

はじめに

- Meetup当日に質問したい事、議論したい事を各見出しの下に書いて下さい。適切な見出しが無い場合は、「その他」に書くか、「見出し1」のレベルで新しい見出しを追加して下さい。
 - 挙がっている質問や議題について何か情報を持っている方は、ぜひコメントを追記して情報を共有して下さい。
 - 「自分もこれに関心がある！」という話題には、「(名前)」を後に付け足して下さい。Meetup当日は、関心を持っている人が多い話題から優先的に取り上げる予定です。
 - 「これは良いね！」という情報には「👍」マークを付けてもいいかもしれません。
 - この資料に書かれた内容は、議事録代わりとしてイベント終了後に公開される可能性があります。機密情報は含めないで下さい。
-

XUL拡張機能からの移植・移行について

- これをやるのにこんなAPIが欲しい、という提案はどうすればいい？ どういう提案なら受け入れてもらいやすい？ (Piro)
 - 却下されやすいものの傾向 ([design-decision-deniedとなったbug](#))
 - XULアドオンでのやり方(生のXUL)に1対1で対応するようなものはまず駄目。FirefoxのUI上の生のイベントを拾う、XUL要素の属性値を変更する、など。
 - 同期的なAPIはまず駄目。原則として非同期。
 - スcopeが絞り込まれていない提案は駄目。userChrome.cssと同じことがしたい、userChrome.jsと同じことがしたい、など。
 - 実際に受理されたものの傾向 ([design-decision-approvedとなったbug](#))
 - Chromeに同じAPIがあって互換性が崩れている、という物はまずChromeに合わせる形で承認される傾向あり。
 - [1376251 - Allow sidebar extensions access to native tab context menu](#), [1396031 - Create browser.menus.show\(contextType, parameters, contextParameters\)](#);: 既存APIに似た設計での提案だと承認されやすい。

- [1352598 - Add an API to list installed search engines](#):「検索エンジンを管理するAPIが欲しい」というざっくりした提案が元になり、その一部を切り出す形で立てられたbug。スコープが絞り込まれていると承認されやすいという例。
 - [1409262 - Updated openerTabId is not notified via tabs.onUpdated if it is changed by tabs.update\(\)](#):既存APIのルールから考えて当然そうあるべき、と納得してもらえると承認されやすい模様。
- 他のアドオンとの連携のための、`runtime.sendMessage()`で呼べる/`runtime.onExternalMessage`で受信できるメッセージベースでのAPIの設計のベストプラクティスは？(Piro)
 - 参考:[ツリー型タブが提供しているAPI](#) 設計の方針は[WebExtensions移行時の解説記事](#)にも記載あり(Piro)
- Web ExtensionsのAPIを提案するBugをFileしたひとはどのくらいいますか？(Piro)
 - 2人(saneyuki_s / Piro)
 - Quantumの影響で、パフォーマンスに影響するものはRejectされるかも(nakano)
- [OSS Gate](#)という、OSSにバグ報告をしたりプルリクエストを送ったりするのを経験者のサポートを得ながら実際にやってみるワークショップがある。これに参加して、Bugzillaに報告する機会にして貰えれば幸いです。(Piro)
 - とはいえBugzillaを使ったコミュニケーションはGitHub上でのコミュニケーションと比べて独特な部分が色々あるので、Bugzillaに特化したOSS Gateのような回をTokyo WebExtensions Meetupの中で儲けても良いかも知れない。(Piro)

Chrome Extensionからの移植について

- ChromeとWebExtensions両対応ってどれくらい意義があるの？ 両対応するとどれくらい影響があるの？(Piro)
- Chrome 拡張機能作者を取り込むには？ 彼らの活発なコミュニティはどこ？(Kohei)
- APIの違いだけでなく、一般的なブラウザ間の非互換にも気をつけないといけない。(Piro)
 - 例えば[KeyboardEvent.keyCodeの違い](#)は深刻なので、[KeyboardEvent.key](#)もしくは[KeyboardEvent.code](#)を使うことが強く推奨される。(keyCodeなんて使ってるのはむしろレガシーなXULアドオンから移植した場合か？)(Piro)
-

拡張機能の新規作成(開発開始)時

- 新規に拡張機能を作り始める時のrails newや、バックグラウンドページを追加する時などのrails generate scaffoldのような仕組みは無いの？(Piro)
 - 個人的には、既存の自作アドオンの中で構成が似ている物からファイルを拾い上げて書き換えて作っています。(Piro)
 - 現状では未対応([web-extの該当するissue](#))
-

拡張機能の開発時・デバッグ時

(関心を持っている人が多いようなので当日はここを中心に話していこうかと思います:Piro)

