

Power Apps Hands on

2. 環境準備/ログイン

2.1. Power Apps コミュニティプランの新規作成

既にOffice 365 のアカウントをお持ちだったり
既にコミュニティプランをお持ちの方は
スキップしていただいて構いません。

1. Power Apps コミュニティプラン サイトにアクセスする
<https://powerapps.microsoft.com/ja-jp/communityplan/>
2. 「無料で今すぐ開始」をクリックする

Microsoft | Power Apps 概要 料金設定 パートナー 考察 コミュニティ

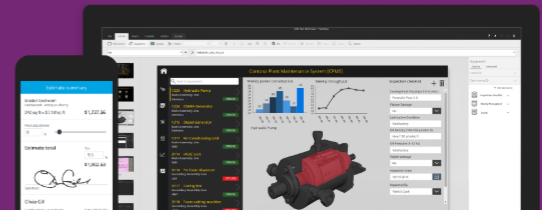
Power Apps Community Plan

個人使用向けの無料環境を入手

- Power Apps と Power Automate 技術の学習および向上
- すべての機能が備わった開発環境の詳細と作成
- Common Data Service で、単一の拡張可能なデータのビューにアプリを構築

無料で今すぐ開始

既存のユーザーの場合個人環境の作成



ビジネス アプリとワークフローの作成

コードを書かずにアプリやフローを作成できます。100を超える組み込みコネクタで、複数のデータソースに同時に接続できます。



個人での使用は無料

Power Apps、Power Automate および Common Data Service についての詳細をご覧ください。この環境では、ユーザーとデータストレージに制限があります。



よりスマートなアプリの開発と配布

CDS に構築されたアプリおよびフローでは、Dynamics 365、Office 365、Azure Active Directory などの既存のデータを使用できます。AppSource にソリューション

3. サインアップを行う

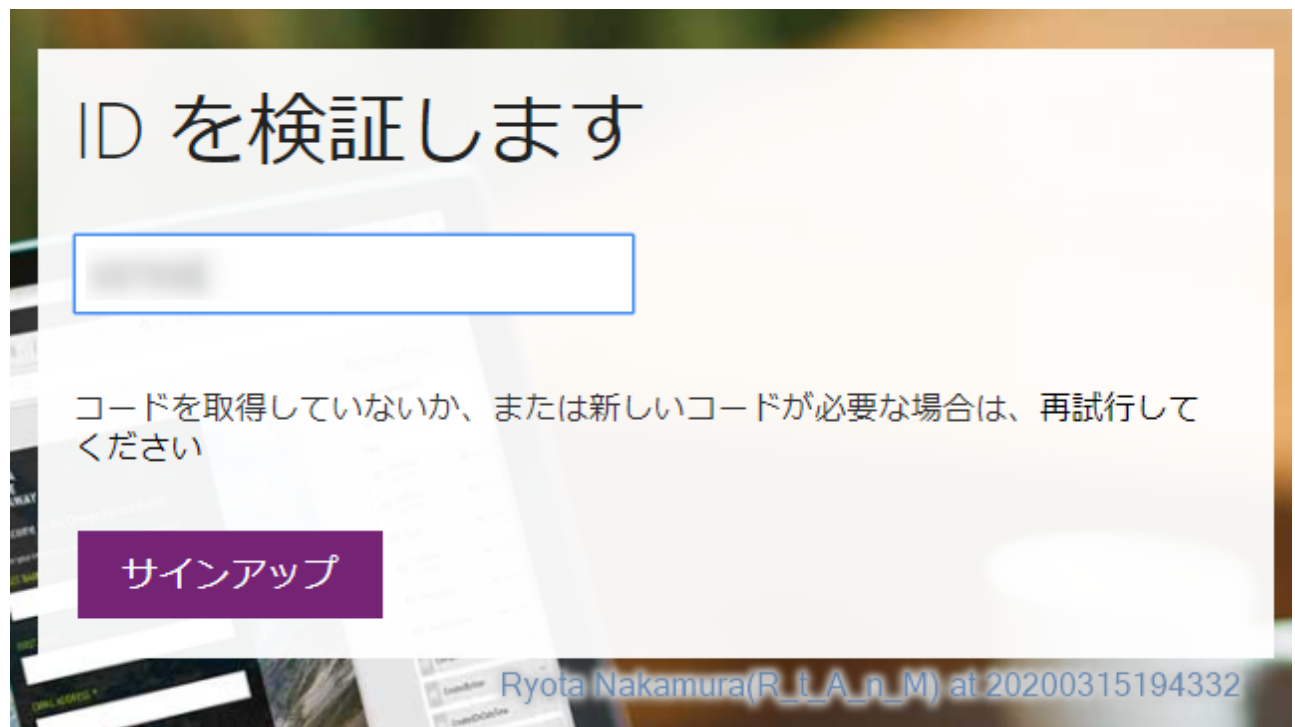


※gmailやhotmailなどは使用できません。

4. `電話もしくはSMS認証を行う



5. 取得した認証コードを入力して認証を行う



6. アカウントの詳細を入力して「開始」をクリックする

最後に...

姓 (例: 田中) 名 (例: 太郎)

パスワードの作成

パスワードの確認

に確認コードが送信されました。コードを入力して、サインアップを完了してください。

確認コード [新規登録コードの再送信](#)

国または地域 ▼
[国または地域が表示されていない場合](#)

開始を選択すると、[契約条件](#)と [Microsoft プライバシー ポリシー](#)に同意し、お客様のメール アドレスが組織に関連付けられている (かつ、個人用またはコンシューマーのメール アドレスではない) ことを確認したものと見なされます。また、お客様の組織の管理者がお客様のアカウントとデータの管理を担当することがあり、お客様の名前、メール アドレス、試用組織名が組織内の他のユーザーに表示されることを理解したものと見なされます。[詳細をご確認ください](#)。

開始

Ryota Nakamura(R_t_A_n_M) at 20200315194623

2.2. Azure の新規登録

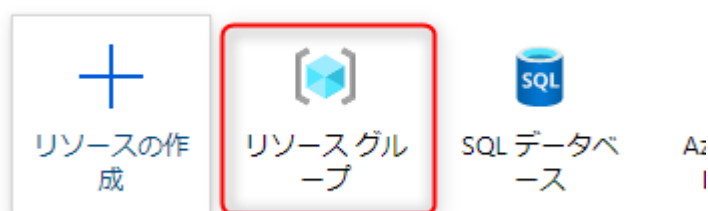
既にAzureをご利用中の場合はスキップして頂いて構いません。

3. Azure側の準備

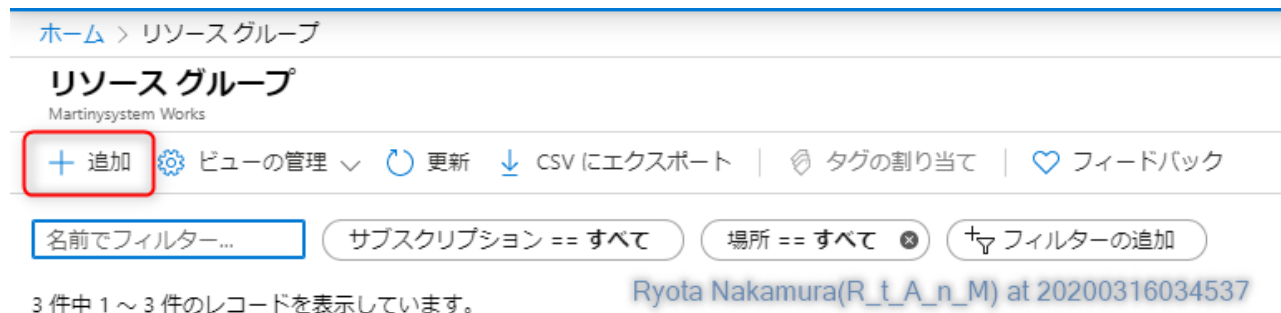
3.1. リソースグループの作成

1. リソースグループをクリックする

Azure サービス



2. 「追加」をクリックする



3. リソースグループのパラメータを指定する

リソースグループを作成します

基本 タグ 確認および作成

リソースグループ - Azure ソリューションの関連リソースを保持するコンテナ。リソースグループには、ソリューションのすべてのリソースを含めることも、グループとして管理したいリソースのみを含めることもできます。組織にとって最も有用なことに基づいて、リソースグループにリソースを割り当てる方法を決めてください。 [詳細情報](#)

プロジェクトの詳細

サブスクリプション * ⓘ

従量課金

①サブスクリプションを指定する

リソースグループ * ⓘ

technight

②任意のリソースグループ名を入力する

リソースの詳細

リージョン * ⓘ

(Asia Pacific) 東日本

③リージョンを指定する(東日本推奨)

④確認及び作成をクリックする

確認および作成

< 前へ

次: タグ >

Ryota Nakamura(R_t_A_n_M) at 20200316034700

4. 作成をクリックする。

リソースグループを作成します

✓ 検証に成功しました。

基本 タグ 確認および作成

基本

サブスクリプション	従量課金
リソースグループ	technight
リージョン	東日本

作成

< 前へ

次へ >

3.2. ストレージアカウントの作成

1. 作成したリソースグループから「追加」をクリックする

△ > リソースグループ > technight

technight
リソースグループ

検索 (Ctrl+/) << **+ 追加** ≡ 列の編集 🗑 Delete res

概要
アクティビティ ログ
アクセス制御 (IAM)
タグ

サブスクリプション... : 従量課金
サブスクリプション ID : 7f4e1ac3-021
タグ (変更) : タグを追加す

Ryota Nakamura(R_t_A_n_M) at 20200316035454
名前フィルター... (種類 == す

2. ストレージアカウントを選択する

ホーム > リソースグループ > technight > 新規

新規

① Storage と入力する

Storage

- StorSimple Virtual Device Series
- StorSimple Physical Device Series
- Nutanix Storage
- Storage Resource Monitor
- Storage account - blob, file, table, queue**

② Storage Account 選択する

ブロックチェーン
Compute
コンテナ
データベース

SQL Database
クイックスタートとチュートリアル

Ryota Nakamura(R.T.A.n.M) at 20200316035634

3. ストレージアカウントのパラメータを指定する

ストレージ アカウントの作成

基本 ネットワーク 詳細 タグ 確認および作成

Azure Storage は、高可用性、セキュリティ、耐久性、スケーラビリティ、冗長性を備えたクラウド ストレージを提供する Microsoft が管理するサービスです。Azure Storage には、Azure BLOB (オブジェクト)、Azure Data Lake Storage Gen2、Azure Files、Azure Queues、Azure Tables が含まれます。ストレージ アカウントのコストは、使用量と、下で選ぶオプションに応じて決まります。 [Azure ストレージ アカウントの詳細](#)

