

Manuel d'utilisateur

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

L'extension permet d'exécuter l'action sur des `Bookmark` déjà 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.

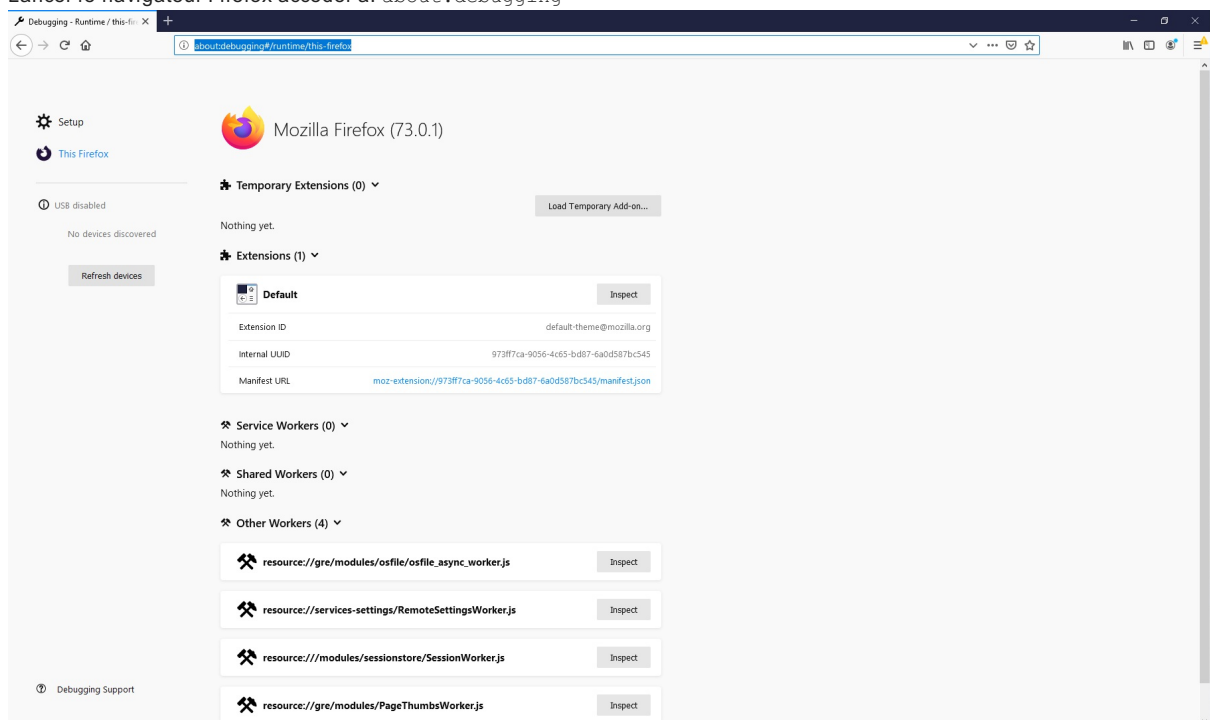
Le code de l'application permet aussi de faire des tests sur des `Marker` stockés en une base de données, des `Markers` en fichier texte ou, par défaut, sur des `Markers` qui contiennent l'information requise sous forme d'un `String` associé à l'attribut `Name` grace à une expression régulière.

