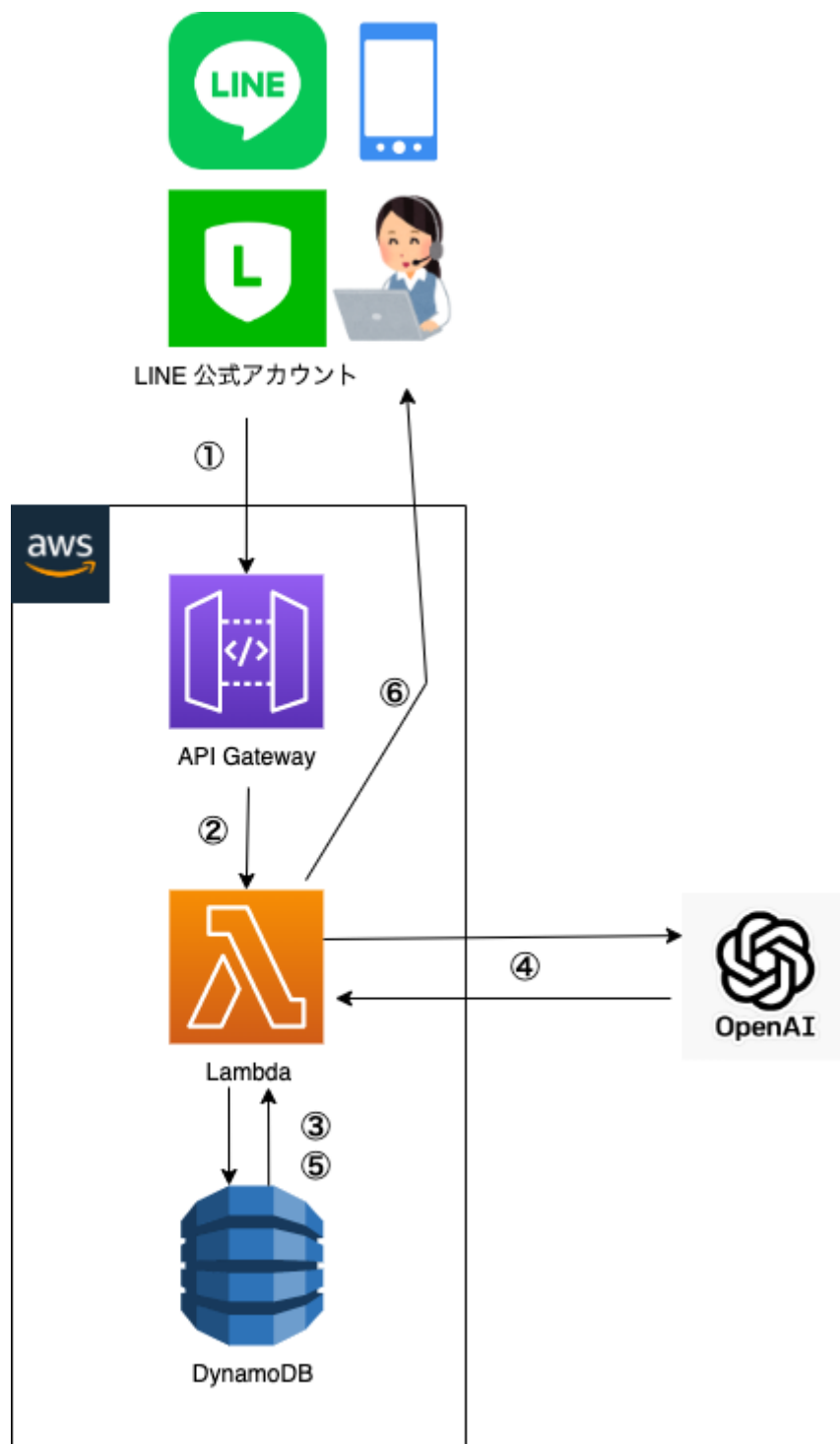


# LINE で OpenAI 社の API と連携するチャットボットを作ってみた (ChatGPT)

## 構成とやりとりの流れ



1. LINE 公式アカウントの **MessagingAPI** を用いて AWS の **API Gateway** に紐付いている URL に Webhook イベントを送信する
2. AWS **Lambda** に連携
3. AWS **DynamoDB** に会話を保存 + 過去の会話履歴を新しい方から適当な分だけ取得
4. OpenAI 社の API を叩いて返答を得る