プロジェクトの詳細

デプロイされているリソースとコストを管理するサブスクリプションを選択します。フォルダーのようなリソース グループを使用して、すべてのリソースを整理し、管理します。

サブスクリプション *

リソースグループ *

[新規作成](#)

インスタンスの詳細

既定の展開モデルは Resource Manager であり、これは最新の Azure 機能をサポートしています。代わりに、従来の展開モデルを使った展開も選択できます。 [クラシック展開モデルを選択します](#)

ストレージ アカウント名 * ①

場所 *

パフォーマンス ① ☒ Standard ☐ Premium ② Standard を選択する

アカウントの種類 ① ③ BlobStorage を選択する

レプリケーション ① ④ ローカル冗長ストレージ (LRS) を指定する

⑤ クールを選択する

アクセス層 (既定) ① ☒ クール ☐ ホット

⑥ 確認および作成をクリックする

確認および作成

< 前へ

次: ネットワーク >

Ryota Nakamura(R_t_A_n_M) at 20200316040218

4. 作成をクリックする。

ストレージ アカウントの作成

✓ 検証に成功しました

基本 ネットワーク 詳細 タグ 確認および作成

基本

サブスクリプション	従量課金
リソース グループ	technight
場所	東日本
ストレージ アカウント名	technightstg
デプロイ モデル	Resource Manager
アカウントの種類	BlobStorage
レプリケーション	ローカル冗長ストレージ (LRS)
パフォーマンス	Standard
アクセス層 (既定)	クール

ネットワーク

接続方法 パブリックエンドポイント (すべてのネットワーク)

詳細

安全な転送が必須	有効
BLOB の論理的な削除	無効
BLOB の変更フィード	無効
階層構造の名前空間	無効
NFS v3	無効

作成

< 前へ

次 Ryota Nakamura(R_t_A_n_M) at 20200316040639

3.3. Face APIの作成

1. 作成したリソースグループから「追加」をクリックする

△ > リソースグループ > technight

technight
リソースグループ

検索 (Ctrl+/)

+ 追加

列の編集

Delete res

概要

アクティビティ ログ

アクセス制御 (IAM)

タグ

サブスクリプション... : 従量課金

サブスクリプション ID : 7f4e1ac3-021

タグ (変更) : タグを追加す

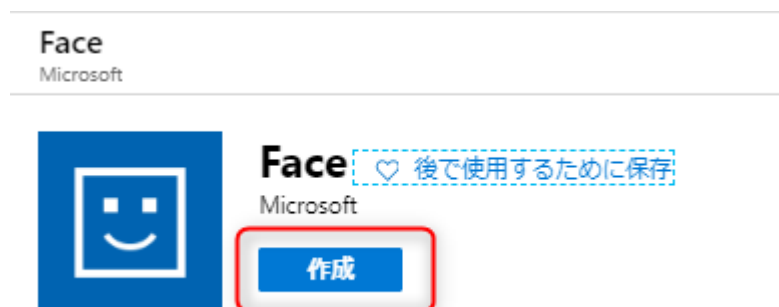
Ryota Nakamura(R_t_A_n_M) at 20200316035454

名前フィルター... (種類 == す

2. Faceを選択する



3. 作成をクリックする



4. Faceのパラメータを指定する

ホーム > technight > 新規 > Face > 作成

作成

Face

名前 * ①任意の名称を指定する

technightface

サブスクリプション *

従量課金

場所 * ②東日本を指定する

(Asia Pacific) 東日本

価格レベル (価格の詳細を表示) * ③F0を指定する

リソースグループ *

technight

新規作成

④作成をクリックする

作成 Automation options
Ryota Nakamura(R_t_A_n_M) at 20200316041345

4. Power Apps のみで顔認証アプリを作成する

4.1. Azure 上の準備

1. 作成済みのストレージアカウントを開く

ホーム > PowerAppsAIService > nakamuratest2

nakamuratest2
ストレージ アカウント

検索 (Ctrl+/) << Explorer で開く → 移動

概要

- アクティビティ ログ
- アクセス制御 (IAM)
- タグ
- 問題の診断と解決
- データ転送
- イベント

リソースグループ (変更) : Powe

状態 : プラ

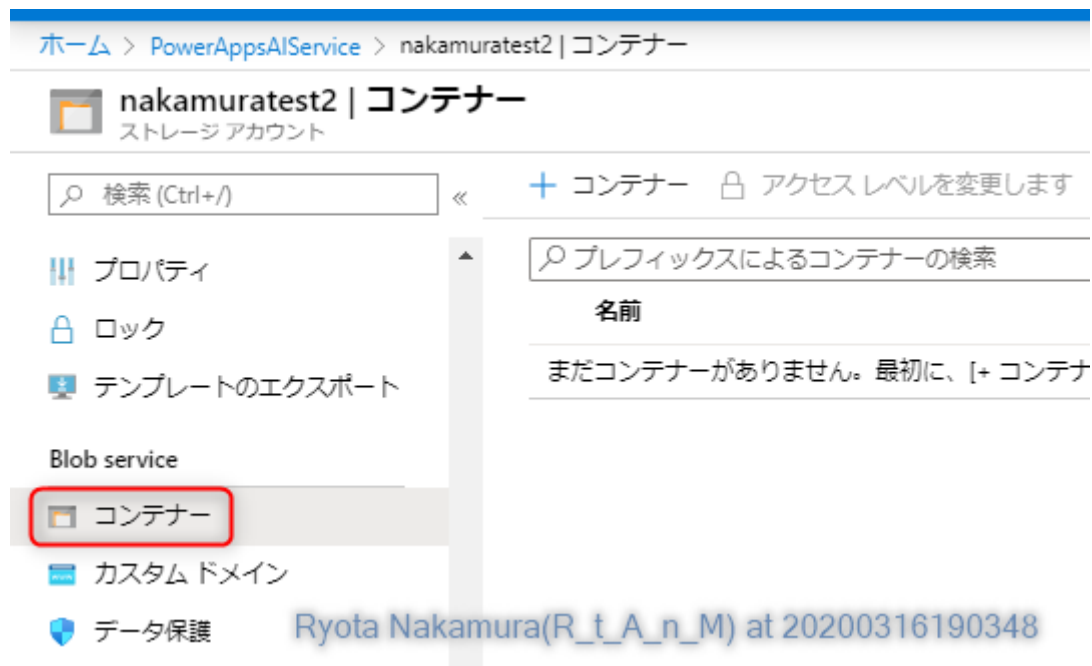
場所 : 東日

サブスクリプション... : 従量

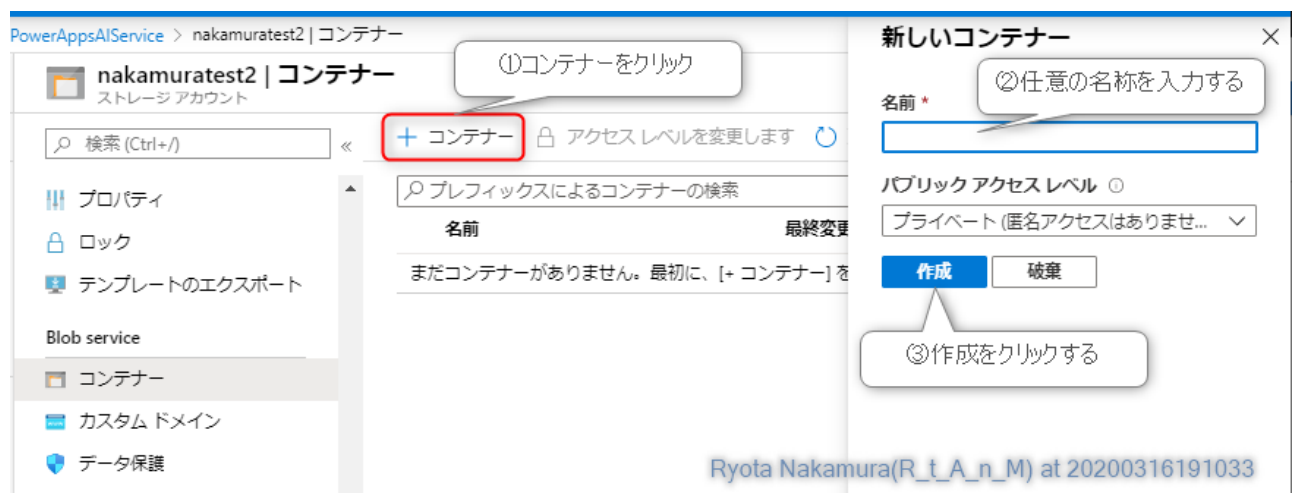
サブスクリプション ID : 7f4e1

タグ (変更) : タグ

2. コンテナを開く



3. コンテナを作成する



4. ストレージアカウントのアクセスキーを取得する

ホーム > PowerAppsAIService > nakamuratest2 | アクセス キー

nakamuratest2 | アクセス キー
ストレージ アカウント

検索 (Ctrl+/)

概要
アクティビティ ログ
アクセス制御 (IAM)
タグ
問題の診断と解決
データ転送

①アクセスキーをクリックする

設定
アクセス キー
geo レプリケーション
CORS
構成
暗号化
Shared Access Signature
ファイアウォールと仮想ネ...
プライベートエンドポイン...

