

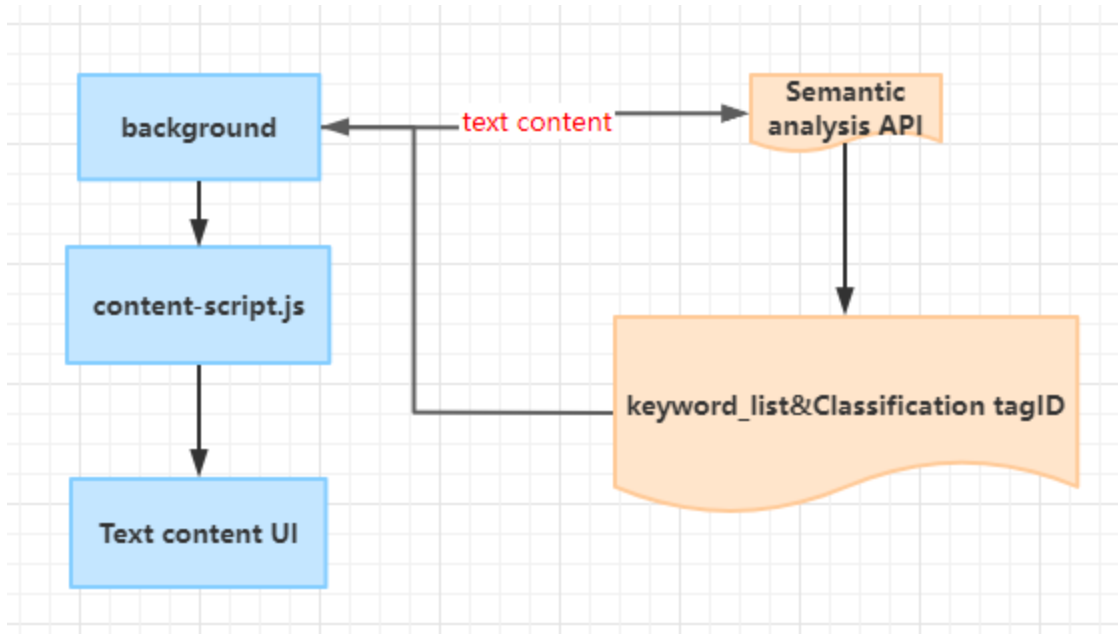


Browser UI Editor

<https://github.com/wjy1919da/FirefoxExtensionDemo>

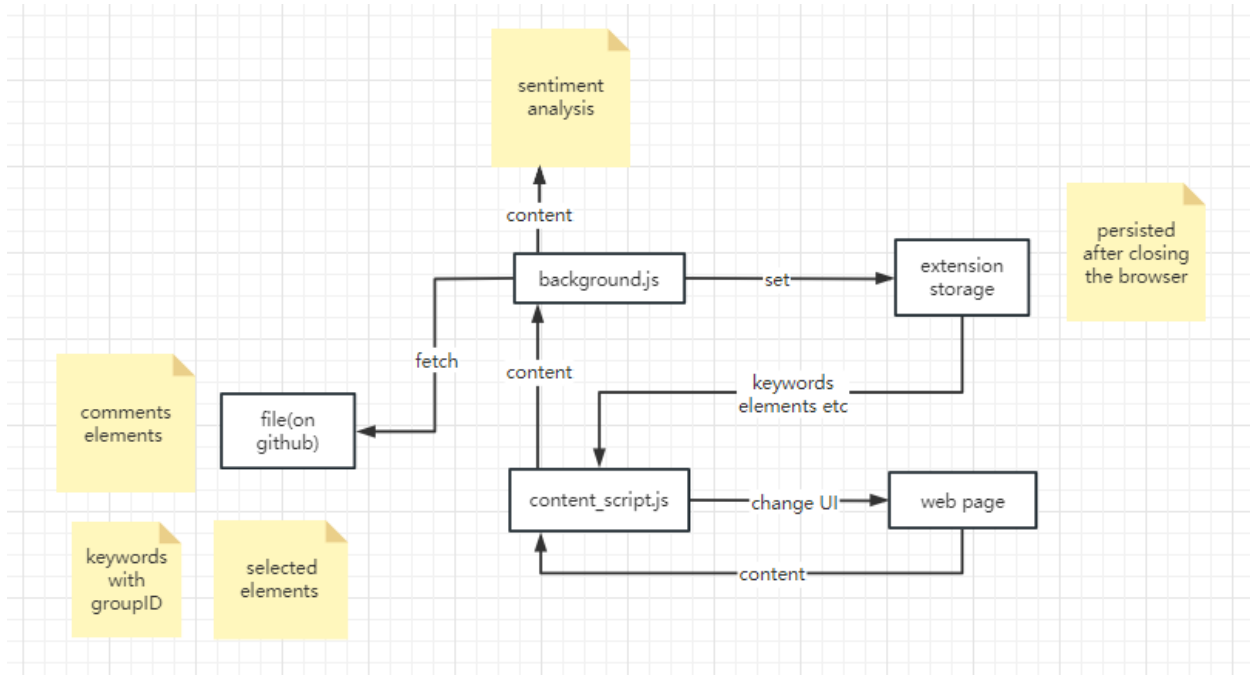
Project propose

1. Allow users to modify the browser page according to their needs.
2. Allow users to modify the browser page according to their needs.
3. Can be used in conjunction with an online streaming sentiment analysis system (which will provide real-time feedback of user page text data to the backend, allowing for changes to the page's text UI based on a keywords list).

