

CloudflareによるPoC制作

自己紹介

1. ITエンジニアではありません

- ・コロナ禍の在宅ワーク期間、通勤の時間が浮いたので、趣味でPythonの学習を開始。
- ・セキュリティとか、ネットワークとか、今だによく分からぬ。
- ・新規事業に於けるPoCレベルのものは自作可能。

2. データサイエンティストではありません

- ・Kaggleは、コロナ禍の在宅ワーク期間、時間があったので趣味の一環として遊びました。
- ・数学を勉強したのは、大学院時代が最後(20年以上前)で、高度な数学的な素地がある訳ではない。
- ・半導体技術者時代に「シックスシグマ」に触れて以来、データの活用自体には興味があった。

3. 主なキャリアパスは、半導体開発設計、コンサルタント、新規事業開発とマーケティングです

- ・グローバルメーカーで、開発・設計・品質管理・生産技術管理等に従事。
- ・コンサルタント時代は、金融業界、保険業界、メディア業界、IT業界で、様々なPJに従事。
- ・直近の10年以上は、新規事業開発(本業・副業含む)とマーケティング、海外市場開拓に従事。

なぜ、Cloudflareに目を付けたか？

1. そもそも、僕がしたいことは？

- ・新規事業開発のPSFフェーズに於けるPoCを自作したい。（このフェーズでは、出戻りが多いため）
- ・できれば、無料(or低料金)で、爆速で制作し、想定ジョブに対する仮説検証を行いたい。

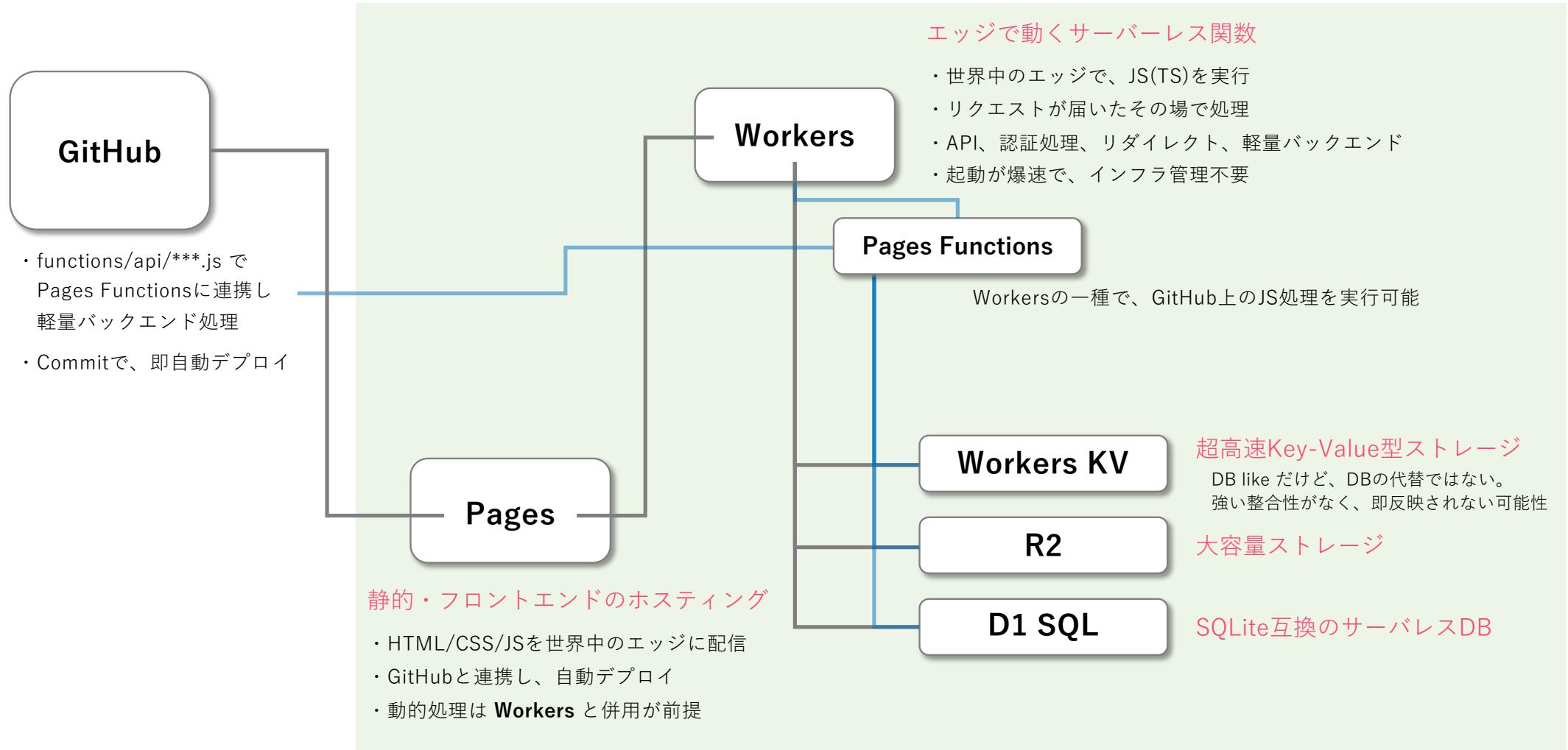
2. Cloudflareとは？

- ・無料でもPoC前提の「フロントエンド、軽量バックエンド、簡易DB、ストレージ」を単体で完結できる。
- ・GitHubと連携可能で、Commitしたら、すぐに自動デプロイ。
- ・Zero Trust等の認証機能もある。
- ・生成AI API連携ももちろん可能。