この Azure ストレージ アカウントに対して要求を行う際は、アプリケーションの認証にアクセス キーを使用します。アクセス キーは、(たとえば Azure Key Vault を使用して) 安全に保存し、共有しないでください。アクセス キーは、定期的に再生成することをお勧めします。アクセス キーは 2 つ提供されるため、一方のキーを再生成しながら、もう一方のキーを使用して接続を維持することができます。

アクセス キーを再生成する場合、その新しいキーを使用するには、この Azure ストレージ アカウントにアクセスする Azure CLI ツールとアプリケーションをすべて更新する必要があります。この操作は、ストレージ アカウントの他の設定を変更することはありません。 [ストレージ アクセス キー](#)

②コピーアイコンをクリックし、ストレージアカウント 名をコピーした後、メモ帳などにペーストしておく

ストレージ アカウント名

key1

キー

③コピーアイコンをクリックし、キーをコピーした後、メモ帳などにペーストしておく

接続文字列

DefaultEndpointsProtocol=h

key2

キー

接続文字列

DefaultEndpointsProtocol=h

Ryota Nakamura(R_t_A_n_M) at 20200316192115

5. 作成済みのFace APIを開き、アクセスキーを取得する

ホーム > PowerAppsAIService > NakamuraPPFFaceAPI | クイック スタート

NakamuraPPFFaceAPI | クイック スタート
Cognitive Services

検索 (Ctrl+/)

概要
アクティビティ ログ
アクセス制御 (IAM)
タグ
問題の診断と解決
データ転送

①クイックスタートが開く

リソース管理
クイック スタート
Keys and Endpoint
価格レベル
Virtual network
Identity
サブスクリプション

完了しました。クイックスタートのガイダンスを調べて、Face を開始します。

1 開始するためのキーとエンドポイント

Key1

②コピーアイコンをクリックし、キーをコピーした後、メモ帳などにペーストしておく

エンドポイント

https://

③コピーアイコンをクリックし、エンドポイントをコピーした後、メモ帳などにペーストしておく

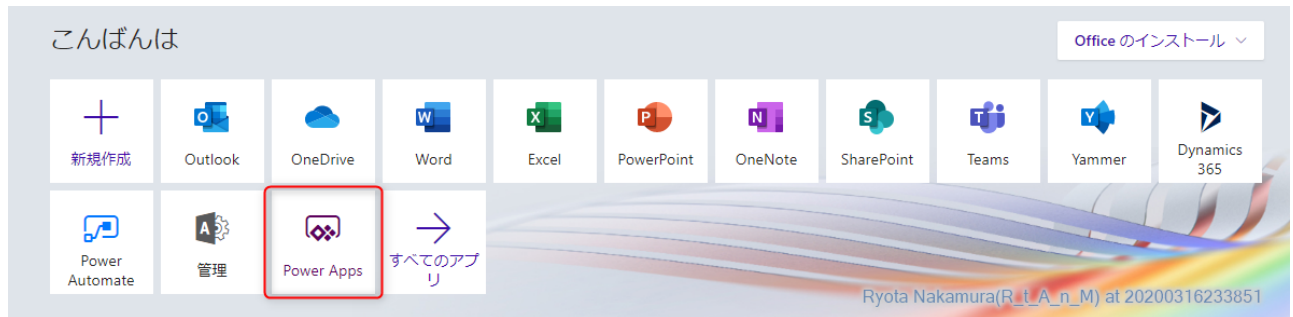
Face の Web API 呼び出しと Docker コンテナのアクティビティを行うために、このガイドの [アクティビティ](#) セクションで提供されているコマンドを実行する必要があります。Web API の場合、アクセス キーが必要です。Docker コンテナの場合は、エンドポイントが必要です。

