

Programming Boot Camp

應用編

東京工業大学 2021/11/27

Ryo Imahashi, Naotake Kyogoku

目次

- ふりかえりと今回やることの確認
- 外部連携
- ノーコードツールでの共同作業のTips
- チーム開発演習
- まとめ

ふりかえりと今回やることの確認

- これまでの4回のレクチャーでは、AdaloとBubbleという2つのノーコードツールを使ったアプリの開発方法を学んできました。
- 今回はまず、AdaloとBubbleのそれぞれを外部のサービスと連携させる方法を学びます。
- 次に、共同作業に関するコツや注意点をご紹介します。
- 最後に、Development Phaseのチームでアプリ開発の演習と発表を行います。

外部連携

AdaloやBubbleだけで実現できないことがある場合、外部サービスと連携することでそれを実現できるかもしれません。

そのための方法をご紹介します。

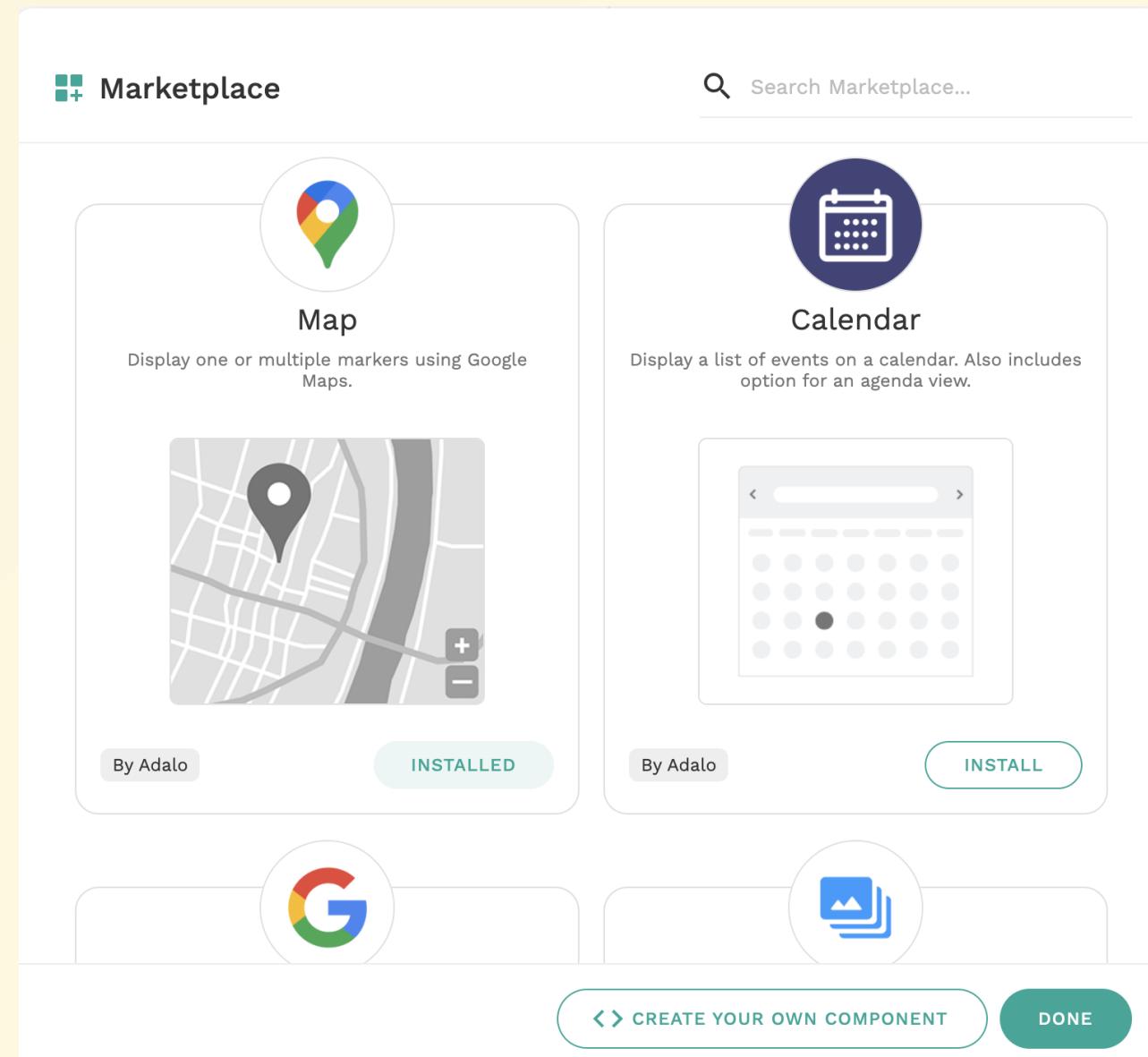
Adaloでの外部連携

以下3種類の外部連携の方法を紹介します。

- Marketplaceの外部連携コンポーネント
- Custom Action
- External Collection

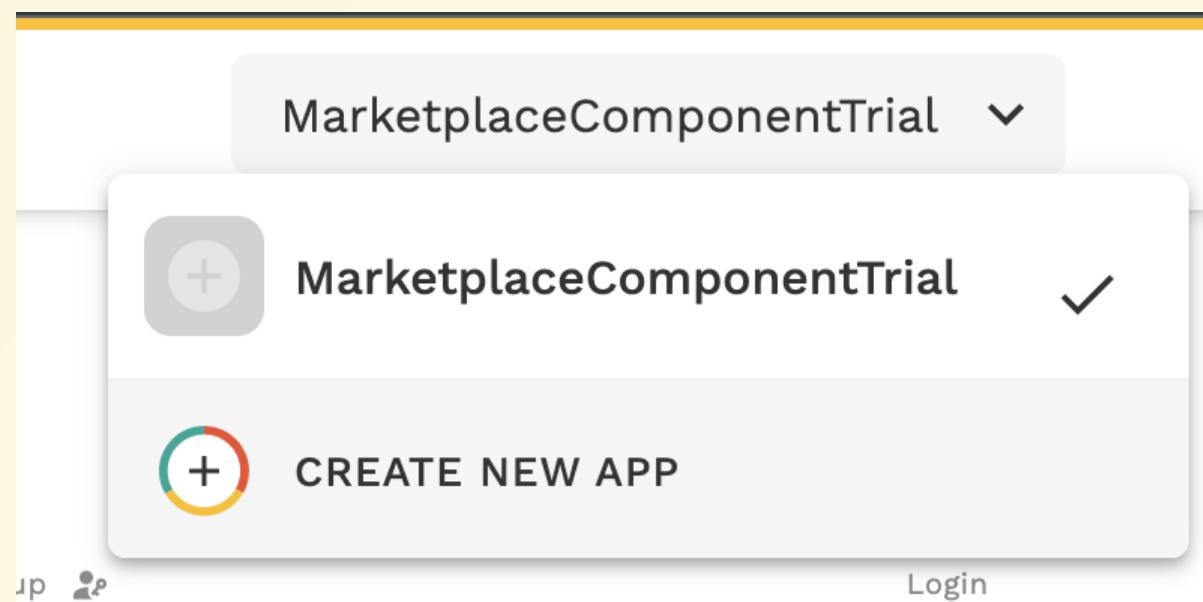
Marketplaceの外部連携コンポーネント

Marketplaceから外部連携を可能にするコンポーネントを追加できます。

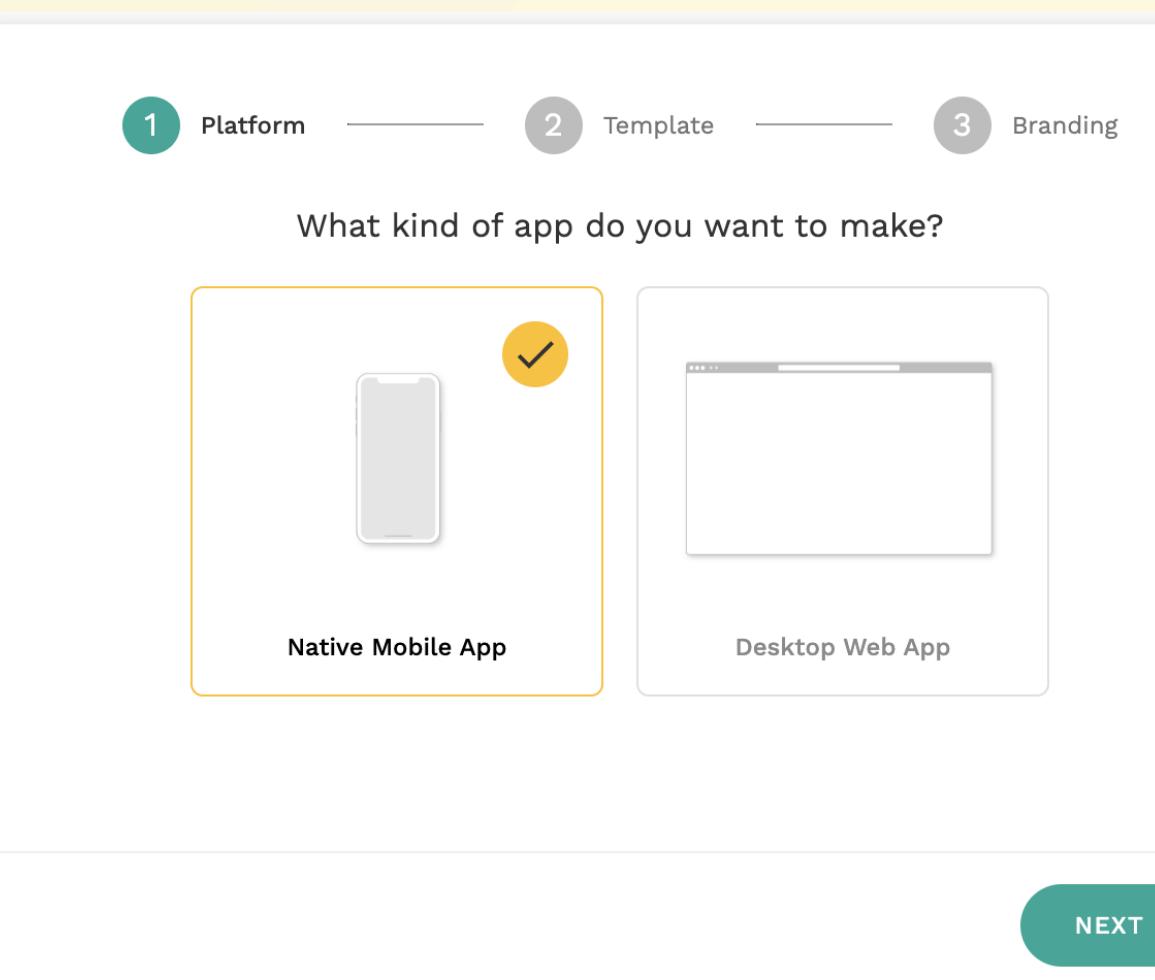


まずはアプリを新規作成しましょう。

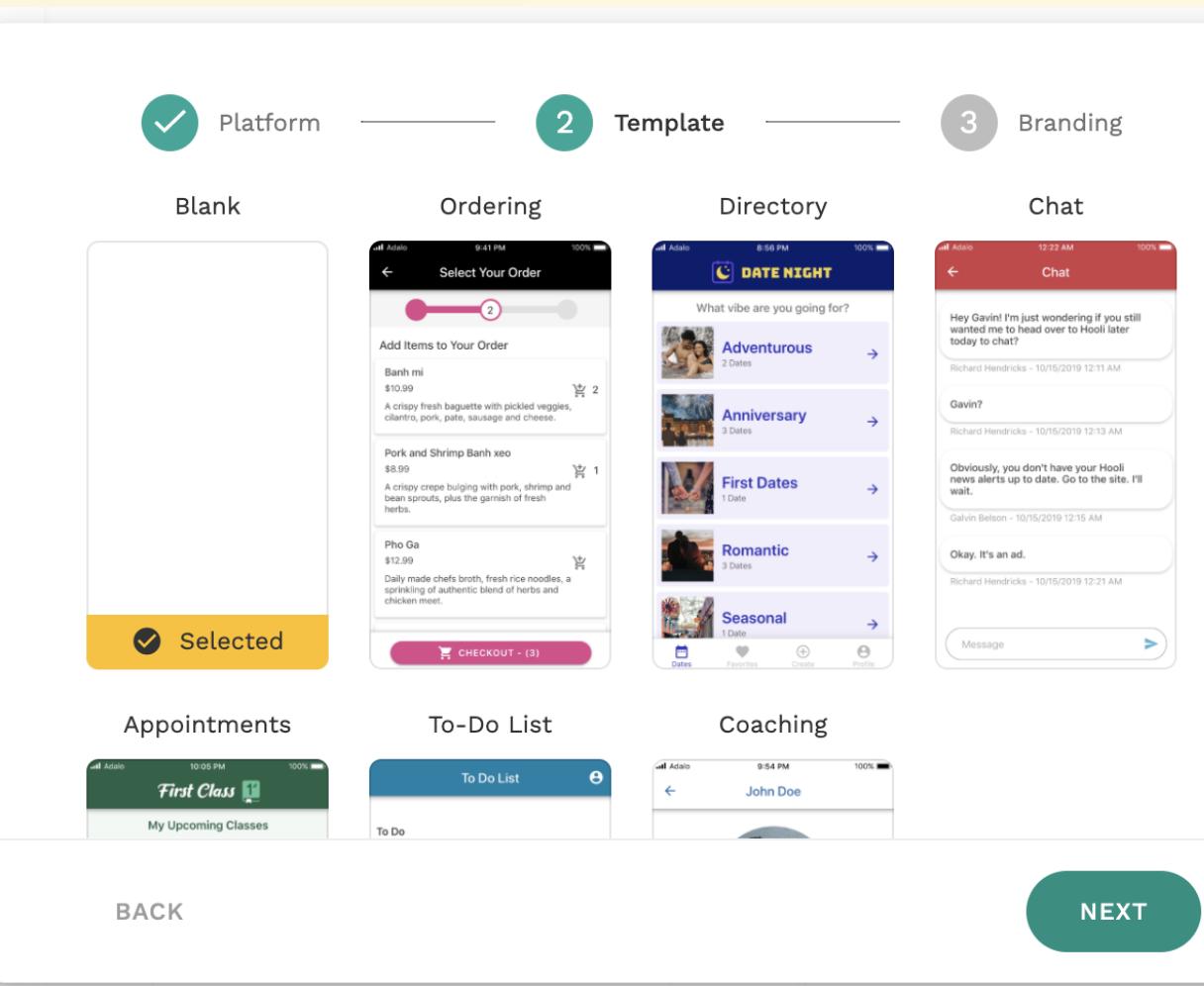
- Adaloにログイン
<https://app.adalo.com/login>
- CREATE NEW APPをクリック



