey/ViolentMonkey/… to Chrome Extension
- More enhancements than scripts managers
- Big Market and platform



- **No portable**
 - **compare to website(which cross browsers)**
 - **locked to chrome ecosys**
- **install before usage**



Chrome Extension Structure

manifest.json is the key

```
.  
├── bundle.css  
├── bundle.js  
├── bundle.js.map  
├── icon.jpg  
└── index.html  
   └── manifest.json
```

像index.html实际上是manifest.json里配置的popup页面，具体用什么名字都可以根据自己的喜好配置。
icon.jpg是action button的图标，也是在manifest.json里配置。
引用的js，要么是在manifest.json里配置，要么是配置的popup页面的依赖。
总之，manifest.json是整个Extension的总控和入口。

versions of manifest.json



Parceljs helps

`parcel build manifest.json --config @parcel/config-webextension`

```
manifest.json:

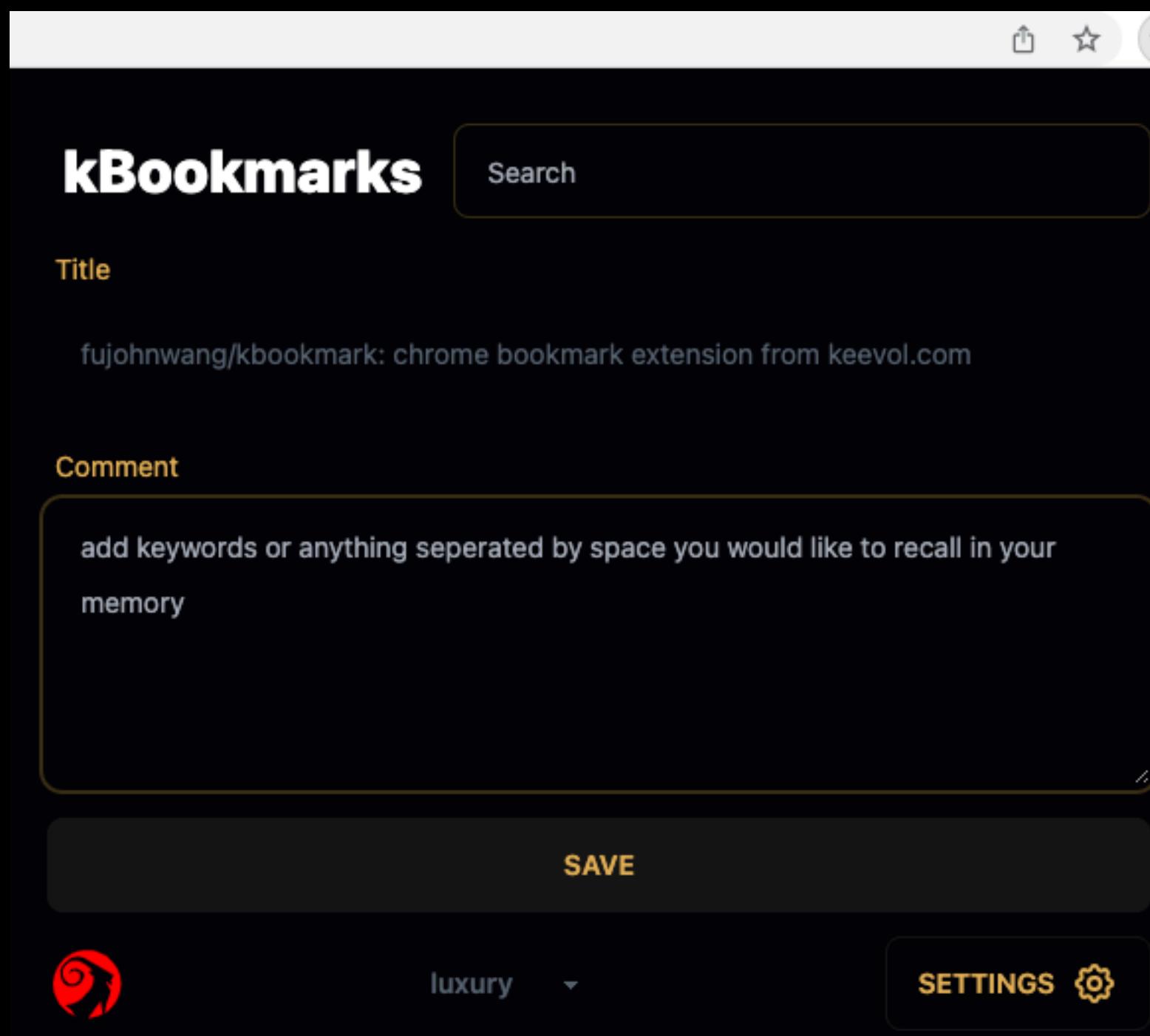
{
  "manifest_version": 3,
  "name": "Sample Web Extension",
  "version": "0.0.1",
  "background": {
    "service_worker": "background.ts",
    "type": "module"
  },
  "content_scripts": [
    {
      "matches": ["*://github.com/parcel-bundler/*"],
      "js": ["parcel-content-script.ts"]
    }
  ]
}
```

