

端末の健康情報データを取得する方法

歩数など

・ GCPプロジェクトの設定

・ 下記リンクより、新しくGCPプロジェクトを作成する。

https://console.cloud.google.com/cloud-resource-manager?walkthrough_id=resource-manager--create-project&start_index=1&hl=ja&_ga=2.38436195.131999896.1678957153-1897555526.1674891975#step_index=1

※Android対応時、Google Fit APIの利用権限、APIの有効化が必要なため。

「+プロジェクトを作成」をクリック

リソースの管理

+ プロジェクトを作成

+ フォルダを作成

移動

削除

タグ

≡ フィルタ フィルタ

<input type="checkbox"/>	名前	ID	最終アクセス日	≡	↓
<input type="checkbox"/>	▶ No organization		2022年5月2日		

削除保留中のリソース

プロジェクト名を入力して、「作成」をクリック

新しいプロジェクト



割り当て内の残りのプロジェクト数は 15 projects 件です。プロジェクトの増加をリクエストするか、プロジェクトを削除してください。[詳細](#)

[MANAGE QUOTAS](#)

プロジェクト名 *

TestProject



プロジェクト ID: testproject-381014。後で変更することはできません。 [編集](#)

場所 *

組織なし

[参照](#)

親組織またはフォルダ

作成

キャンセル

．下記リンクより、画面の指示に従い、「Fitness API」を有効にする。
<https://console.cloud.google.com/flows/enableapi?apiid=fitness&hl=ja>

下記画面にて「次へ」をクリック

API へのアクセスの有効化

1 プロジェクトを確認

2 API を有効にする

プロジェクト「testproject-health」に変更を加えようとしています。これが使用したいプロジェクトでない場合は、上にあるプロジェクトセレクタを使用して別のプロジェクトを選択するか作成できます。

[次へ](#)

「有効にする」をクリック

API へのアクセスの有効化

✓ プロジェクトを確認

2 API を有効にする

「Fitness API」を有効にしようとしています。

有効にする

下記のような表示になることを確認

API へのアクセスの有効化

✓ プロジェクトを確認

✓ API を有効にする

「Fitness API」を有効にしようとしています。

有効にする

・下記リンクより、認証情報の画面に進む。

<https://console.cloud.google.com/apis/credentials?referrer=search&hl=ja>

・[新しい認証情報]をクリックし、[OAuth クライアント ID]を選択する。

API API とサービス

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ページの使用に関する契約

認証情報

+ 認証情報を作成

削除

削除した認証情報を復元

有効な API にアクセス

必ず、API

API キー

名前

表示する API キー

API キー

シンプル API キーを使用してプロジェクトを識別し、割り当てとアクセスを確認します

OAuth クライアント ID

ユーザーのデータにアクセスできるようにユーザーの同意をリクエストします

サービス アカウント

ロボット アカウントによるサーバー間でのアプリレベルの認証を有効にします

ウィザードで選択

使用する認証情報の種類を決定するため、いくつかの質問をします

作成日 ↓

OAuth 2.0 クライアント ID

- ・ OAuth 同意画面の設定が必要な場合は、画面の指示に従い進める。
「同意画面を設定」をクリック

Google Cloud

TestProject-Health

スラッシュ (/) を使用してリソース、ドキュメント、プロダクトな...

検索

2

?

:

d

API API とサービス

← OAuth クライアント ID の作成

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ページの使用に関する契約

クライアント ID は、Google の OAuth サーバーで個々のアプリを識別するために使用します。アプリが複数のプラットフォームで実行される場合、それぞれに独自のクライアント ID が必要になります。詳しくは、[OAuth 2.0 の設定](#) をご覧ください。OAuth クライアントの種類の[詳細](#)

OAuth クライアント ID を作成するには、まず同意画面で設定を行う必要があります

同意画面を設定

User Typeは基本的には外部を選択する。

Google Cloud

TestProject-Health

API API とサービス

OAuth 同意画面

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ページの使用に関する契約

アプリをどのように構成および登録するか（ターゲットユーザーを含む）を選択します。プロジェクトに関連付けることができるアプリは1つだけです。

User Type

☐ 内部 ?

組織内のユーザーのみが使用できます。確認を受けるためにアプリを送信する必要はありません。[ユーザーの種類の詳細](#)

☒ 外部 ?

Google アカウントを持つすべてのテストユーザーが使用できます。アプリはテストモードで起動し、アプリを使用できるのは、テストユーザーのリストに追加されたユーザーに限られます。アプリを本番環境に移す準備ができたなら、アプリの確認が必要となる場合があります。[ユーザーの種類の詳細](#)

作成

下記画面では、「アプリ名」「ユーザーサポートメール」「デベロッパーの連絡先情報」を設定する。
その他の項目は未入力でも良い。
ページ一番下の「保存して次へ」をクリック

Google Cloud

TestProject-Health

API API とサービス

アプリ登録の編集

有効な API とサービス

ライブラリ

認証情報

1 OAuth 同意画面 — 2 スコープ — 3 テストユーザー — 4 概要

アプリ情報

OAuth 同意画面

ページの使用に関する契約

この情報は同意画面に表示されるため、デベロッパーのユーザー情報とデベロッパーへの問い合わせ方法をエンドユーザーが把握できます。

アプリ名 *

同意を求めるアプリの名前

ユーザー サポートメール *

ユーザーが同意に関して問い合わせるために使用

アプリのロゴ

This is your logo. It helps people recognize your app and is displayed on the OAuth