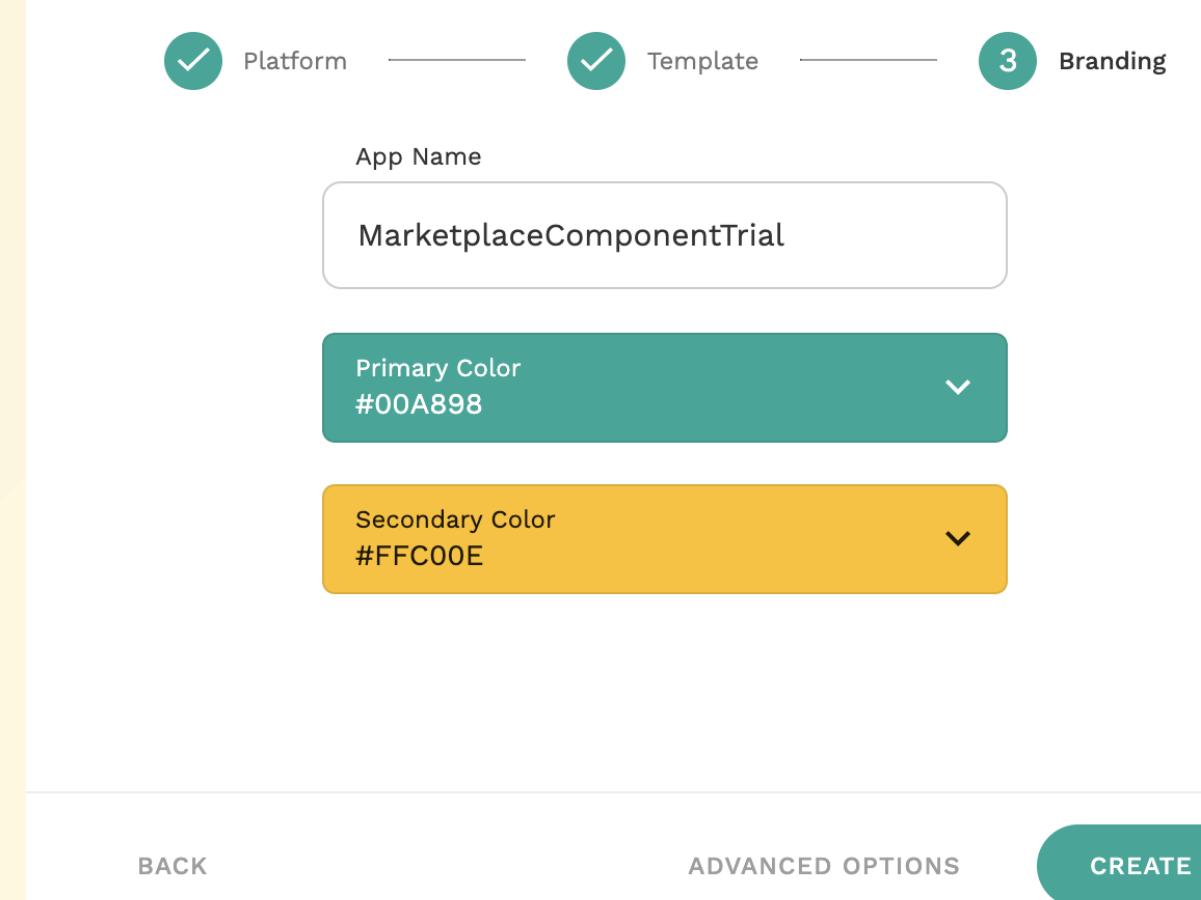
- PlatformはNative Mobile Appを選択



- TemplateはBlankを選択

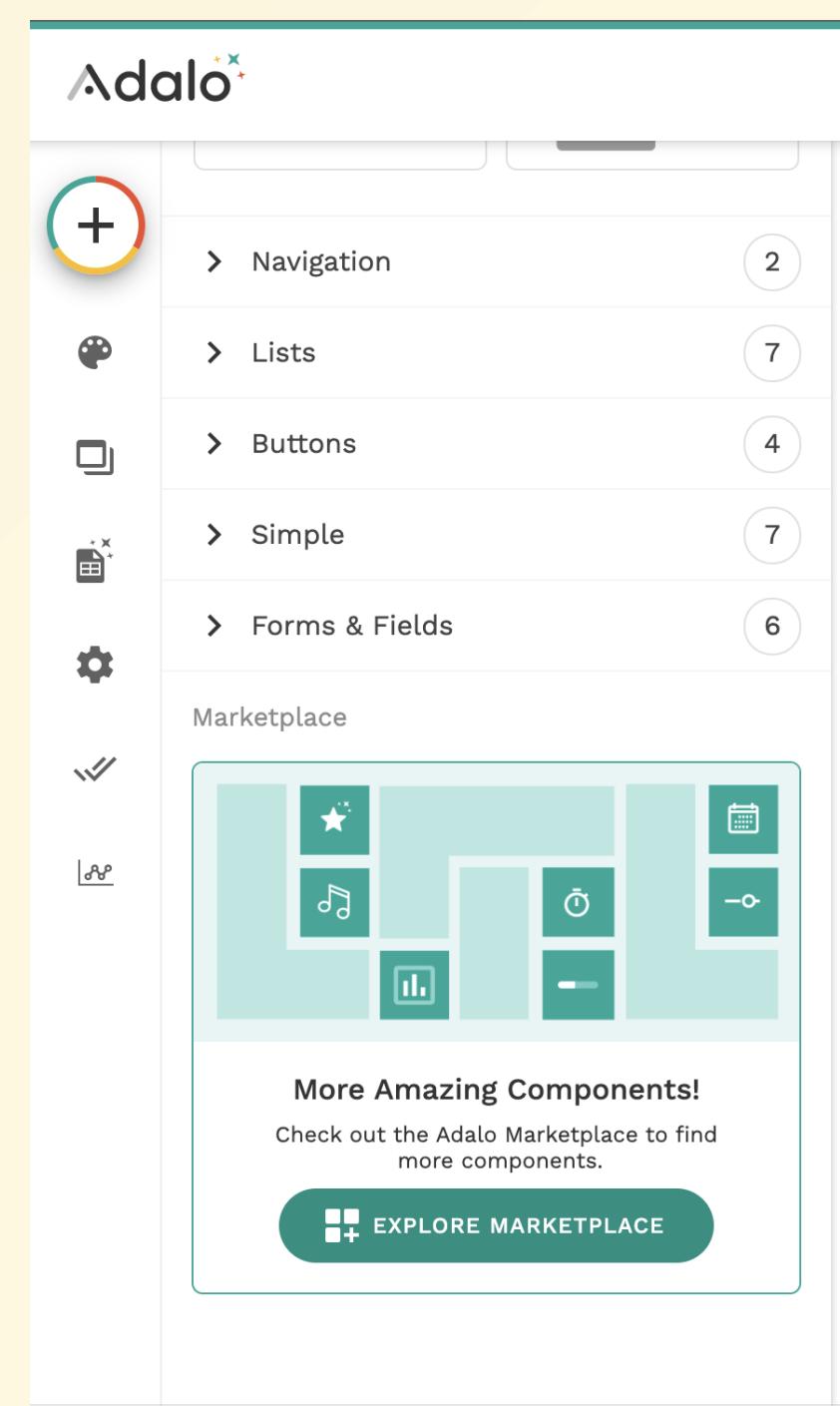


- App Nameには MarketplaceComponentTrial を入力
- Colorは自由に設定してください

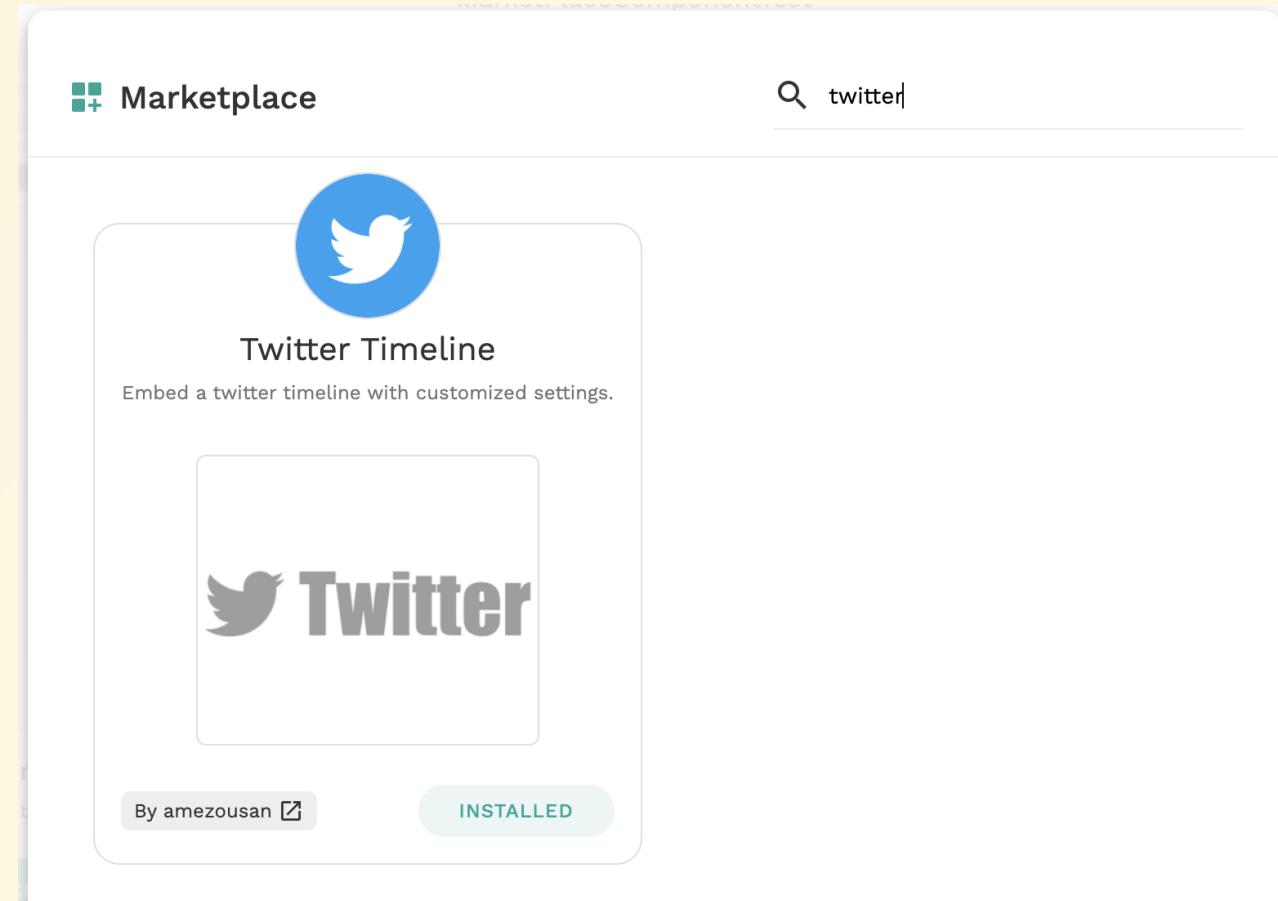