- 皆さんの開発スタイルを知りたい(Piro)
- デバッガーを使ってる方どのくらいいますか？(Piro)
 - (会場では挙手2名ほど)
 - async/awaitを使うのでコンソールログ出したほうが早い(saneyuki_s)
- about:debuggingから開いたデバッガのコンソールで、「フィルタ」に入力した内容を含む行だけを絞り込んで表示するのではなく、通常のページ内検索のように見つかった行までスクロールするようにはできないの？(Piro)
 - 直しましょう！ > Firefox 開発陣
(たぶんこれ。https://bugzilla.mozilla.org/show_bug.cgi?id=992594)
- Reactとか使って拡張機能を開発している方の知見を聞きたいです(Piro)
 - 大量のリストを扱ったり、設定画面を作るのは楽です。
- Babel使ってます？(Piro)
 - 自分はbabelでES6のtranspileとjsxのtransformをして、Webpackで一つの.jsにまとめてます(ueokande)
 - Gecko上でほぼ最新のJSの仕様をそのまま使えるのにトランスパイルの必要そもそもあるの？(Brian)→まだ入ってない機能もあるし、だいいちType Script等の型のある言語で書きたいというニーズがある(saneyuki_s)→WE APIでPromiseが返される事が多くなり、生の値とPromiseを取り違える場面が増えたので、型は確かに欲しくなる(Piro)

- eslintやprettier使ってます？(Piro)
 - web-ext lintはmanifest.jsonのチェックもしてくれます。
 - JSON だけなら Firefox で直接開けば文法エラー確認できますよね (Kohei)
- web-ext buildとトランスパイラやWebpack等はどうやって連携するのがベストプラクティス？(Piro)
 - 連携の順番は意識しなくても良いと思う
- Windowsでの開発の仕方のベストプラクティスは？ web-extなどCLIのツールはいくつかあるが、Windowsではどう実行するのがおすすめ？(Piro)
 - 自分は、web-ext runだけはWindows版Node.jsのコンソールでインストールしたものを使っていて、それ以外のビルドやeslintなどはWSLのUbuntuで実行してます (Piro)
 - [MozillaBuild](#)便利ですよ (saneyuki_s)<https://design.firefox.com/photon/welcome.html>
 - ファイルの変更を監視して自動再読み込みや自動トランスパイル、のような事はWSLでもできるの？(Piro)→WSLでもできる。それができるようになったというのが最近の更新でアピールされていた。(kato)
- Manifest.jsonの編集もdebugger windowでできるように👍
- Iconなんかはブラウザからドラッグ & ドロップして追加したい👍
- 使用したAPIに対応するpermissionsは自動でmanifest.jsonに追加して欲しい👍
 - 不要なpermissionsもついでに削除
 - この機能を呼ぶにはこのpermissionが必要です、とwarningで教えてくれる開発専用モードみたいなものがあるとよい？(Piro)👍👍
- chrome.~とbrowser.~どっち派？(Piro)
 - 自分はbrowser.~派(Promise好きでコールバック関数は使いたくない)(Piro)
 - もっと言えばasync/await多用する派(Piro)
 - async/await 大変便利。[Nightly Tester Tools](#)でも多用してます(Kohei)
 - Chromeでもbrowser.~を使えるようにするPolyfillのためのライブラリがあるので、chrome.~はもう誰も使ってない疑惑(少なくとも会場にいた人はそうだった)
 - <https://github.com/mozilla/webextension-polyfill/> のプルリクが滞留しているのが気になっています。(熊本)
- https://developer.mozilla.org/en-US/Add-ons/WebExtensions/manifest.json/browser_action のbrowser_styleの扱いがショボいけどもう少しなんとかならないものか (saneyuki_s)👍

- そもそもこれが何なのかよく分かっていなかった。HTMLの要素に特定の名前のclassを指定するとブラウザのUIと同じ見た目のスタイルが適用されるという物だとの事。ちゃんとHTML側にclassを指定していないと意味がない。(Piro)
- <https://firefoxux.github.io/StyleGuide/>には 推奨とあるけれども、後者には特にコード例がない問題
- これ本当に将来的にメンテされるん？→絶対放置される奴や(Piro)
- てかWeb Componentがshipされたらそっちに切り替えてほしい気持ちがあります
👍

拡張機能の翻訳(言語リソース)について

(次点。外部協力者からのコントリビュートを受けるきっかけになりやすいポイントとして。:Piro)