Guide d'installation

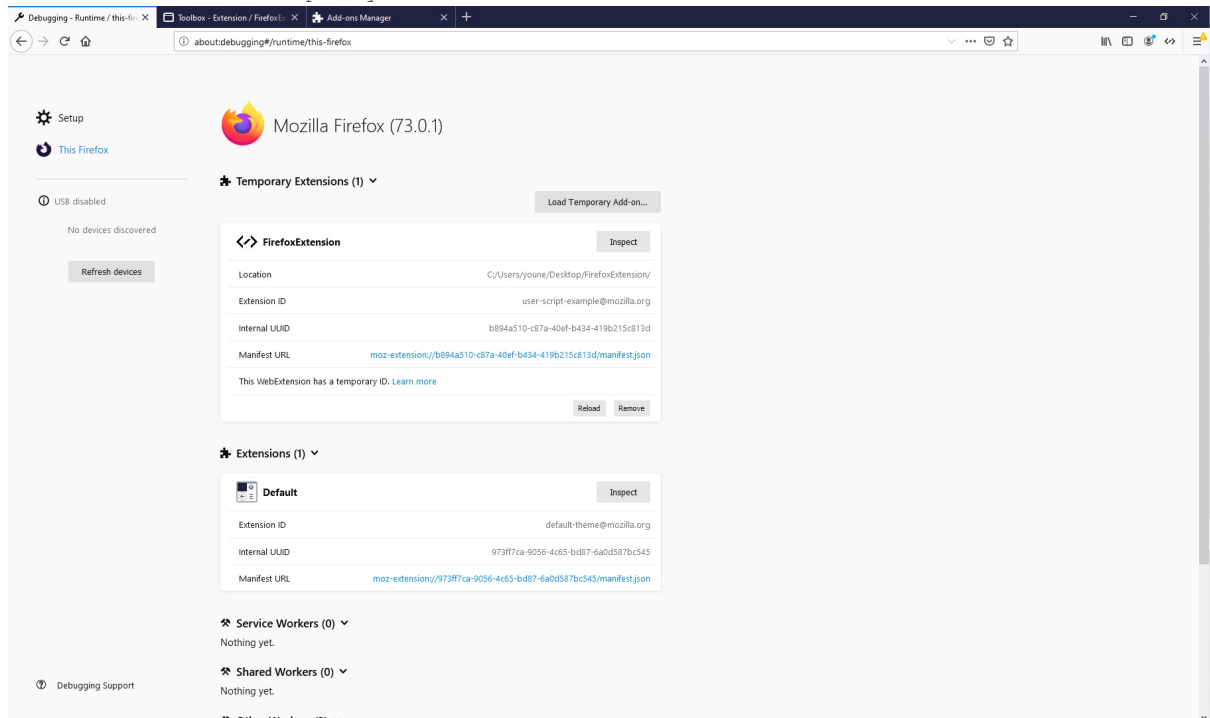
Le fichier le plus intéressant pour cette étape est le `manifest.json` qui se situe dans la racine du projet.

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

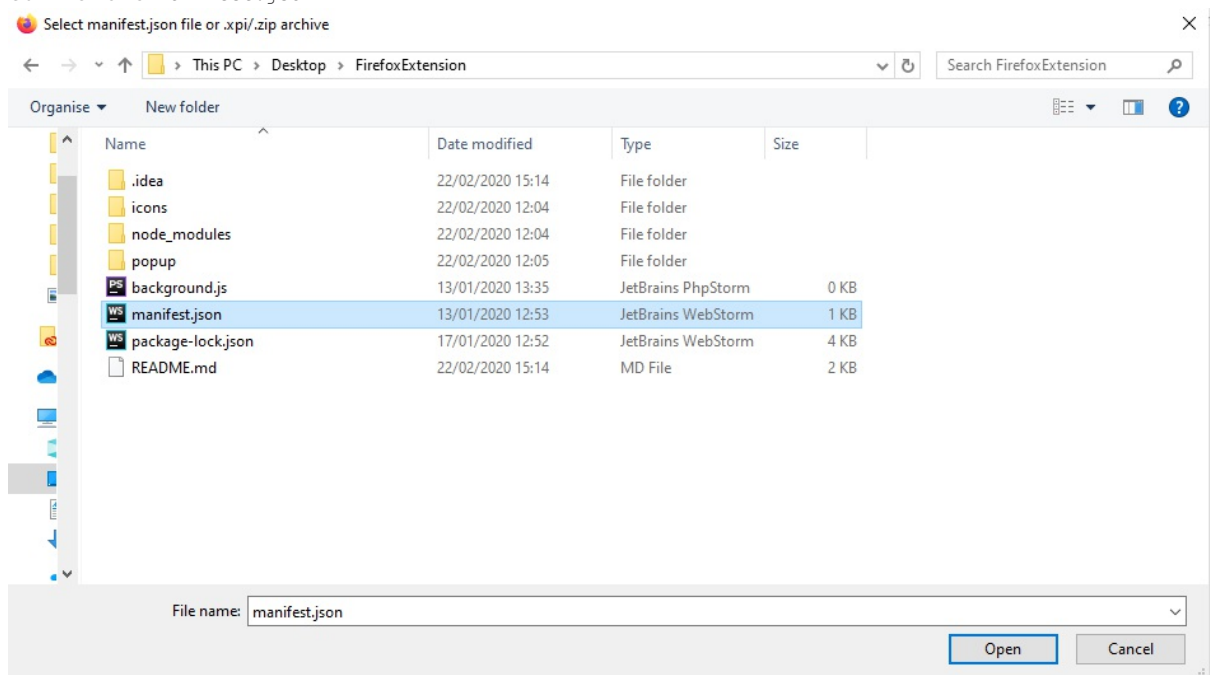
- Copier le dossier `FirefoxExtension` dans un serveur local par exemple: `..\XAMPP\htdocs`
- Avec un SGBD créer une base de données de test `firef` et lancer la requete dans le fichier:
`..\FirefoxExtension\popup\firef.sql`
- Lancer le navigateur Firefox accéder à: `about:debugging`



