

# Go + Geminiな サービスを爆速で作ってみた

2025.03.11 Fukuoka.go なんと10周年

近藤 うちお (@udzura)

SmartHR プロダクトエンジニア

こんにちは！



# 自己紹介

近藤 宇智郎 (@udzura)

株式会社 SmartHR (2024/11入社)

プロダクトエンジニア 労務基本機能担当

マイブーム: WebAssembly、中学受験

好きなGoのパッケージ: `x/time/rate`



# SmartHRの 技術といえば...



ですが今日は  
Goを使った話です！

2025-01-21

## 年末恒例第2回SmartHRハッカソン開催レポート —— テーマはLLM!

イベント・協賛 PdE devrel AI

2024年12月25日～27日の3日間にわたって、年末恒例第2回SmartHRハッカソンを行いました。このレポートでは、その模様をお伝えします。

### 目次

---

- [目次](#)
- [今年のテーマはLLM！](#)
- [事前にハンズオン](#)
- [開発](#)
- [発表会](#)
- [審査](#)

<https://tech.smarthr.jp/entry/2025/01/21/115835>

# 前提知識





# SmartHR Plus アプリの紹介

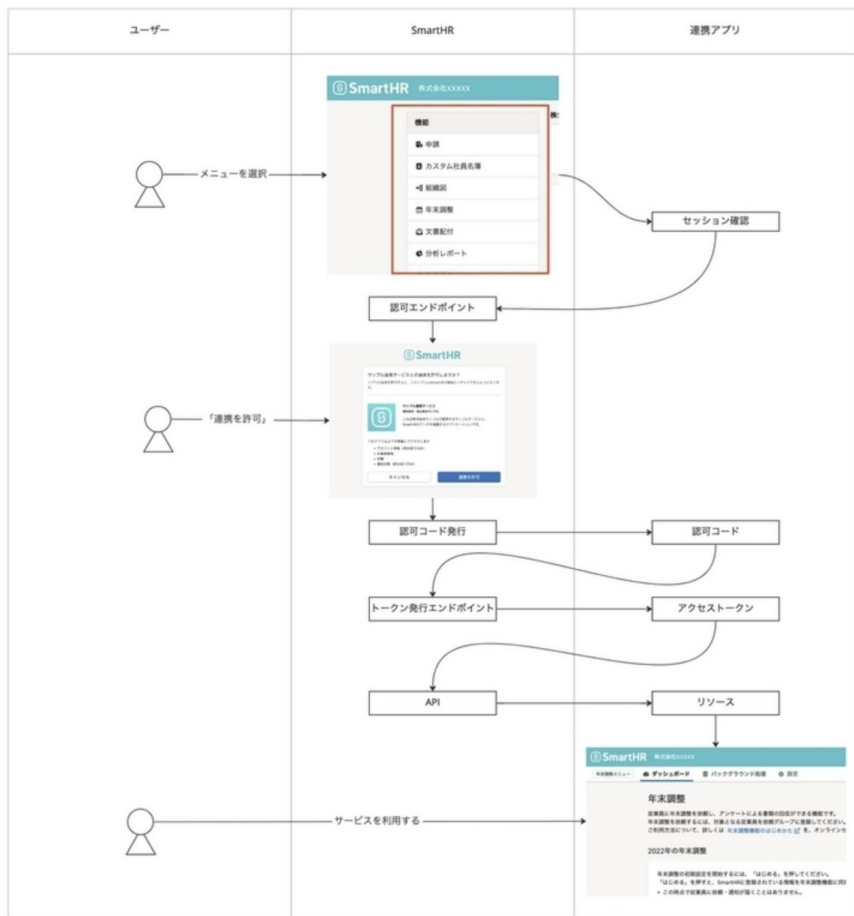
- ・ SmartHRにある、サービス公開のためのプラットフォーム
- ・ 特徴:

