

LINE対話

はじめてのLINEBOT作成

前提

- LINEアカウントがある
- 48v.me にアカウントがある
 - FTPでプログラムを置ける

GitHub

使用するプログラムをダウンロードするまで

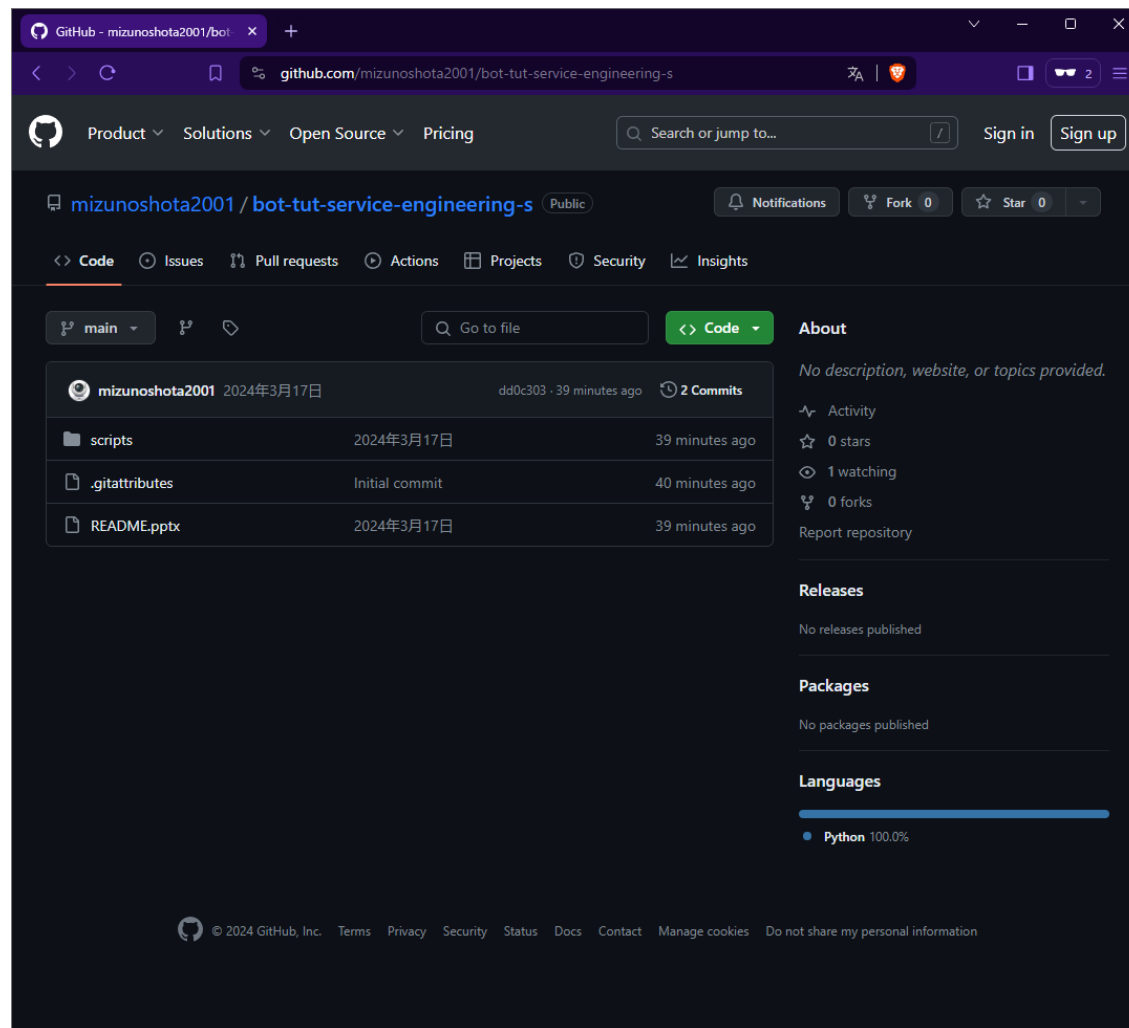
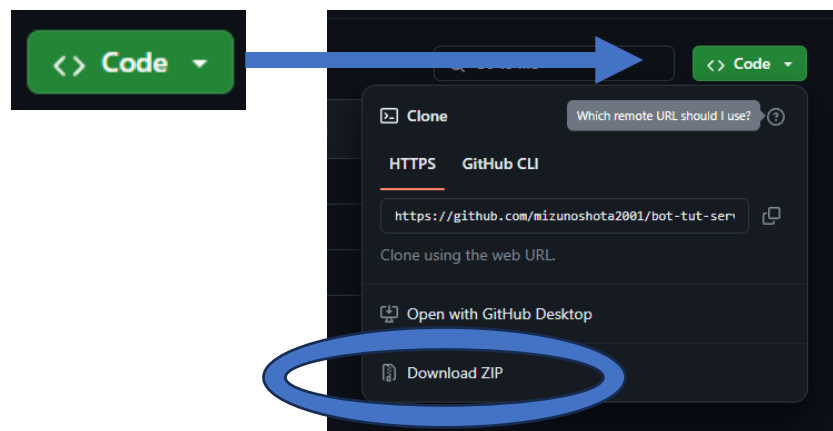
GitHub