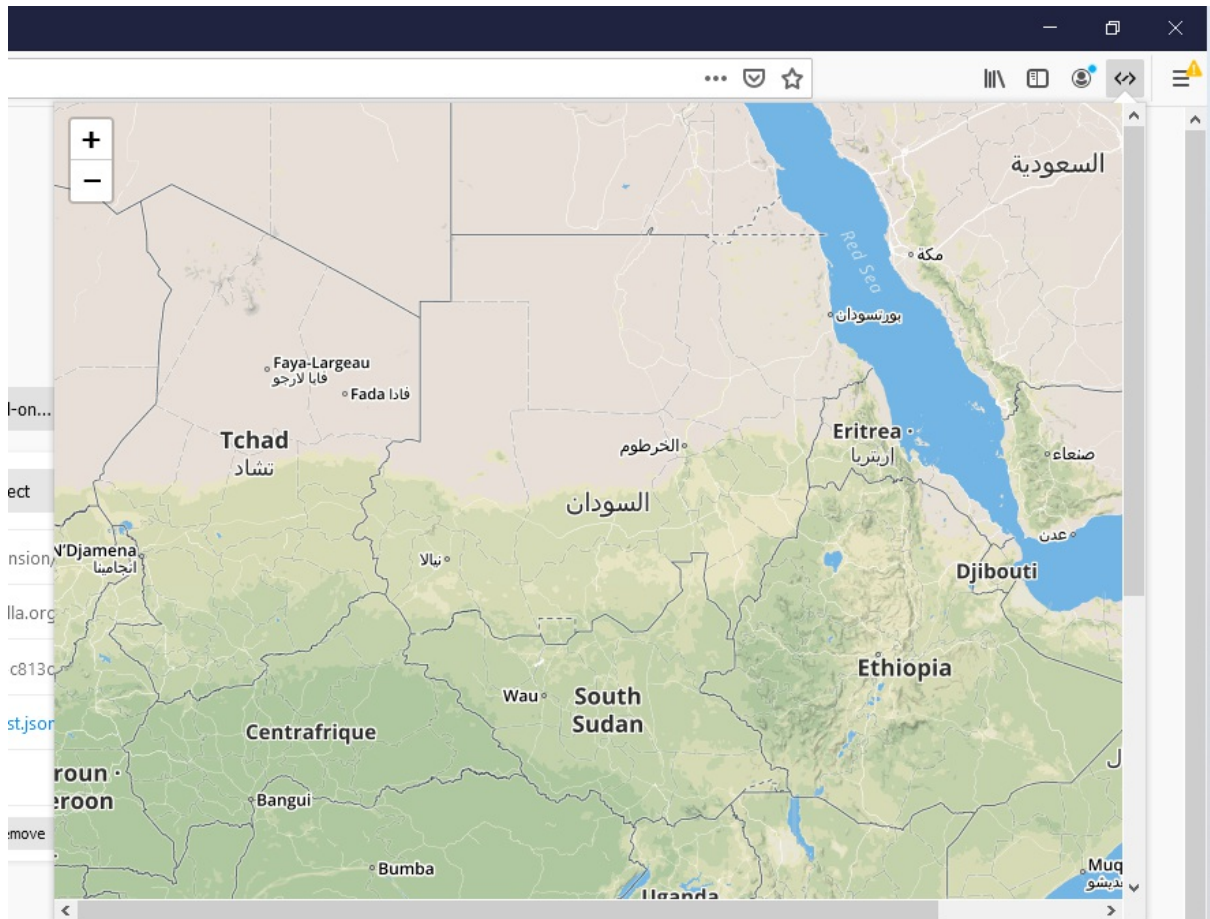
- Cliquez sur le bouton: Load Temporary Add-on