5. AWS **DynamoDB** に返答を保存

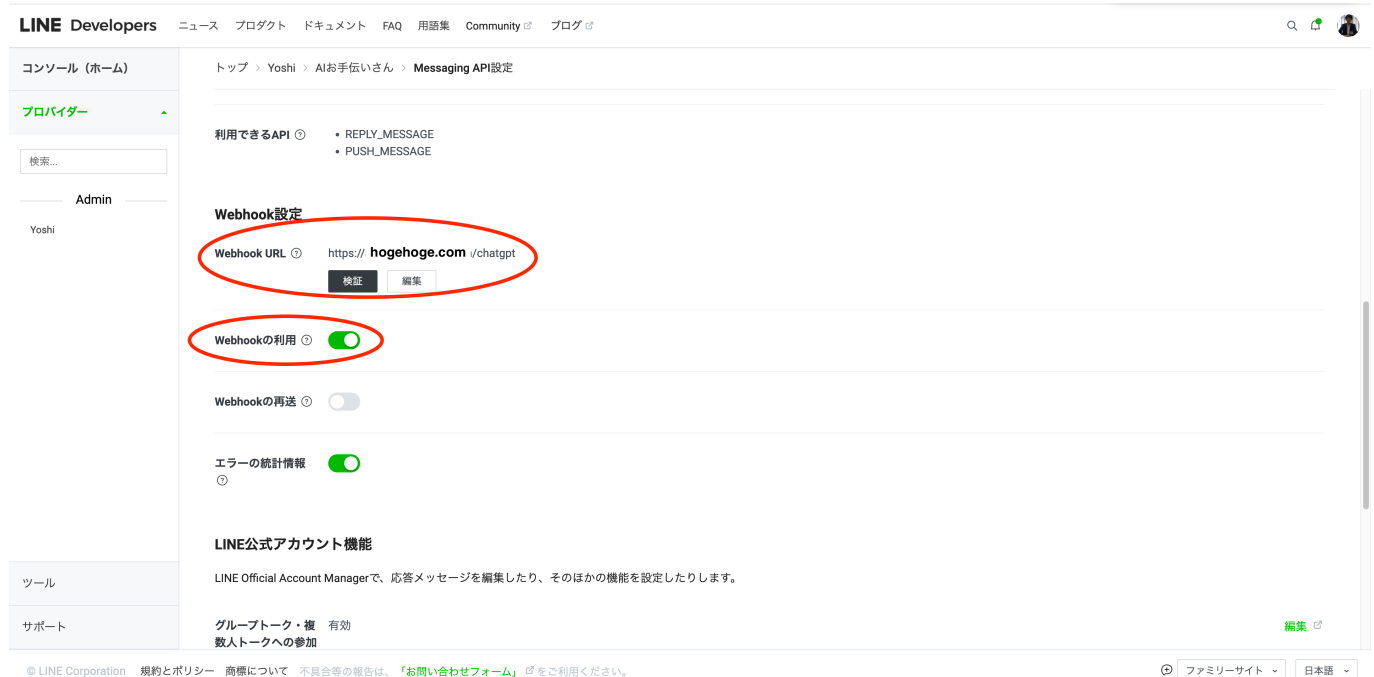
6. **MessagingAPI** の **Reply API** を叩いて返事を LINE に送信

## 1. MessagingAPI で Webhook イベント送信 -> 2. Lambda に連携

LINE 公式アカウントを作る



**Messaging API**設定にて AWS の **API Gateway** に紐付いている URL を Webhook URL に設定し、利用を有効にする



LINE 公式アカウントとのトークでメッセージを送信

今日

既読  
午後 10:39

AWS Lambda のメリット・デメリットを完結にまとめよ。

この発言内容が API Gateway 経由で Lambda に連携される

3. DynamoDB に会話を保存 + 過去の会話履歴を取得

Lambda で受け取った発言内容をまずは DynamoDB に保存 + 過去の会話履歴を取得する

aws

サービス

検索

[オプション+S]

AWS Cost Explorer

EC2

CodeBuild

Elastic Container Service

CloudWatch

Route 53

IAM

API Gateway

DynamoDB

三

DynamoDB > 項目: chatgpt-support-line--chat-history > 項目の編集

項目の編集

項目の属性を追加、削除、または編集できます。属性は、最大 32 レベルの深さまで他の属性内にネストできます。 [詳細はこちら](#)

フォームJSON ビュー

属性

新しい属性の追加

属性名	値	タイプ	
id - パーティションキー	1bafcdf2-efc4-4aa8-9a4c-88c12f499741	文字列	
content	AWS Lambda のメリット・デメリットを完結にまとめよ。	文字列	削除
role	user	文字列	削除
typedAt	2023-04-06T22:39:25.978685	文字列	削除
userId	xxxxxx	文字列	削除