Docker コンテナを実行する (プレビュー) | Japaneast

Ryota Nakamura(R_t_A_n_M) at 20200316192822

4.2. Power Apps 編集画面を起動

1. Power Apps を開く



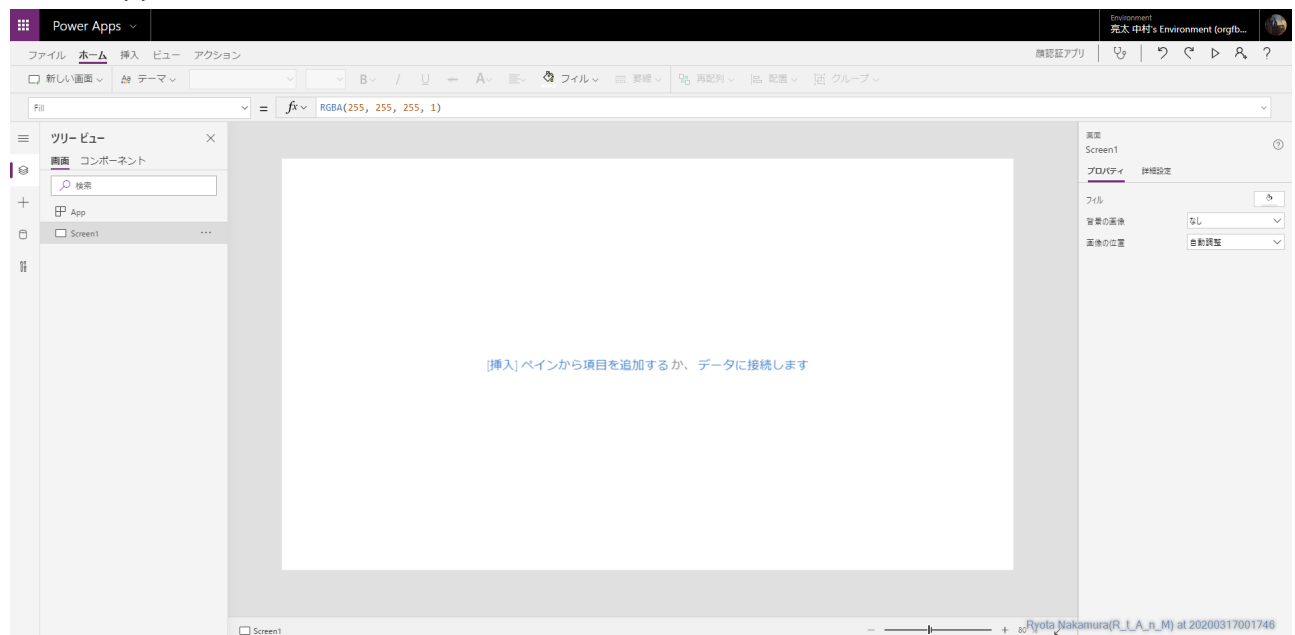
2. キャンパスアプリを一から作成をクリック



3. アプリをタブレットモードで作成する



4. Power Apps 編集画面が起動する

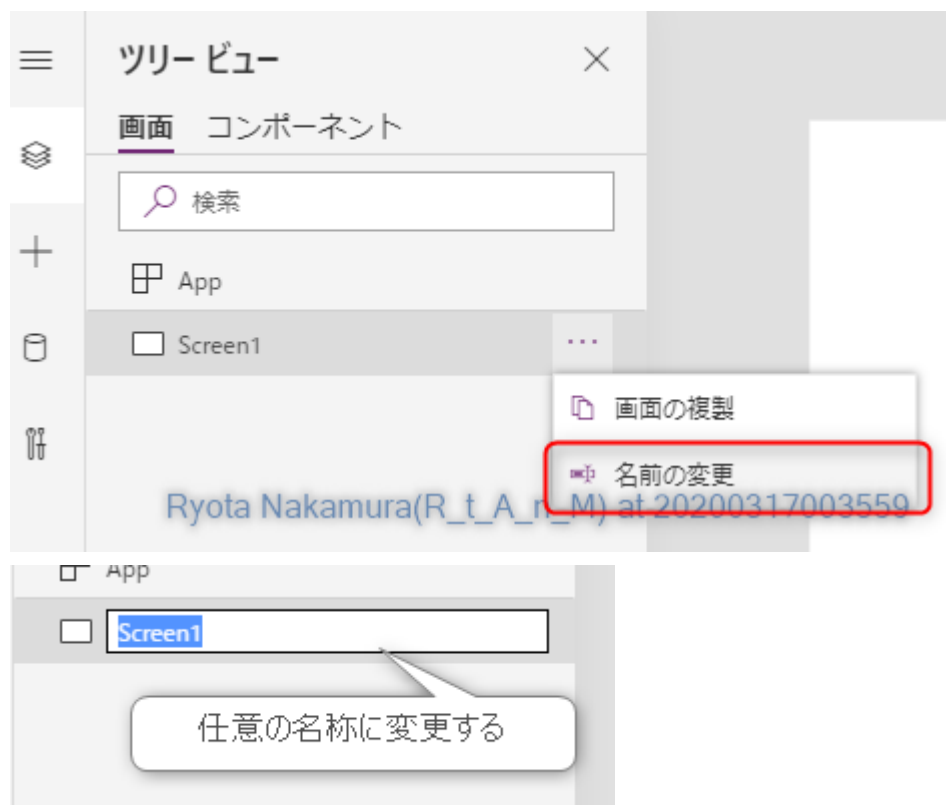


4.3. 各種画面の作成

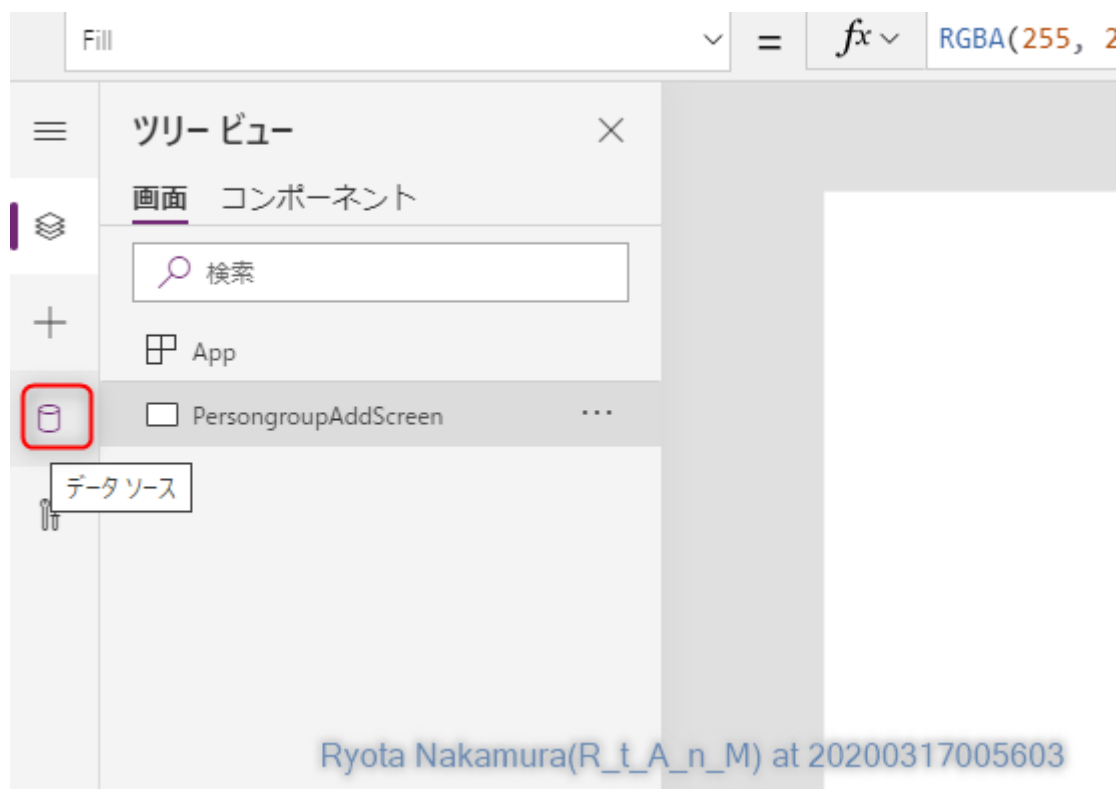
4.3.1. Person Group 作成画面

Face API の顔認証を行う為には、ユーザーを登録するための Person Groupをあらかじめ作る必要があるため、その画面を作成します。

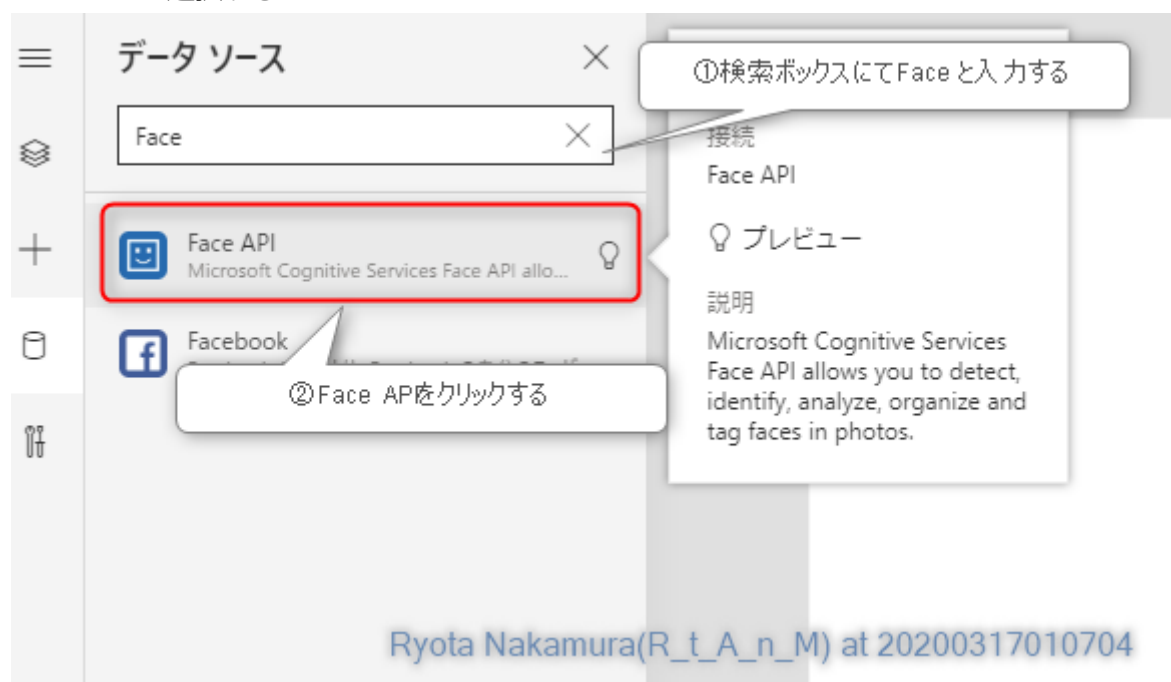
1. 画面名の変更



2. データソースを開く



3. Face API を選択する



4. 初回利用時は認証情報が必要となるので、メモ帳にペーストした内容を適宜入力する

データに接続してください

Face API

プレビュー

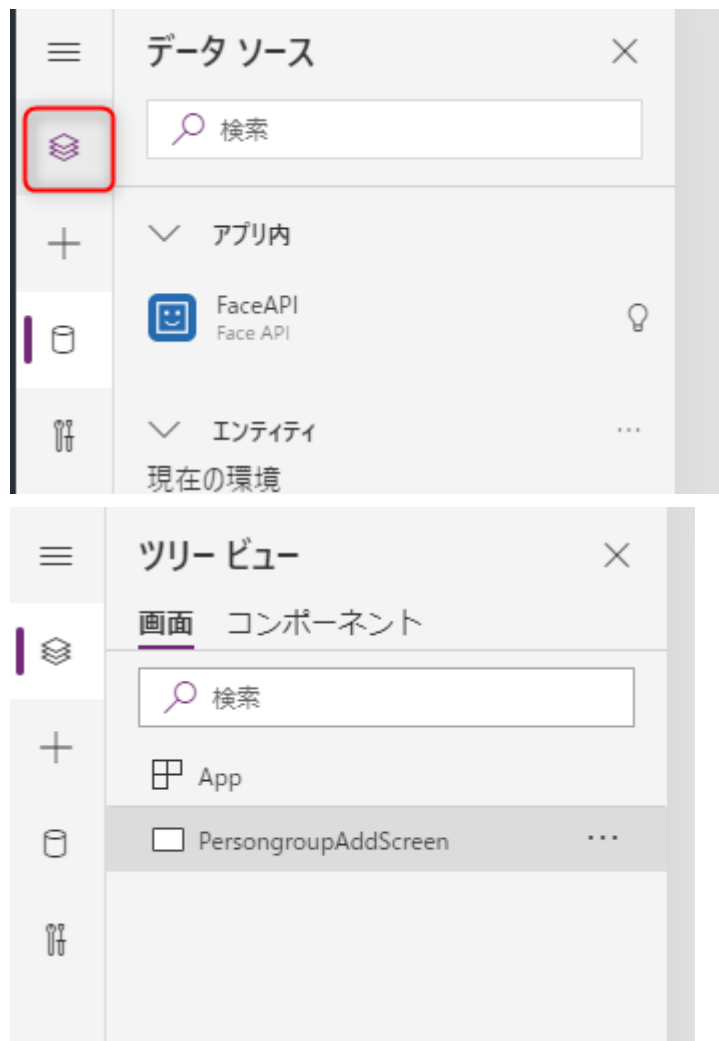
Microsoft Cognitive Services Face API allows you to detect, identify, analyze, organize and tag faces in photos.