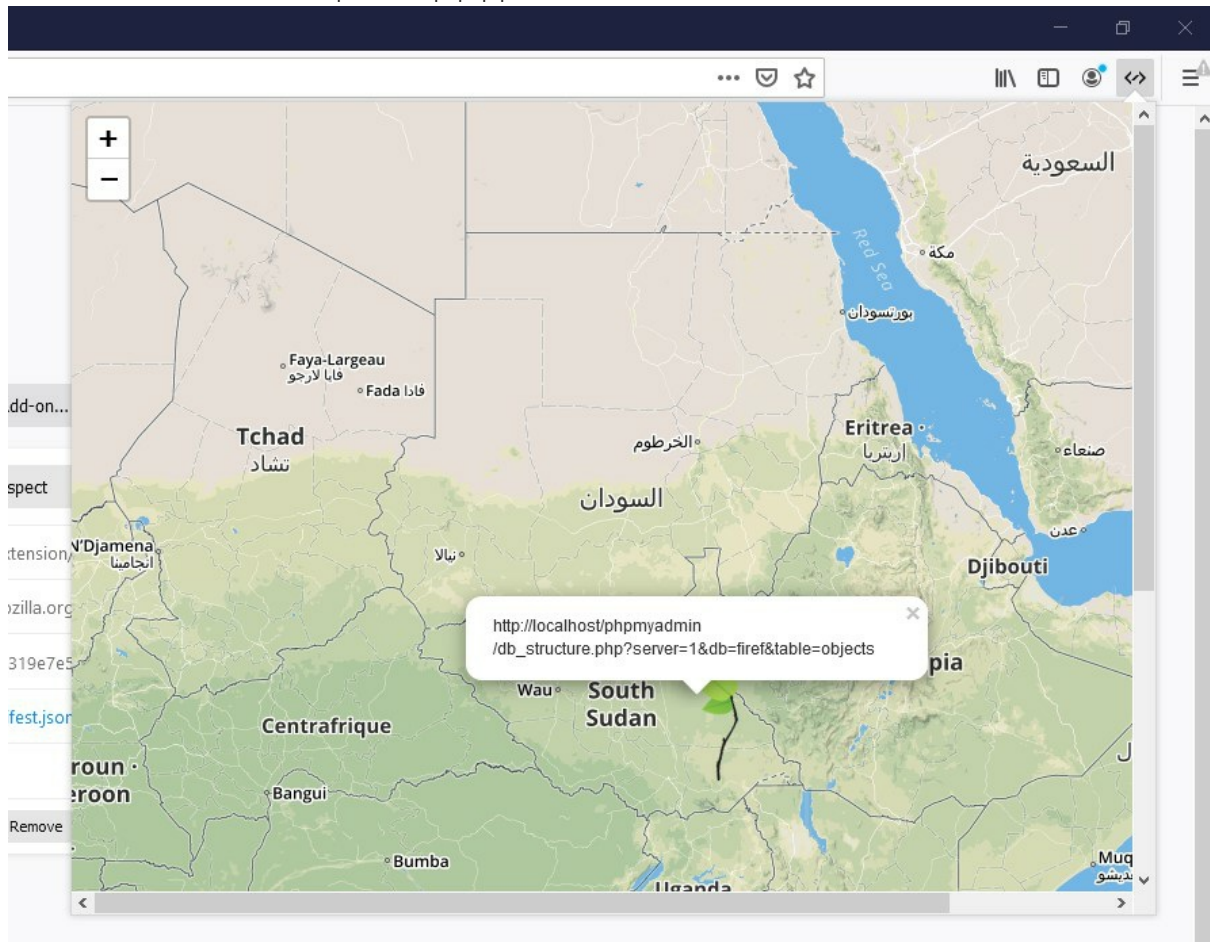
- Ouvrir le fichier manifest.json



- 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 sur l'extraction de longitude et latitude qui sont sous la forme :