Twitter Timeline コンポーネント

- +ボタンを押してADD COMPONENTの中の Explore Marketplaceをクリック

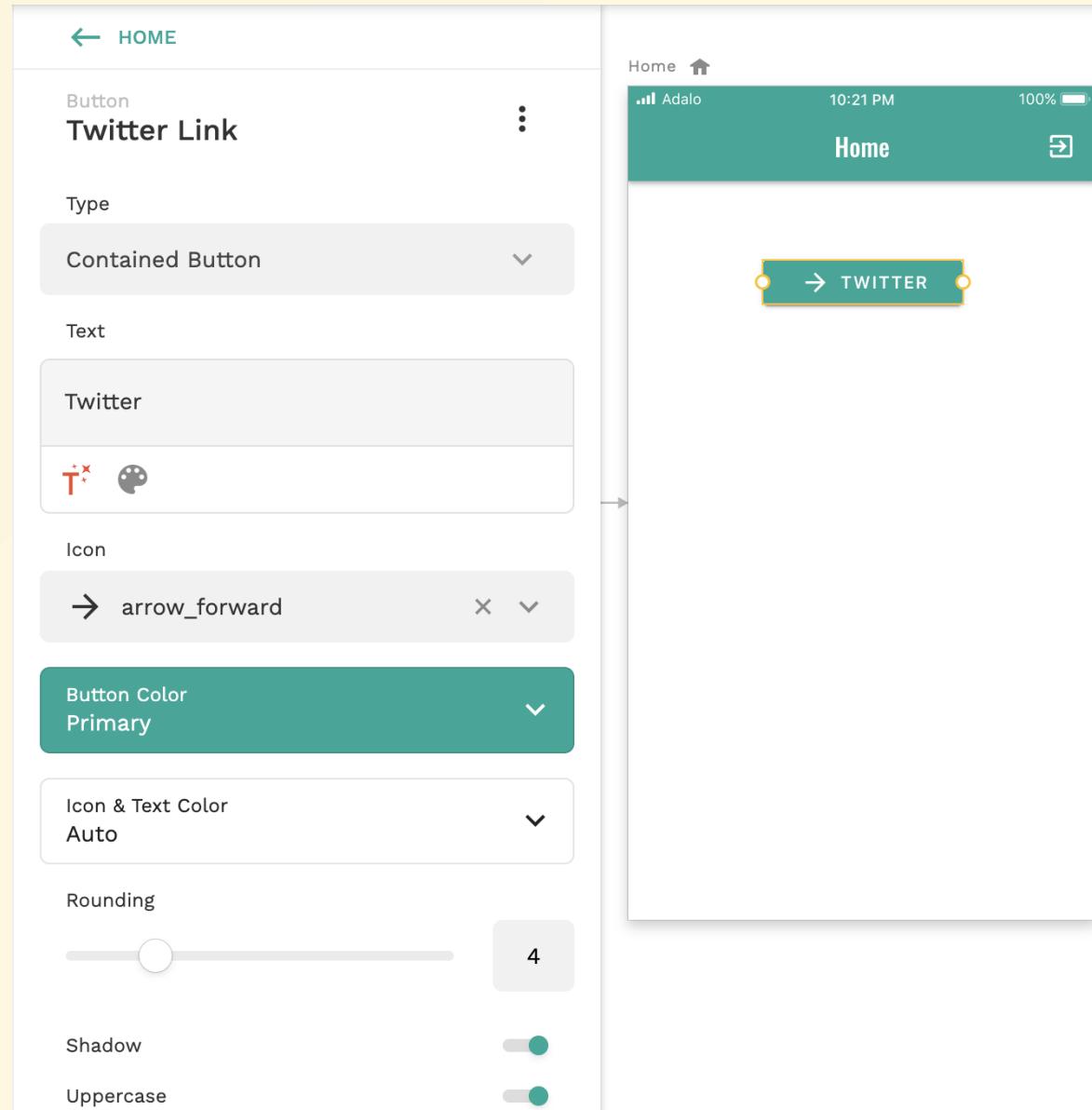


- Twitter TimelineコンポーネントをINSTALL

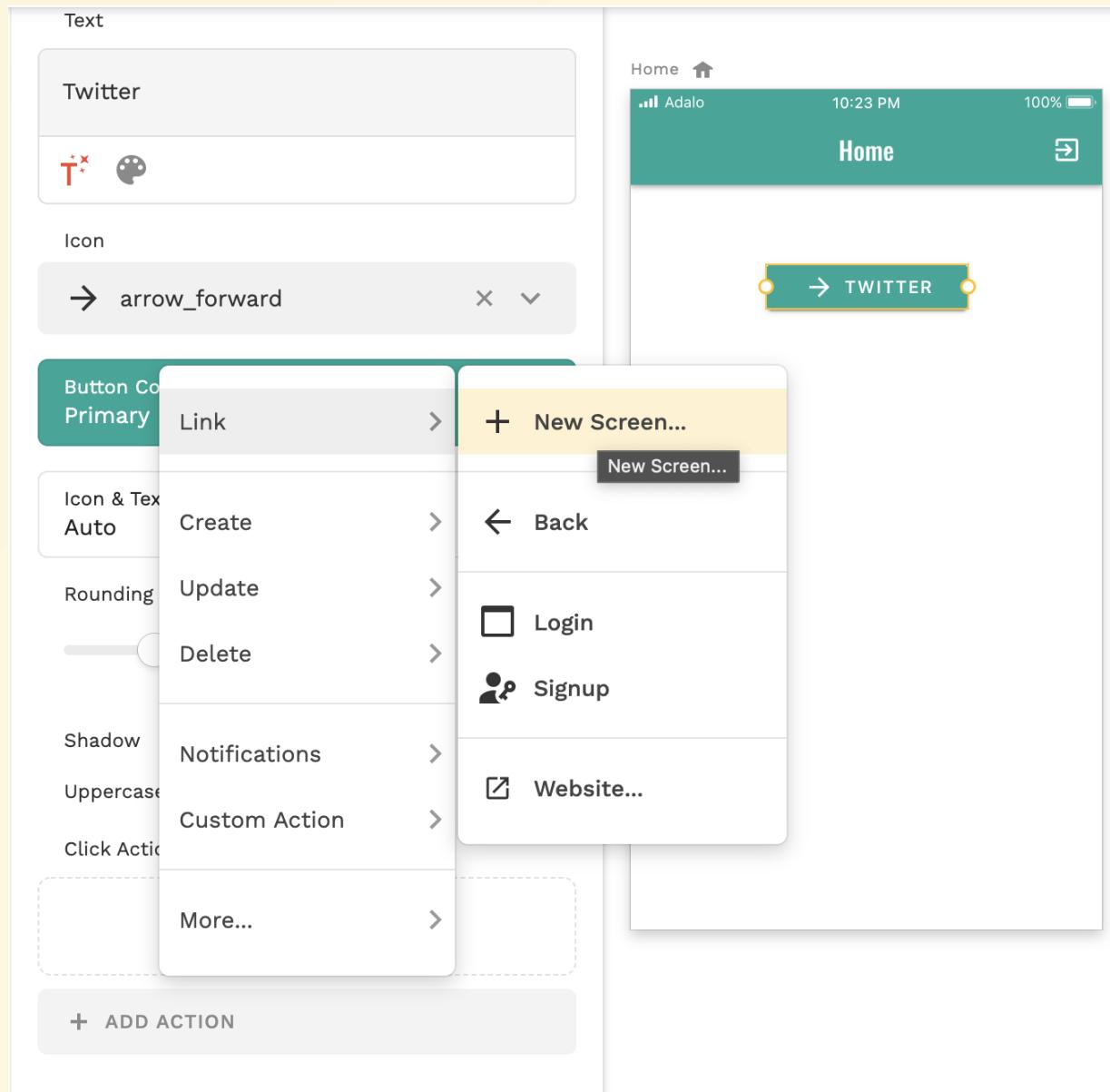


Twitter Timelineコンポーネントを配置します。

