Rapport POM : GITHUND - Extension Thunderbird pour GitHub

Paulin BAT, Léo HENINI

Mai 2021

Résumé : Le projet Githund vise à créer une extension Thunderbird pour Github. Elle propose une interface plus allégée mais qui ne concerne que les mails venant de GitHub (en particulier les mails threadés). L'extension apporte un confort dans le traitement de ces mails : tri par date, regroupement des mails par threads, suppression d'un thread et des messages liés, affichage d'un mail ou d'un thread, opération github (via l'API) sur ces derniers (abonnement/désabonnement à un thread, ouvrir/fermer un ticket, …).

Lien du dépôt du projet : https://github.com/zacun/githund

Mots-clefs : Thunderbird, Github, extension, api (rest), thread, issue

1. Introduction

Ce Projet d'Orientation en Master s'inscrit dans le cadre du Master 1 Informatique. Le travail a été intégralement réalisé en distanciel par deux étudiants, Paulin BAT et Léo HENINI, et encadré par Gabriel RADANNE. Il s'agit de la réalisation d'une extension Thunderbird pour GitHub et ce projet s'intitule Githund. L'extension doit permettre une administration simplifiée des mails venant de GitHub via une interface allégée permettant aussi d'interagir avec l'API de GitHub afin de réaliser certaines actions comme par exemple l'abonnement/désabonnement à un thread d'une issue.

Beaucoup de logiciels proposent une bibliothèque d'extensions permettant d'étendre les fonctionnalités de base de ces derniers, l'objectif de ce projet était donc de découvrir comment réaliser une extension pour un gros logiciel tiers tout en essayant d'obtenir une application fonctionnelle et utilisable pour le public.

2. Présentation du contexte de l'extension

GitHub est un service d'hébergement et de gestion de développement des logiciels se basant sur le logiciel de gestion de version Git. Il permet donc d'avoir un suivi détaillé du développement de nos applications tout en mettant en place un système de discussion autour de chaque changement apporté (pull request, commit) mais permet aussi à la communauté de remonter des problèmes et bugs (issue) puis d'en discuter.

Thunderbird est un client de messagerie libre et gratuit. Il permet donc de consulter des mails de manière simple et est une alternative viable à Outlook par exemple. C'est un logiciel très utilisé, notamment par les personnes travaillant avec un système d'exploitation Linux (la suite Microsoft office n'étant pas disponible sur les systèmes UNIX). De plus, Thunderbird est extensible, il permet donc d'ajouter des extensions développées par la communauté afin d'en étendre les fonctionnalités proposées. En effet, Thunderbird propose une base centralisée d'extension qui peuvent s'installer en quelques clics.

Actuellement, GitHub donne la possibilité de recevoir des notifications par mails lorsque nous sommes abonnés à une discussion d'un projet (issue, pull request) afin de savoir si de nouveaux messages ont été

postés. Cela peut faire beaucoup de mails à traiter pour certaines personnes très impliquées et hébergeant leurs projets sur cette plateforme.

Malheureusement, l'interface native de Thunderbird n'est pas pratique car les messages doivent être lus un par un et des actions comme "ignorer un thread" (une discussion) par exemple demandent à l'utilisateur d'ouvrir son navigateur et de se connecter à son compte GitHub pour le faire manuellement. Cela rend donc le suivi des bugs difficiles pour les contributeurs impliqués dans de nombreux projets.

L'extension permettra donc d'apporter plus de simplicité dans la gestion des mails de GitHub avec Thunderbird et permettrait même, avec plus de temps, de réaliser des fonctionnalités via l'API de GitHub comme être notifié qu'une pull-request a été intégrée par exemple et cela tout en restant sur le client de messagerie.

3. Objectifs du projet

L'objectif principal du projet était de créer une extension Thunderbird afin de mieux traiter les mails venant de GitHub en proposant une interface plus allégée et spécifique pour les suivis de discussions.