ソースコードをダウンロードしよう

<https://github.com/mizunoshota2001/bot-tut-service-engineering-s>

このURLにアクセスしてください。

ベースとなるプログラムとその説明を公開しています。ZIPでダウンロード

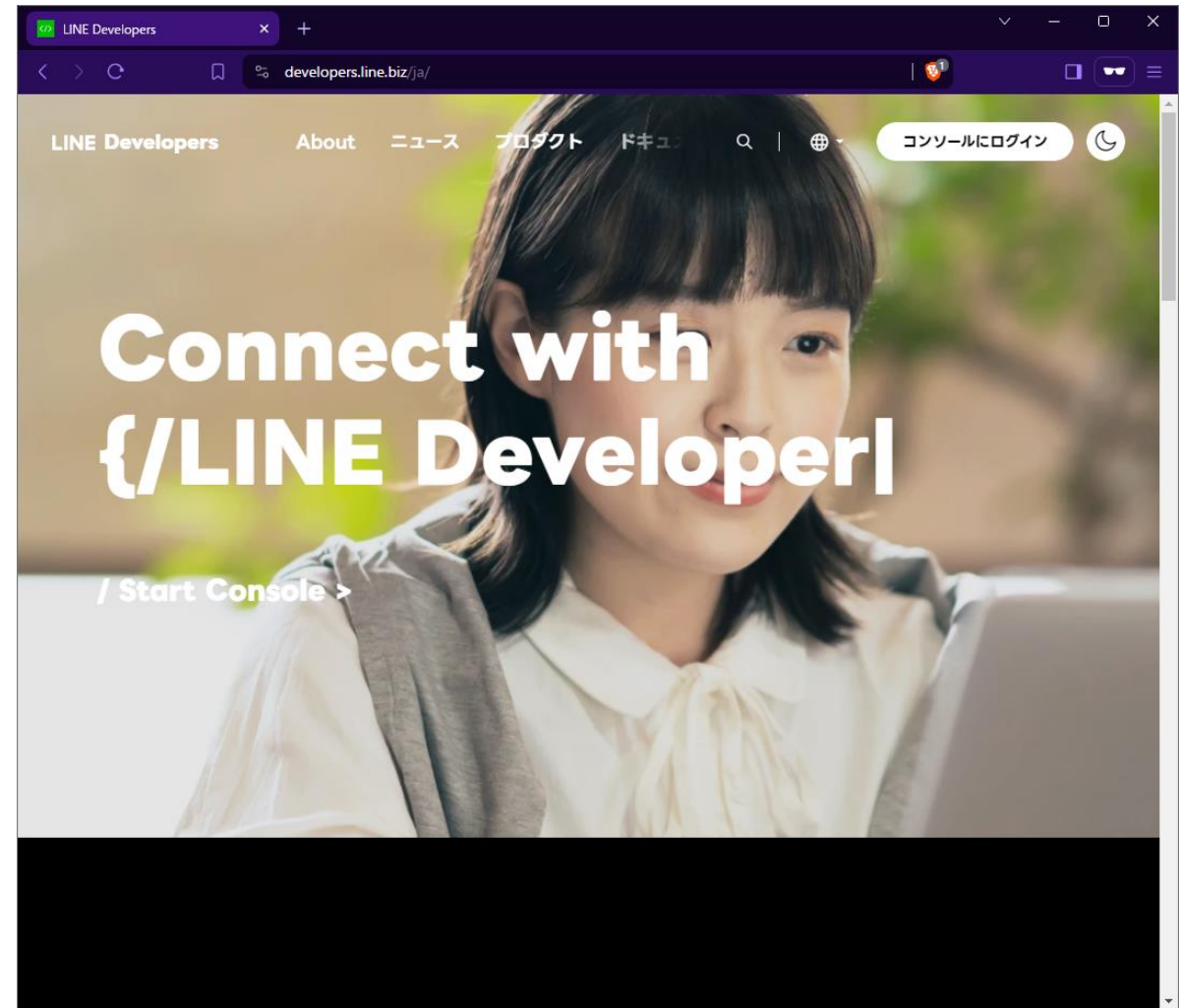


LINE Developers

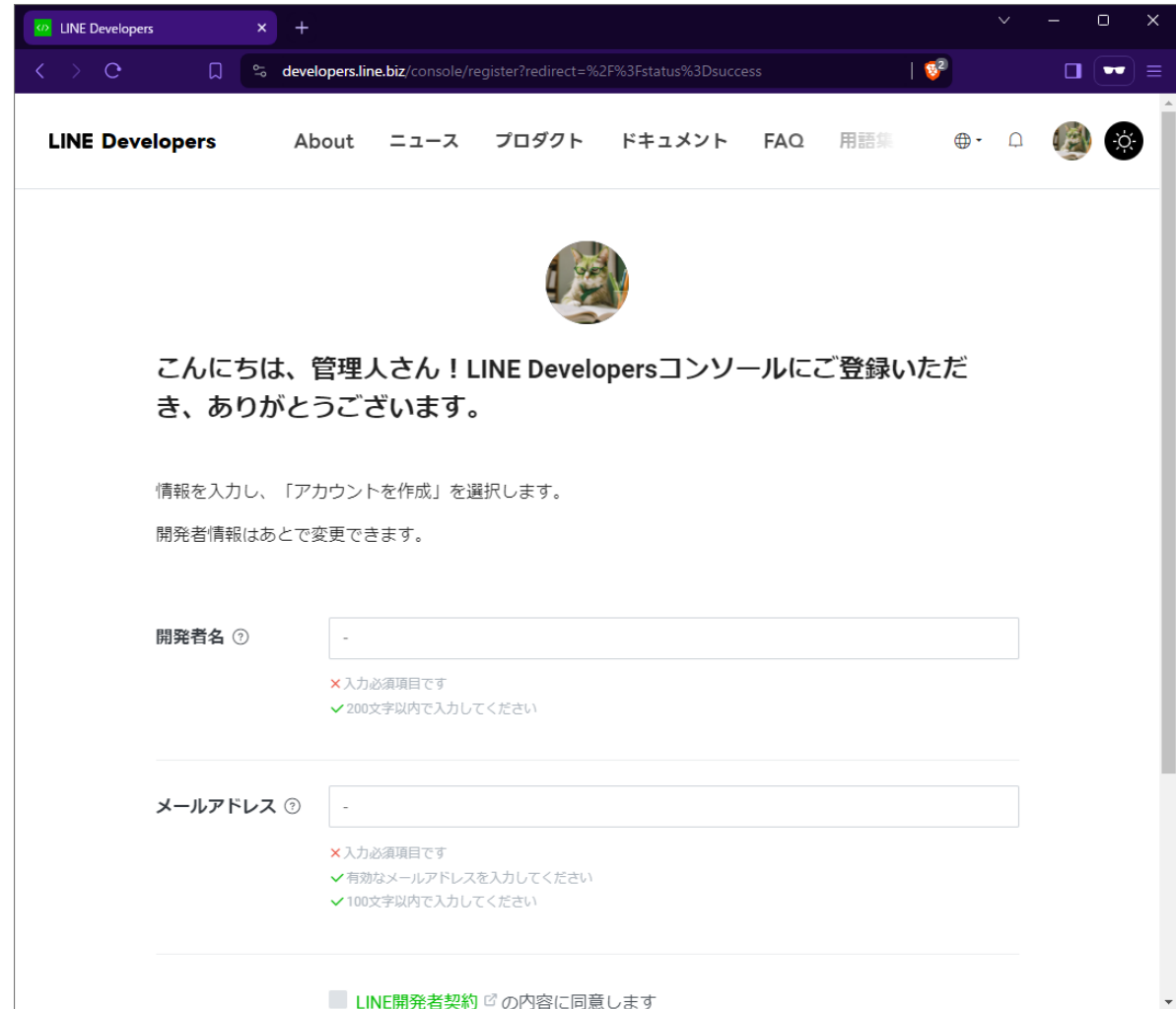
公式LINE-BOT用のチャネルを作成するまで

LINE Developers ログインしよう

<https://developers.line.biz/ja/>



LINE Developers アカウントを作成



The screenshot shows a web browser window with the URL `developers.line.biz/console/register?redirect=%2F%3Fstatus%3Dsuccess`. The page header includes the "LINE Developers" logo and navigation links: "About", "ニュース", "プロダクト", "ドキュメント", "FAQ", and "用語集". A user profile picture of a cat is visible in the top right corner.

The main content area features a greeting: "こんにちは、管理人さん！LINE Developersコンソールにご登録いただき、ありがとうございます。" (Hello, Manager! Thank you for registering on the LINE Developers console.)

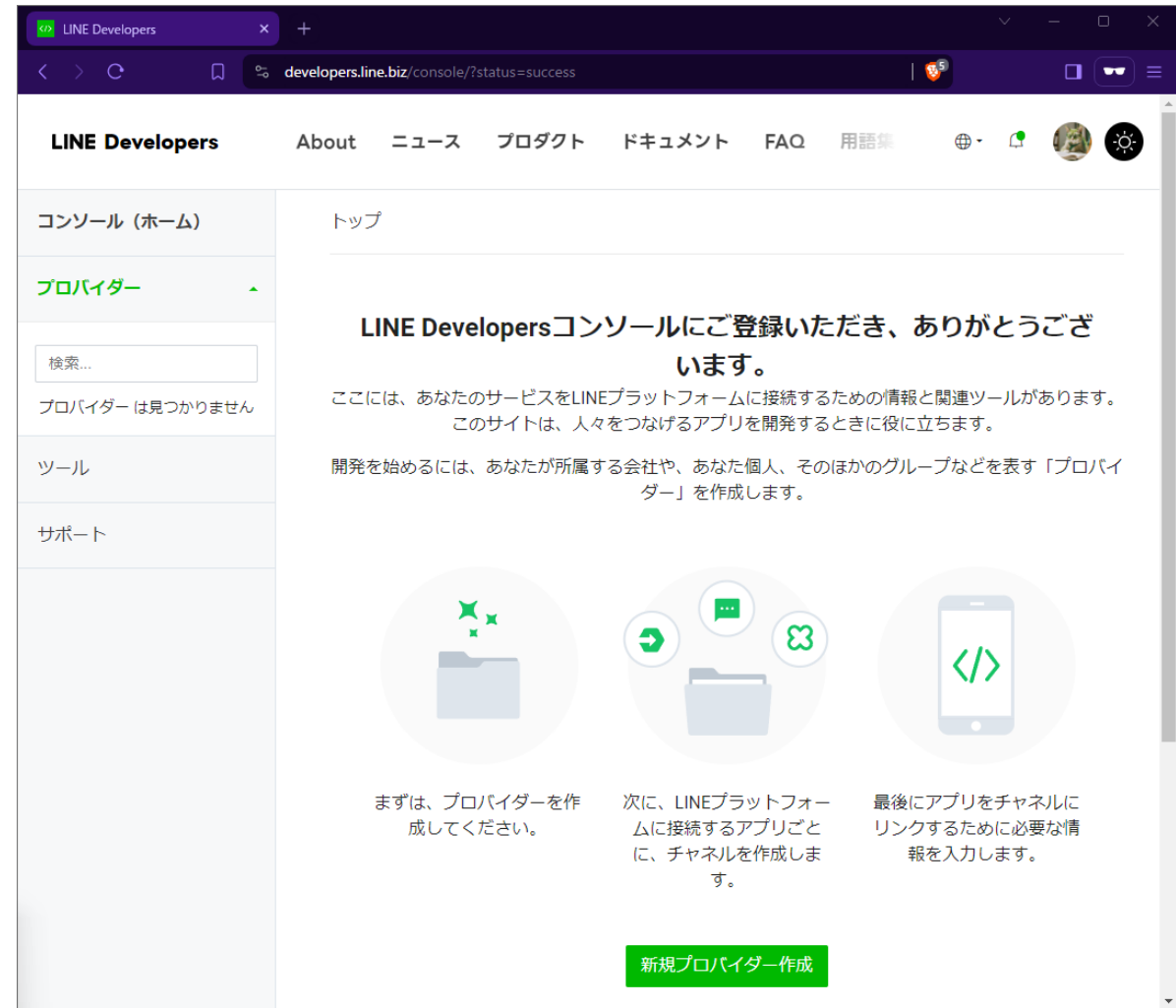
Below the greeting, instructions are provided: "情報を入力し、「アカウントを作成」を選択します。" (Enter the information and select "Create Account".) and "開発者情報はあとで変更できます。" (Developer information can be changed later.)