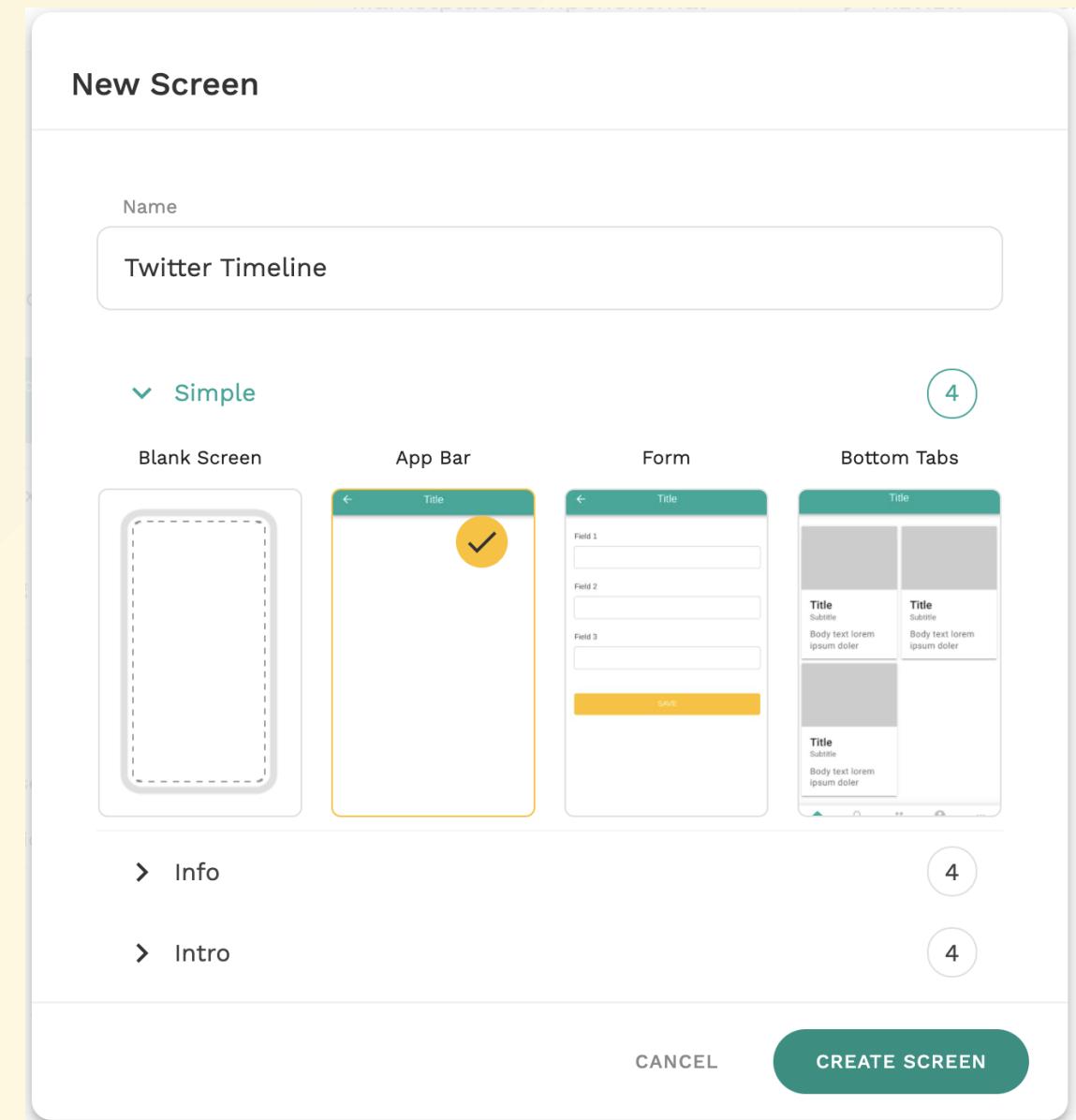
- Home ScreenにTwitter用画面へのLinkボタンを追加



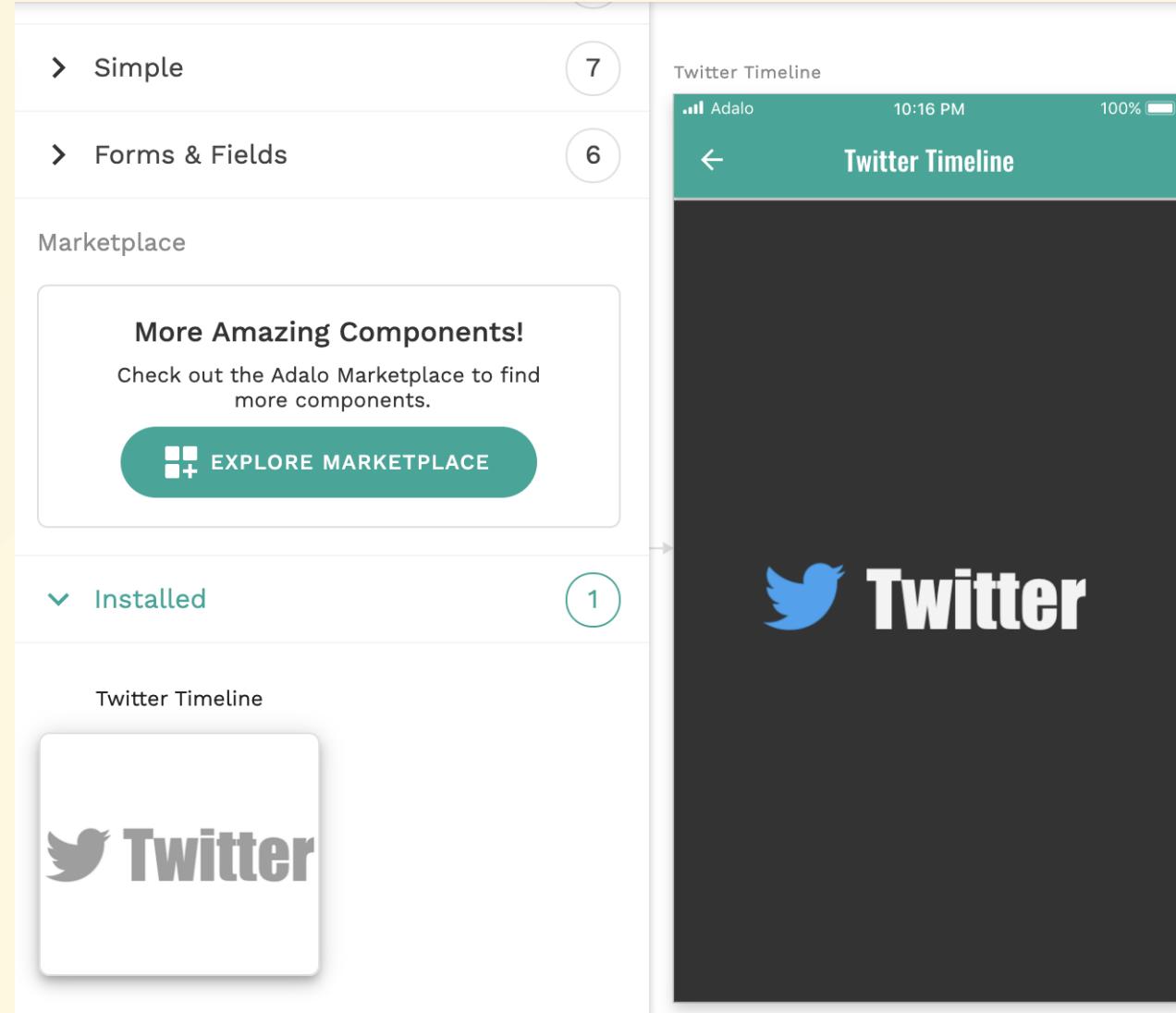
- ADD ACTIONからNew ScreenへのLinkを追加



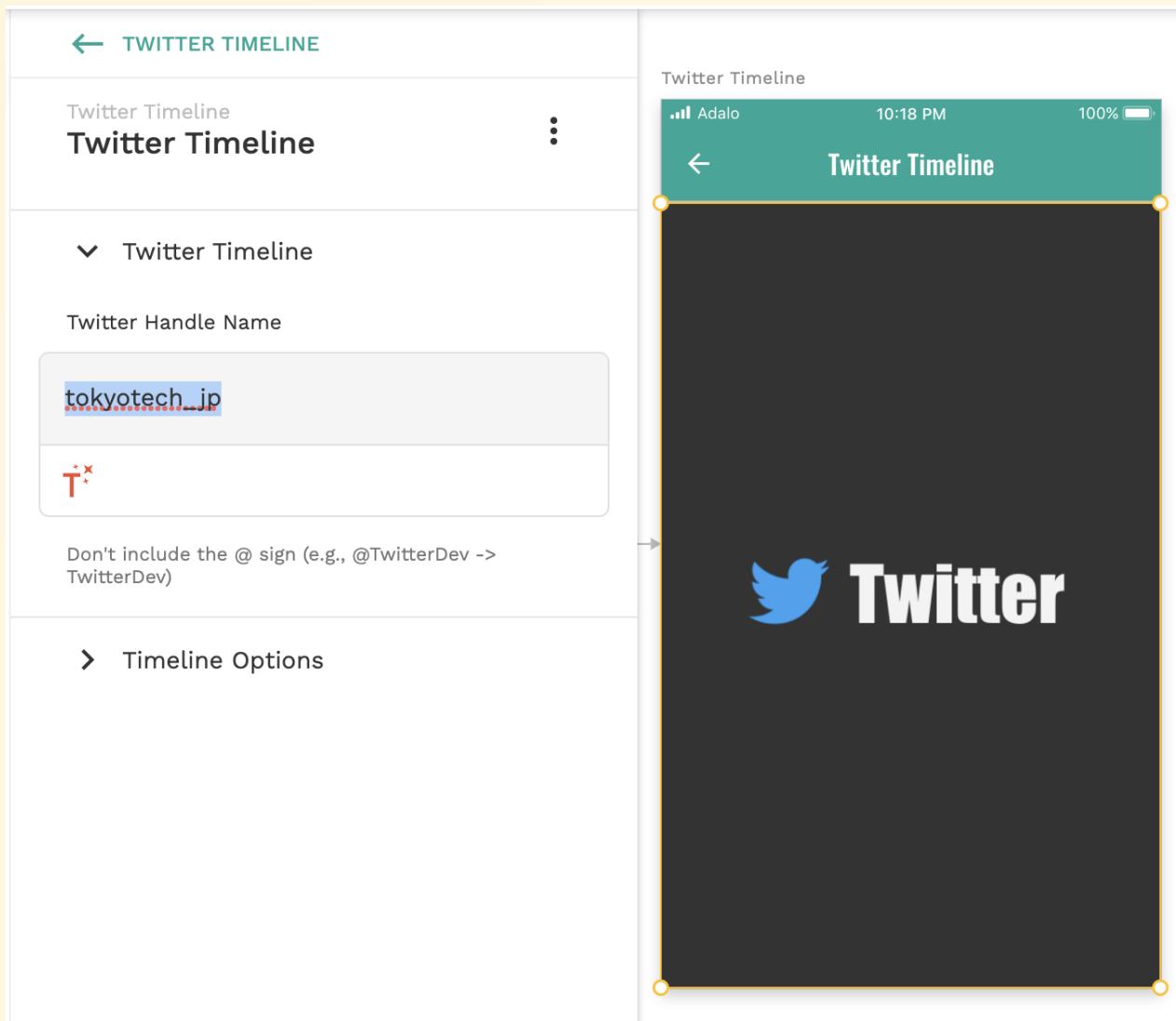
- Templateで"App Bar"を選択し、"Twitter Timeline" 画面を作成