API Key *

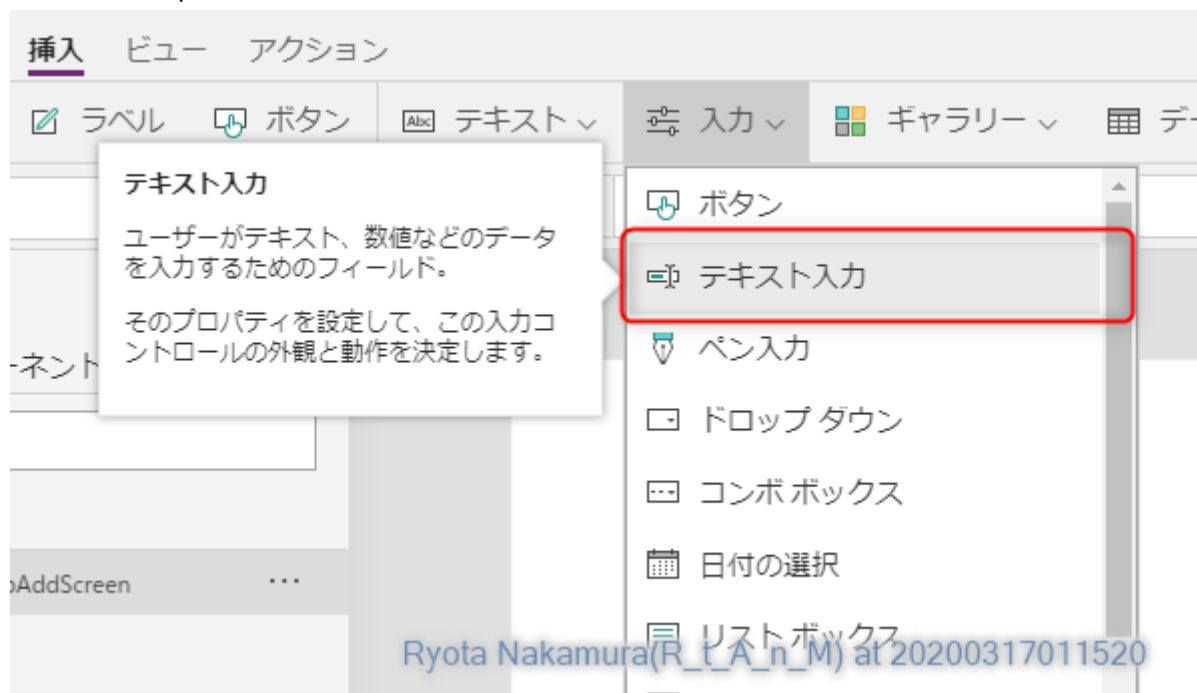
Site URL

接続 キャンセル

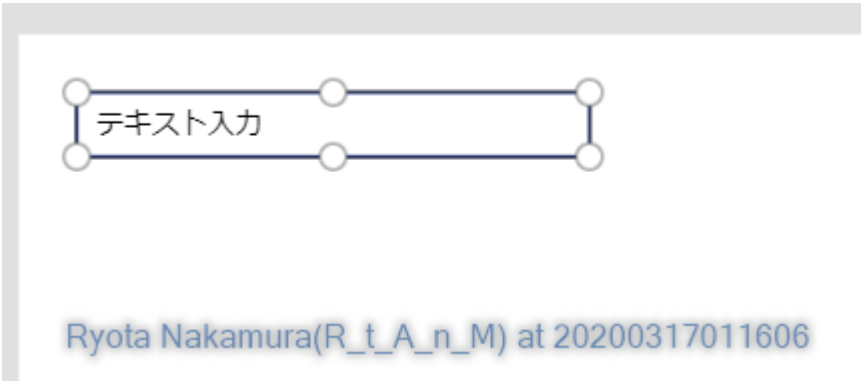
5. ツリービューに戻る



6. Person Group ID を指定する入力ボックスを作成する

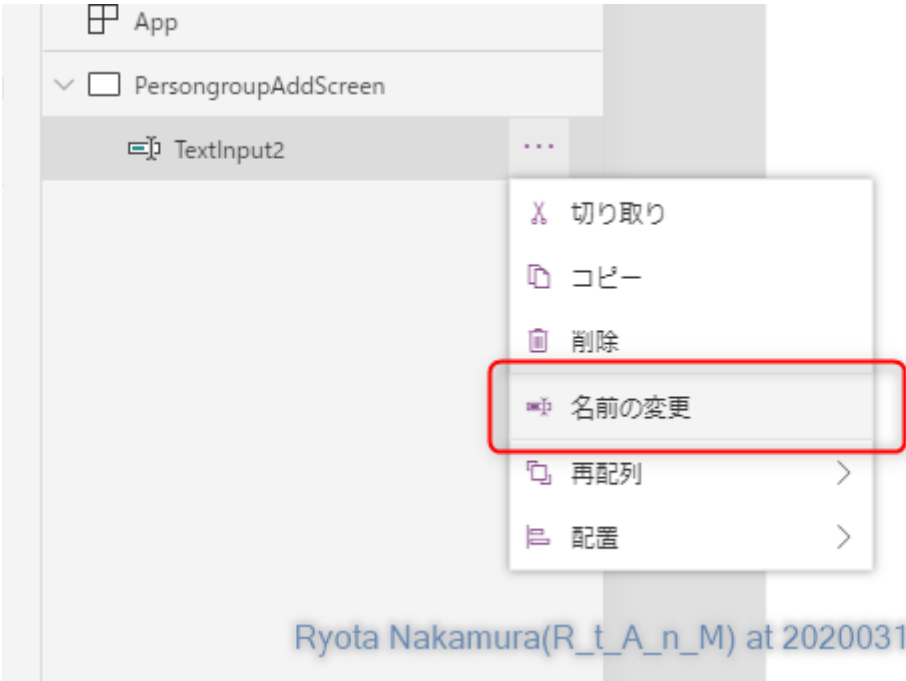


Ryota Nakamura(R_t_A_n_M) at 20200317011520

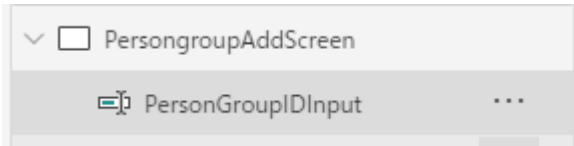


Ryota Nakamura(R_t_A_n_M) at 20200317011606

7. 作成した入力ボックスの名称を「PersonGroupIDInput」に変更する

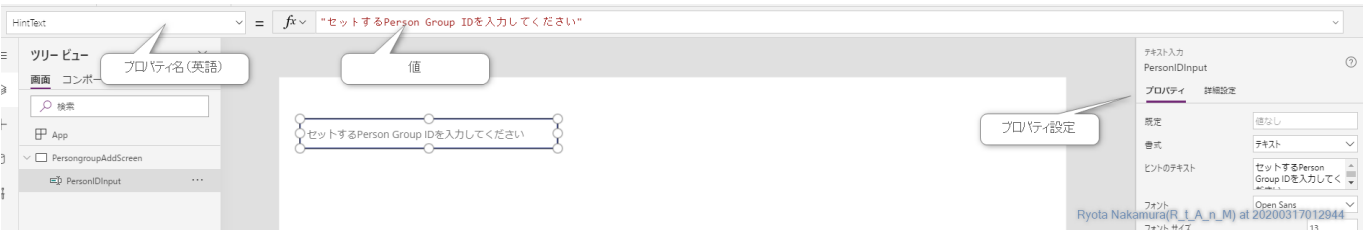


Ryota Nakamura(R_t_A_n_M) at 20200317011700



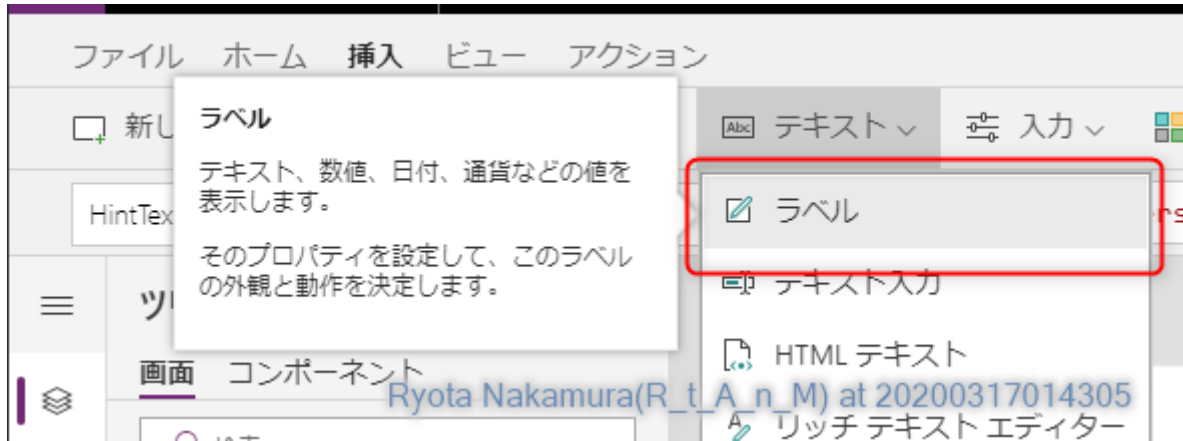
8. PersonGroupIDInputのプロパティを以下のように変更する。

プロパティ名	日本語名	値
Default	既定	
HintText	ヒントのテキスト	セットするPerson Group IDを入力してください



Ryota Nakamura(R_t_A_n_M) at 20200317012944

9. PersonIDInputのラベルを作成する



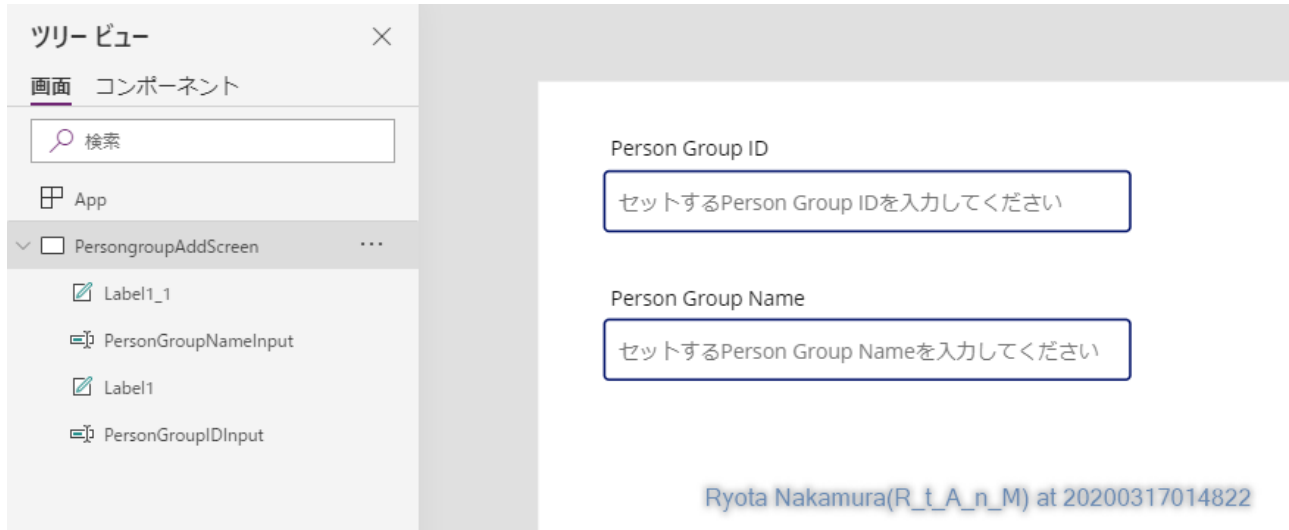
プロパティ名	日本語名	値
Text	テキスト	Person Group ID

Person Group ID

セットするPerson Group IDを入力してください

Ryota Nakamura(R_t_A_n_M) at 20200317014456

10. 同様にPerson Group Nameを入力するテキストボックス(PersonGroupNameInput)とラベルを作成する。

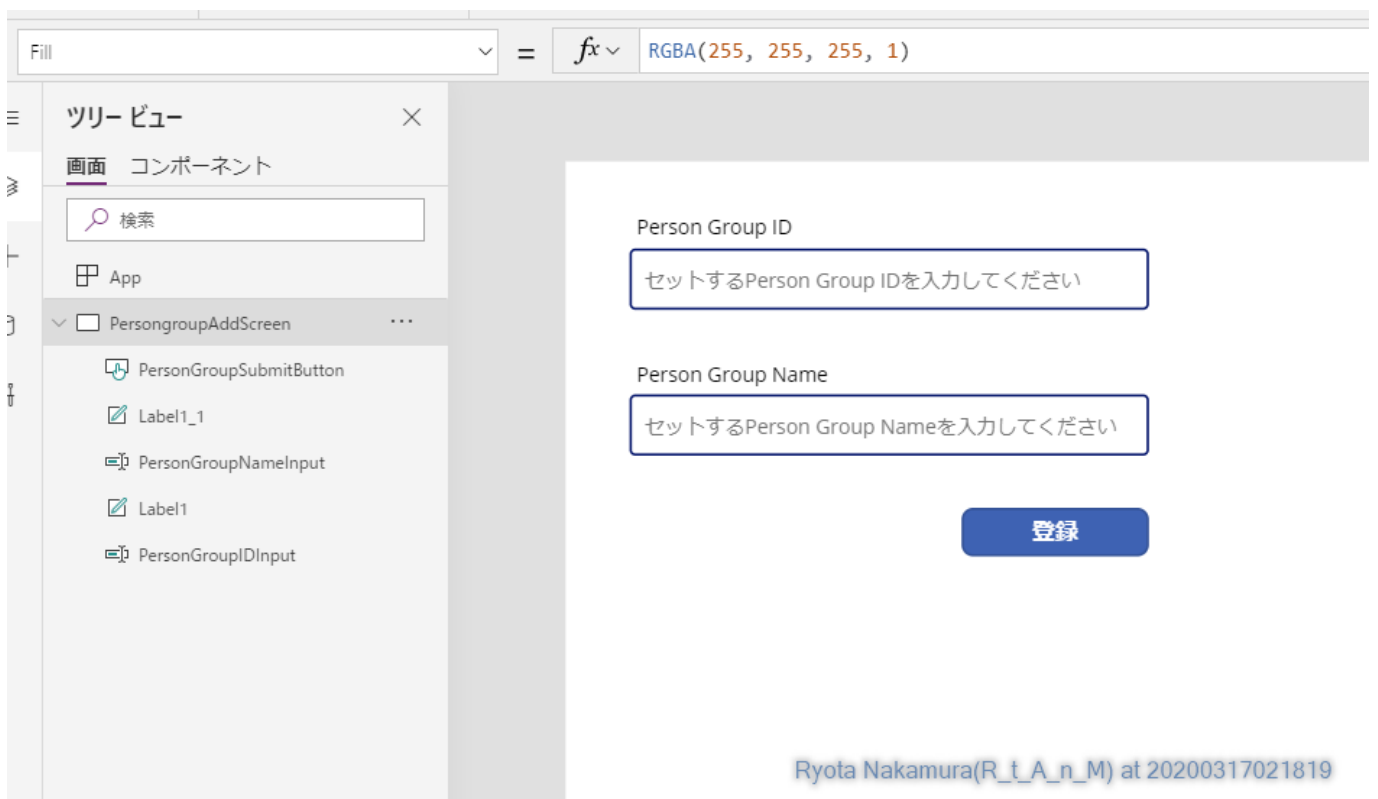


Ryota Nakamura(R_t_A_n_M) at 20200317014822

11. 登録用のボタン(PersonGroupSubmitButton)を作成する。



プロパティ名	日本語名	値
Text	テキスト	登録



12. PersonGroupSubmitButton の OnSelect プロパティに以下を指定する。

```
//Person Group 作成処理
FaceAPI.CreatePersonGroup(PersonGroupIDInput.Text,PersonGroupNameInput.Text);
```