Les objectifs étaient :

- Sélectionner les dossiers mails dans lesquels se trouvent les mails en provenance de Github.
- Réaliser un groupement des mails venant d'un même thread (un thread est un fil de discussion, en somme il s'agit ici de mails/messages appartenant au même sujet d'une même discussion) et proposer un affichage ergonomique des mails dudit thread.
- Permettre l'abonnement/désabonnement à une conversation (sans passer par le navigateur).
- Permettre la réouverture/fermeture d'un ticket (issue) sans passer par le navigateur.
- Suppression/archivage de tous les mails venant d'un même thread.
- Créer une interface la plus ergonomique possible.

Outre le développement d'une extension, ce projet devait aussi permettre d'approfondir nos connaissances dans l'univers des extensions étant donné que de plus en plus de logiciels libres permettent d'en utiliser mais aussi de renforcer nos compétences en Javascript, langage très utilisé. Le projet nous permet aussi d'explorer l'API REST de GitHub

4. Outils et technologies utilisés

Le langage utilisé est Javascript pour la logique métier de l'extension. Nous utilisons aussi du HTML5 / CSS pour gérer l'interface graphique. Nous utilisons aussi Mustache.js, moteur de template, afin d'intégrer les données plus facilement.

Nous travaillons avec le client Mozilla Thunderbird pour les développeurs. C'est une version qui demande de télécharger tout le code source "Mozilla Base Code", de le build puis de le lancer via un terminal, cela permet d'avoir accès à la console de développement afin de débugger nos applications. Nous avons ensuite accès à l'API de Thunderbird afin de réaliser tout un tas d'actions sur le logiciel. Nous avons aussi accès aux API web de Mozilla comme par exemple la Web Storage API pour stocker des données.

Finalement, nous utilisons l'API REST de GitHub afin de faire communiquer, via le client de messagerie, des actions à réaliser sur les dépôts GitHub.

5. Réalisation du projet

Le projet s'est déroulé en deux grosses parties :

- L'étude des technologies afin de comprendre les outils utilisés.
- Le développement de l'extension.

5.1 Etudes des technologies

L'étude des technologies a été une partie très importante du projet. Elle a permis de prendre en compte et de découvrir les possibilités offertes par les API de Github et de Thunderbird : ce qui était possible ou non de réaliser par rapport à notre cahier des charges.

Pendant cette étude, tout un tas de tests a été effectué permettant de nous familiariser avec les API de manière concrète.

Cependant, on notera que l'étude seule n'a pas été suffisante étant donné que des problèmes auxquels nous n'avions pas pensé sont survenus nous obligeant à de multiples reprises à refaire tout notre modèle de données ou de trouver de nouvelles façons pour contourner ces problèmes. En particulier :

- l'absence de l'ID de la notification/thread du mail reçu dans ce dernier et donc ne permettant pas de faire appel à certaines fonctionnalités de l'API GitHub (désabonnement d'un thread par exemple), nous avons dû passer par d'autres systèmes (mailing list) plutôt que de simples appels à l'API. En effet, ne pouvant donc faire appel aux fonctions de l'API à cause du manque de données reçues dans les mails, nous sommes passés par un système d'envoi de mail à une adresse automatique qui désabonne l'utilisateur.
- limitation de l'API de Thunderbird ne permettant pas, de manière simple et efficace, de changer l'interface graphique déjà existante. Nous avons donc décidé de créer un nouvel onglet et de réaliser une nouvelle interface de A à Z.

Afin de résoudre ces problèmes, nous avons fait appel à la communauté (StackOverflow, création d'une discussion Github, chatroom officielle de Thunderbird pour les développeurs) qui a permis d'apporter quelques indices de réponses sans pour autant proposer de vraies solutions efficaces, cependant nous avons su trouver des solutions alternatives que nous présentons dans la partie 5.3 - Implémentation des fonctionnalités.