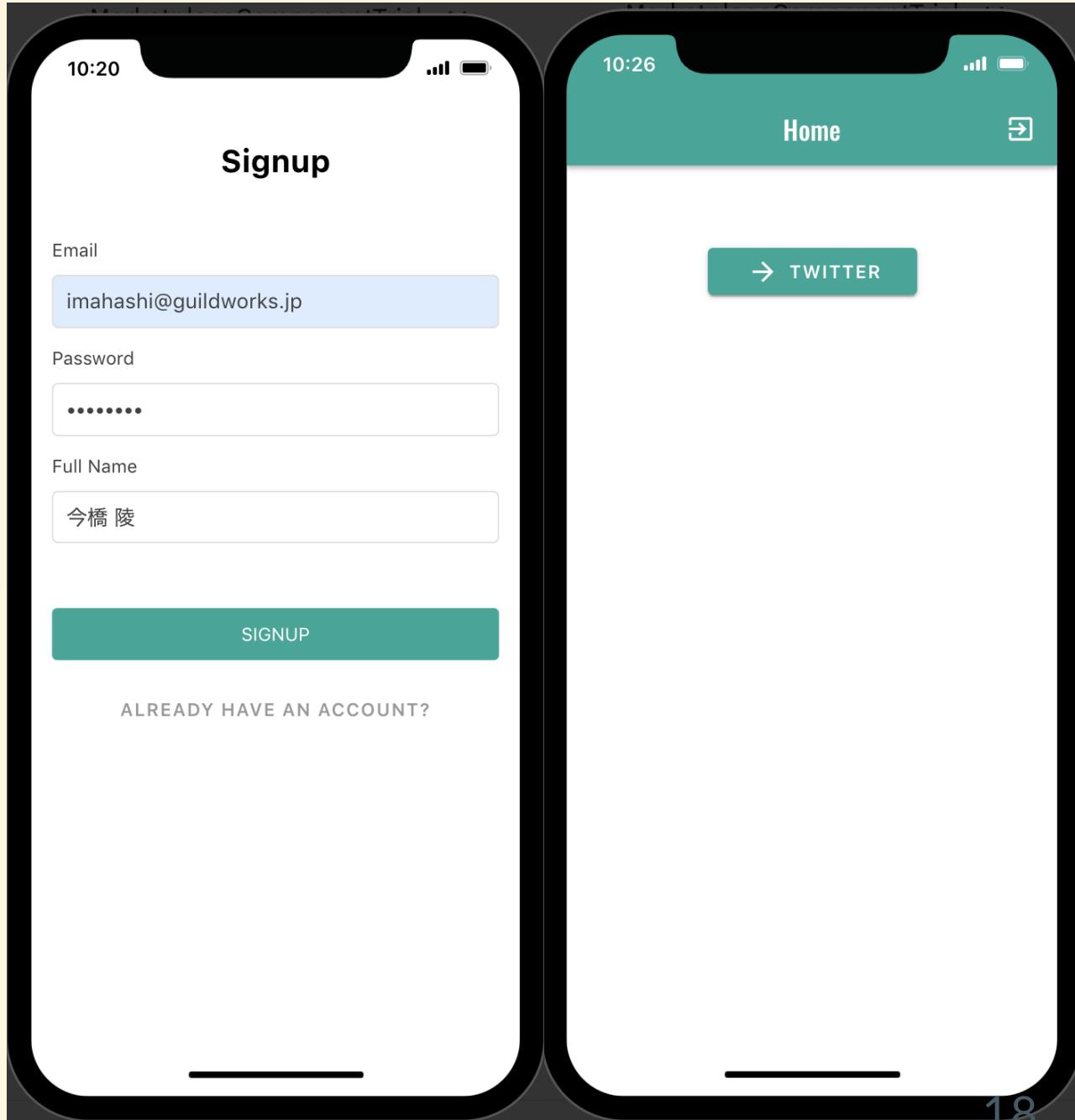
- Twitter Timelineコンポーネントを配置



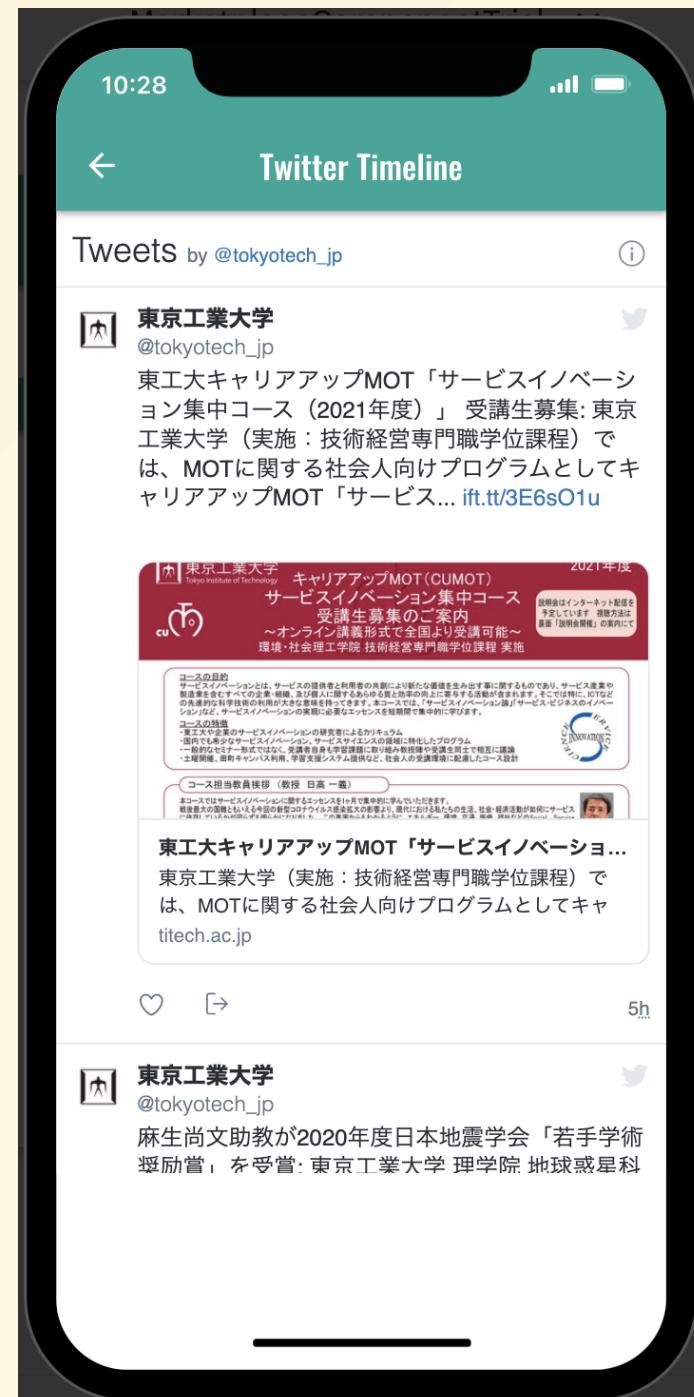
- Twitter Handle Nameに "tokyotech_jp" と入力
 - 好きなTwitterアカウントのHandle Nameに変更してもOKです



- Preview機能でSignupして、Twitterボタンをクリック



- 入力したHandle NameのTwitterアカウントの投稿が一覧表示されます



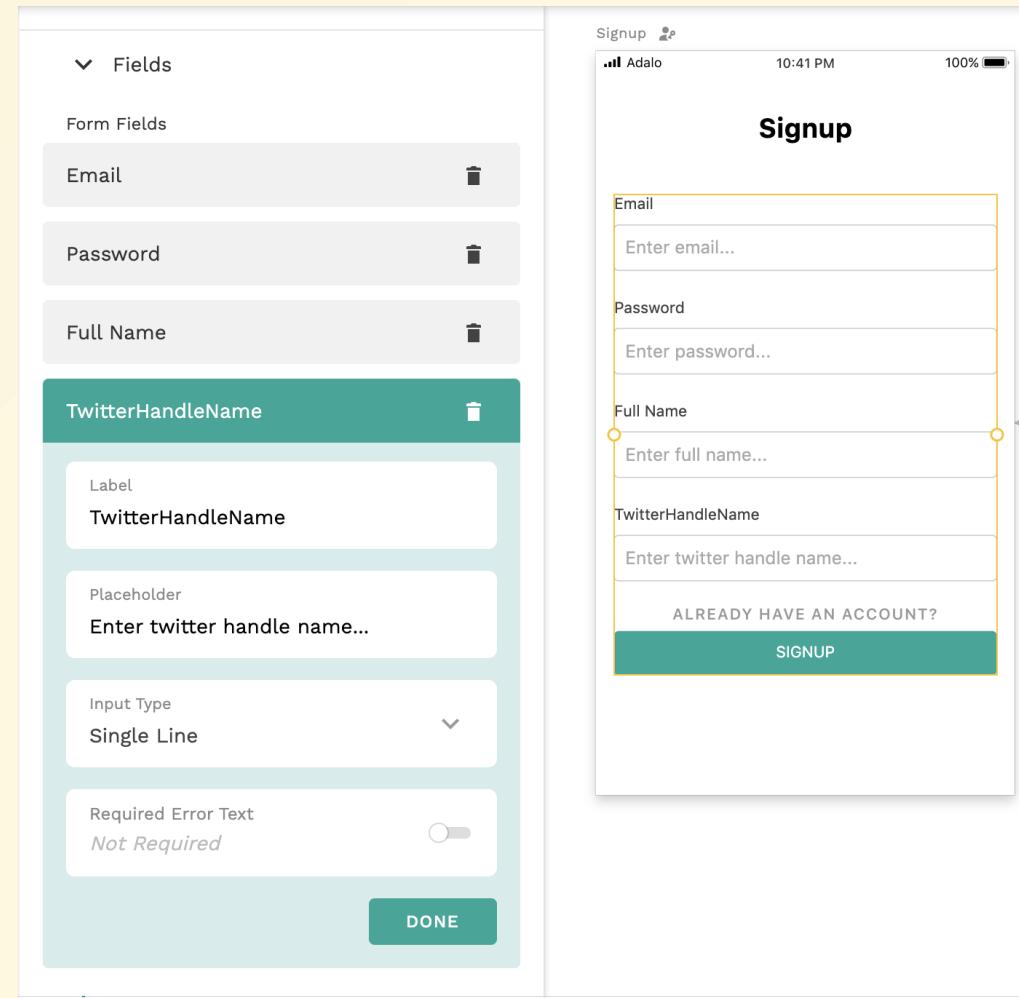
~~ログインしたユーザー自身の
Twitterアカウントの投稿が
一覧表示されるように修正しま
しょう。~~

スライドの枚数的に時間が足
りなくなりそうなので、割愛
します...

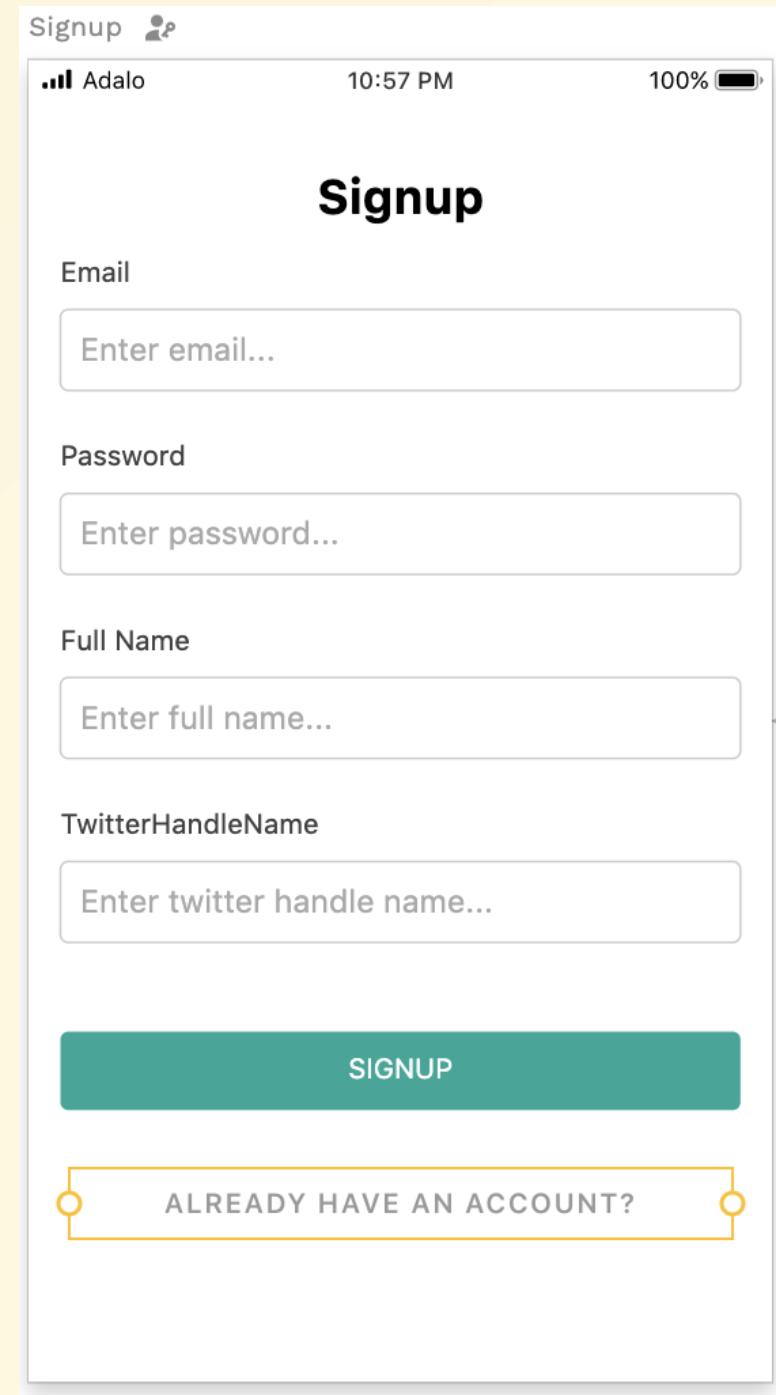
- Users Collectionに
TwitterHandleName
Propertyを追加
 - TypeはTextを選択

The screenshot shows the MongoDB Compass interface. On the left is a sidebar with icons for creating a new database, collection, or index, and for deleting or updating documents. The main area displays the 'Database Collections' section, with 'Users' selected. A red callout highlights the 'New Property' dialog box. This dialog has fields for 'Type' (set to 'Text') and 'Name' (set to 'TwitterHandleName'). At the bottom of the dialog are 'CANCEL' and 'SAVE' buttons, and a 'ADD PROPERTY' button at the bottom right.

