太田直希

生成AIを用いたフードロス削減を支援する料理支援システムの使用方法

Github: <https://github.com/Naoki-02/sotuken_dev>

準備するもの

* メールアドレスとパスワード（アカウントを作成する際に使用）
* クレジットカード（API使用に登録するカード）
* ChatGPT APIのAPI Key（OpenAIのアカウント）
* Google cloud vision API のAPI Key（Googleのアカウント）
* Githubからダウンロードするデータ
* Docker Desktopのインストール
* javascriptを動かせる環境

ChatGPT APIのAPI Key作成手順

1. OpenAIアカウントの作成

ChatGPT APIを利用するには、OpenAIのアカウントが必要になります。

アカウントをお持ちでない場合は、以下の手順で作成してください。

1. OpenAIの公式サイトにアクセスする。

https://platform.openai.com/docs/overview

1. 画面右上「Sign Up」または「Log in」をクリックする。
2. メールアドレス、Googleアカウント、またはMicrosoftアカウントで登録する。
3. 登録後、OpenAIのAPIページにログインする。
4. ログインの際登録したメールアドレスに検証コードが送られるので入力する。
5. クレジットカードの登録

ChatGPT APIを利用するには、クレジットカード情報の登録が必要です。以下の手順で登録してください。

https://platform.openai.com/settings/organization/billing/overview

1. OpenAIの課金設定ページにアクセスします。
2. 画面左側「Billing」をクリックする。
3. 「Add a payment method」ボタンをクリックする。
4. クレジットカード情報を入力し、「Save」ボタンをクリックする。
5. 登録が完了したら、「Add to credit balance」ボタンをクリックし10ドルほどチャージしておくといいでしょう。
6. API Keyの作成

https://platform.openai.com/docs/overview

1. OpenAIのAPI管理ページにアクセスする。
2. 右上Dashbordまたは設定画面から「API Keys」を選択する。
3. API Keysの画面右上の緑色のボタン「Create new secret key」ボタンをクリックする。
4. Keyの名前（オプション）とProject（Default project）を選択する。
5. 「Create secret key」ボタンを選択する。
6. 表示されたAPI Keyをコピーし、安全な場所に保存をする。

※：API Keyは一度しか表示されないため、忘れてしまった場合は、今あるKeyを削除し、新しく作成してください。

Google Cloud Vision のAPI Key作成手順

※クレジットカードの登録が必要になった場合は登録してください。

1. Googleアカウントの作成
2. Googleのサイトにアクセスします。

<https://www.google.com/search?q=https://accounts.google.com/signup>

1. 画面の指示に従って、氏名、ユーザ名パスワードなどの情報を入力し、Googleアカウントを作成します。
2. Google Cloud Platform（GCP）プロジェクトの作成
3. 作成したGoogleアカウントで、<https://console.cloud.google.com/>にアクセスします。
4. GCPプロジェクトを作成します。（まだ作成していない場合）
5. GCPコンソールに初めてアクセスする場合、プロジェクトの作成を促されることがあるので、画面の指示に従ってプロジェクトを作成してください。
6. Vision APIを有効にします。（まだ有効にしていない場合）
7. 画面左上のハンバーガーメニュー（三本線アイコン）をクリックします。
8. 表示されたメニューから「APIとサービス」を選択し、さらに表示されるサブメニューから「ライブラリ」をクリックします。
9. 検索バーに「Cloud Vision API」と入力し、検索結果から「Cloud Vision API」をクリックします。
10. 「有効にする」ボタンが表示されている場合は、それをクリックして有効にします。
11. API Keyの作成
12. 左上のハンバーガーメニューをクリックし、「APIとサービス」を選択、さらに表示されるサブメニューから「認証情報」をクリックします。
13. 「認証情報」画面の上部にある「＋認証情報を作成」ボタンをクリックします。
14. ドロップダウンメニューから「APIキー」を選択します。
15. 「APIキーが作成されました」というポップアップが表示され、APIキーが表示されます。
16. API Keyをコピーし安全な場所に保管してください。
17. 作成されたAPIキーは「認証情報」画面の一覧に表示されます。