下記画面、スコープは設定しなくて良い。そのまま保存して次へをクリック。

Google Cloud

TestProject-Health

API	API とサービス	アプリ登録の編集
有効な API とサービス		<div> <div>OAuth 同意画面</div> <div>2 スコープ</div> <div>3 テストユーザー</div> <div>4 概要</div> </div>
ライブラリ		<p>スコープとは、アプリのユーザーに許可を求める権限を表します。スコープを定めることで、プロジェクトからユーザーの Google アカウントにある特定の種類のプライベートなユーザーデータへのアクセスが可能になります。詳細</p> <div>スコープを追加または削除</div>
認証情報		
OAuth 同意画面		
ページの使用に関する契約		

テストユーザーに、アプリからログインしようとしているGoogleアカウント（歩数情報を取得したいユーザーのGoogleアカウント）を設定する。

Google Cloud

TestProject-Health

API	API とサービス	アプリ登録の編集
有効な API とサービス		<div> <div>OAuth 同意画面</div> <div>スコープ</div> <div>3 テストユーザー</div> <div>4 概要</div> </div>
ライブラリ		<div>テストユーザー</div> <p>公開ステータスが「テスト中」に設定されている間は、テストユーザーのみがアプリにアクセスできます。アプリの確認前の許可済みユーザー数の上限は 100 で、この上限はアプリの全期間でカウントされます。詳細</p> <div>+ ADD USERS</div>
認証情報		
OAuth 同意画面		
ページの使用に関する契約		

下記の画面になったら、次の手順へ。

Google Cloud

TestProject-Health

・再度下記リンクへ遷移し、「認証情報を作成」→「OAuthクライアントID」を選択する。
<https://console.cloud.google.com/apis/credentials?referrer=search&hl=ja>

・ [アプリケーションの種類] で [Android] を選択する。
また、アプリの SHA-1 フィンガープリントとパッケージ名を入力する。
Debug用フィンガープリントの取得コマンド

☰

Google Cloud

TestProject-Health ▼

API API とサービス

OAuth クライアント ID の作成

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

クライアント ID は、Google の OAuth サーバーで個々のアプリを識別するために使用します。アプリが複数のプラットフォームで実行される場合、それぞれに独自のクライアント ID が必要になります。詳しくは、[OAuth 2.0 の設定](#)をご覧ください。OAuth クライアントの種類の[詳細](#)

アプリケーションの種類 *
Android ▼

名前 *
Android クライアント 1

OAuth 2.0 クライアントの名前。この名前はコンソールでクライアントを識別するためにのみ使用され、エンドユーザーには表示されません。

パッケージ名 *

AndroidManifest.xml ファイル内。

SHA-1 証明書のフィンガープリント *

SHA-1 署名証明書フィンガープリントによって Android アプリのみに使用を制限します。
[詳細](#)

Use this command to get the fingerprint.

```
$ keytool -keystore path-to-debug-or-production-keystore -list
```

注: 設定が有効になるまで 5 分から数時間かかることがあります

作成 キャンセル

- ・ 作成をクリックする。

Flutterプロジェクト作成

まずはFlutterのプロジェクトを作成する。
この中では、歩数データ取得に必要な対応について解説する。

- ・ Android向けパーミッション取得の処理を実装する。
Androidで android.permission.ACTIVITY_RECOGNITION の権限許可を求めるために、permission_handlerパッケージを使用導入する。

- ・ パーミッション状態の判定

```
PermissionStatus status = await Permission.activityRecognition.status;
```

- ・ ユーザーへのパーミッションリクエスト

```
PermissionStatus newStatus = await Permission.activityRecognition.request();
```

また、AndroidManifestに下記を記載する。

```
<manifest ...>  
  <uses-permission android:name=android.permission.ACTIVITY_RECOGNITION />  
  ...  
</manifest>
```

- ・ iOS向け設定の追加

ios/Runner/Info.plistに下記を記載する。

```
<key>NSHealthShareUsageDescription</key>
<string>私たちは、お客様の健康データを使用して、お客様にパーソナライズされた推奨コンテンツを提供します。</string>
<key>NSHealthUpdateUsageDescription</key>
<string>私たちは、お客様の健康データを使用して、お客様にパーソナライズされた推奨コンテンツを提供します。</string>
```

また、XCodeの設定より、CapabilitiesのHealthKitを追加する。
(事前にApp IDの作成と、作成時にHealth KitのCapabilitiesを追加しておいてください。)

・ healthパッケージの導入

healthパッケージを使うためのインスタンス取得と、歩数取得のtypeの定義を行う。

```
final health = HealthFactory();
final types = [HealthDataType.STEPS];
```

healthパッケージが利用する健康情報へのアクセスをユーザーにリクエストする。

```
final _isAuthorized = await health.requestAuthorization(types);
```

健康状態へのアクセスを許可されれば、下記のようなコードで期間を指定して歩数情報を取得できる。

```
DateTime startDate = DateTime.now().subtract(const Duration(days: 31));
DateTime endDate = DateTime.now();

List<HealthDataPoint> healthData = await health.getHealthDataFromTypes(
  startDate,
  endDate,
  types,
);
```