5.2 Fonctionnement de l'extension

Le fonctionnement de l'extension est simple, d'un point de vue développement :

- un fichier manifest.json qui permet de définir les propriétés de l'extension (auteurs, nom, autorisations nécessaires, fichier d'entrée, ...)
- un fichier background.js qui lance un script lorsque l'extension est activée.
- un bouton qui ouvre un nouvel onglet qui fonctionne comme une page web normale, l'extension se comporte donc comme une application web (HTML5 et CSS3 pour l'interface graphique, JS pour les scripts). C'est dans les fichiers JS inclus dans cette page web que le code principal se trouve.

D'un point de vue utilisateur :

- une fois l'extension activée, il faut choisir les dossiers dans lesquels se trouvent les mails de Github (via un clic droit dessus)
- cliquer sur le bouton "Open Githund" pour ouvrir un nouvel onglet contenant l'interface de l'application.

Une fois les dossiers choisis, l'application parcourt tous les mails de ces dossiers et ne garde que ceux venant de GitHub (en recherchant un en-tête précis se situant dans les informations des mails en provenance de GitHub). Une série de traitements est effectuée sur ces derniers afin de les classer dans les bonnes variables. Finalement, nous utilisons un moteur de template afin de mettre à jour la page HTML.

5.3 Implémentation des fonctionnalités

5.3.1 Modèle de données

La principale difficulté a été de trouver un bon modèle de données afin de représenter nos données. Après de multiples essais qui se sont révélés inadéquats, nous avons choisi d'opter pour deux principaux objets Mails et Threads.

L'objet Mails contient des clés faisant référence aux dossiers dans lesquels se trouvent les mails, chaque dossier est lui-même un objet contenant des threads et chaque thread contient un objet data avec des informations sur le dit thread et un tableau de mails associés au thread.

La variable Threads est un tableau d'objet qui contient la liste des threads à afficher (triée par date), et permet d'aller chercher dans l'objet Mails les bons mails à afficher.

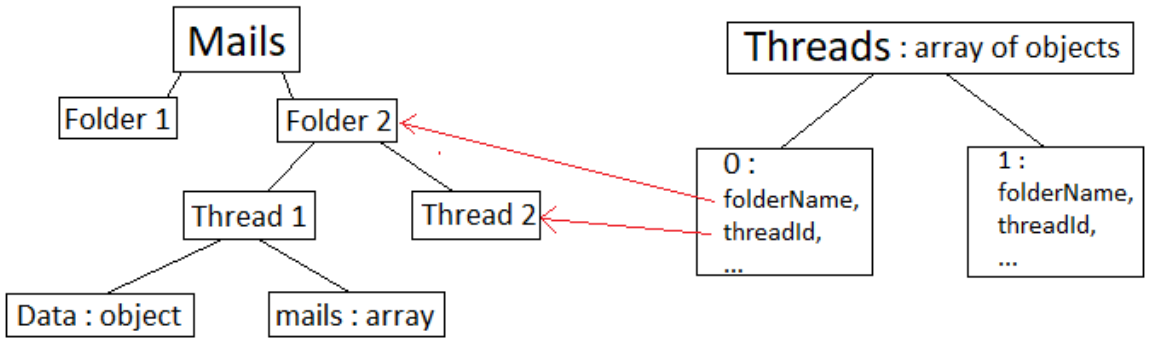

Figure 1 : modèle de données utilisé dans l'extension

5.3.2 Affichage par thread

L'un des intérêts principaux de cette extension est de permettre à l'utilisateur d'avoir une interface plus ergonomique afin de voir les différents threads (groupement de mails pour une même discussion) GitHub présents dans sa boite mail. Avec notre modèle de données, il suffit d'envoyer au template uniquement le premier message de chaque thread. Ainsi, lors de l'ouverture de l'extension, l'utilisateur a accès à un affichage similaire à celui de Thunderbird par défaut. Cependant, si l'utilisateur clique sur un thread, il pourra visionner, dans la partie inférieure de l'interface, tous les messages présents dans ce thread.