The registration form consists of two main sections:

- 開発者名 ①** (Developer Name ①): A text input field with a placeholder "-". Below the field, there are two lines of feedback text: "× 入力必須項目です" (Red X: Required field) and "✓ 200文字以内で入力してください" (Green checkmark: Please enter within 200 characters).
- メールアドレス ②** (Email Address ②): A text input field with a placeholder "-". Below the field, there are three lines of feedback text: "× 入力必須項目です" (Red X: Required field), "✓ 有効なメールアドレスを入力してください" (Green checkmark: Please enter a valid email address), and "✓ 100文字以内で入力してください" (Green checkmark: Please enter within 100 characters).

At the bottom of the form, there is a checkbox labeled "LINE開発者契約" (LINE Developer Agreement) followed by the text "の内容に同意します" (I agree with the contents).

LINE Developers 新規プロバイダー作成



LINE Developers 新規プロバイダー作成

The screenshot shows the LINE Developers console interface. A modal dialog titled '新規プロバイダー作成' (New Provider Creation) is displayed in the center. The dialog contains a form for creating a new provider, with the 'プロバイダー名' (Provider Name) field filled with '管理人'. Below the input field, there are three green checkmarks indicating validation rules: '入力必須項目です' (Required input), '特殊な文字（4バイトのユニコード文字）を使用しないでください' (Do not use special characters (4-byte Unicode characters)), and '100文字以内で入力してください' (Enter within 100 characters). Below the form, there is a paragraph explaining that a provider is someone who provides services and obtains user information, with a link to the 'ドキュメント' (Documentation). At the bottom of the dialog, there are two buttons: 'キャンセル' (Cancel) and '作成' (Create). The background of the console shows a sidebar with 'コンソール (ホーム)', 'プロバイダー', 'ツール', and 'サポート'. The main content area has a header 'LINE Developers' and a navigation bar with 'About', 'ニュース', 'ブログ', 'ドキュメント', 'FAQ', and '用語集'. The main content area also has a 'トップ' (Top) link and a section titled 'LINE Developersコンソールにご登録いただき、ありがとうございます' (Thank you for registering on the LINE Developers console).

LINE Developers

About ニュース ブログ ドキュメント FAQ 用語集

コンソール (ホーム)

プロバイダー

検索...