13. PersonGroupSubmitButton の DisplayMode プロパティに以下を指定する。

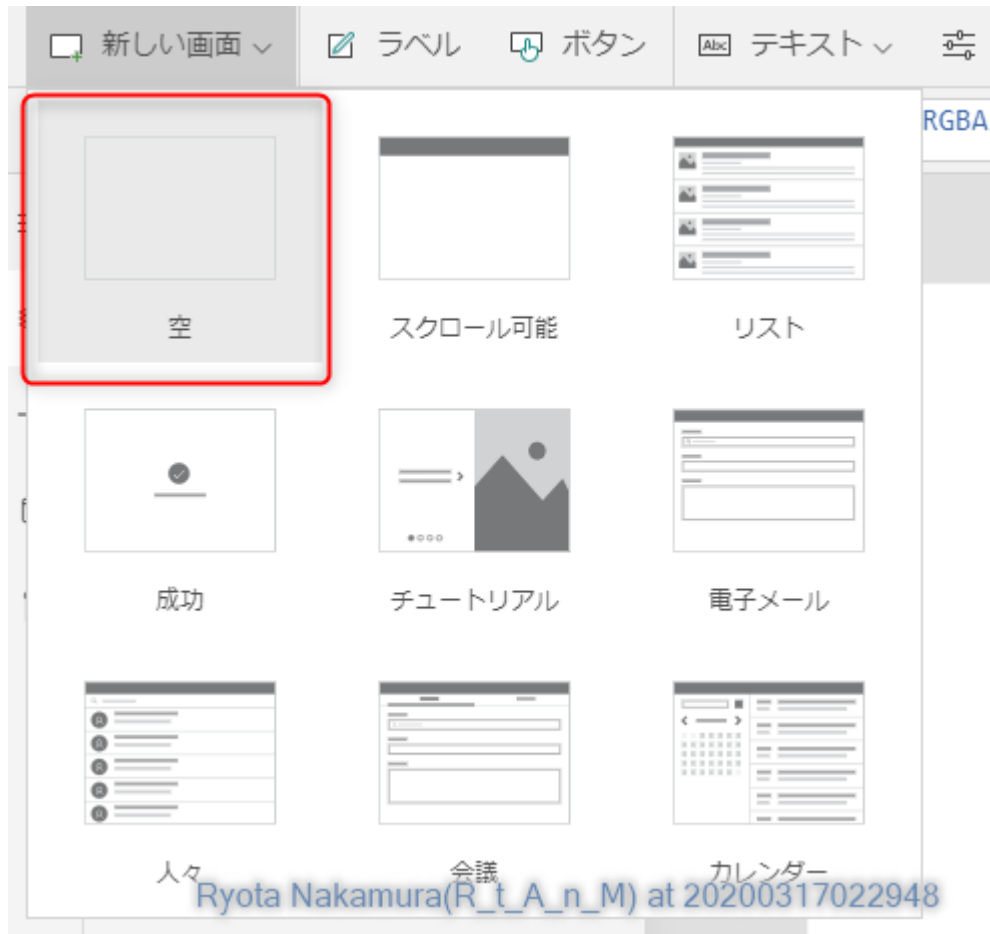
```
//IDとNameが未入力の場合はボタン操作を無効化する
If(Or(IsBlank(PersonGroupIDInput.Text),IsBlank(PersonGroupNameInput.Text)),Display
Mode.Disabled,DisplayMode.Edit)

//PersonAddScreen に移動
```

```
Navigate(PersonAddScreen,ScreenTransition.Fade)
```

4.3.2. Person 作成画面

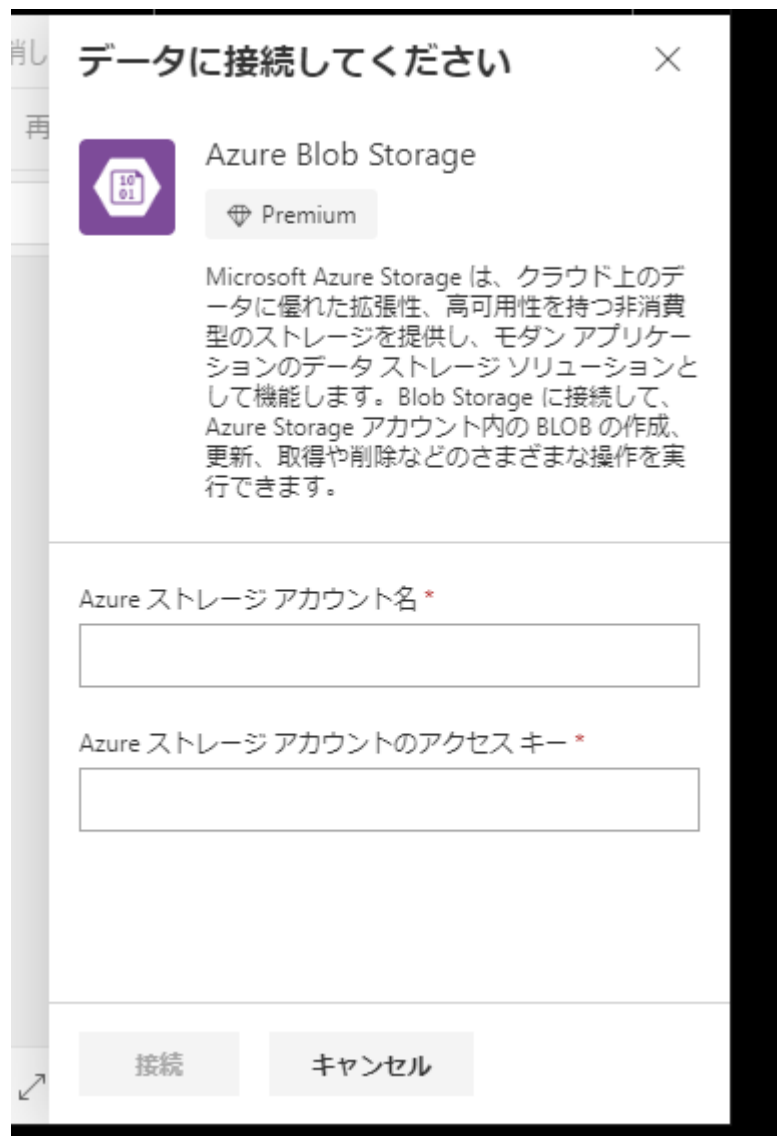
1. 新たに空の画面を作成し、PersonAddScreenとする。



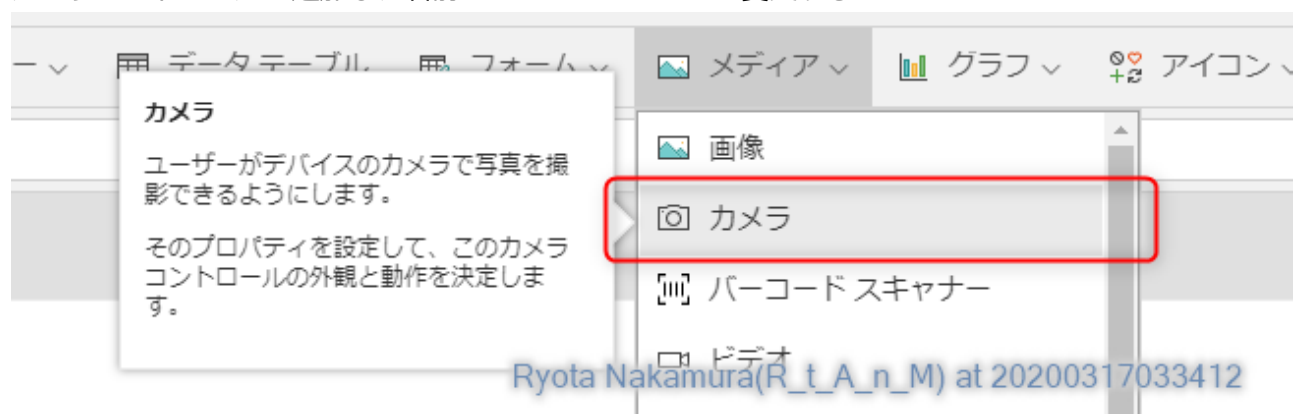
2. データソースを開く
3. Azure Blob Storage を選択する