L'affichage a été long à réaliser car nous sommes partis d'une page web blanche et nous avons dû confectionner une interface semblable à celle de ThunderBird afin de ne pas dépayser les utilisateurs (cf. figures 2 & 3). Cela impliquait de devoir recréer des fonctionnalités basiques comme la possibilité de changer la taille des différents panels via la souris par exemple.

A noter que dans la liste des mails, seuls les threads y sont listés et non tous les mails, ainsi l'interface est bien plus allégée.

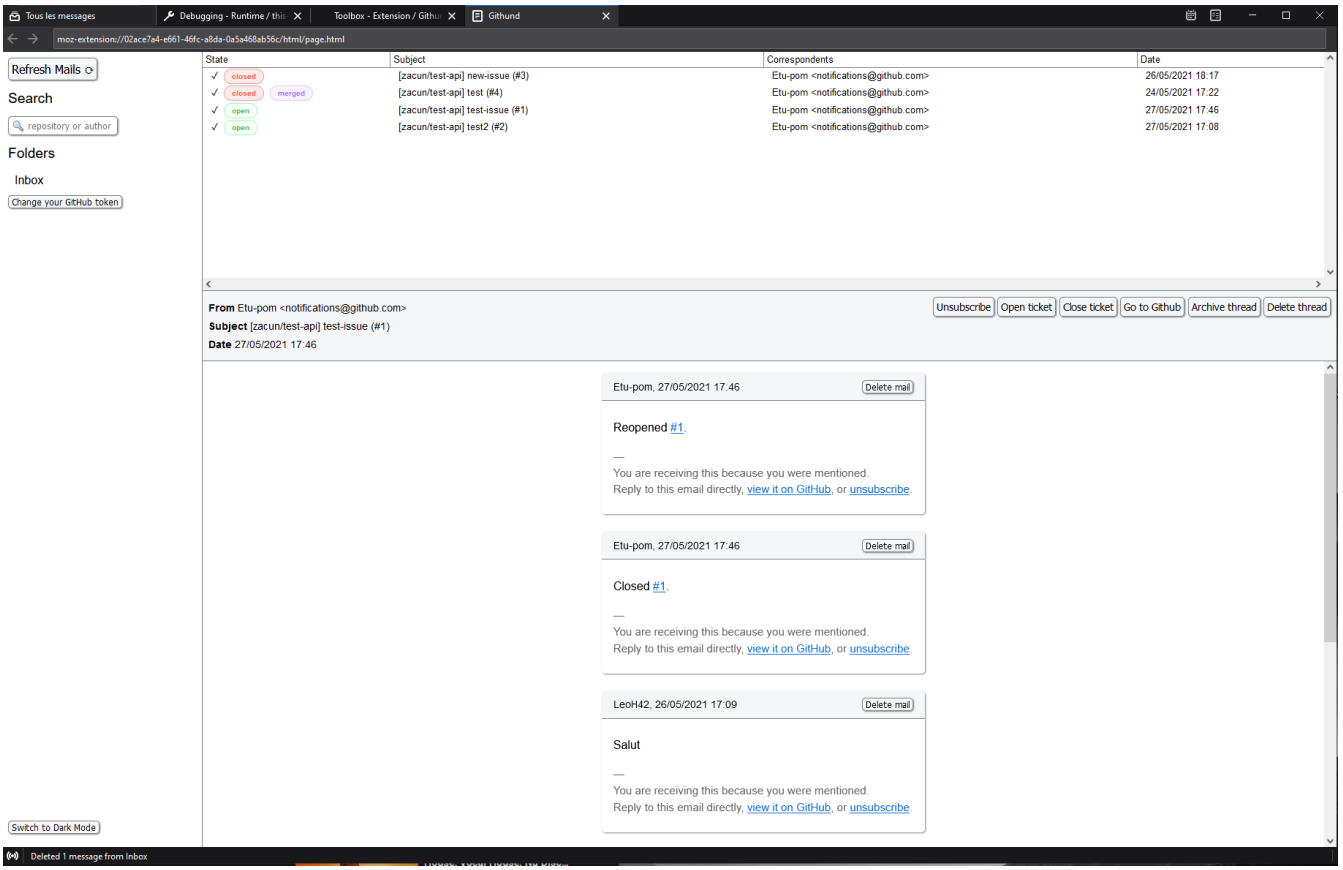

Figure 2 : Mode clair de l'interface de l'extension

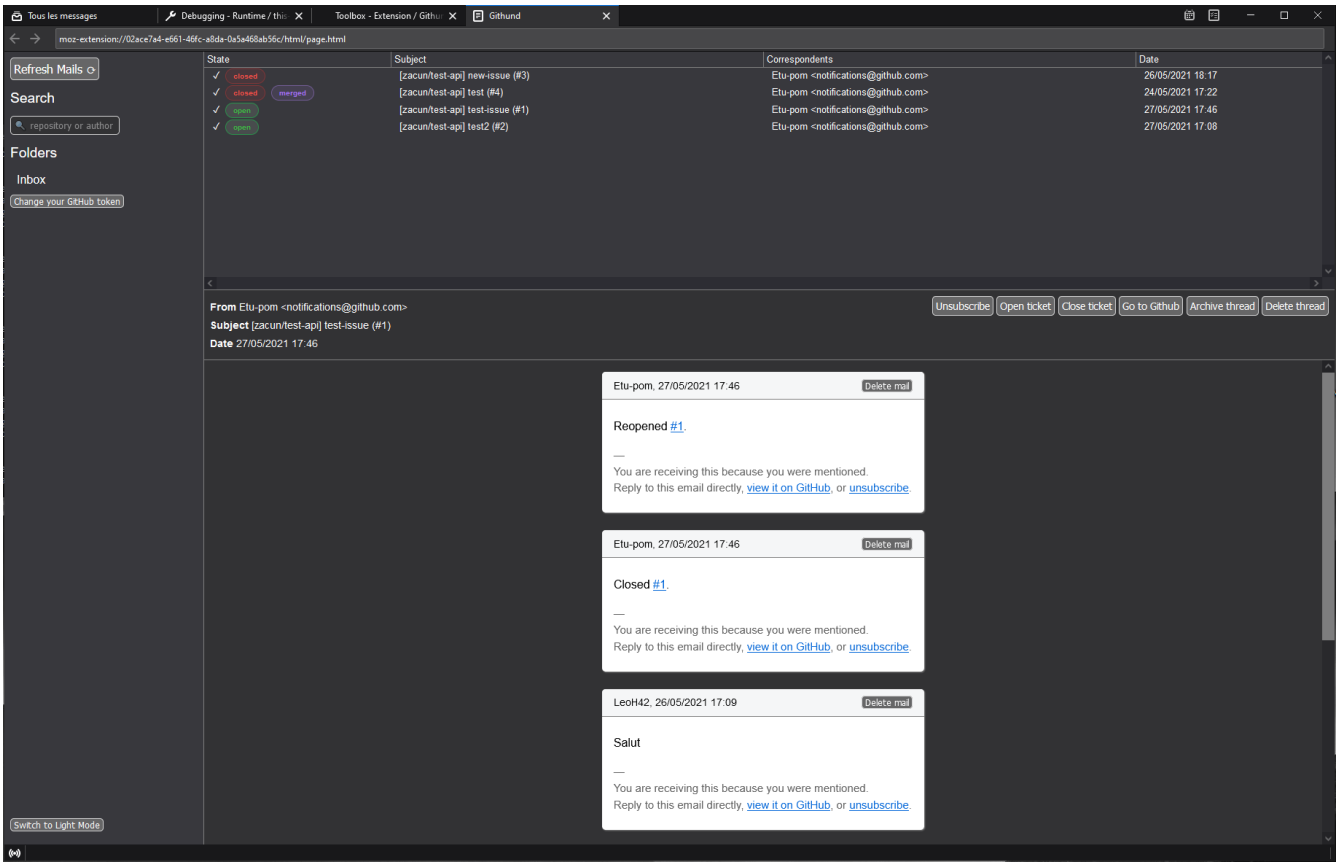

Figure 3 : Mode sombre de l'interface de l'extension

5.3.3 Tris

Par dossier : l'utilisateur peut choisir de n'afficher que les mails venant seulement d'un certain dossier.