疎結合なアーキテクチャ

基盤部分(労務サービス)とAPIで連携できる

1機能に特化してリリースできる

参考: <https://tech.smarthr.jp/entry/platform-app-store>



認可コードフローの例

選びたければ、言語は自由  
※templateはある

じゃあ、Goで...  
作っちゃおっかな

# Why Go?



# 今回Goを選んでみた理由

## 1. LLM利用縛りがある

GeminiのSDKはPython or Goが充実してそう→Goで！

## 2. PoCレベルを素早く作りたい

DBすら不要なので、Railsよりさらに早く作れるかも？

フロントはSmartHR UIなので、APIのみなのもあった

ハッカソン参加！

どんなサービス？



# 採用面接のペインをなくすためのサービス！

- ・面接の動画をアップする

1. 面接内容の文字起こしをしてくれる
2. 面接官にフィードバックもしてくれる！

→ 評価がしやすくなり、面接の生産性も上がる！

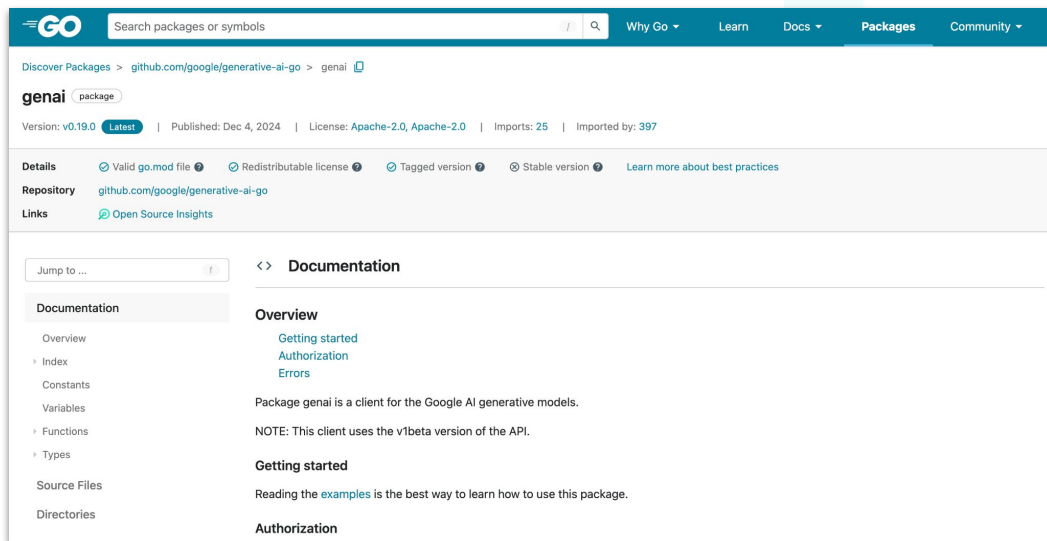
# 使った技術

## 主な技術要素

- ・言語: Backend = Go / Frontend = TypeScript (Next.js)
- ・AI: Gemini (Google Generative AI)
- ・(インフラはGoogle Cloudになるだろうけどまだ...)

# GoからGeminiを使う

- ・Goの公式SDKがよくできてるのですぐ使える



<https://pkg.go.dev/github.com/google/generative-ai-go@v0.19.0/genai>

# コードの例

```
import "github.com/google/generative-ai-go/genai"

func dummy(ctx context.Context) {
    client, _err := genai.NewClient(ctx, option.WithAPIKey(os.Getenv("API_KEY")))
    model := client.GenerativeModel("gemini-2.0-flash-exp")
    fileURIs, _err = uploadVideo(ctx, client, path, "video/mp4")
    _err = waitForVideosReady(ctx, client, fileURIs)
    session := model.StartChat()
    session.History = []*genai.Content{ ... } // Fileを添付
    resp, err := session.SendMessage(ctx, genai.Text(`
        まず、面接全体について、発話者を明示したうえでなるべく正確に文字起こしをしてください。
    `))
    for _, part := range resp.Candidates[0].Content.Parts { ... }
}
```

# コードの例

```
import "github.com/google/generative-ai-go/genai"
```

```
func dummy(ctx context.Context) {  
    client, _err := genai.NewClient(ctx, option.WithAPIKey(os.Getenv("API_KEY")))  
    model := client.GenerativeModel("gemini-2.0-flash-exp")  
    fileURIs, _err = uploadVideo(ctx, client, path, "video/mp4")  
    _err = waitForVideosReady(ctx, client, fileURIs)  
    session := model.StartChat()  
    session.History = []*genai.Content{ ... } // Fileを添付  
    resp, err := session.SendMessage(ctx, genai.Text(`  
        まず、面接全体について、発話者を明示したうえでなるべく正確に文字起こしをしてください。  
    `))  
    for _, part := range resp.Candidates[0].Content.Parts { ... }  
}
```