キャンセル変更を保存

<input type="checkbox"/>	1bafcdf2-efc4-4aa8-9a4c...	AWS Lambda のメリット・デメリットを完結にまとめよ。	user	2023-04-06T22:39:25.978685
<input type="checkbox"/>	3121319a-0fe7-45b0-a19...	そのような期限付きで柔軟なデータを保存するテーブルについ...	assistant	2023-03-31T13:59:08.034438
<input type="checkbox"/>	37448dcb-5828-4b0e-9cc...	リレーショナルデータベースにて、テーブル名の命名で悩んで...	user	2023-03-31T13:57:50.233076
<input type="checkbox"/>	584ba8a0-564e-468d-aef...	Redis のバージョンによって、脆弱性の存在有無が異なります...	assistant	2023-03-30T15:53:55.577392
<input type="checkbox"/>	d9f9e2c7-6872-4807-9e5...	Redis のバージョンで脆弱性がないものを教えてください。	user	2023-03-30T15:53:23.273598
<input type="checkbox"/>	a86daaa7-dd8a-4669-9ff...	246 です。	assistant	2023-03-30T10:55:20.819364
<input type="checkbox"/>	4e0a1b2d-5412-49cd-93...	123 + 123 =	user	2023-03-30T10:55:10.833758
<input type="checkbox"/>	4054d587-c30d-434b-8e...	mocopi (モコピ) は、音声合成技術を用いて、人間の声に近...	assistant	2023-03-30T10:31:51.803371
<input type="checkbox"/>	24ae128b-2e7b-4fc8-b6b...	ソニーのmocopiについて教えて	user	2023-03-30T10:31:19.080161
<input type="checkbox"/>	5f166eae-ae7f-4acf-90bd...	こんばんは！何かお困りのことはありますか？私にできること...	assistant	2023-03-29T22:53:07.285956
<input type="checkbox"/>	87c0e6ef-9a35-45cc-b58...	こんばんは。	user	2023-03-29T22:52:53.135613

過去の発言を新しい方からいくらか取得する理由は、4.で OpenAI 社の API を叩く際に、AI が会話の文脈を理解できるようにするためです

また、カラムは

- ID (ランダム文字列)
- 本文
- ロール
- 発言日時
- ユーザーID (LINEユーザーごとの一意の値)

を設けていますが、一例なので過不足はあるかもしれません  
ロールについては後述します

4. OpenAI 社の API と連携 -> 5. DynamoDB に返答を保存

いよいよ OpenAI 社の API を叩きます  
その際必要になるものは主に以下のようなものがあります

- model : モデル。後述
- maxTokens : API から返却される最大のトークン数 (≒ 文字数 (厳密には違う))
- temperature : 高い値だとより複雑な文章を生成できるが不自然になる可能性も含んでいる
- messages : 会話内容
  - role : ロール。後述
  - content : 内容

OpenAI 社では用途に応じて様々なモデルを用意してくれています  
以下が主なモデルです

base model	model	max tokens	特徴
GPT-3.5	text-davinci-003	4,097	最も高性能
GPT-3.5	gpt-3.5-turbo	4,096	davinci より小規模で高速でコストも davinci の 1/10
GPT-4	gpt-4-32k	32,768	どの GPT-3.5 モデルよりも能力が高く、より複雑なタスクを実行できる
Codex	code-davinci-002	8,001	自然言語をコードに変換するのが得意
Whisper	whisper-1	-	音声認識モデルであり、多言語の音声認識、音声翻訳、言語識別を実行できる

メッセージには本文の他にロールも渡します。ロールには以下の3つがあります。

- system : AI の設定などを記述
  - ex. ○○という設定で会話をします / 口調のサンプルは「～～」です など
- assistant : AI の発言
- user : ユーザーの発言

API を叩く際には API key が必要になるので、予め OpenAI 社のコンソールで確認して Lambda に設定しておきましょう

# API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

SECRET KEY	CREATED	LAST USED
sk-...0604	2023年3月11日	2023年3月30日

+ Create new secret key

API を叩くとしばらくして返事が返ってくるので、次の発言時に過去の会話を投げられるようにするため、AI からの返答を保存しておきます。

DynamoDB > 項目: chatgpt-support-line--chat-history > 項目の編集

項目の編集

項目の属性を追加、削除、または編集できます。属性は、最大 32 レベルの深さまで他の属性内にネストできます。 [詳細はこちら](#)

