Manuel d'utilisateur

Cette extension permet la l'ajout et la visualisation des Marker pour chaque Bookmark ajouté au navigateur.

L'extension permet d'éxecuter l'action sur des Bookmark dejà géolocalisés, c'est à dire qui ont une information qui renseigne la position géographique

sous forme de latitude et longitude en degrés.

Les Marker cartographiés génère des popups contenant l'URL de la page correspondante et qui sont cliquable pour ouvrir cette page dans un nouvel onglet.

Le code de l'application permet aussi de faire des tests sur des Marker stockés en une base de données, des Markers en fichier texte ou, par defaut, sur

des Markers qui contienne l'information requise sous forme d'un String associé à l'attribut Name grace à une expression régulière.

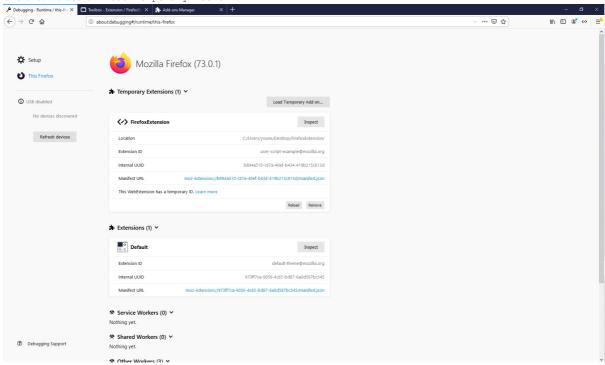
Guide d'installation

Le fichier le plus intéressant pour cette étape est le manifest.json qui se situe dans la racine du projet.

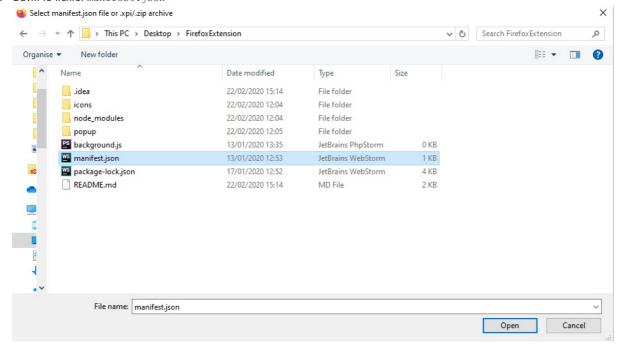
Pour pouvoir lancer et le test et l'extension il faut configurer le coté Backend et pour le faire on doit:

- Copierle dossier FirefoxExtension dans un serveur local par exemple: ..\XAMPP\htdocs
- Avec un SGBD créer un base de données de test firef et lançer la requete dans le fichier: ..\FirefoxExtension\popup\firef.sql
- Lancer le navigateur Firefox accéder à: about : debugging (←) → ℃ ŵ **☆** Setup Mozilla Firefox (73.0.1) this Firefox ★ Temporary Extensions (0) ∨ Load Temporary Add-on... ♣ Extensions (1) ∨ Refresh devices Default Inspect Internal UUID 973ff7ca-9056-4c65-bd87-6a0d587bc545 ☆ Shared Workers (0) ∨ resource://gre/modules/osfile/osfile_async_worker.js Inspect resource://services-settings/RemoteSettingsWorker.is Inspect ② Debugging Support resource://gre/modules/PageThumbsWorker.js Inspect

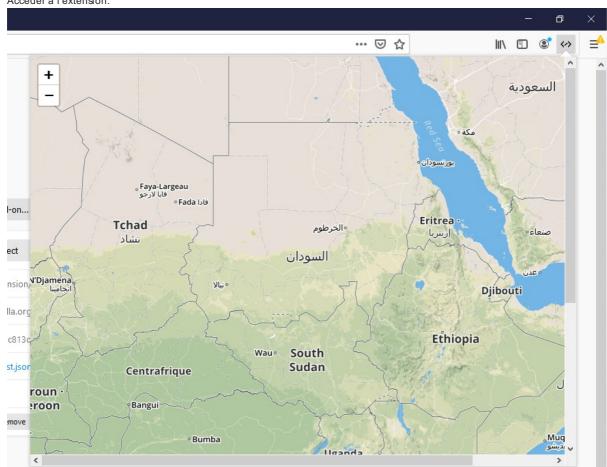
• Clicker sur le bouton: Load Temporary Add-on