プロバイダー は見つかりません

ツール

サポート

トップ

LINE Developersコンソールにご登録いただき、ありがとうございます

新規プロバイダー作成

プロバイダー名 ① 管理人

- ✓ 入力必須項目です
- ✓ 特殊な文字（4バイトのユニコード文字）を使用しないでください
- ✓ 100文字以内で入力してください

プロバイダーとは、サービスを提供し、利用者の情報を取得する個人の開発者、企業、組織のことです。詳細は、[ドキュメント](#) を参照してください。

キャンセル 作成

まずは、プロバイダーを作成してください。

次に、LINEプラットフォームに接続するアプリごとに、チャンネルを作成します。

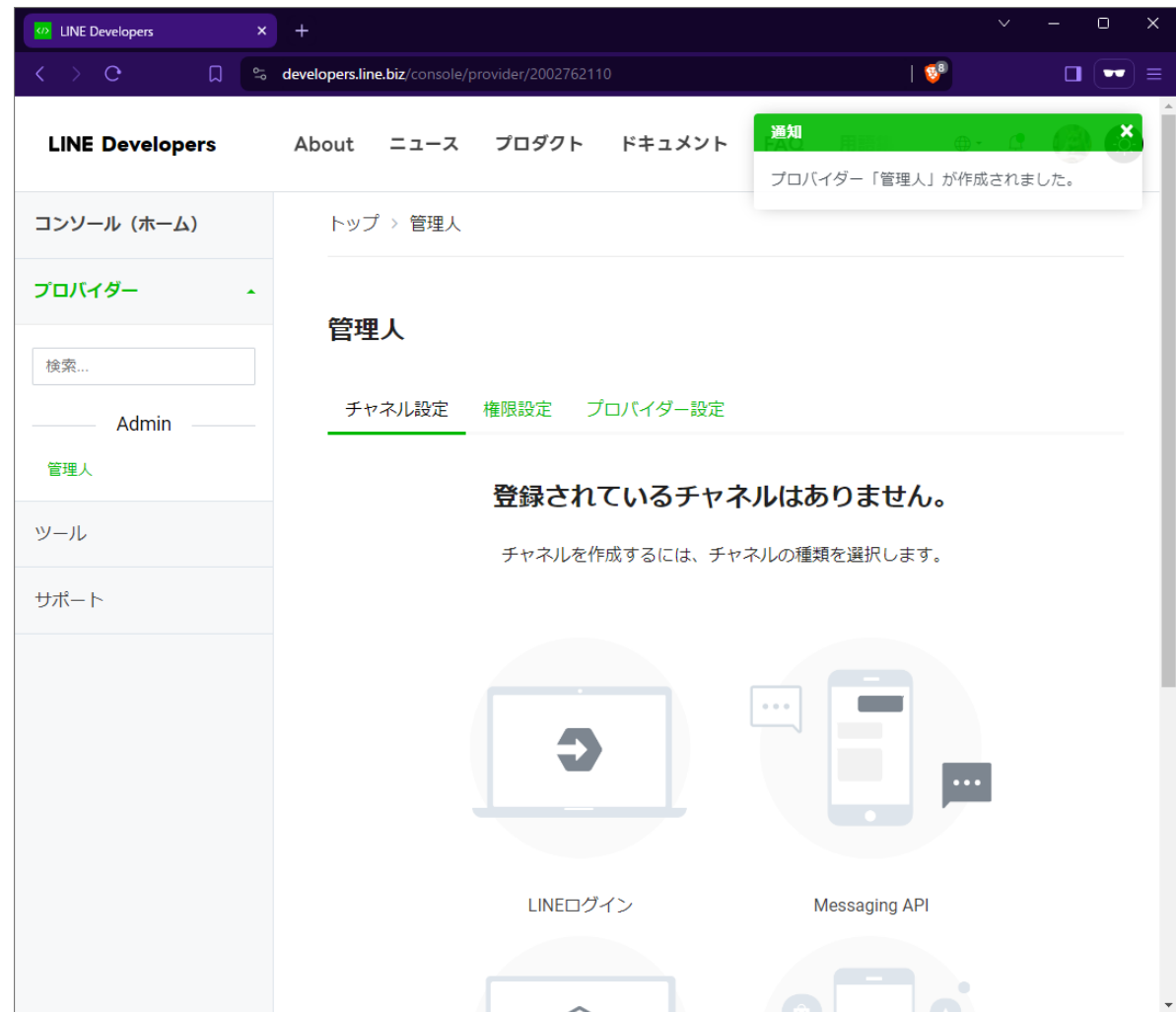
最後にアプリをチャンネルにリンクするために必要な情報を入力します。

新規プロバイダー作成

LINE Developers 新規チャネル作成

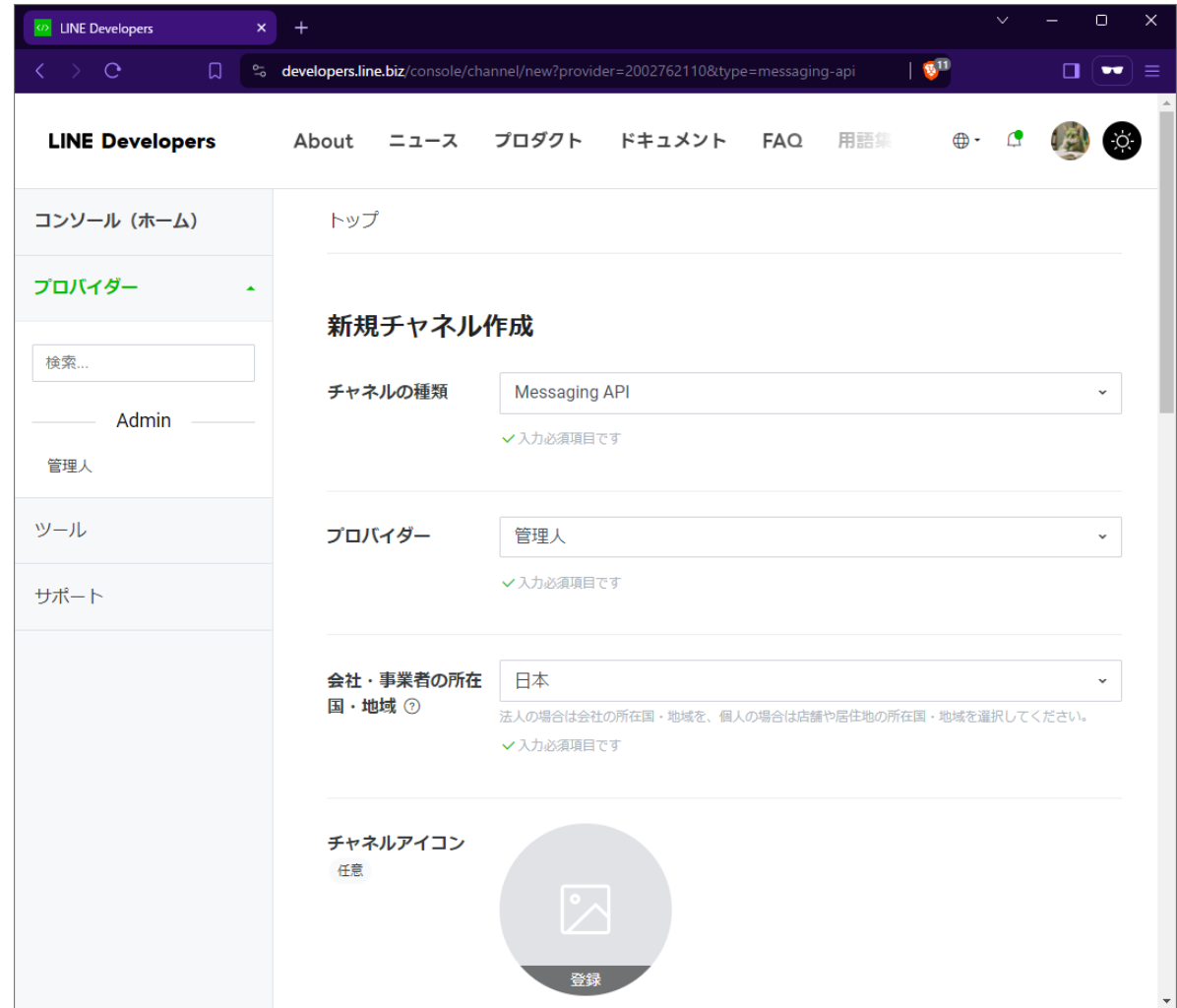
チャネルを作成するには、チャネルの種類を選択します。

今回はMessaging API



LINE Developers 新規チャネル作成

必要な項目を入力

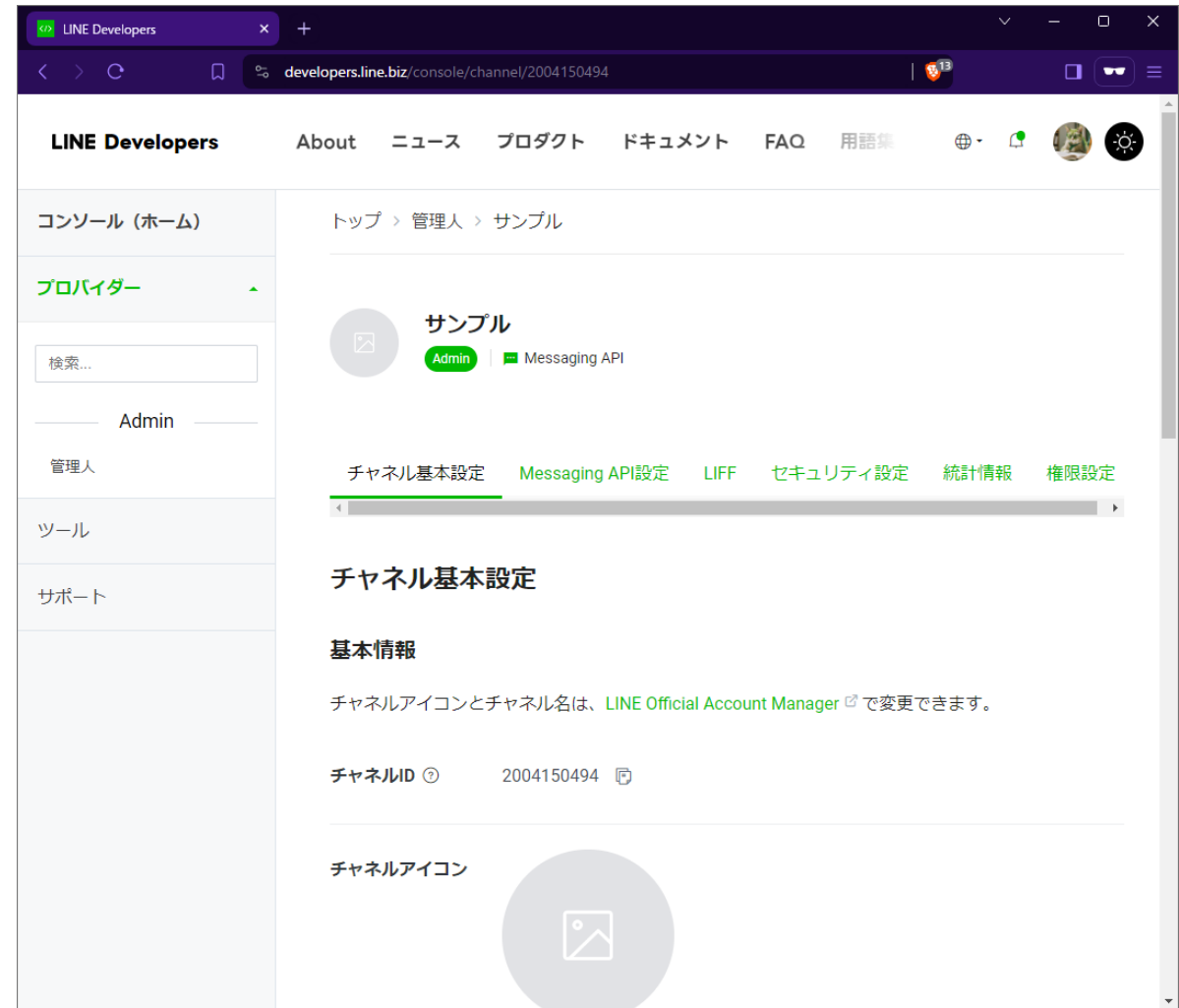


The screenshot shows the LINE Developers console interface for creating a new channel. The browser address bar displays `developers.line.biz/console/channel/new?provider=2002762110&type=messaging-api`. The page title is "LINE Developers". The navigation menu includes "About", "ニュース", "プロダクト", "ドキュメント", "FAQ", and "用語集". The left sidebar contains "コンソール (ホーム)", "プロバイダー", "検索...", "Admin", "管理人", "ツール", and "サポート". The main content area is titled "新規チャネル作成" (New Channel Creation) and contains the following form fields:

- チャンネルの種類** (Channel Type): A dropdown menu with "Messaging API" selected. A green checkmark and the text "入力必須項目です" (Required input item) are shown below the field.
- プロバイダー** (Provider): A dropdown menu with "管理人" (Admin) selected. A green checkmark and the text "入力必須項目です" (Required input item) are shown below the field.
- 会社・事業者の所在 国・地域 ①** (Company/Business Location Country/Region ①): A dropdown menu with "日本" (Japan) selected. A green checkmark and the text "入力必須項目です" (Required input item) are shown below the field. A note below the field states: "法人の場合は会社の所在国・地域を、個人の場合は店舗や居住地の所在国・地域を選択してください。" (For legal entities, select the country/region of the company; for individuals, select the country/region of the store or residence).
- チャンネルアイコン** (Channel Icon): A section with the text "任意" (Optional) and a large circular button with a camera icon and the text "登録" (Register).