Par date : l'utilisateur peut classer les threads par ordre croissant ou décroissant (les mails d'un même threads sont forcément classés du plus récent au moins récent).

Mais aussi à l'aide d'une barre de recherche recherchant dans les parties "subject" et "correspondents". Il a fallu implémenter ce genre de fonctionnalités basiques car l'interface custom ne possède aucune de ces fonctionnalités.

5.3.4 Lien avec GitHub

L'extension est avant tout un moyen de gérer les mails de notifications en provenance de GitHub et de réaliser certaines actions sans avoir à passer par le navigateur web et de se connecter au site. Pour cela, nous utilisons la fonctionnalité Fetch de JavaScript afin de facilement requêter l'API REST de GitHub.

Pour le moment, cette API sert exclusivement à ouvrir ou fermer une issue GitHub mais avec plus développement on pourrait imaginer une vraie interface d'administration permettant de réaliser bien plus de tâches sans avoir à passer par le site web.

Nous utilisons aussi l'API afin de récupérer "l'état" d'un thread : c'est à dire si la discussion à laquelle il est lié est encore ouverte, close ou si les changement demandés ont été "merged" (ils ont été ajoutés au projet principal) dans le cas d'une pull request (lorsqu'un contributeur demande à ce que le code qu'il a réalisé sur une application soit vérifié et ajouté à celle-ci). Une fois l'état récupéré nous pouvons l'afficher à l'aide d'un indice visuel à côté du sujet du mail.

L'abonnement/désabonnement d'un thread ne passe pas l'API car les mails reçus ne procurent pas les informations nécessaires, cependant ces mails peuvent aussi fonctionner en mode "mailing list", en effet, ils contiennent les informations d'une adresse mail automatique à contacter afin de se désabonner. Nous utilisons donc cette méthode : l'utilisateur envoie un mail à l'adresse fournie et se désabonne de cette manière.

En ce qui concerne l'authentification, l'utilisateur doit créer un token sur GitHub en spécifiant les droits qu'il souhaite puis l'ajouter dans l'extension afin que cette dernière puisse faire des requêtes. L'extension permet de remplacer le token ajouté ou de le supprimer si souhaité (en mettant un champ vide lors de l'édition du token via un bouton présent sur l'interface).

5.3.5 Actions réalisables sur les threads

Une série d'actions est disponible sur les threads affichés via des boutons.

- L'abonnement/désabonnement
- L'ouverture/fermeture d'un ticket
- Se rendre sur la page web de thread en question
- L'archivage du thread et des mails associés
- La suppression du thread et des mails associés (il est aussi possible de supprimer les mails de manière individuelle

5.3.6 L'interface graphique

L'interface graphique est entièrement custom. Nous avons essayé de nous rapprocher du style de ThunderBird (cf. figures 2 & 3 page 5).

Nous avons ajouté la possibilité de changer le thème de l'interface entre un mode clair et un mode sombre. Nous avons aussi développé la possibilité de redimensionner les trois panels principaux mais aussi les colonnes dans la partie contenant la liste des threads.

5.4 Contraintes rencontrées

De nombreuses contraintes ont été rencontrées durant le développement du projet, notamment dues aux API qui ne permettaient de réaliser simplement ce que nous souhaitions.