Modelを指定、  
ファイルアップロード

# コードの例

```
import "github.com/google/generative-ai-go/genai"

func dummy(ctx context.Context) {
    client, _err := genai.NewClient(ctx, option.WithAPIKey(  
model := client.GenerativeModel("gemini-2.0-flash-exp  
fileURIs, _err = uploadVideo(ctx, client, path, "video.mp4",  
_err = waitForVideosReady(ctx, client, fileURIs)  
session := model.StartChat()  
session.History = []*genai.Content{ ... } // Fileを添付  
resp, err := session.SendMessage(ctx, genai.Text(`  
    まず、面接全体について、発話者を明示したうえでなるべく正確に文字起こしをしてください。  
`))  
for _, part := range resp.Candidates[0].Content.Parts { ... }  
}
```

Chatのセッションを開始して  
ファイルを添付し、  
プロンプトを送る

## 留意点：動画はactiveになるのを待つ必要あり

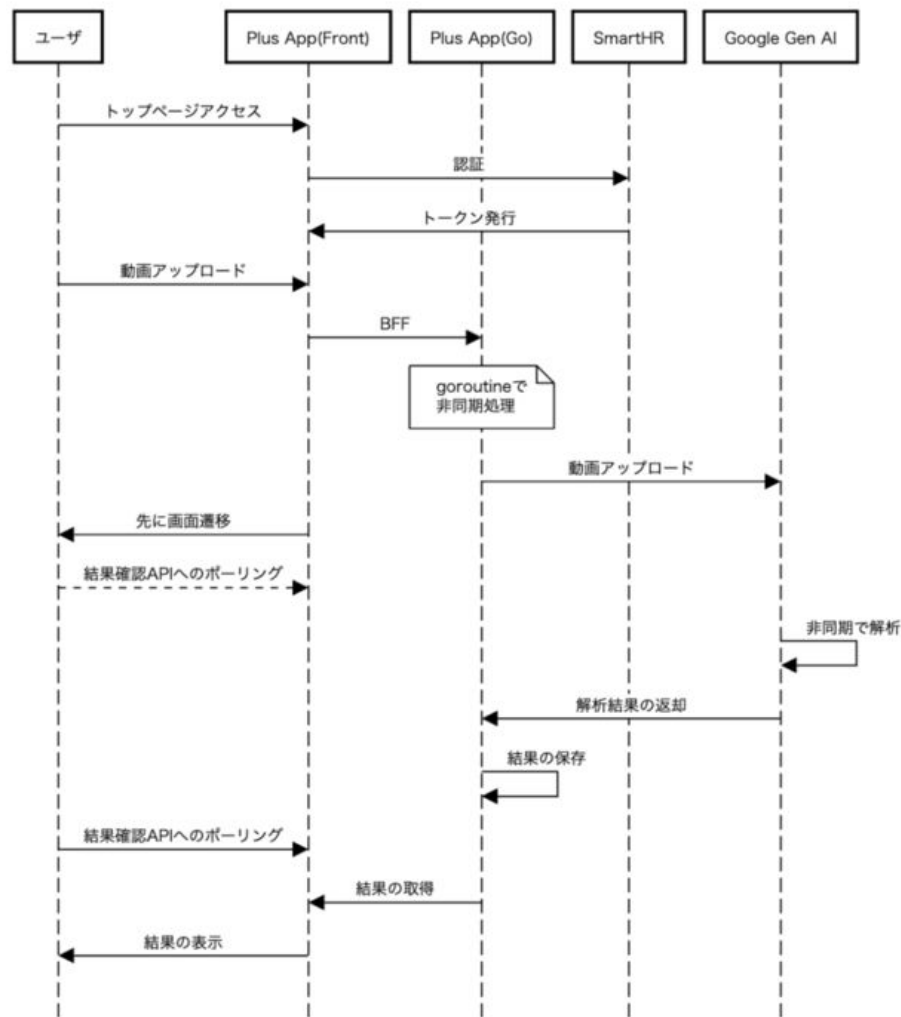
```
func waitForVideosReady(ctx context.Context, client *genai.Client, l []*genai.File) error {
    for _, f := range l {
        tick := time.NewTicker(1 * time.Second)
        processing := true
        for processing {
            select {
            case <-tick.C:
                file, err := client.GetFile(ctx, f.Name)
                if err != nil {
                    return errors.WithStack(err)
                }
                if file.State == genai.FileStateActive {
                    processing = false
                } else {
                    log.Printf("info: waiting... state: %s", file.State)
                }
            case <-ctx.Done():
                return errors.Errorf("Context canceled: %v", ctx.Err())
            }
        }
    }
    return nil
}
```