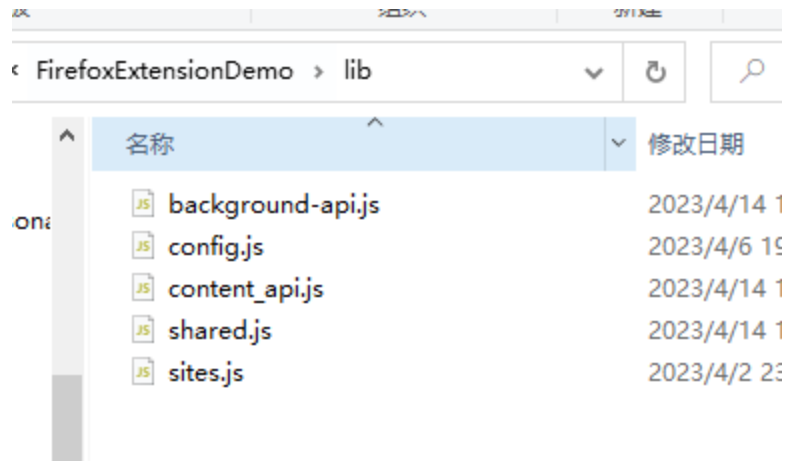


Semantic analysis API is not implemented.

Design



Introduction of the library



The lib contains some tool which will be useful to modify the UI of browser page.
You need to download the lib.zip and put it in your extension.

名称	修改日期
.git	2023/
background	2023/
content_script	2023/
css	2023/
img	2023/
lib	2023/
popup	2023/
w3css	2023/
framing.md	2023/

Then embeded the file in manifest.json

Firefox extention:

add this code after your content-scripts.js

```
{
  "matches": ["<all_urls>"],
  "run_at": "document_start",
  "all_frames": true,
  "match_about_blank": true,
  "js": ["lib/content_api.js"]
}
```

It should be like this:

```
"content_scripts": [  
  {  
    "css": [  
      "css/content.css"  
    ],  
    "js": [  
      "content_script/content-script.js"  
    ],  
    "matches": [  
      "http://*/**",  
      "https://*/**"  
    ],  
    "run_at": "document_end"  
  },  
  {  
    "matches": ["<all_urls>"],  
    "run_at": "document_start",  
    "all_frames": true,  
    "match_about_blank": true,  
    "js": ["lib/content_api.js"]  
  }  
],
```

Introduction of content_api.js

- **insertStylesIntoPage()**

Reference:

<https://github.com/grantwinney/hide-comments-everywhere>

Hide comments everywhere, which is suitable for many web sites.

test_sites.json includes the elements of comments in many web sites. I put it on the github so everyone are able to contribute to it.

call it in content-scripts.js

```
insertStylesIntoPage();
```

- **updateHtmlPage**({ 'keywords': keywordsWithGroupId })

Function used to change the UI of specific words.

Get the keywordsWithGroupId from extension storage:

```
let keywordsWithGroupId;
browser.storage.local.get("global_keywordslist").then(result => {
  keywordsWithGroupId = JSON.parse(result.global_keywordslist);
});
updateHtmlPage({ 'keywords': keywordsWithGroupId });
```

The css style of the keywords is related to the groupId

"Accessibility": 0,	.highlighted.style-0 {
"Keep in mind": 0,	background: #4c7bd9;
"covid": 0,	}
"Mask": 0,	

- **hideKeywordComments**({ 'keywords': keywordsWithGroupId })
- **hideWords**({ 'keywords': keywordsWithGroupId })
- **replaceContent**({ 'keywords': keywordsWithGroupId })

Function used to hide the comments with keywords.

Input: keywordsWithGroupId

```
hideKeywordComments({ 'keywords': request.keywordsWithGroupId });
```

- **analysisDomText**(Node node)

Traverse the DOM tree, store nodes of text type in an array, and send them to the background.

It is more convenient to embed the removeUI function here, with the option parameter corresponding to different remove functions.

```
analysisDomText(document.body)
```

Page refresh,

```
const observer = new MutationObserver((mutations) => {
  mutations.forEach((mutation) => {
    if (mutation.addedNodes && mutation.addedNodes.length > 0) {
      for (let i = 0; i < mutation.addedNodes.length; i++) {
        const newNode = mutation.addedNodes[i];
        analysisDomText(newNode)
      }
    }
  });
});
observer.observe(document.body, {
  childList: true,
  subtree: true
});
```

- **removeHighlights(node)**
- **removeHideWords(node)**
- **showComment()**
- **removeReplaceContent()**
- **removeHideContent()**

Remove the insert UI style of browser page.

Input: document.body, function* f (remove function)

```
analysisDomText(document.body, function* f) // f can be any remove function
```

Bottleneck and future work

- Asynchronization
- MVCC (comments elements and content elements)

Update the global version when git commit the file; fetch the global version first, if the local version need to update, fetch all the data from github.

A timer to get the update

- Back-end model: gpt-3.5
give the fame and content , rewrite the content base on the frame
- need to rewrite content function