L'extension se base sur les threads mais il est impossible de récupérer l'id de ces threads via les mails reçus afin d'effectuer des opérations via l'API de GitHub, il a donc fallu utiliser d'autres moyens comme l'utilisation de mailing list.

Les autres contraintes viennent principalement de l'API de Thunderbird. Notre extension demandait de changer l'interface graphique de l'affichage des mails en mode thread (groupement de mails) mais cela n'étant pas possible via l'API seulement, il aurait fallu faire appel à des fonctions beaucoup plus poussées venant à modifier le code source du logiciel ce qui aurait demander un approfondissement dans l'étude des technologies et donc de passer encore plus de temps dessus.

5.5 Pistes d'amélioration

A ce stade, l'extension n'en est qu'à sa première version, on pourrait même dire qu'elle en est au stade de bêta. Il y a encore beaucoup à faire afin de délivrer une extension encore plus professionnelle.
Voici quelques pistes d'amélioration :

- Un vrai système d'authentification via OAuth pour lier l'extension et GitHub (actuellement l'utilisateur doit générer lui-même un token sur Github puis le donner à l'extension)
- Une synchronisation plus poussée des données (notamment des mails) de l'extension avec celles de ThunderBird afin de ne pas avoir à parcourir tous les mails à chaque démarrage.
- Amélioration des performances (temps de chargement) notamment via une meilleure synchronisation mais sinon difficilement améliorable car nous avons besoin de parcourir chaque mail pour les trier et pour ce faire nous devons faire des appels bloquant à l'API de ThunderBird.
- Rendre des fonctionnalités plus "souples" comme par exemple n'afficher des boutons que lorsqu'une action est réellement possible (bouton open/close seulement si le thread appartient à l'utilisateur)
- Création d'un menu contextuel (clic droit) personnalisé

6. Conclusion

Nous avons choisi ce projet car le développement d'une extension sur un logiciel tel que Thunderbird est très formateur et valorisant. De plus, nous avons eu à utiliser différentes API, ce qui est une pratique courante dans le monde du développement et qui nous sera utile par la suite.

Au niveau technologique, et particulièrement au niveau du langage Javascript, ce projet a été très enrichissant. Que ce soit en termes de syntaxes, d'opérateurs, ou même de façon de coder, nous avons appris ou consolidé de nombreux concepts et principes de programmation. Ce projet nous a permis de découvrir davantage les possibilités qu'offre ce langage et comment s'en servir au mieux.

Nous avons pu aussi mettre un premier pied dans le monde des extensions : même si toutes les extensions ne se développent pas de la même façon suivant le logiciel concerné, ce projet nous a rappelé que tout n'était pas réalisable et qu'il y avait des limites (parfois contournables, parfois non) : une bonne étude de faisabilité est donc fortement recommandée avant de se lancer.

Au niveau humain, ce projet a été très instructif. Même si nous n'étions que 2, nous avons dû coordonner notre travail et construire ce projet en équipe, notamment lors de la partie étude des technologies, où chacun était chargé d'expliquer à l'autre sa partie. La plupart des objectifs de départ ont été réalisés mais nous restons toutefois avec un sentiment d'inachevé. En effet, nous sommes convaincus qu'avec plus de temps consacré à ce projet, nous aurions pu réaliser la totalité des objectifs fixés, de manière plus souple et plus profonde permettant de livrer une extension bien plus complète et solide.

Il est aussi intéressant de noter que lorsqu'on développe un projet open source ayant pour but d'être livré publiquement, ce sentiment d'inachevé devient encore plus grand par peur de ne pas avoir été à la hauteur.

Pour conclure, ce projet a été particulièrement intéressant à étudier et à réaliser. Il nous a notamment permis de toucher à certains aspects de la réalisation d'un projet, comme la documentation et l'étude d'une API, auxquels nous ne passons d'habitude pas beaucoup de temps. Ce projet a globalement été une belle expérience à réaliser en équipe et nous remercions notre encadrant de nous l'avoir proposé et de nous avoir accompagné durant cette période.