

chrome插件开发调研

- 架构
 - Manifest

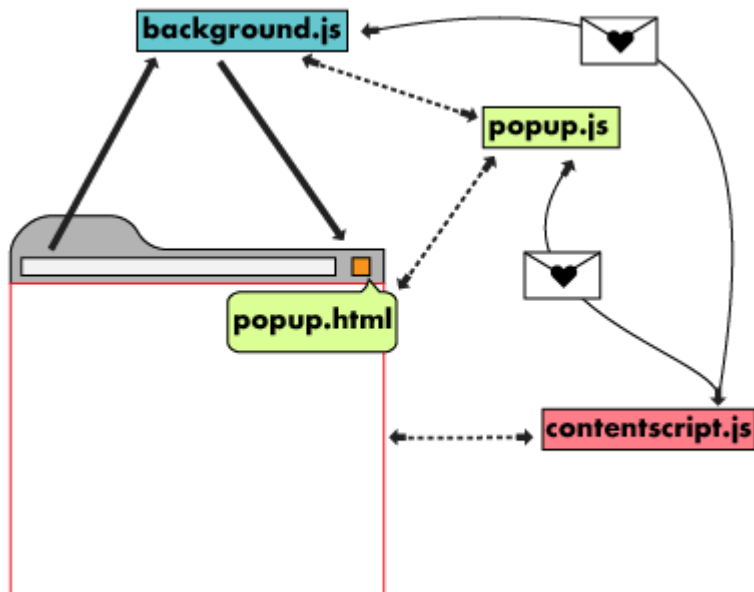
在项目根目录创建[manifest.json](#)文件
 - Background Script

定义插件事件处理逻辑
 - UI Elements

定义插件弹框UI
 - Content Script

跟浏览器当前打开页面交互
 - Options Page

为插件提供定制化
- 插件交互图



tronlink分析

tronlink基于React开发

manifest分析

- API权限申请

```
"permissions": [
  "storage",
  "unlimitedStorage",
  "clipboardWrite",
  "activeTab"
]
```

- 弹框UI

```
"browser_action": {
  "default_popup": "packages/popup/build/index.html",
  "default_title": "TronLink"
}
```

- 扩展程序管理界面图标

```
"icons": {
  "128": "packages/popup/static/icon.png"
}
```

- 事件handler

```
"background": {
  "scripts": [
    "dist/backgroundScript.js"
  ],
  "persistent": true
}
```

- 跟当前打开的页面交互

```
"content_scripts": [
  {
    "js": [
      "dist/contentScript.js"
    ],
    "run_at": "document_start",
    "all_frames": true
  }
]
```

- content_scripts需要加载的资源

```
"web_accessible_resources": [
  "dist/pageHook.js"
]
```

backgroundScript分析

- 模块
 - NodeService
- 网络节点管理

- StorageService

状态存储

- WalletService

公链API调用

- 代码分析

- index.js

加载时绑定所有事件

```
run() {  
  BackgroundAPI.init(duplex);  
  this.bindAnalytics();  
  // 公链相关事件  
  this.bindPopupDuplex();  
  // 签名事件  
  this.bindTabDuplex();  
  // 钱包设置事件  
  this.bindWalletEvents();  
}
```

- WalletService

- Account.js

创建，导入，导出账户；
添加通证；
存款，取款；
转账；

- index.js

钱包操作入口；
签名白名单管理；
账户状态维护；

UI

- popup目录为基于react开发的UI

contentScript分析

- index.js

加载pageHook，为浏览器当前页面注入tronWeb对象

```
inject() {  
  const injectionSite = (document.head || document.documentElement);  
  const container = document.createElement('script');  
  
  container.src = extensionizer.extension.getURL('dist/pageHook.js');
```

```
container.onload = function() {
  this.parentNode.removeChild(this);
};

injectionSite.insertBefore(
  container,
  injectionSite.children[ 0 ]
);

logger.info('TronLink injected');
}
```

- pageHook

为浏览器页面提供tronWeb对象

```
window.tronWeb = tronWeb;
```

提供签名入口

```
sign(transaction, privateKey = false, useTronHeader = true, callback =
false)
```