使用@parcel/config-webextension， parcel允许在编译的时候使用manifest.json作为输入源（source）

这将极大便利Extension的开发和编译。

Extensions I built

<https://github.com/fujohnwang/kbookmarks>



Select'n'Copy

Selector Expression

Evaluate

Result Content

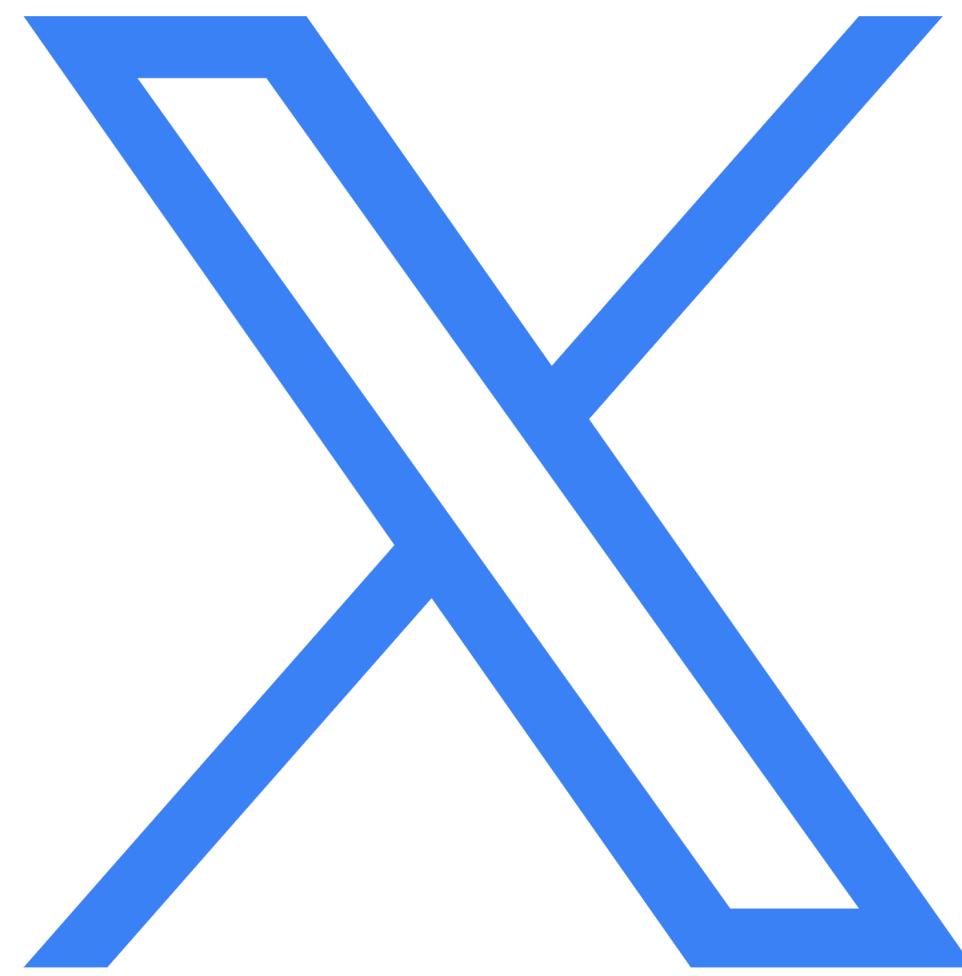
0 elements are found before joined as string

Copy to Clipboard

A screenshot of the Select'n'Copy tool. It features a large purple "Evaluate" button. The "Result Content" area is currently empty. Below it, a message says "0 elements are found before joined as string". A "Copy to Clipboard" button is at the bottom.

<https://github.com/fujohnwang/selectncopy>

Any Questions?



<https://x.com/fujohnwang>