# 工夫1: 回答の スキーマを 選べる

```
var schemaForTranscript *genai.Schema = &genai.Schema{
    Type: genai.TypeObject,
    Properties: map[string]*genai.Schema{
        "文字起こし": {
            Type: genai.TypeArray,
            Items: &genai.Schema{
                Type: genai.TypeObject,
                Properties: map[string]*genai.Schema{
                    "発話者": {
                        Type: genai.TypeString,
                    },
                    "発言内容": {
                        Type: genai.TypeString,
                    },
                },
            },
        },
        "フィードバック": {
            Type: genai.TypeArray,
            Items: &genai.Schema{
                Type: genai.TypeObject,
                Properties: map[string]*genai.Schema{
                    "発話者": {
                        Type: genai.TypeString,
                    },
                    "フィードバック内容": {
                        Type: genai.TypeString,
                    },
                },
            },
        },
    },
}
```

## 工夫2: 非同期化 ... (無理やり)



# 画面のサンプル

## AI面接評価

ファイルの選択

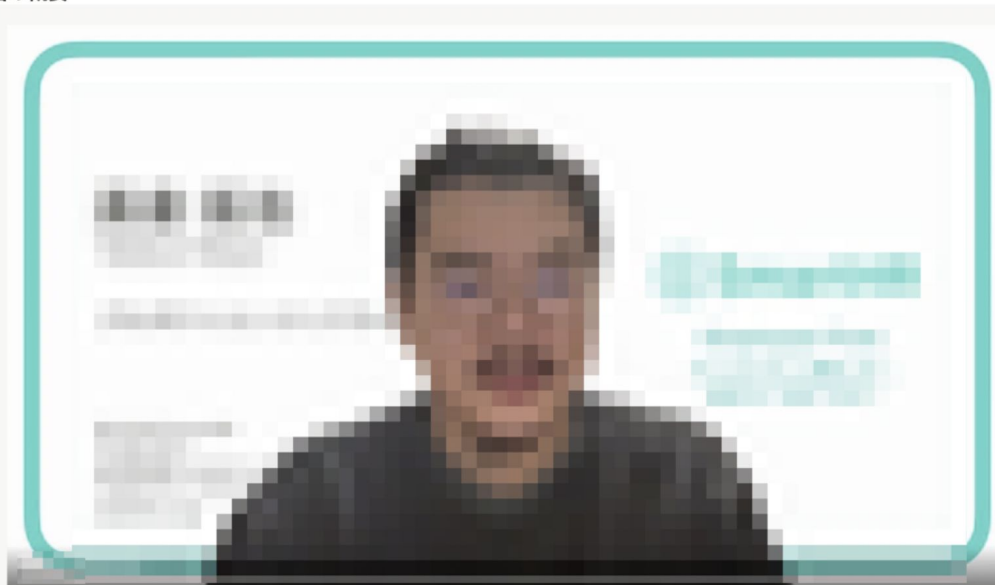
 面接の録画ファイルを選択

面接を評価する



## AI面接評価 解析結果

### 動画の概要



作成日: 2024/12/27 13:00:01

長さ: 02:00:00

### 文字起こしの結果



始まりました。

あ、こんにちは。

こんにちは。

Uchio KONDO

、こんにちは。

Uchio KONDO

はい、え、あるいは、えっと、面接を始めさせていただきます。

お願いします。

Uchio KONDO

じゃ、ちょっとさっきに我々側から自己紹介をさせていただきます。ええっと、ローム、ロームプロ、ロームシーン読んでしまいました。え、ロームプロダクト本部の、え、プロダクトエンジニアしています近藤と申します。社内ではうずらと呼ばれているので、お気軽に呼びください。あ、じゃ、[REDACTED]さんの方もお願いします。

はい、え、同じくプロダクトエンジニアの[ ]です。え、[ ]と呼んでください。よろしくお願いします。

Uchio KONDO

はい。

Uchio KONDO

えっと、ではそうですね、今日はまあ、5分弱ぐらい、えっと、ま、させていただこうかと思うのですが、え、ちょっと最初に  
さんの方から簡単な自己紹介からお願いできますか？。

まとめ



**SmartHR では  
Go も使えます！  
(※個人の感想です)**