

目的

AI面接官Webアプリケーションにて取得した「設問」「面接者の回答」をGoogleスプレッドシートに自動記録し、評価・判定・スコア管理などを容易に行えるようにする。

基本設計

1. データ取得の流れ

ユーザーが面接画面で回答を入力（または音声認識で文字起こし）

HTML + CSS + JavaScript のフロントエンドから JSON 形式でデータを生成

fetch により Cloudflare Workers へ POST

Workers が API キー付きで Google Sheets API へ書き込みリクエストを実行

指定スプレッドシートに1行ずつ追記

2. インフラ構成

項目	内容
フロントエンド	HTML / CSS / JavaScript（Cloudflare Pages にデプロイ）
API中継	Cloudflare Workers
データベース	Google スプレッドシート（API v4）
認証管理	Googleサービスアカウント（JWT認証）
秘密鍵管理	wrangler secret により暗号化保持

3. スプレッドシート構成（フォーマット案）

A列	B列	C列	D列	E列	F列
日時	面接者ID	設問	回答	判定	スコア

（備考：任意で「面接官コメント」や「音声ファイルリンク」などの追加も可能）

詳細設計

1. Cloudflare Workers API設計

URL例：https://ai-interview-worker.example.workers.dev/submit

メソッド：POST

認証：なし（シークレットキーで制限 or 認証ヘッダー追加検討）

入力フォーマット（全て必須 / 型はstring）：

```
{
  "timestamp": "2025-04-13T15:00:00",
  "intervieweeld": "user-001",
  "question": "志望動機は？",
  "answer": "御社の理念に共感しました。",
  "score": "A",
  "judgment": "明瞭な回答"
}
```

レスポンス形式：

```
{
  "status": "success",
  "message": "Data appended to Google Sheets."
}
```

2. Google Sheets API 書き込み処理

使用API：spreadsheets.values.append

モード：RAW

インサート方法：INSERT_ROWS

JWT認証でOAuthトークン取得 → Authorization Bearerで送信

シートID・シート名・範囲はconfigまたはsecretにて切替（dev/prod）

3. 環境分離（開発／本番）

項目	開発環境	本番環境
Sheets ID	dev用スプレッドシート	本番用スプレッドシート
Worker URL	devサブドメイン	本番サブドメイン
APIキー	GOOGLE_KEY_DEV	GOOGLE_KEY_PROD

補足事項

Google Sheetsはログ兼DBとして柔軟な運用が可能

スプレッドシートが共有しやすく、評価フローにも流用可能

Cloudflare側でログ（console.log）も確認できる

面接のセッション単位管理が必要な場合は、別途セッションIDやフェーズ管理列を追加

必要に応じて、この設計からAPI仕様書やスプレッドシートテンプレート、Cloudflare Workersコードに落とし込むことが可能です。