- Signup画面のFormをクリック
- Fields > ADD VISIBLE FIELD > TwitterHandleName
- Not Requiredに変更

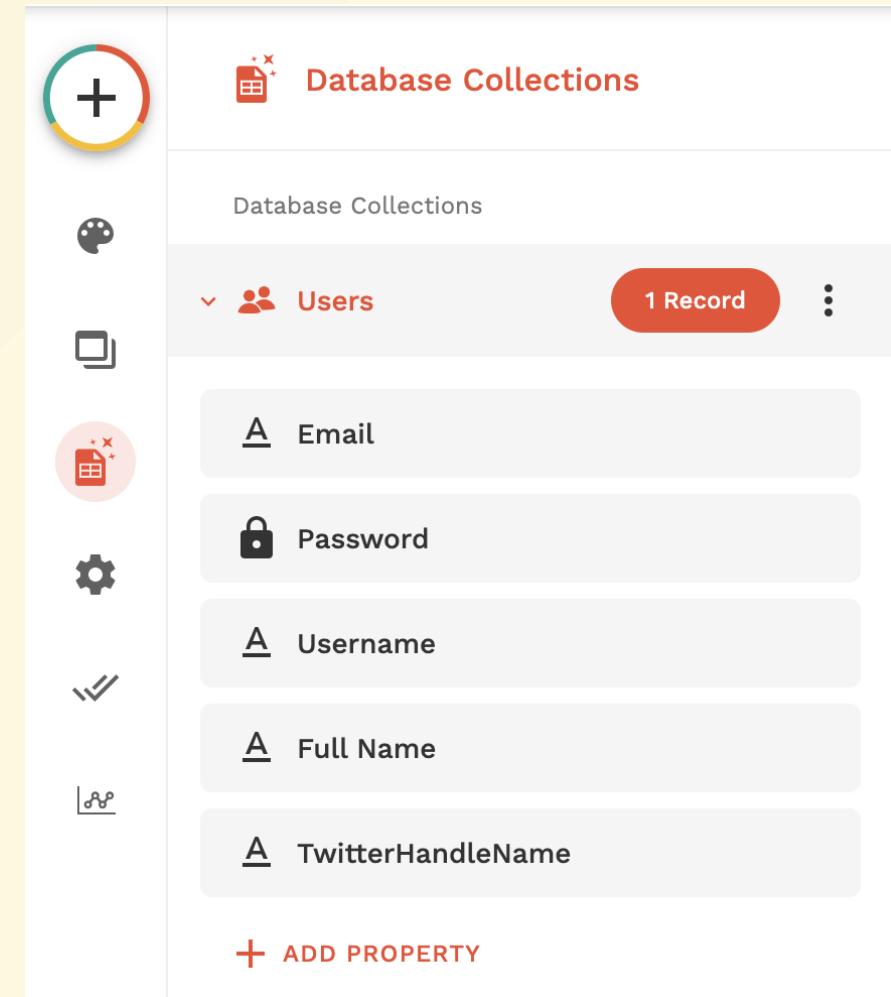


- "ALREADY HAVE AN ACCOUNT?"というリンクがフォームに重なってしまっていたので、下に移動



登録済のユーザーに
TwitterHandleNameを設定し
ておきましょう

- Users Collection の "1 Record" ボタンをクリック



- 登録済のユーザーのレコードをクリック
- TwitterHandleNameを入力
(自分のアカウントでも、好きなアカウントでもOK)