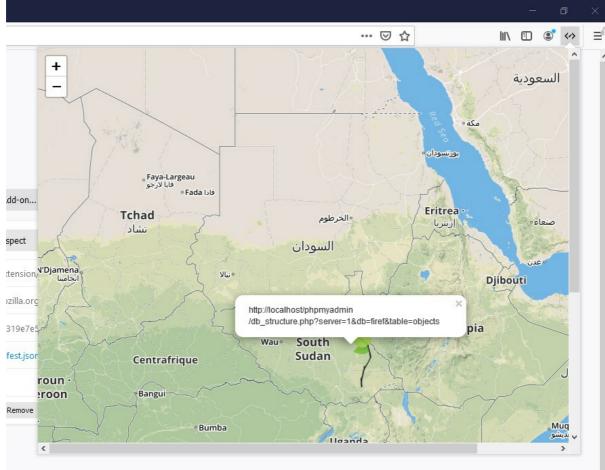
• Ouvrir le fichier manifest. json



Accéder à l'extension:



Faire un hover sur le Marker et cliquer sur le popup pour l'ouvrir



Guide dev:

L'approche par titre se base su l'extraction de longitude et latitude qui sont sous la forme :

• "texte + -lon: + float1 + -lat + float2 + text"

Cette extraction se base sur les expressions regulières:

```
let myreg=/(?<=-lon:)(\w[0-9][.,][0-9]*)/;
let lon=parseFloat(myreg.exec("text"));</pre>
```

L'expression ci-dessus permet d'extraire float1 et de meme pour float2 avec text le nom du Bookmark recupéré depuis l'API Bookmark API (https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/bookmarks):

```
let myreg=/(?<=-lat:)(\w[0-9][.,][0-9]*)/;
let lat=parseFloat(myreg.exec("text"));</pre>
```

Les tests unitaires associés sont faits par mocha dans le Backend grace à Nodejs comme le montre l'illustration suivante

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
FirefoxExtension > icons > test.js
                                                                                                        F
   🔳 Project🗸 🗘 🚆
                       🌣 — 🥻 package-lock.json × 🕻 package.json × 👪 test.js × 👪 user-script.js
                                                                                                     # README.md
                                       const assert = require('asse
     FirefoxExtension C:\Users
        icons
                                       let myreglon=/(?<=-lon:)(\w[0-9][.,][0-9]*)/;
           debug.PNG
           🗂 Extension.PNG
                                       let myreglat=/(?<=-lat:)(\w[0-9][.,][0-9]*)/;</pre>
           f_source_code_10371
           📇 index.js
           ₫ LICENSE
           🛃 Load.PNG
                                           assert.equal(parseFloat(myreglon.exec(mytext)),62,2496633 )
           marker.png.jpg
           test.js
        node_modules
          popup
           🟭 ajax.php
           firef.sql
           📇 jquery-3.4.1.js
           🛃 leaf-green.png
          aleaflet-src.js
. Z: Structure
   Run:
                                                                                                                          ů.
           0 17 12
                                  Oms "C:\Program Files\nodejs\node.exe" C:\Users\youne\Desktop\FirefoxExtension\node_

✓ Test Results

             correctly gets the lat

✓ correctly gets the Ion 0 ms

   ==

    Windows Defender configuration updated

     4: Run ≔ 6: TODO 🗷 Terminal
                                                                                                                     Event Loa
Tests passed: 2 (a minute ago)
                                                                                                12:1 CRLF UTF-8 4 spaces 🧣 🖶
```

Pour la réutilisation du code, il faut savoir que:

- Toute bibliothèque externe est à télécharger localement sinon vous risquez des problèmes liés au CORS. <u>StackOverFlow answer (https://stackoverflow.com/a/8456586)</u>
- La partie base de données ajoutée sert tout simplement pour offrir un outil de visualisation des Markers et une extension possible du code si on dispose d'une base contenant des Bookmark qui nous appartiennent pas pour des éventuels besoins. Du coup pour la mettre en oeuvre il fallait opter pour un serveur permettant de tourner du code php en asynchrone (JQuery installé en local) avec l'extension sur le browser. Voir le fichier popup/ajax.php.
- Le serveur Nodejs permet de tester unitairement et intégralement les fonctions grace à Mocha (Jasmine est installé aussi au cas ou).
- Un répértoire public github est associé au projet pour toutes mises à jour, pulls request, vulnérabilités ou modifications sous l'URL <u>FireFoxExtension</u> (https://github.com/izaganami/FireFoxExtension)