新しい属性の追加 ▼

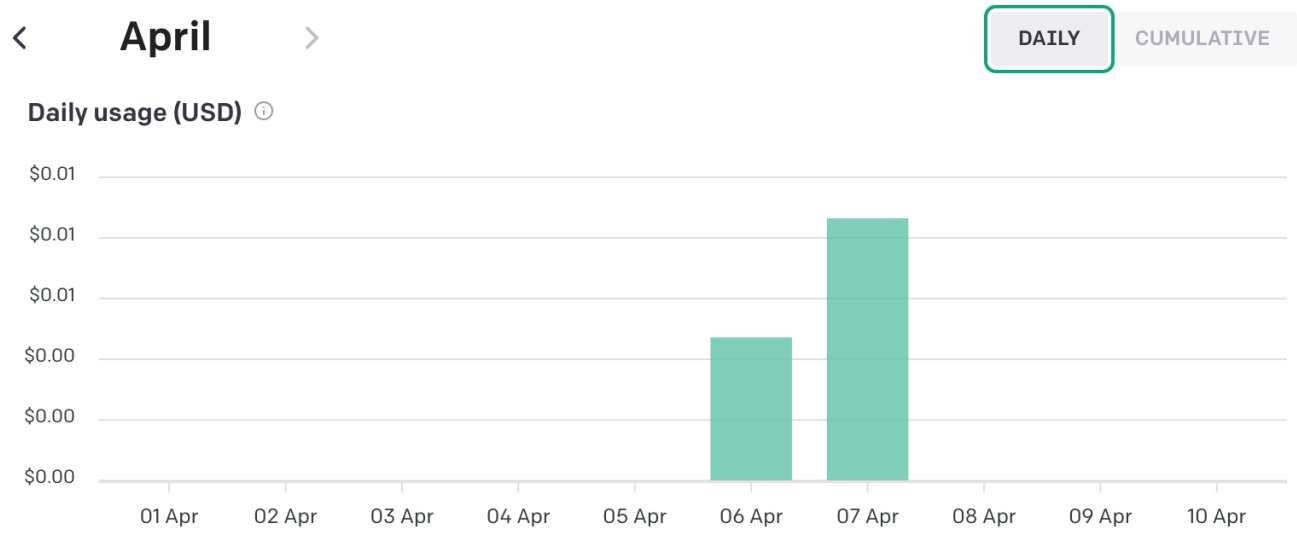
属性名	値	タイプ	
id - パーティションキー	c258e75e-9fc5-4815-ae4e-29be77b51436	文字列	
content	<div>あまり呼ばれないAPIを運用することには向いている場合があります。以下に理由を挙げます。  1. コストが低い: あまり呼ばれないAPIを運用する場合、従量課金制の場合はほとんど費用がかかりません。一方、毎秒数千回以上APIが呼ばれるAPIは、コストが倍増する可能性があります。  2. キャッシュやプリコンパイルが認められる: あまり呼ばれないAPIを運用する場合、一度レスポンスが生成されれば、そのレスポンスをキャッシュし、次回以降はキャッシュから取得することによ</div>	文字列	削除
role	assistant	文字列	削除
typedAt	2023-04-10T22:24:29.227854	文字列	削除
userId	IGNORE_U020e6bedcf25b9ca5e8e14e8a265c122	文字列	削除

キャンセル 変更を保存

なお、今回用いているのは **gpt-3.5-turbo** で、1,000トークンあたり **0.002ドル**（約0.26円）です。  
普段遣いの軽い会話くらいだと大してコストは掛からないようです。

# Usage

Below you'll find a summary of API usage for your organization. All dates and times are UTC-based, and data may be delayed up to 5 minutes.



## Free trial usage

\$0.48 / \$18.00

GRANT #	CREDIT GRANTED	EXPIRES (UTC)
Grant 1	\$18.00	2023年5月1日

## 6. MessagingAPI の Reply API を叩いて返事を LINE に送信

最後に返事を LINE のトークに返却します。  
使用するのは **MessagingAPI** の **Reply API** です 必要になるのは、まず LINE 公式アカウントのコンソールから確認できる **長期アクセストークン** です

[トップ](#) > [Yoshi](#) > [AIお手伝いさん](#) > **Messaging API設定**

## エラーの統計情報



?

