

# WebビューアやURLから挿入ステップの使いどころの考察

# 目次

- Webビューア
  - Webビューアを使う目的
  - Webビューアのまとめ
- URLから挿入ステップ(curlコマンド)
  - curlを使う際に知っておくと良いこと1
  - curlを使う際に知っておくと良いこと2
  - curlで頑張って実装しないこと

# はじめに

- ・ 私が個人的に感じたことなので、参考程度に聞いて頂ければと思います。
- ・ お客様に納品することを前提に考えています。
  - ・ なるべく安定して使えるように。
- ・ 間違っている点があれば、ご指摘お願いします。。
- ・ 「私は〇〇を気をつけているよ。」という点があれば、教えていただきたいです。

# きっかけ



**Teruhiro Komaki**

@trhrkmk



Webビューアからアクセスするのは、良くない実装方法だと思っている。

もっと言うとFileMakerのcurlコマンドも、あまり良くないと思う。

LambdaとかCloud Function使うほうが良い。  
それが難しいなら、素直にコネクトを使えば良い。

午後6:54 · 2021年6月30日 · Twitter Web App

|| ツイートアクティビティを表示

6 件のいいね



Webビューア

# Webビューアを使う目的

- FileMakerの苦手な部分を補う使い方(主にJavaScriptライブラリ)
- ビューアとしての使い方
- スクレイピング(WebサイトにAPIがない場合)
- WebビューアのJavaScriptエンジンを計算に活用

# Webビューアを使う目的

## FileMakerの苦手な部分を補う使い方(主にJavaScriptライブラリ)

- グラフ -> Chart.js / Amazon QuickSight
- ピボットテーブルなどの計算 -> PivotTable.js / Amazon QuickSight
- ポータルでフィルタリングやソート -> Grid.js / Cheetah Grid / Grid関連
- ポータルの列の並び替え -> Grid.js / Grid関連
- カレンダー表示 -> FullCalendar
- QRコードやバーコードの生成 -> qrcode.js / JsBarcode

# Webビューアを使う目的

## FileMakerの苦手な部分を補う使い方(主にJavaScriptライブラリ)

- ドラッグ&ドロップで並び替え -> JavaScript(@p388cellさん)
- 音声を再生, カメラ -> JavaScript(若林さん)
- レスポンシブレイアウトのような表現 -> CSS
- お絵描きツール -> Fabric.js
- ICTツール -> Twilio



# Webビューアを使う目的

## FileMakerの苦手な部分を補う使い方(主にJavaScript)

- ・ 実現するには…
  - ・ JavaScriptのライブラリを利用
  - ・ アドオンの利用

# Webビューアを使う目的

## FileMakerの苦手な部分を補う使い方(主にJavaScript)

- 課題や考慮すること…
  - 各ライブラリのバージョンを最新にしていると動かなくなる可能性がある。
  - FileMakerのバージョンと「FileMaker.PerformScriptWithOptions」に注意。
  - 大きなバイナリファイルのやりとりやPSOSの引数の制限などにも注意。
  - リソースをCDNから取得していて、ロードに時間がかかる場合に「定義していない変数がある。」などのエラーが発生する可能性がある。

# Webビューアを使う目的

## ビューアとしての使い方

- ・ PDFやメディアなどのビューア
- ・ Googleマップなどの埋め込む機能があるWebサイトの表示

# Webビューアを使う目的

## スクレイピング(WebサイトにAPIがない場合)

- ・ 特定のWebサイトのソースを取得したい。
- ・ (例)取引先が公開しているWebサイトにログインし、CSVをダウンロードして、FileMakerに取り込みたい。

# Webビューアを使う目的

## スクレイピング(WebサイトにAPIがない場合)

- ・ 実現するには…
  - ・ MBSプラグインの活用
  - ・ 即時関数(即時実行関数式)の活用
  - ・ AppleScriptを活用
  - ・ GetLayoutObjectAttribute ( "Webビューア" ; "content" ) を活用

# Webビューアを使う目的

## スクレイピング(WebサイトにAPIがない場合)

- ・ 実現するには…
  - ・ スクレイピング用のツールやライブラリを活用
    - ・ Puppeteer(Node.js)、Selenium(Python)などを使う。
- ・ なぜ、Webビューアを使わないのか？
  - ・ Webビューアでアクセスして、正常に表示されないケースがある。
  - ・ CSVファイルをダウンロードして、そのCSVファイルにアクセスして…などの処理をFileMakerで書くのは大変。
  - ・ 実現したいことによるが、専用のツールを使う方が、情報も多く効率が良い。