**Docker Desktopとjavascript環境のインストール方法について**

Docker Desktop:以下のURLの下の方にある「DockerDesktopをダウンロードする」からインストーラをダウンロードし実行する。

<https://www.docker.com/ja-jp/products/docker-desktop/>

javascript:以下のURLの「Node.js(LTS)をダウンロードする」からインストーラをダウンロードし実行する。

https://nodejs.org/ja

**OpenAIのAPI KeyとGoogleのAPI Keyが準備できたら**

1. Githubからコードデータをダウンロードしてくる。
2. 緑色の「＜＞Code」ボタンをクリックし、「Download Zip」をクリックしダウンロードする。
3. 適当な場所にZipファイルを解凍する。
4. VSCodeなどのエディタを開き、先ほど解凍したフォルダを開く。
5. 開いた先の「django」フォルダの中に新規で「.env」ファイルを作成する
6. 「.env」ファイルには以下のコードを書いて保存してください。

|  |
| --- |
| MYSQL\_ROOT\_PASSWORD=root  MYSQL\_DATABASE=db  MYSQL\_USER=db-user  MYSQL\_PASSWORD=password  TZ='Asia/Tokyo'  OPENAI\_API\_KEY='　　　'  VISION\_API\_KEY='　　　' |

※OPENAI\_API\_KEYとVISION\_API\_KEYのシングルクォーテーションの中には先ほど取得したAPI Keyを入力してください。

1. 次に「docker-compose.yml」などが置いてある階層に新規フォルダとして「mariadb」を作成し、その中に新規ファイル「.env」を作成してください。
2. 作成した「.env」には以下のコードを書いて保存してください。

|  |
| --- |
| MYSQL\_ROOT\_PASSWORD=root  MYSQL\_DATABASE=db  MYSQL\_USER=db-user  MYSQL\_PASSWORD=password  TZ='Asia/Tokyo' |

1. 次にエディタのターミナルから「docker-compose.yml」ファイルがある階層で、以下のコードを順に実行してください。

|  |
| --- |
| docker compose build  docker compose up -d |

　　これでサーバの起動はできたので、次にサーバの設定を行います。

1. データベースの設定を反映するために以下のコードをターミナルで順に実行してください。

|  |
| --- |
| docker-compose exec django ./manage.py makemigrations  docker-compose exec django ./manage.py migrate |

※2行目のコードを実行するとエラーのような赤い文字が表示される時があるので、その時はもう一度、1行目を実行しなおしてから２行目を実行してください、繰り返すことで赤い文字がなくなると思います。

1. サーバの準備はできたので、フロント側のサーバを準備をします。
2. 「docker-compose.yml」ファイルがある階層に「react\_front」フォルダがあるので、ターミナルで「react\_front」の中に移動してください。
3. 「react\_front」フォルダにターミナルで入れたら、以下のコードを実行してください。

|  |
| --- |
| npm i |

※実行した後「react\_front」フォルダに「node\_modules」フォルダが作成されたことを確認してください。

1. 確認が出来たら、以下のコードを実行し、「Local:」に書かれたURLへアクセスしてください、少し時間を待つとサイトが表示されると思います。

|  |
| --- |
| npm run dev |

1. 以上で事前準備、サーバの構築は終了です。

ここまで出来ている場合(オープンキャンパスなどで見せるためにサーバを起動する方法)

1. Docker Desktopを開く
2. エディタでフォルダを開きターミナルで以下のコードを実行する

|  |
| --- |
| docker compose up -d |

1. 「react\_front」の階層にターミナルで移動し、以下のコードを実行する。

|  |
| --- |
| npm run dev |

1. 「Local:」に書かれたURLにアクセスする
2. ログイン画面に移動し、サインアップまたはログインをする。

※メールアドレスはhogehoge@hoge.comなどの存在しないもので大丈夫です。

1. ログインするとアプリが使用できるようになります。

アプリの使用説明

1. ログインする
2. レシート読み取りから写真を選択しレシートの登録を行う。

（付属の画像を使うと説明しやすい）

1. 登録が完了したら、食材一覧ボタンから登録された食材を確認することができる。
2. 料理提案ボタンから提案を行う。