3. デメリット/制約条件は？

- ・Pythonは正式サポートされておらず、JavaScript/TypeScriptが前提。
- ・軽量バックエンドなので、機械学習関連のタスクは不可（Google Cloud Runで処理するAPI連携は可能）

Cloudflareのイメージ



家計簿アプリの概要

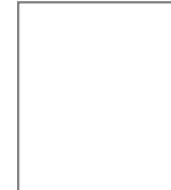
login.html



index.html



monthly_sum.html



admin.html



- ログイン成功後に遷移
- 購入内容、分類と金額を入力
- ヘッダーに月次累計画面と管理画面のリンクあり

- 過去に入力してきた金額を月次単位で、表形式で表示
- 各分類の小計も表示
- 月次単位の合計も表示
- 各項目に対し編集も可能
- 対象月の検索も可能

- Cloudflare KVを使用
- 購入分類名称の編集・追加
- 月次単位の一括削除
- 全てのデータの一括削除

- ・ID=kenta、Pass=kenkenとし、CloudflareのVariables and Secretsではなく、GitHub/リポジトリ/functions/api/login.js にハードコードしようか。
- ・style.cssは、別途、アップロードしたものを生かしてください。
- ・「購入内容」とは、商品名の時もあれば、「〇〇との食事代」「ランチ」「コンビニ」等、多岐に渡る。（ユーザーが自由に入力可能にした方が良い？）
- ・購入の「分類」とは、例えば「家賃」「電気代」「食費」といった、いわゆる『家計簿としてよくある分類』のこと（代表的なものを用意してください）
- ・管理画面では、本来、ユーザーとは別の管理者だけが入れる様にすべきだけど、面倒だから、ここでは全ユーザーが入れるようにする。
(本来は、管理者ID、管理者Passを用意した方が良いのはわかっている)
- ・シンプルにするため、Cloudflare KVを使い、DB like に使用したい。D1 SQLの方が良いだろうけど。

注意)

- Set-Cookieは、RFC的に、複数値をカンマ結合できないので、必ず append で複数行として返してください（この点を満たさないコードは不合格）
- PoCのため、ID/Pass は、login.js にハードコードで良いけど、cookie処理は正しく実装すること
- ログイン後に画面がログイン画面へ戻らないことを必須要件とします。
- /api/login 成功時は Set-Cookie: session=... がブラウザに保存されること。
- /api/me はその session cookie を読み、KV の session:<token> が存在すれば必ず 200 を返すこと（未ログイン時の 401）。
- 重要：Set-Cookie は必ず Headers.append("Set-Cookie", ...) で返すこと（カンマ結合は不合格）。
- 重要：共通レスポンスヘルパー（例：json()）を実装する場合、Headers を破壊せず Set-Cookie が確実にレスポンスに残ること（Headers を object 展開して消す実装は禁止。不合格）。
- 受け入れ条件：ブラウザNetworkで /api/login の Response Headers に set-cookie が存在し、直後の /api/me が 200 になること。

PoC制作の手順（コードは全て、生成AIに書かせましょうか）

Cloudflare環境
各種設定

- Cloudflareへの登録。
(基本、全て無料ですけど、Cloudflare R2だけは、クレジットカード情報の入力が必要です)

GitHub
リポジトリの作成

- Cloudflare Pages(Pages functions)と連携するリポジトリを用意。
(PrivateでもOK)

関連資料の用意
プロンプト

- 関連するコード(html, css, JS)は、全て生成AIに書かせるため、プロンプトだけでも良いけど・・・
簡単な要件を記載した資料を用意した方が、出戻りが少ないです。

ChatGPT-5.2
コーディング

- いわゆる「バイブルコーディング」?!
まぁ、PoCだし、雰囲気が分かれば・・・・

生成AIのコード
GitHubに貼付け

- 生成AIが指定するフォルダ構成、ファイルをコピペし、関連するリポジトリに保存(Commit)。
(CSSは結構時間が掛かるので、こだわる人は最初にデザインだけ生成AIとFixしておいた方が良い)

Cloudflare
設定・デプロイ

- Cloudflare側の設定(Pages、Pages Functions、KV、Bindings)を行い、
Pages&Workersからデプロイ処理を実行。

| 実際に、使用的するプロンプト

Cloudflareを使用し、アップロードした写真のような「家計簿アプリ(PoC)」を制作したいと思っています。

フロント → Cloudflare Pages

軽量バックエンド → Cloudflare Pages Functions(GitHubのリポジトリ/functions/api/直下のjsファイル

データ保持 → Cloudflare KVで、DB likeに

こんなイメージで考えています。

まずは、アップロードした写真を読み込み、要件を全て理解してください。

その後、

1. GitHubのリポジトリの、フォルダ/ファイル構成を可視化してください。(README.mdに図と構成を反映)
2. Cloudflare側の処理 (PagesのCreate Applicationとか、KVのCreate Instance、KVとのBinding等)も、
名称等も含めて提案してください。
(素人でも理解できるように)
3. 必要なファイルを全て出力してください。
(コピペしたいので、ソースコードは「全て」出力してください)

質問がある時は、ソースコードの出力を始める前に、ちゃんと質問してください。

尚、最後に、全ファイルについて、フォルダ構成を保持したまま ZIP ファイルとして生成し、
一括でダウンロードできる形で出力してください。

| 果たして、結果は・・・

全くコードを見ていないから、URLをクリックするまで分からない。

でも・・・

仮にエラーが起きたら、そのキャプチャーを元に会話し、

追加要件があったら、そのキャプチャーを元に会話し、

GitHubのファイルを追加・編集することで、簡単に更新できる！