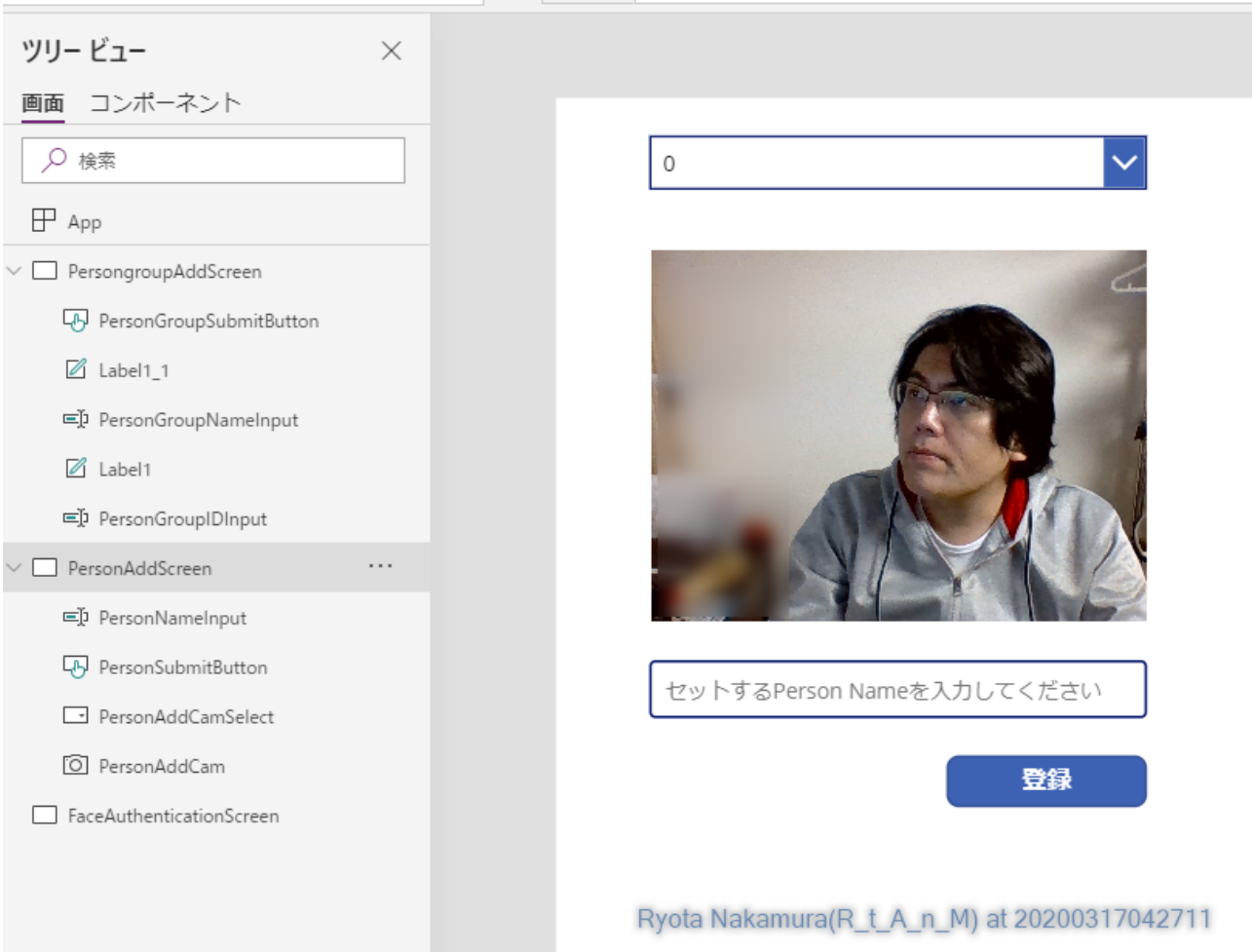


4. 初回利用時は認証情報が必要となるので、メモ帳にペーストした内容を適宜入力する

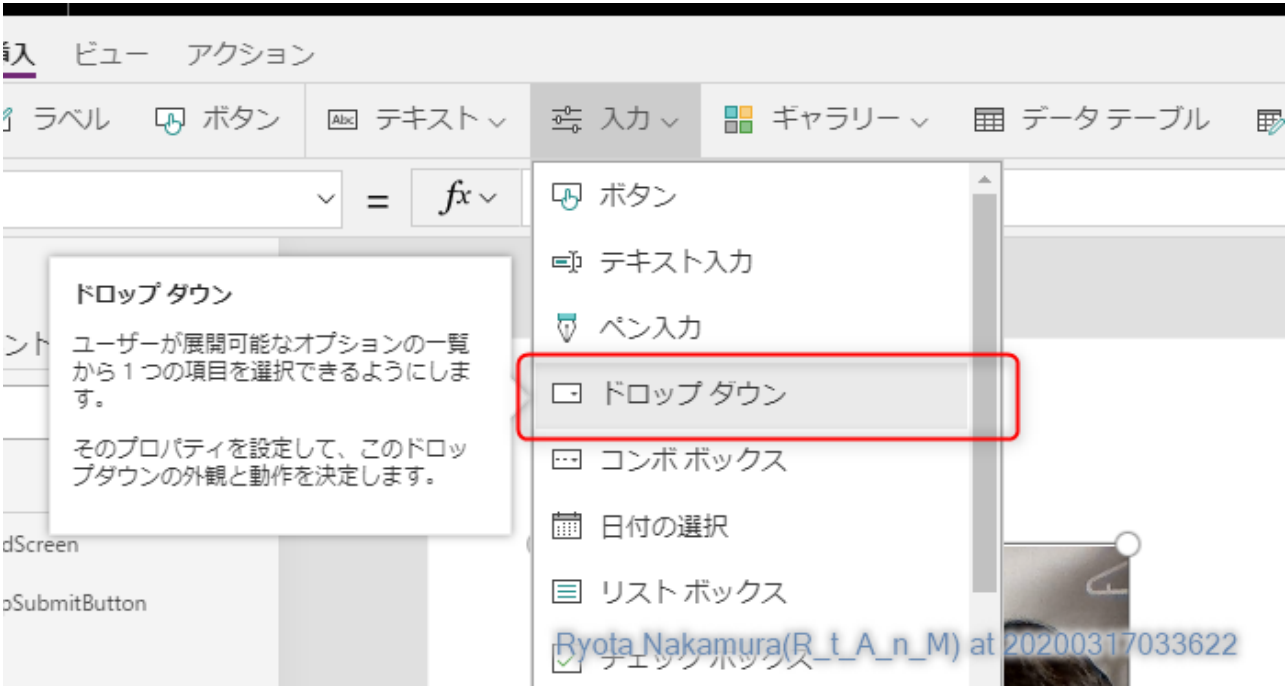


5. カメラコントロールを追加し、名前を PersonAddCam に変更する。





6. 複数のカメラがある場合の選択用として、ドロップダウンボックス（PersonAddCamSelect）を作成する



プロパティ名	日本語名	値
Items	--	[0,1,2]
Default	既定	0

7. PersonAddCam のプロパティを以下のように変更する

プロパティ名	日本語名	値
Camera	カメラ	PersonAddCamSelect.SelectedText.Value
StreamRate	ストリームレート	100

8. Person Nameを入力するテキストボックス(PersonNameInput)とラベルを作成する。

プロパティ名	日本語名	値
Default	既定	
HintText	ヒントのテキスト	セットするPerson Nameを入力してください

9. 登録用のボタン(PersonSubmitButton)を作成する。

プロパティ名	日本語名	値
Text	テキスト	登録

ツリー ビュー

画面 コンポーネント

検索

App


PersongroupAddScreen

PersonGroupSubmitButton
Label1_1
PersonGroupNameInput
Label1
PersonGroupIDInput

PersonAddScreen

PersonNameInput
PersonSubmitButton
PersonAddCamSelect
PersonAddCam

0



セットするPerson Nameを入力してください

登録

Ryota Nakamura(R_t_A_n_M) at 20200317042734

10. PersonSubmitButton の OnSelect プロパティに以下を指定する。

```
//Person データ作成
Set(PersonID,FaceAPI.CreatePerson(PersonGroupIDInput.Text,PersonNameInput.Text));
```

```
//顔認証登録用イメージアップロード
AzureBlobStorage.CreateBlockBlob("<コンテナ名>", "learn.jpg", PersonAddCam.Stream);

//顔認証登録用イメージURL生成
Set(learnimageuri, AzureBlobStorage.CreateShareLinkByPath("<コンテナ名>/learn.jpg"));

//顔認証用イメージ登録
FaceAPI.AddPersonFace(PersonGroupIDInput.Text, PersonID.personId, learnimageuri.WebUrl);

//FaceAuthenticationScreen に移動
Navigate(FaceAuthenticationScreen, ScreenTransition.Fade)
```

11. PersonSubmitButton の DisplayMode プロパティに以下を指定する。

```
//Nameが未入力の場合はボタン操作を無効化する
If(IsBlank(PersonNameInput.Text), DisplayMode.Disabled, DisplayMode.Edit)
```

4.3.3. 顔認証画面

1. 新たに空の画面を作成し、FaceAuthenticationScreenとする。
2. ドロップダウンボックス (AuthenticateCamSelect) を作成する

プロパティ名	日本語名	値
Items	--	[0,1,2]
Default	既定	0

3. カメラコントロールを追加し、名前を AuthenticateCam に変更する。

プロパティ名	日本語名	値
Camera	カメラ	AuthenticateCamSelect
StreamRate	ストリームレート	100

4. `認証用のボタン(AuthenticateSubmitButton)を作成する。

プロパティ名	日本語名	値
Text	テキスト	認証

5. AuthenticateSubmitButton の OnSelect プロパティに以下を指定する。

```
//顔認証用イメージアップロード
AzureBlobStorage.CreateBlockBlob("<コンテナ名>
```

```

>","authenticate.jpg",AuthenticateCam.Stream);

//顔認証用イメージURL生成
Set(authenticateimageuri,AzureBlobStorage.CreateShareLinkByPath("<コンテナ名
>/authenticate.jpg"));

//イメージ情報取得
Set(FaceIDdata,FaceAPI.Detect(authenticateimageuri.WebUrl,{returnFaceId:"true"}));

//認証結果取得
Set(FaceVerify,FaceAPI.Verify(First(FaceIDdata).faceId,PersonGroupIDInput.Text,Per
sonID.personId))

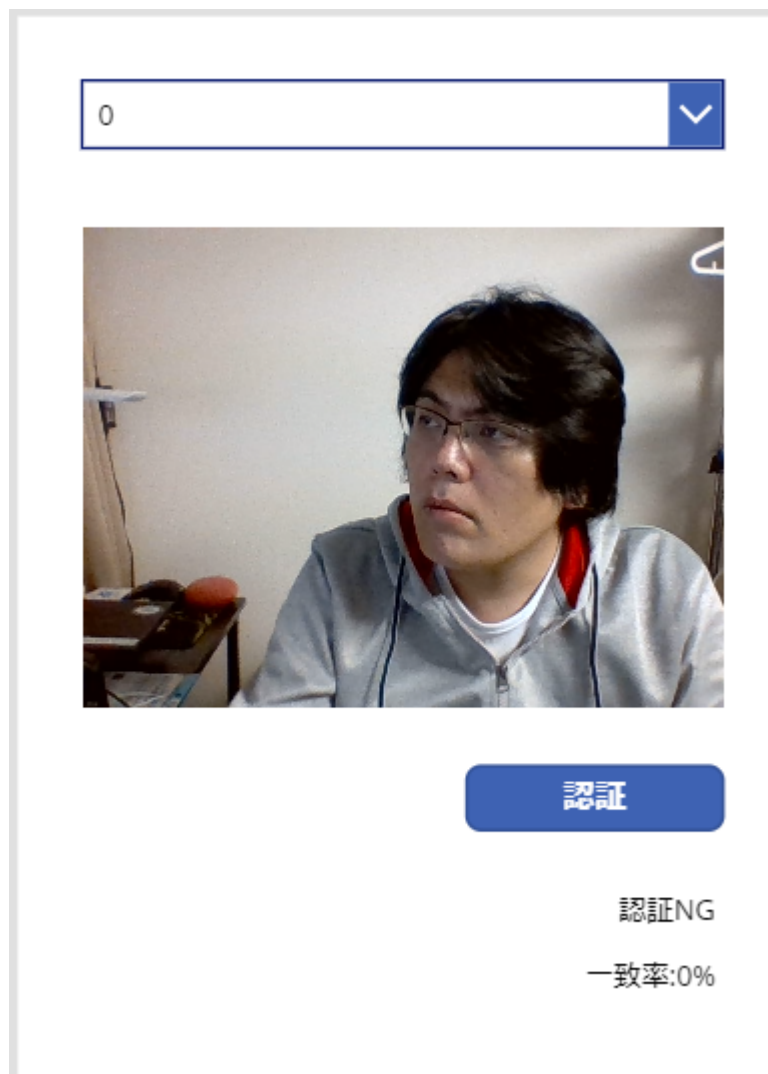
```