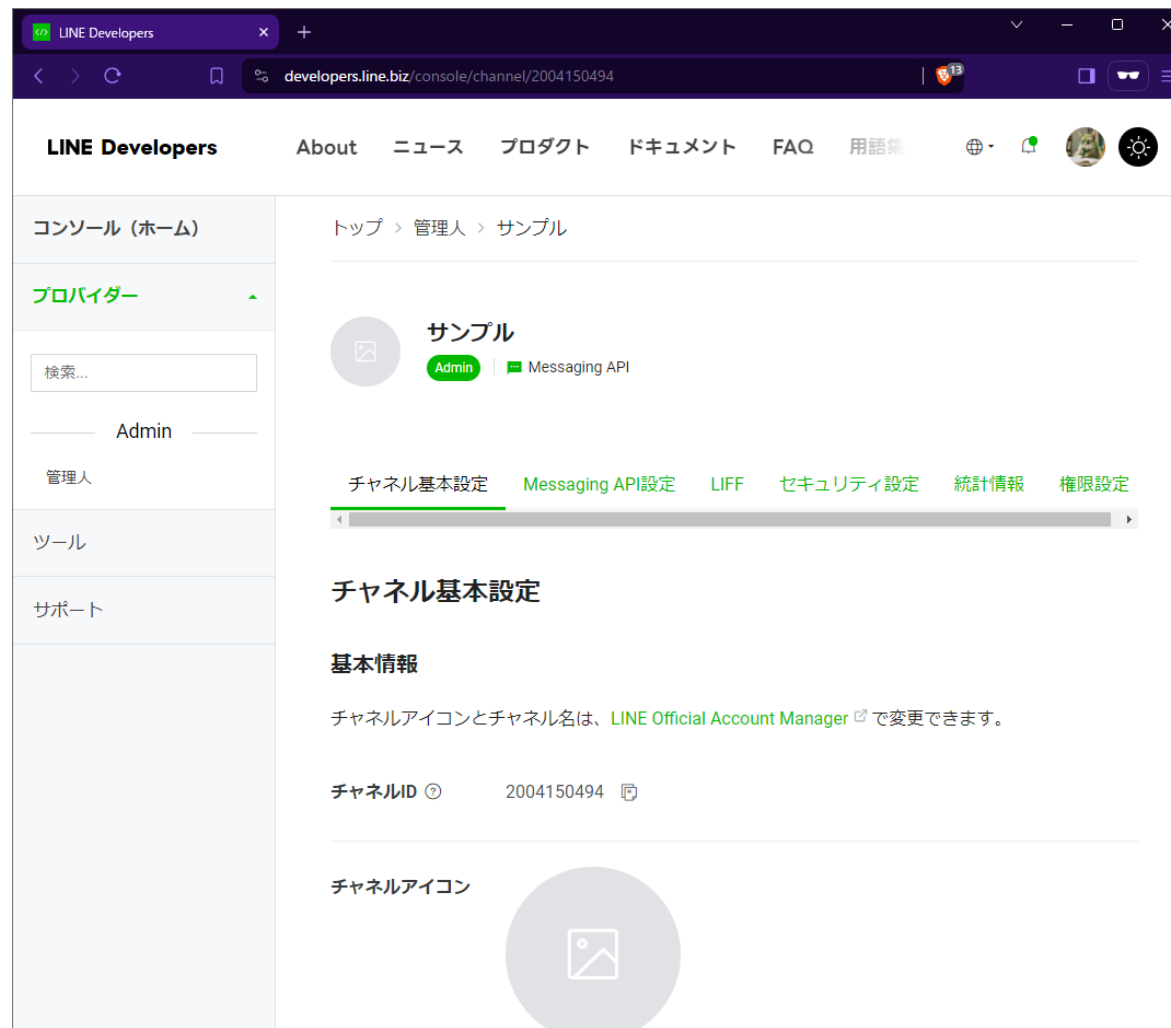
LINE Developers 新規チャネル作成

作成完了後にはチャネルの設定ページに遷移します。



LINE Developers チャンネルを確認しよう

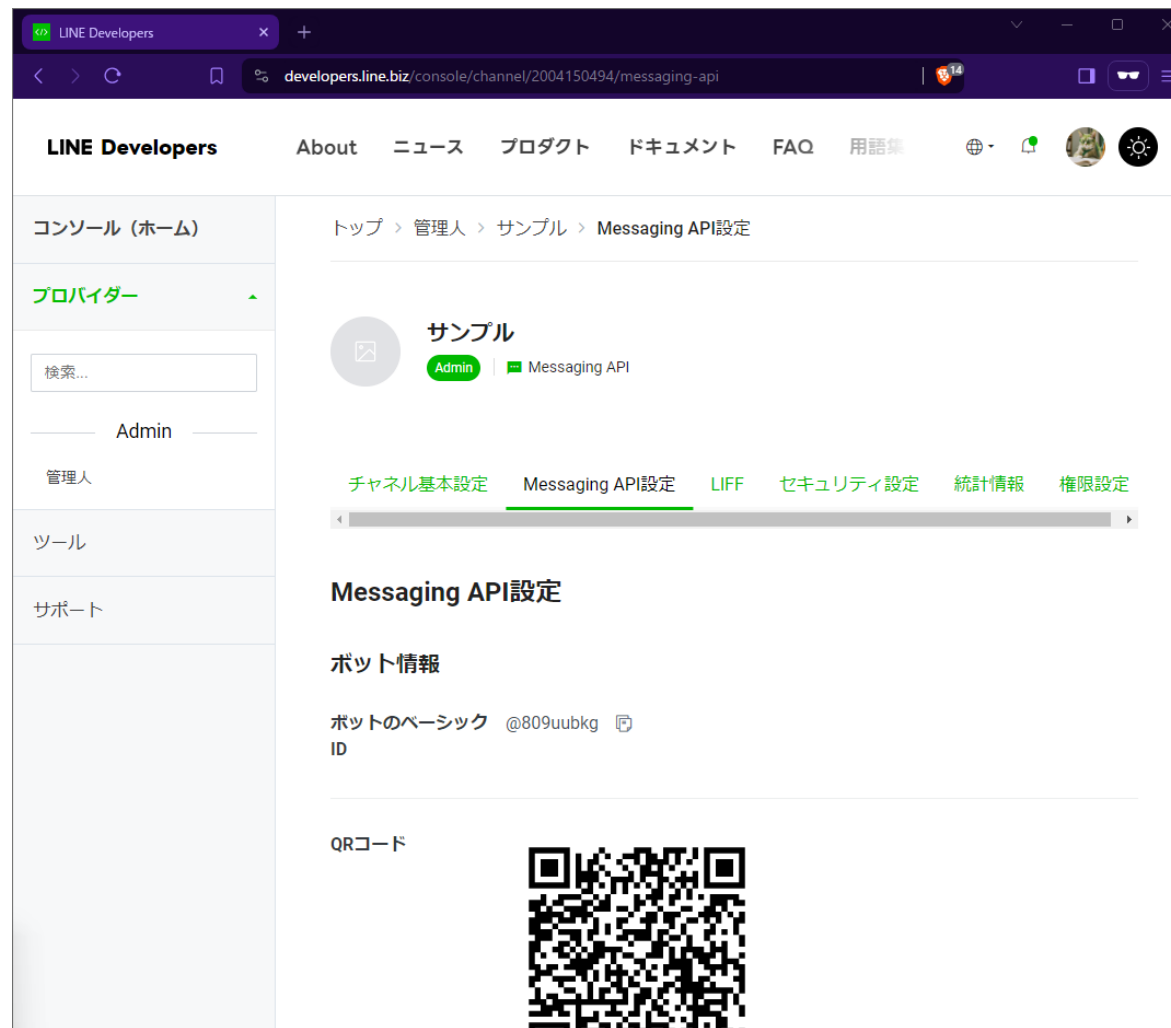
チャンネルの詳細画面から、**Messaging API設定**をクリック



LINE Developers チャンネルを確認しよう

QRコードから作成したチャンネルを友達登録

初期設定では、LINEが用意しているあいさつメッセージが表示される

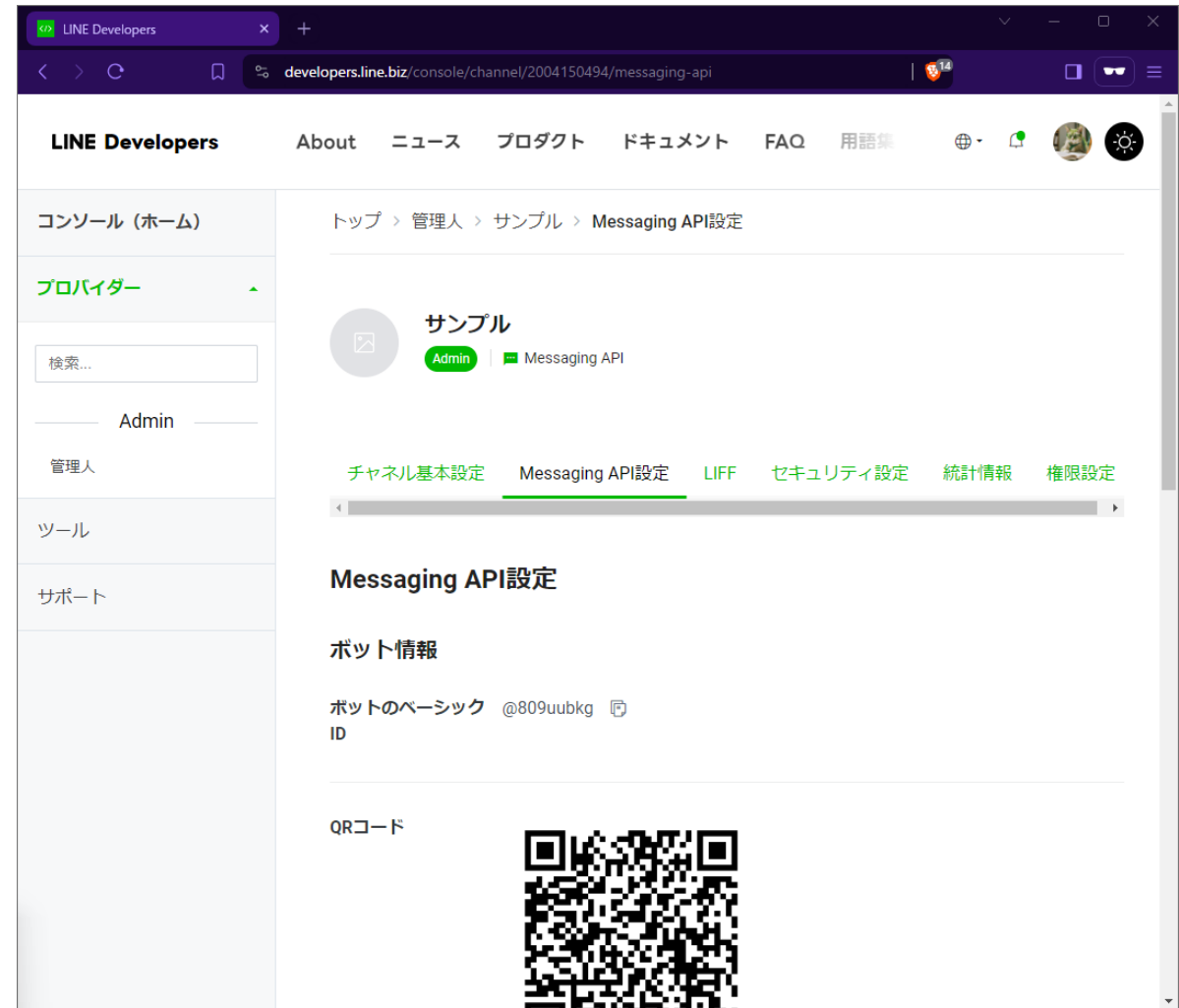


トークン発行

チャンネルアクセストークンを発行するまで

トークン発行

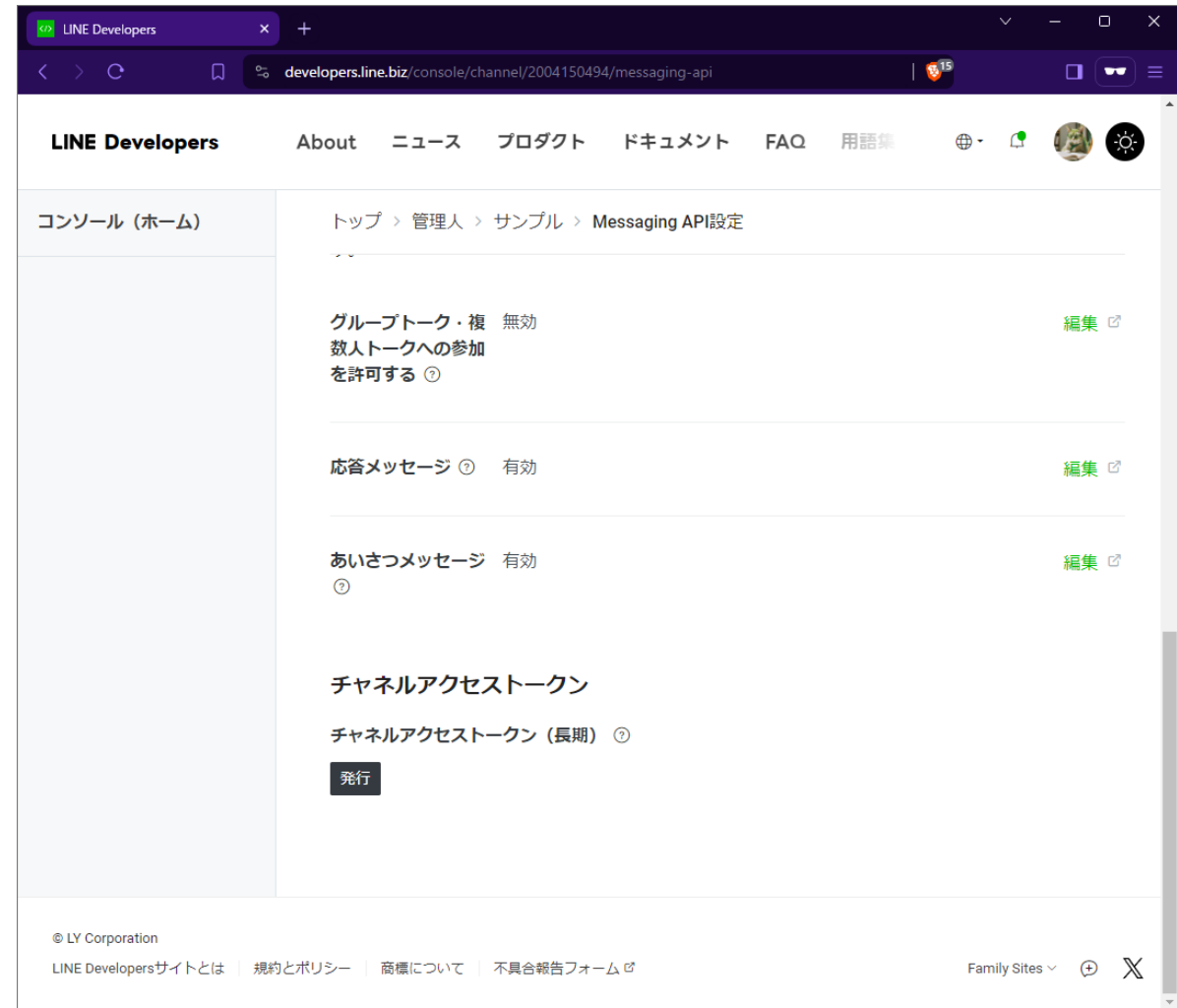
チャンネルの詳細画面から、Messaging
API設定をクリック



トークン発行

ページ最下部からアクセストークンを発行できる。