- ロケールを追加するPRをくれた人が、英語のロケールにだけ新規の項目を追加されても分かりにくくて辛い、というコメントをくれた。翻訳で自分が分からない言語のリソースはどう管理してる？(Piro)
 - GitHubでリビジョンやタグ間の差分を出すリンクの例
<https://github.com/piroor/treestyletab/compare/2.4.8...master> や、[Java製のロケールエディタ](#)や[Chrome拡張機能向けのSublime Text用プラグイン](#)を紹介した。(Piro)
- 以前(Firefox 57が出た前後の頃)はできていたロケールのテストが、いつからかできなくなった。[説明](#)の通りにやっても、ロケールが変更されなくなってしまった。(Azicore)
 - 文字列型の設定「intl.locale.requested」を作成して「ja」などの値を設定するとアドオンマネージャ上での表示名と説明文だけはロケールが変わった。e10sのせいかと思ひ「extensions.webextensions.remote」を「false」にしても状況は変わらず。regressionかも？(Piro)
 - おそらくリグレッション。自分も遭遇した。about:addons 内の紹介文だけは問題なかったと思う(Kohei)
 - Filed : https://bugzilla.mozilla.org/show_bug.cgi?id=1443068
- 多言語化はどこまで考慮するのがよいのでしょうか？また、母国語以外の言語(英語含む)はみなさんどのように対応しているのでしょうか？(機械翻訳？自力？他人からのPR？)(Azicore)

- 英語と日本語は自分で用意して、それ以外はプルリクエストがあればマージという方針。母語でない言語を機械翻訳でやるのは、誤訳に気付けないし、翻訳語の文章の権利が不明瞭という問題もあるので、やらないのがオススメ。(自力翻訳の補助に使うのは有り)(Piro)
- ネイティブのレビューが入ったほうがいい(marsf)
- プルリクエストに頼る場合、文言を大幅に変更したり追加したりした場合は、変更分・不足分はどうしていますか？(Azicore)
 - 新規追加分については前述の通り。真偽値の設定の意味が逆だったので直した、のような大きな変更は、他の言語リソースからその部分を削除(そうすると既定の言語にfallbackしてくれるので)して、次のPRが来るまで未訳のままにする事が多いです。間違った翻訳があるよりは、読める人が減っても正しい情報が表示されている方が良いという考え方。(Piro)
- `gettextize` みたいなコマンドが欲しい
 - ソースファイル中の `getMessage()` を `enumerate` して `messages.json` を作るようなやつ
 - JSのファイル中に埋め込まれる文字列は、画面表示用の物とそうでない物との見分けが付かないので難しそう。HTMLファイルに書かれたテキストノードを収集して `messages.json` を作るくらいはできそう。(Piro)
- パネルやアドオン画面など HTML コンテンツをローカライズする方法を標準で提供して欲しい。今のところスクリプトで動的に書き換えるしかない(参考:[Piroさんのライブラリ](#)、[Nightly Tester Toolsのコード](#))(Kohei)
- AMOのサイト上で翻訳→ロケールを提供ができるといいね(Piro)
 - こういうライセンスで同意した物と見なす、みたいなルールは必要そう(Piro)
 - Mozillaの公式ローカライズツール [Pontoon](#) が活用できれば非常に便利(Kohei)
 - そこへ誘導する「この拡張機能を翻訳する」のようなラベルのリンクがあればいいのかも(Piro)
- 用語の統一、UI要素を何と呼ぶか、英語や日本語でどう表現するのか？→Chrome拡張機能として作られてFirefoxに移植された物はChromeの語彙、Firefox拡張機能として作られChromeに移植された物はFirefoxの語彙になっている事が多い模様。→定番の単語については[翻訳チームの語彙表](#)がある。

拡張機能の動作テスト、実装のテスト時

- ユニットテストのベストプラクティスは何がある？ (Piro)
- chromeオブジェクトをmockできる[sinon-chrome](#)の実用性はどうか？ (ueokande)(Piro)
- TravisCI等でもUIを含むテストってできるの？ (Piro)
 - 手前味噌ですが、自分はe2eテストの自動化を始めてます
<https://i-beam.org/2018/02/25/webextensions-e2e-testing/>
(ueokande)。👍👍
 - コンテンツ領域のテストはできるが、サイドバーやツールバーのパネルについてはやった事が無いので分からないとのこと。でもコンテンツ領域内にiframeを作って読み込ませればなんとかなるのではないかという気もする。(Piro)
- 皆さんドッグフーディングどうやってやってますか？ (Piro)
 - 自分は[web-extを改造して、普段使いのプロファイルに対して複数の開発中アドオンを一時的なアドオンとして読み込むようにした物](#)を使っています。トランスパイラを使わない方針で開発してないと使えないけど。(Piro)
-