## LINE公式アカウント機能

LINE Official Account Managerで、応答メッセージを編集したり、その他の機能を設定したりします。

グループトーク・複数人トークへの参加を許可する ①

応答メッセージ ② 有効

あいさつメッセージ 有効  
②

チャンネルアクセストークン

チャンネルアクセストークン (長期) ②

[illegible]

再発行

もう一つが、当初メッセージを受け取ったときに一緒に受け取っている `replyToken` です

```
body:
{
  "destination": "xxxxxxxxxxxxxx",
  "events": [
    {
      "type": "message",
      "message": {
        "type": "text",
        "id": "111111111111",
        "text": "なるほど、メリットが多いんですね。"
      },
      "webhookEventId": "ABABABABABABAB",
      "deliveryContext": {
        "isRedelivery": false
      },
      "timestamp": 1681133223393,
      "source": {
        "type": "user",
        "userId": "xxxxxxxxxxxxxx"
      },
      "replyToken": "fugafugafugafugafugafugafugafuga",
      "mode": "active"
    }
  ]
}
```

この2つを用いて返信用のエンドポイント(<https://api.line.me/v2/bot/message/reply>)に POSTすると...以下のように LINE にて返事を受け取ることができます



AWS Lambdaのメリットとデメリットは以下の通りです。

#### 【メリット】

1. サーバーレスアーキテクチャー：サーバーが不要であり、AWSがインフラへの専念を通じてLambdaを管理するため、AWS Lambdaはサーバーレスアーキテクチャーで構成され、スケーラビリティと信頼性を提供します。
2. 低コスト：AWS Lambdaは、実行されたコードに基づいて課金されるため、料金が発生するのは実際に使用されたリソースのみです。また、EC2などの仮想サーバーよりも低コストで実行できます。
3. 高速起動：AWS Lambdaは、高速な起動時間を持つため、リクエストの量や処理時間に基づいてスケールすることができます。こ



れは、需要が急増するときにサービスのパフォーマンスが低下することを防止するために役立ちます。

4. マネージドサービス：AWS Lambdaは、AWSのマネージドサービスとして提供されているため、アプリケーションのデプロイ、運用、管理などを負担する必要がありません。

5. シームレスな統合：AWS Lambdaは、他のAWSサービスと簡単に統合できます。たとえば、API Gateway、DynamoDB、S3など、他のさまざまなサービスとシームレスに統合できます。

#### 【デメリット】

1. ローカル開発の難しさ：AWS Lambdaはクラウド上で処理を実行することが必要であるため、ローカルでの開発を行う際にはシミュレータやツールの導入が必要になる場合があります、環境構築に時間がかかることがあります。

2. 通信の遅延：AWS Lambdaはネットワークを通じて実行されるため、通信速度が遅い場合や位置情報などのリアルタイム性が必要な場合には、遅延が発生する可能性があります。

3. ランタイムの制限：AWS Lambdaは、サポートされているランタイムの中から選択する必要があります。また、ランタイムによってサポートされている機能が異なるため、特定のミドルウェアやパッケージを使用することができない場合があります。



メッセージを入力

なお、先述のように直近の会話も一緒に投げているので、以下のように会話を続けると細かい説明無しで文脈も把握した上で返事をしてくれます。

既読  
午後 10:27

なるほど、メリットが多いんですね。

↓ 返事

## ここから未読メッセージ



はい、AWS Lambdaにはメリットが多くあります。特に、低コストや高速起動など、アプリケーションのスケラビリティと信頼性を向上させることができる点が大きなメリットです。

ただし、AWS Lambdaはサーバーレスアーキテクチャーであるため、クラウド上でコードを実行するというアプローチになります。そのため、アプリケーションによっては挙動が異なったり、通信の遅延が発生する場合があります。また、ローカル開発に時間がかかることもデメリットの一つです。

あまり呼ばれないAPIを運用する場合には、AWS Lambdaが向いている場合もあれば、AWS上で適切な仮想マシンを立ち上げるほうが向いている場合もあります。APIの利用状況や、コスト、開発環境などを考慮し、最適な選択肢を検討することが重要です。

午後 10:27

## TODO

- 実装が暫定のかたまり
  - 途中で力尽きたので近々直します
- 過去の会話を投げる部分は `max_tokens` を数えてから投げないとエラーになって DB 修正をしないと再起不能になる
- グループに公式アカウントを招待しても使えることは確認済だが、グループ内の会話だけを見分ける方法はあるか？
  - グループに所属しているユーザーの ID が判別できればよさそうだが...
  - LINE 公式アカウントにお詳しい方がいらっしゃいましたら教えて下さい。
- 現在 **GPT-4** 申し込み中 (wktk)

## DEMO

デモる