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

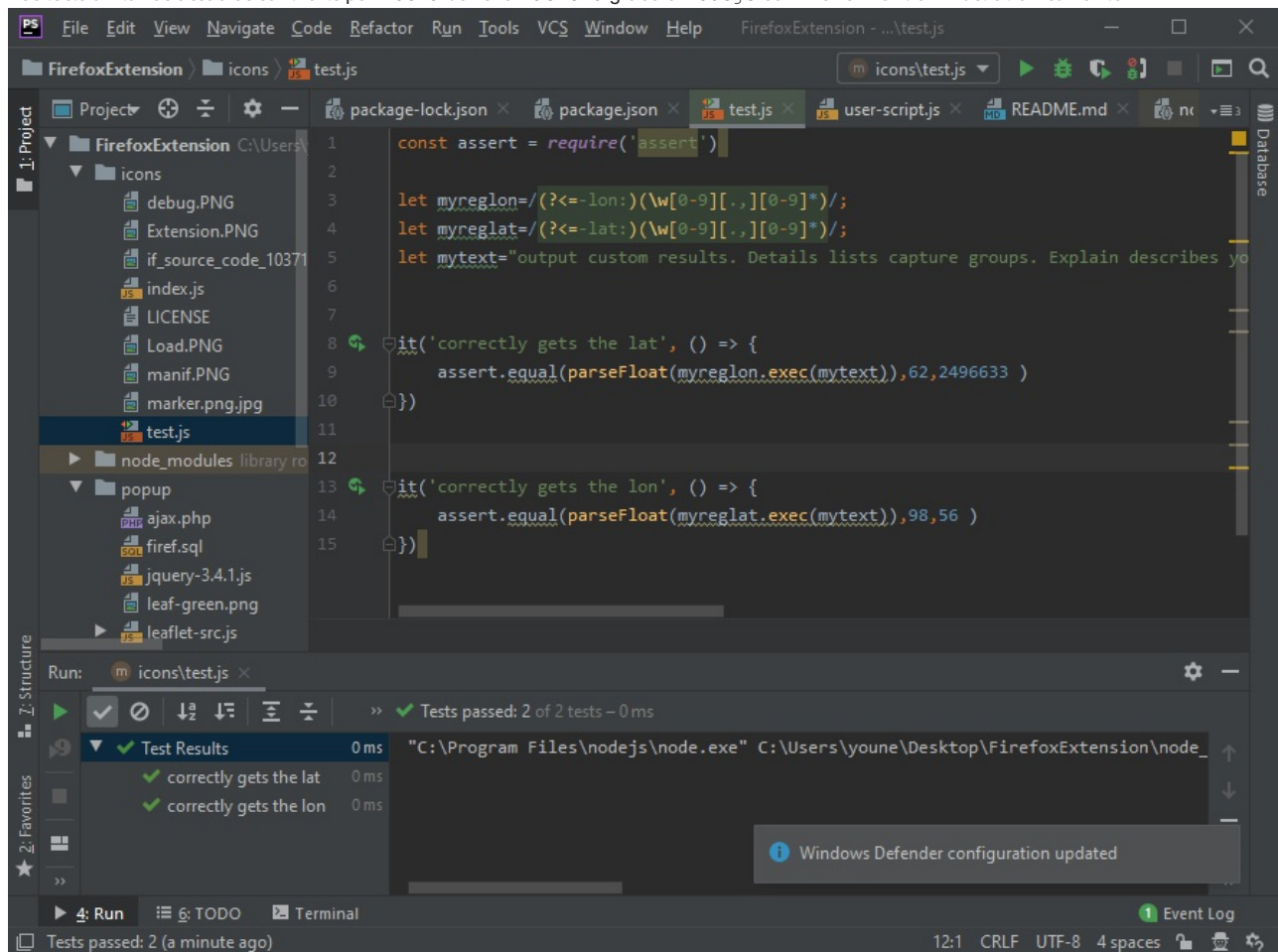
Cette extraction se base sur les expressions régulières:

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

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

Les tests unitaires associés sont faits par mocha dans le Backend grâce à Node.js comme le montre l'illustration suivante



Pour la réutilisation du code, il faut savoir que:

- Toute bibliothèque externe est à télécharger localement sinon vous risquez des problèmes liés au CORS. [StackOverflow answer \(https://stackoverflow.com/a/8456586\)](https://stackoverflow.com/a/8456586)
- 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 Node.js permet de tester unitairement et intégralement les fonctions grâce à Mocha (Jasmine est installé aussi au cas ou).
- Un répertoire public github est associé au projet pour toutes mises à jour, pulls request, vulnérabilités ou modifications sous l'URL [FireFoxExtension \(https://github.com/izaganami/FireFoxExtension\)](https://github.com/izaganami/FireFoxExtension)