拡張機能のリリース時

- gem releaseやnpm publishのように(これらを使うと、rubygems.orgやnpmjs.org上のページの更新まですべて自動で済ませる事ができる)、addons.mozilla.orgのページの更新まで含めてリリースを一発でやる方法ってあるの？ (Piro)
 - 自分は現状ではaddons.mozilla.orgに手作業でファイルをアップロードした後英語と日本語のリリースノートを手動で設定しているのですが、面倒です。(Piro)
 - [web-ext](#) はどうですか？ (daisuke)
 - buildとsignは自動化できるけどAMOのページの更新まではできないのが難点 (Piro)
 - web-ext なりでページの更新まで自動化できると良い？ はたまた AMO ページ内で全部簡単にできるべき？ (daisuke)
 - gemやnpmに倣うと、manifest.jsonのdescription(____MSG____でローカライズしてあるなら各言語版も)を自動で拾ってAMOの短い説明文に

自動で流し込むくらいはしていい気がする。長い説明文は
_locales/XX/description.txt、バージョン情報は
_locales/XX/releasenotes/XX.txtのように、ファイル名で規約を設けてお
いてファイルがあればその内容を使う、みたいになってると悩まなくて良い
(設定より規約)。Chrome1にはそういう仕組みが既にあったりするのだろう
か？(Piro)

- Native Messaging Hostはどうやって開発・リリースするのがおすすめ？(Piro)
 - 自分はWindows用の物は、Goで開発してインストール用バッチファイルと一緒にGitHubのReleasesで配布してます(Piro)

リリース後の宣伝、マーケティングについて

- どこでアナウンスすれば知ってもらえるのか？(Piro)(takedajs)(Azicore)(ueokande)
 - オンラインメディアで取り上げてもらうための取り組み(Kohei)
 - Mozilla Japan コミュニティのソーシャルメディア・ブログを通じた宣伝(Kohei)
- 新機能追加のモチベーション(Kohei)
- AMO内での表示順位を上げるSEO的な方法はあるのでしょうか？AMO内で人気を獲得しやすい作り方、逆に順位を落としやすい作り方はあるのでしょうか？(Azicore)(Piro)
- 自分のアドオンは日本語と英語の2つのロケールで作ったにもかかわらず、ほとんどが日本語環境での利用者です。自分の英語力に問題がある以外で考えられる理由はありますか？(Azicore)
 - スクリーンショットは英語版にしておいた方が良いかも(その人に読めない言語のスクリーンショットしかない、その時点でそっ閉じされてしまう)。(Piro)
- 利用者からの意見や感想を募るベストな方法は何だと思えますか？(Twitter等のSNS、AMOのレビューコメント、PullRequest、アドオン内にアンケートフォームを組み込む、等)(Azicore)
 - 何のために募るのか次第ですが、バグの修正のためという観点だと、複数回の往復ができない場所や一方通行のフォームは不向きに思います。(Piro)
 - バグ報告もほしいですが、一番ほしいのは感想(ここが良かった、ここが気に入らない)です。AMOのレビューコメントは全然増えないので、より良い方法はあるのだろうかと気になりました。自分の場合、Twitterでバグ報告や感想をくれた方が結構いました。(Azicore)

- Twitterだと、[OR検索で名前のバリエーションを色々検索するタブ](#)をピン留めしておくと言及にすぐ気付ける(こういう時、検索性の高い名前だとノイズが少なくて良い)(Piro)
- レビュー欄にわざわざ投稿する・フォームをわざわざ送信するなど積極的にアクションを起こしてくれる人はそもそも少ないものなので、ユーザーの絶対数が少ないと仕方がないかも(Piro)
- バージョンアップ後の新機能などの通知。現在はユーザに通知されないので、新機能に気づかれない。(ueokande)
 - [Background scriptの初期化タイミングで、local storageに保存済みの設定値をコード上で定義している定数と比較して、値が異なっていたら通知用のタブを開く、という処理](#)を自前で書いています(Piro) 👍

その他のこと

- アドオンって呼ぶのと拡張機能って呼ぶのどっちがいいの(Piro)
 - 英語圏でも曖昧だが、アドオンは基本的に、拡張機能、テーマ、検索エンジン、言語パック、スペルチェック辞書、プラグイン(Flash Player)の総称(Kohei)
- 日本語のフォーラムなど気軽に相談できる場所があった方が良い?(Kohei)
 - Slackはスクリーンショットを貼ったりできて気軽に発言できる反面、誰もがMozillaのコミュニティそのものに関わりたいわけではないし、トピックを立てる形式ではないので、使いづらい面もある(Kohei)
- ドキュメントは十分か? 日本語化した方が読みやすい? ナレッジベースを作る?(Kohei)
- webextensionで定義したショートカットキーはchromeみたいにあとで変更できるようになると嬉しいです。(熊本)
 - ショートカットキーは、好みが分かれるところなので。
 - Chromeではインストール済みの各拡張機能が提供するショートカットを包括的に管理するUIがある。Firefoxにも欲しい。