発行をクリック

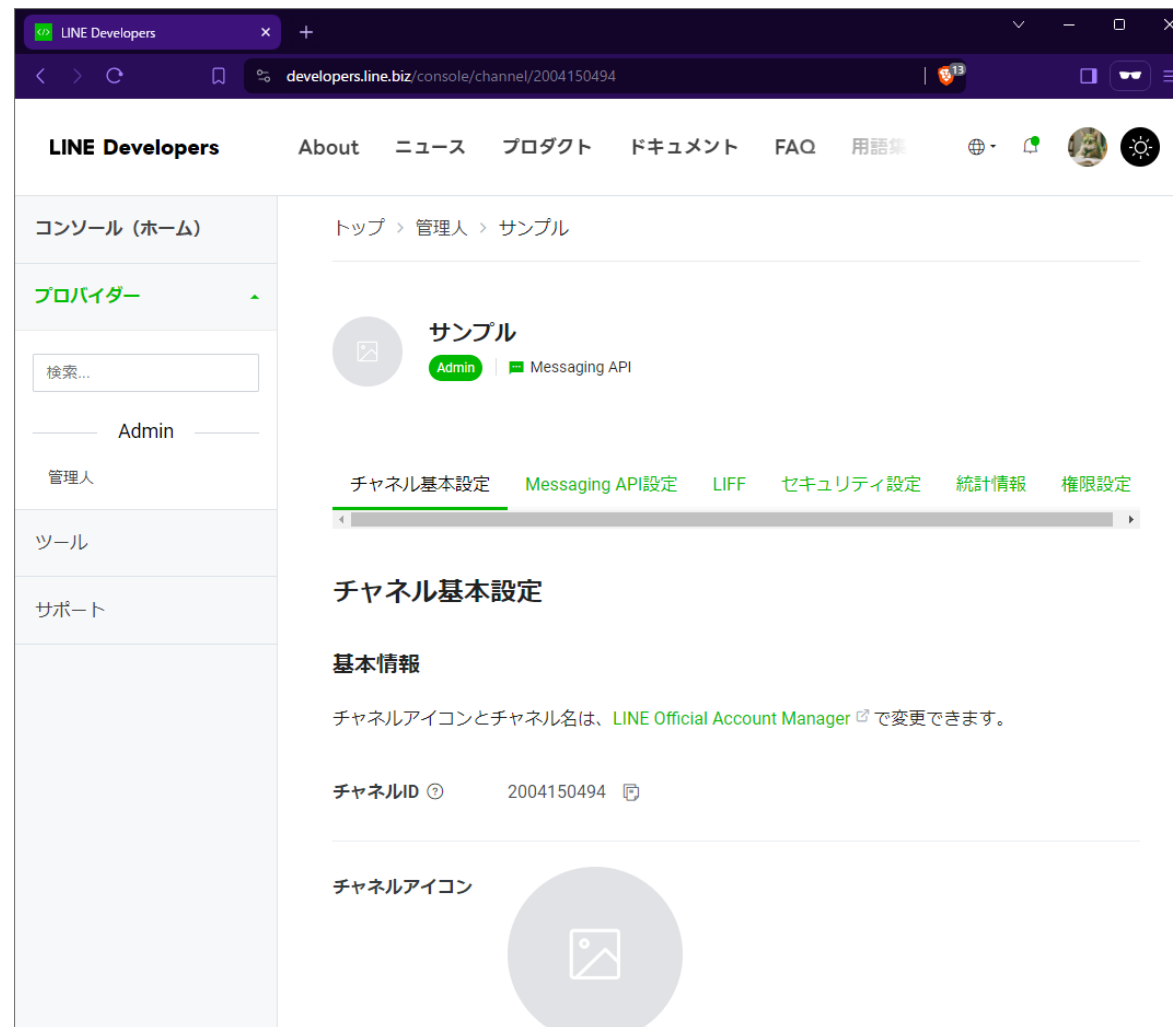


チャンネルシークレットの取得

チャンネルアクセストークンを発行するまで

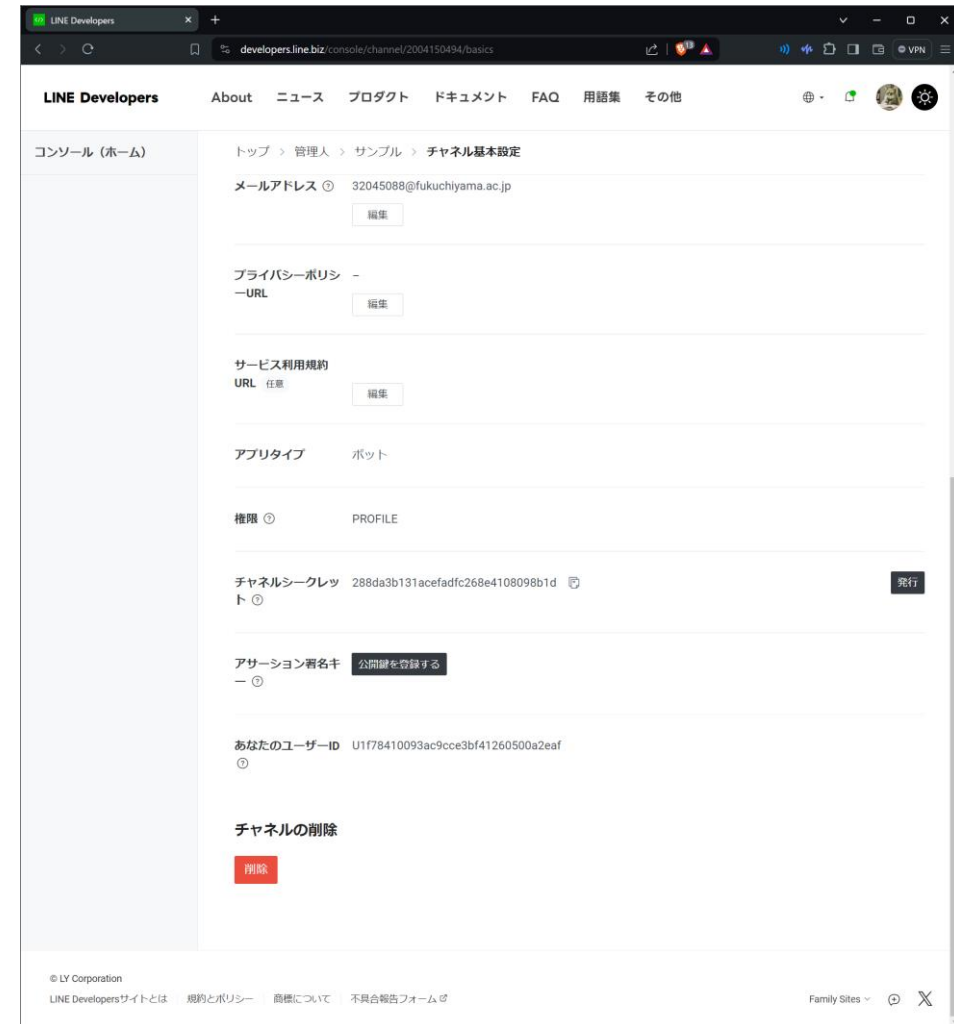
チャンネルシークレットの取得

ページ下部へスクロール



チャンネルシークレットの取得

チャンネルシークレットをメモしておいてください。

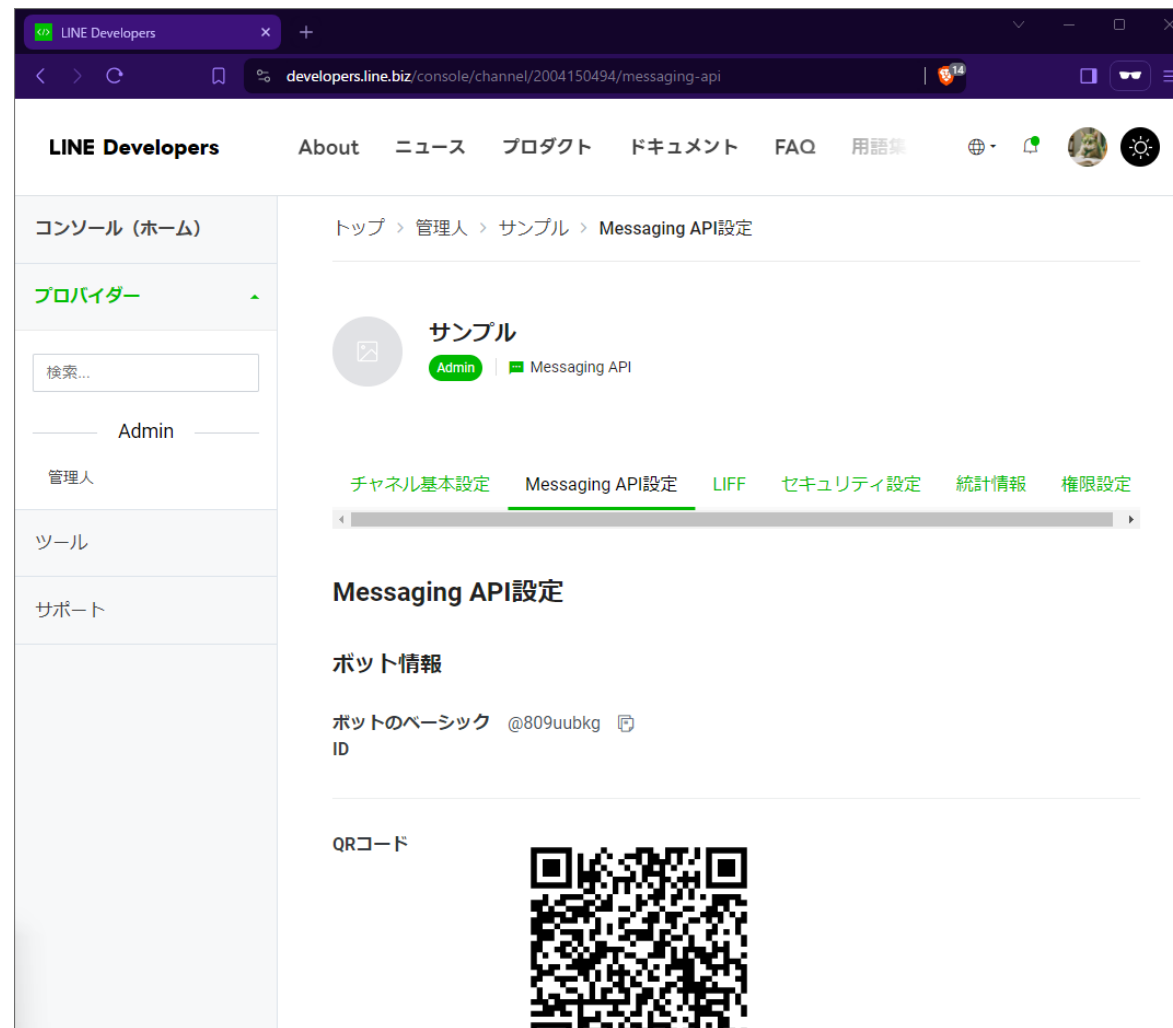


Messaging API設定

Webhook用のURLを設定するまで

Messaging API設定 Webhookを有効にする

チャンネルの詳細画面から、**Messaging API設定**をクリック



Messaging API設定 Webhookを有効にする

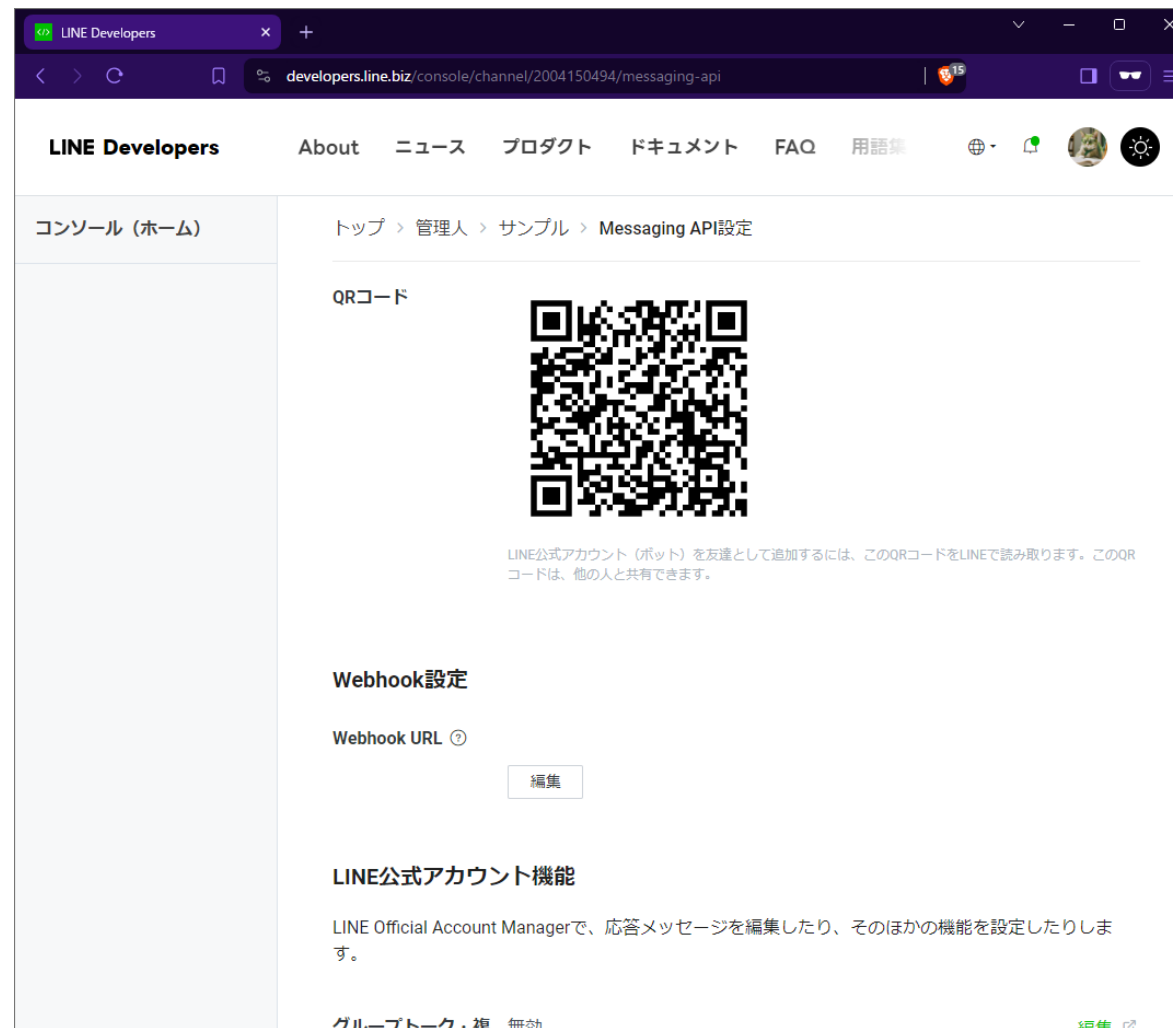
QRコードの下にWebhook設定がある。

編集をクリックし、URLを設定する。

URLは48v.me上に設置した、
「webhook.py」を設定してください。

webhook.py

<https://github.com/mizunoshota2001/bot-tut-service-engineering-s/blob/main/webhook.py>



おうむ返しさせてみよう

サンプルコードを見てみよう

おうむ返しさせてみよう

- どこにユーザが送信したメッセージが格納されていますか。
 - `event.message.text`
- ユーザにメッセージを送信するにはどのようにしますか。
 - `reply_message_with_http_info`
 - `ReplyMessageRequest`

```
• 29行目~  
• @handler.add(MessageEvent, message=TextMessageContent)  
• def handle_message(event):  
•     api_instance.reply_message_with_http_info(  
•         ReplyMessageRequest(  
•             reply_token=event.reply_token,  
•             messages=[TextMessage(text=event.message.text)]  
•         )  
•     )
```

おうむ返しさせてみよう

- `ReplyMessageRequest`
- `replyToken`は`event`に格納されているものを使用します。
 - `event.reply_token`は一度使用すると失効されます。
 - 複数のメッセージを送りたい場合には、`messages`に複数のメッセージを格納してください。

```
• class ReplyMessageRequest(  
•     *,  
•     replyToken: StrictStr,  
•     messages: Any,  
•     notificationDisabled: StrictBool | None  
• )
```