

AIアプリケーションエンジニアとしての初めての仕事

AIアプリケーションエンジニアとしての初めての仕事

MCPの開発の経験を踏まえて

自己紹介

経歴

- 2025年9月からスタートアップでAIアプリケーションエンジニア
- エンジニア歴は5年程 (受託やSaaS開発)
 - バックエンド、フロントエンド、TerraformやGitHub Actionsなどを触っていました

前提: GPT Apps SDKの登場

- 2025年10月にOpenAIより、**GPT Apps SDK**が登場

特徴

- GPTに対してコネクターを作ることができる
- 外部リソースを参照することができる
- そこに独自のUIを導入することができる

条件負荷 1

入社2ヶ月目で開発開始したが…

必要なものが多い：

- CI/CD、インフラリソース
- OAuth連携の開発
- MCPサーバ
- UI

条件負荷 2

情報の少なさ

- 2025年10月にリリースされたGPT AppsのSDKによる開発で**情報が著しく少ない**
- フレームワーク的なものもなく、UIに合わせたものは存在しない
 - フレームワークも作られていたが、実用レベルで使えるものではなかった
 - 結構自力であれこれしないといけなかった

条件負荷 3

時間的制約

- 開発メンバーはほぼ**自分1人**
- 時間的制約が厳しかった
- 初速が大事で、**速度 > クオリティ**
- キャッチアップしながら整備をしつつやる必要があった

大変だったところ 1

開発環境の厳しさ

- UIの確認と連携部分はGPTを通してしかできない
- UIの変更をすると逐一GPTへ取り込んでリロードかけないと反映されない
- さらにそこからプロンプトを入力して変更を確認する必要がある
- GPTも開発が活発

大変だったところ 2

MCPの挙動への制御が難しい

- あくまでコネクターとして連携するので、プロンプトを挟むことができない
- せいぜいMCPのtool部分のdescriptionで制御の指針を示す程度

大変だったところ 3

サービスの制約との解消

- サービスの性質上、**画像の表示に制限が厳しい**
- GPTの画面上にあるiframeのsandboxからの参照を行うことが厳しい一面があった

意識するといいところ

意識すると良いところ 1

MCP開発の知見を共有

開発環境として以下のようなものを紹介する：

- **MCP Inspector**
 - npx で動かすことができるのですが、毎回コマンドを打つのも面倒なので、`package.json`にスクリプトとして、バージョン付きで登録しておくといいです。
- UIを共有できる機構を作る
 - 今回はUIのコンポーネントをビルドしたものを`assets`に配置して、Honoのエンドポイントから参照できるようにしました。

意識すると良いところ 2

ちゃんと役割分担できるようにタスクフォースを組む

- 最低限、インフラやCI/CDとMCPとOAuthを分けることはできたと思う
- リモートMCP開発に精通してゐる人が1人いると心強い

AI開発は楽しい

AI開発は楽しい 1

- このような機会がなければ、MCPを作ることもまだ先になっていたかと思うので
ここで開発できたのは良かった
- LLMのMCP Actionへの誘導についての考え方
日々の開発にも活かせそうな**ガードレールとしての考え方**に通ずるところがある

AI開発は楽しい 2

- AIアプリケーションエンジニアとしてのキャリアとして
MCPの開発も取り入れられたことは良い経験だった

ご清聴ありがとうございました