# Webビューアを使う目的

## スクレイピング(WebサイトにAPIがない場合)

- ・ 課題や考慮すること…
  - ・ Webサイトのコードが突然変わることがあるため、エラー処理に気をつける必要がある。
  - ・ Windowsとmac、ディスプレイサイズによってレスポンスのソースが変わることがある。
  - ・ エラーが発生した場合、Slackに通知するなど、エラーに気づく環境を整える必要がある。
  - ・ Webビューアからのアクセスだと、Webサイト側で、アクセスを制御される可能性がある。
  - ・ 長期的に見て、Webビューア関連の更新時に、影響をうける可能性がある。
  - ・ 実装方法によっては、将来動かなくなる可能性がある。
  - ・ コードに認証情報などは保存しないこと。
  - ・ 規約を確認すること。数秒ごとにアクセスするなどマナーにも気をつけること。

# Webビューアを使う目的

## スクレイピング(WebサイトにAPIがない場合)

- 課題や考慮すること…
  - <https://www.digitalboo.net/post/3571/fm-wp-webview/amp>
  - <https://www.digitalboo.net/post/2358/fm16-notdrag>



# Webビューアを使う目的

## スクレイピング(WebサイトにAPIがない場合)

- どうやってFileMakerと連携するのか…
  - AWS Lambda や Cloud Functions を活用するとよい。

# Webビューアを使う目的

## WebビューアのJavaScriptエンジンを計算に活用

- ・ APIの戻り値のJSONをフィルターしたい。
- ・ 大きなJSONのパーズや計算を高速化したい。
- ・ 「FileMaker Data API を実行」の戻り値を活用したい。
- ・ 動的なJSON(Gmailのレスポンスなど)をパーズしたい。

# Webビューアを使う目的

## WebビューアのJavaScriptエンジンを計算に活用

- 実現するには…
  - WebビューアのJavaScriptエンジンはFileMakerよりも速いのか？
    - WebビューアのJavaScriptエンジンとFileMakerの比較。
  - 即時関数を活用する。
    - Webダイレクトでは機能しないかもしれない。

# Webビューアを使う目的

## WebビューアのJavaScriptエンジンを計算に活用

- 課題や考慮すること…
  - FileMaker.PerformScriptWithOptionsの使い方に注意。
    - オプションを理解すること。
  - 必要があれば、非同期関数を理解すること。

# デモ

- FileMakerでスクレイピング
- Puppeteerでスクレイピング
- JavaScriptエンジンとFileMakerの比較

# Webビューアのまとめ

- ・ライブラリを活用できるようになると、ものすごく便利です！！
- ・スクレイピングなど、外部のデータにアクセスするような場合は、Webビューアを使うことはお勧めしない。
  - ・HTMLを取得する程度であれば「GetLayoutObjectAttribute」で良いかも。
- ・少し勉強が必要ですが、スクレイピング用のツールやライブラリを活用することをお勧めします。
- ・JavaScriptエンジンは高速なので、大きめのJSONを取り扱う場合など、活用したい。
- ・即時関数が便利。

# おまけ

## BIツールを試してみても

- Amazon QuickSight(BIツール)を活用することが増えてきた。
- 安価で、Webビューアに埋め込みもできる。
- FileMakerは必要になるCSVファイルをS3にアップロードするだけで良い。
- QuickSightがS3のCSVデータを定期的に最新の状態に更新してくれる。
- 集計する機能が不要になることもある。
- <https://aws.amazon.com/jp/blogs/news/evolve-your-analytics-with-amazon-quicksights-new-apis-and-theming-capabilities/>

URLから挿入ステップ  
(curlコマンド)



# curlを使う際に知っておくと良いこと 1

- デバッグ方法について
- APIのログファイルは、別ファイルに分割する方が良い。
  - サイズが大変なことになる。
  - 大きめのJSONは、テキストフィールドに入れない方が良い。
- APIやJSONの練習には、以下のWebサイトがお勧め。
  - <https://httpbin.org>
  - <https://badssl.com>
  - <https://github.com/public-apis/public-apis>

# curlを使う際に知っておくと良いこと 2

- スクレイピングと同様で、APIの仕様が変更される可能性があるので、エラー処理を実装すること。
  - Googleのフリープランのアプリに回数制限がかかるようになったことなど。
- AWSの障害など、正常にAPIが実行できないケースを考慮すること。
- エラーに気づける環境を準備する。
- サーバーサイドで実行する場合、ローカルでファイルを開いておく必要があること。
- 規約を確認し、回数制限など確認すること。

# curlで頑張って実装しないこと

- シェルスクリプトでプログラミングをしているようなものなので、curlで頑張って実装するより、ライブラリを活用する方が効率が良い。
- APIのドキュメントにcurlのサンプルがない場合も多い。
- ライブラリのドキュメント、バージョンアップ時の移行方法など、一般的なプログラミング言語を一つ覚える方が効率が良い。
- 習得が難しいのであれば、Claris Connectなどを使う方が良い。

# デモ

- curlの --show-error オプション

# 終わり

- FileMakerやOSのアップデートと同様に、APIも定期的にアップデートされることを考慮しておく。
- (そこまで自信がない場合は)Webビューアを組み込みすぎると、大変になる可能性があるので、控えめに使う。
- お客様にも、技術的な背景を理解してもらう必要がある。