6. 認証結果を表示させるラベルとして AuthenticateResultLabel を作成する

プロパティ名	日本語名	値
Text	テキスト	If(FaceVerify.isIdentical,"認証OK","認証NG")
Align	テキストのアラインメント	Align.Right

7. 一致率を表示させるラベルとして MatchLateLabel を作成する

プロパティ名	日本語名	値
Text	テキスト	Concatenate("一致率:",Text(Round((FaceVerify.confidence*100),2)), "%")
Align	テキストのアラインメント	Align.Right



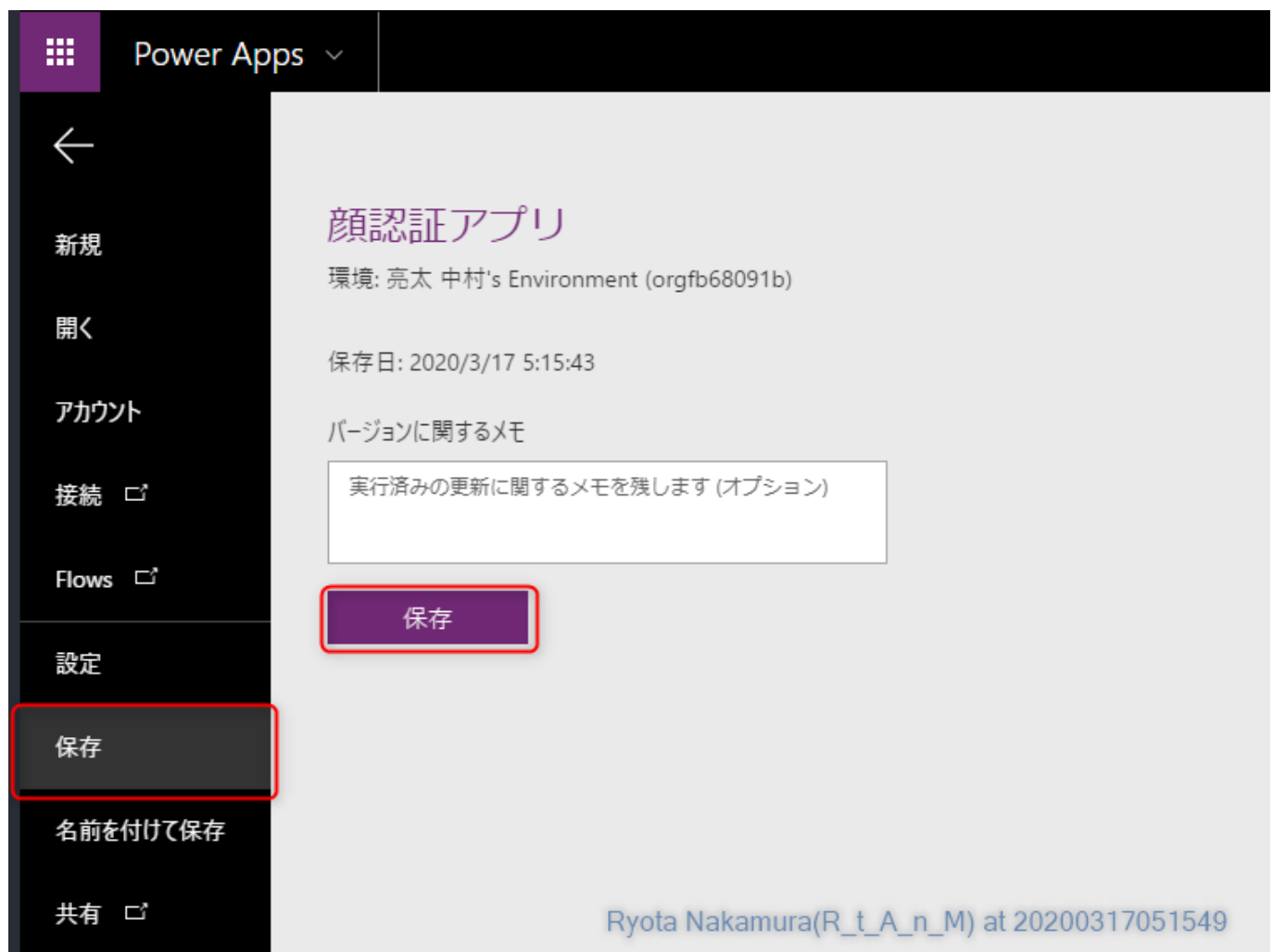
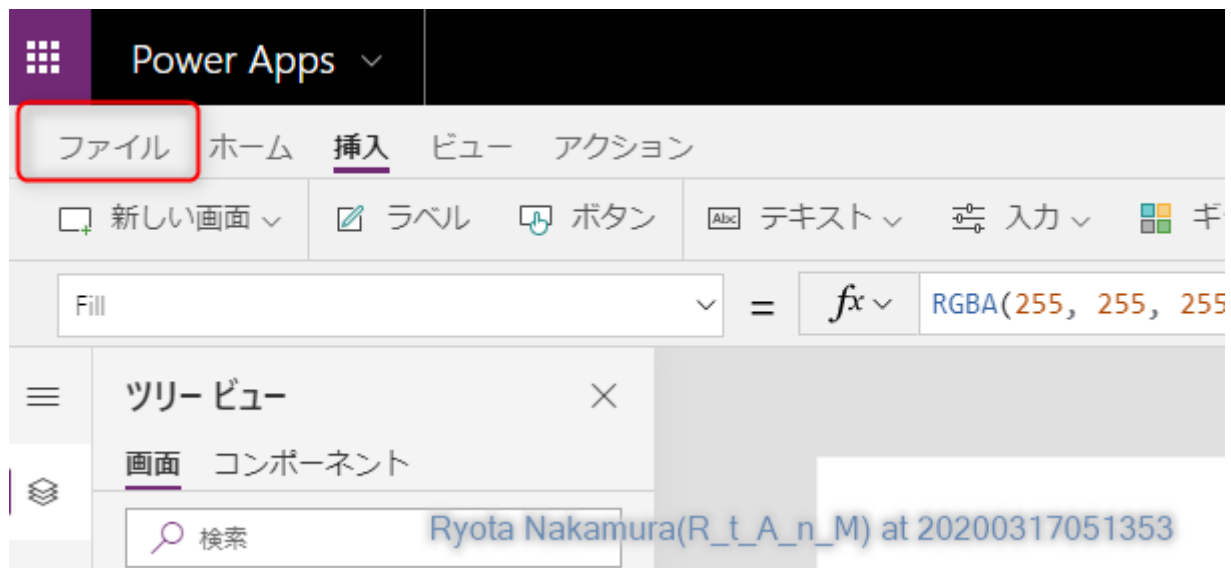
4.4. テスト

画面右上の再生ボタンをクリックすることで 開いている画面からテストを行うことが可能です。

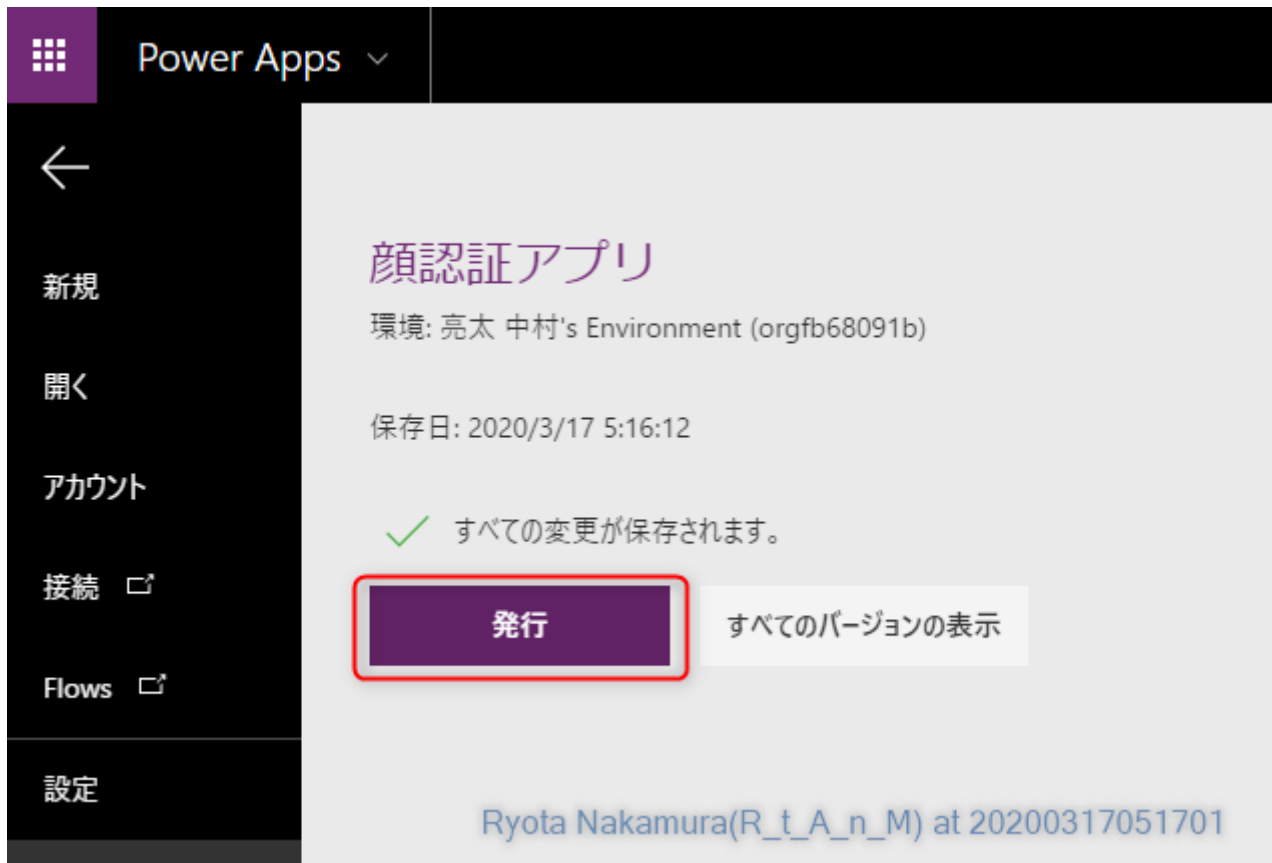


4.5. 保存

ファイルをクリックし、保存をクリックすることで、現在の状態を保存することができます。



また、その後発行をクリックすることで、保存した状態のアプリをスマートフォンで 사용할 ことが可能になります。



5. 試してみる

実際に作ったアプリを使って、顔認証ができるか試してみましょう！