Jser

A Email imahashi@guildworks.jp

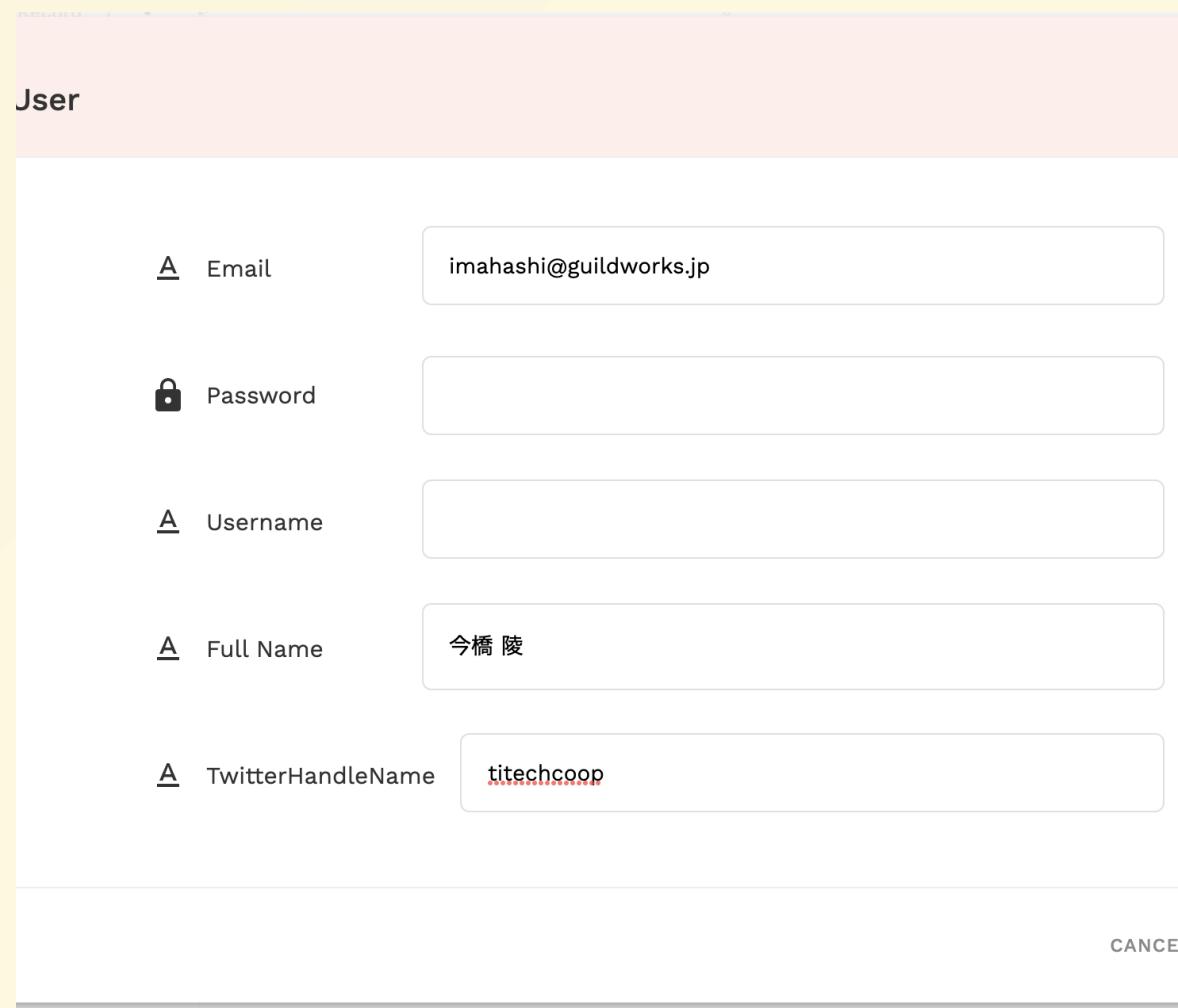
A Password

A Username

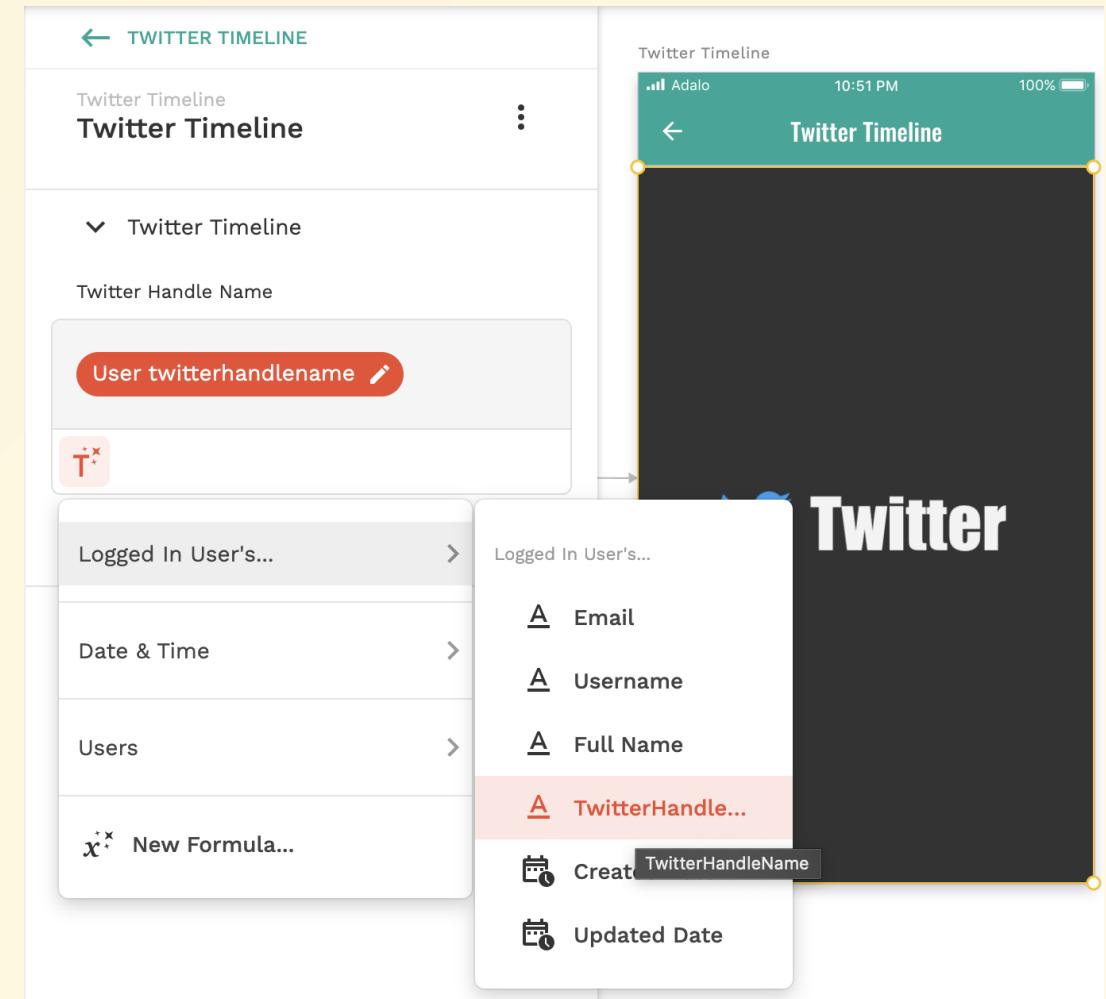
A Full Name 今橋 陵

A TwitterHandleName titechcoop

CANCEL S



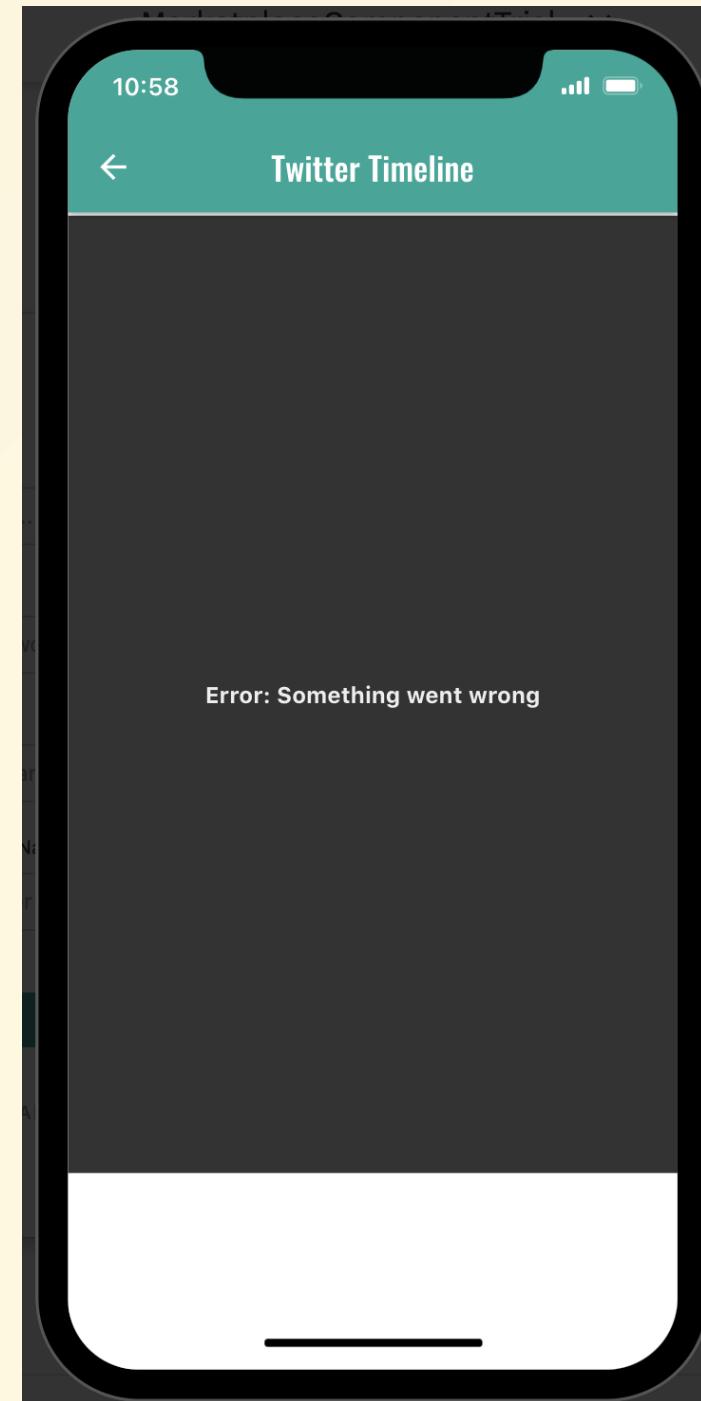
- Twitter TimelineコンポーネントのTwitter Handle Nameを"Logged In User's TwitterHandleName"に変更



理論上はこれでログインユーザー自身が設定したTwitterアカウントの投稿一覧が表示されるはずです。

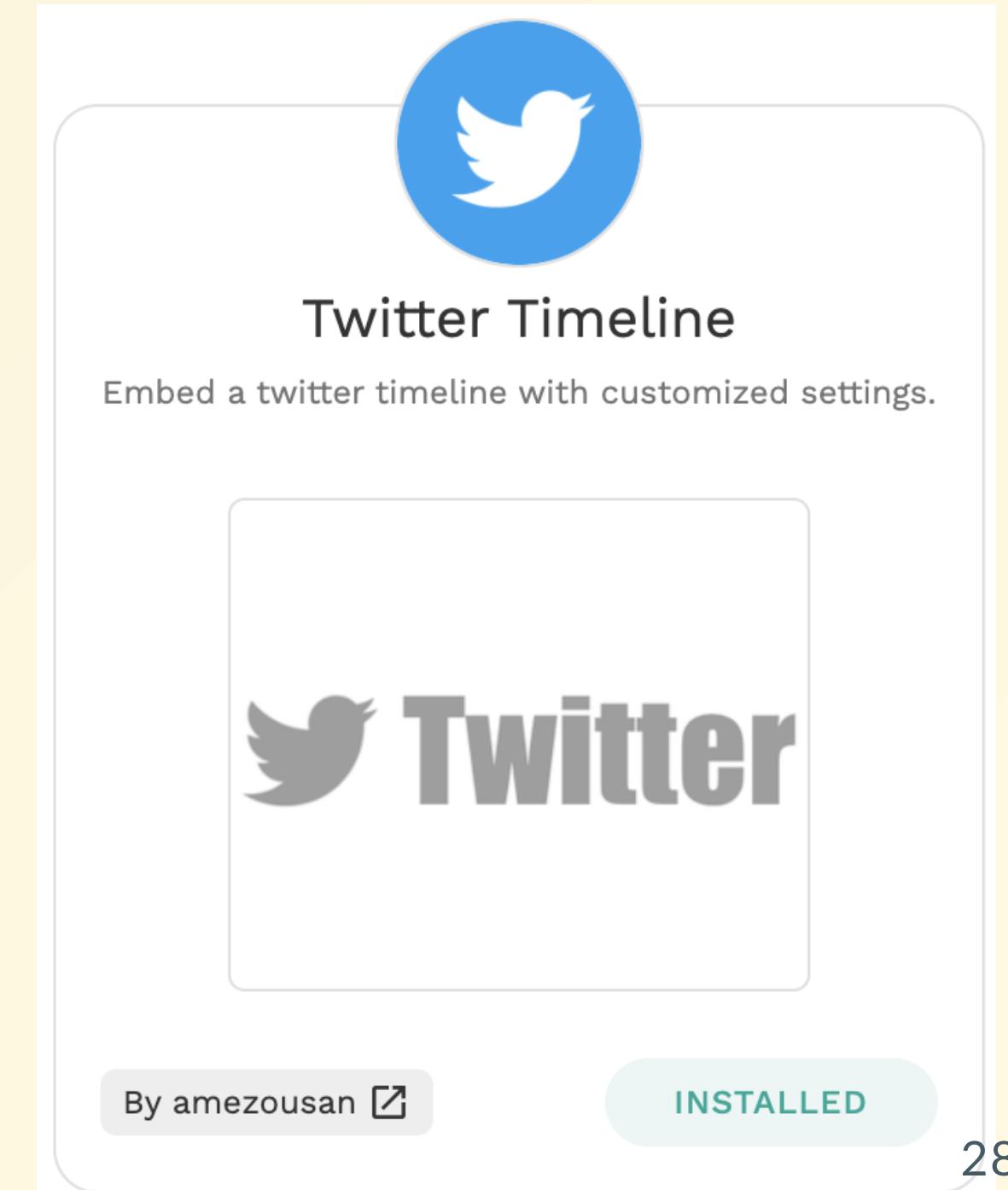
Preview機能で確認してみましょう。

エラーが起きると思います。



Twitter Timelineは有志の方が無償公開してくれているコンポーネントです。

誰かが作ってくれているものが、いつも自分の思い通りに動くとは限りません。



「できないなら紹介しないでよ」と思った方もいるかもしれませんね。ごめんなさいmm

試行錯誤をして「エラーを回避する方法を見つけた！」と思ったので資料に書いたのですが、後からそれが勘違いだったことに気づきました。。(ブラウザにはキャッシュという仕組みがあって、時々幻のようなものを見せてきます)

「そんなこともあるんだな」くらいは覚えておいてもらえると嬉しいです。

参考: Twitter Timelineコンポーネントのコードは公開されているので、修正して使うこともできそうです。(今回はやりません)

<https://github.com/amezousan/adalo-twitter-timeline/>

The screenshot shows the GitHub repository page for 'amezousan / adalo-twitter-timeline'. The repository is public and has 13 commits. The code tab is selected, showing files like README.md, index.js, and package.json. The repository has 1 star, 1 fork, and 1 issue. The description states it's an Adalo plugin for embedding a Twitter timeline. The repository is part of the 'adalo-plugin' organization.

amezousan / adalo-twitter-timeline Public

Code Issues 1 Pull requests Actions Projects Wiki Security Insights

master 1 branch 0 tags Go to file Add file Code

amezousan Fix #11 Add new properties and apply prettier (#12) 8d1f126 on Nov 8, 2020 13 commits

src/components/TwitterTimeline Fix #11 Add new properties and apply prettier (#12) 13 months ago

.gitignore Fix/adalo review (#6) 15 months ago

README.md Update README 13 months ago

index.js Fix #11 Add new properties and apply prettier (#12) 13 months ago

logo.png Initial commit. Working on Adalo Web. 15 months ago

package.json Fix #11 Add new properties and apply prettier (#12) 13 months ago

yarn.lock Fix #11 Add new properties and apply prettier (#12) 13 months ago

README.md

adalo-twitter-timeline

This is an Adalo plugin which allows you to embed a Twitter timeline with customized settings.

Getting Started

Watch 1 Star 0 Fork 1

About

This is an Adalo plugin that allows you to embed a Twitter timeline with customized settings.

adalo-plugin

Readme

Releases

No releases published

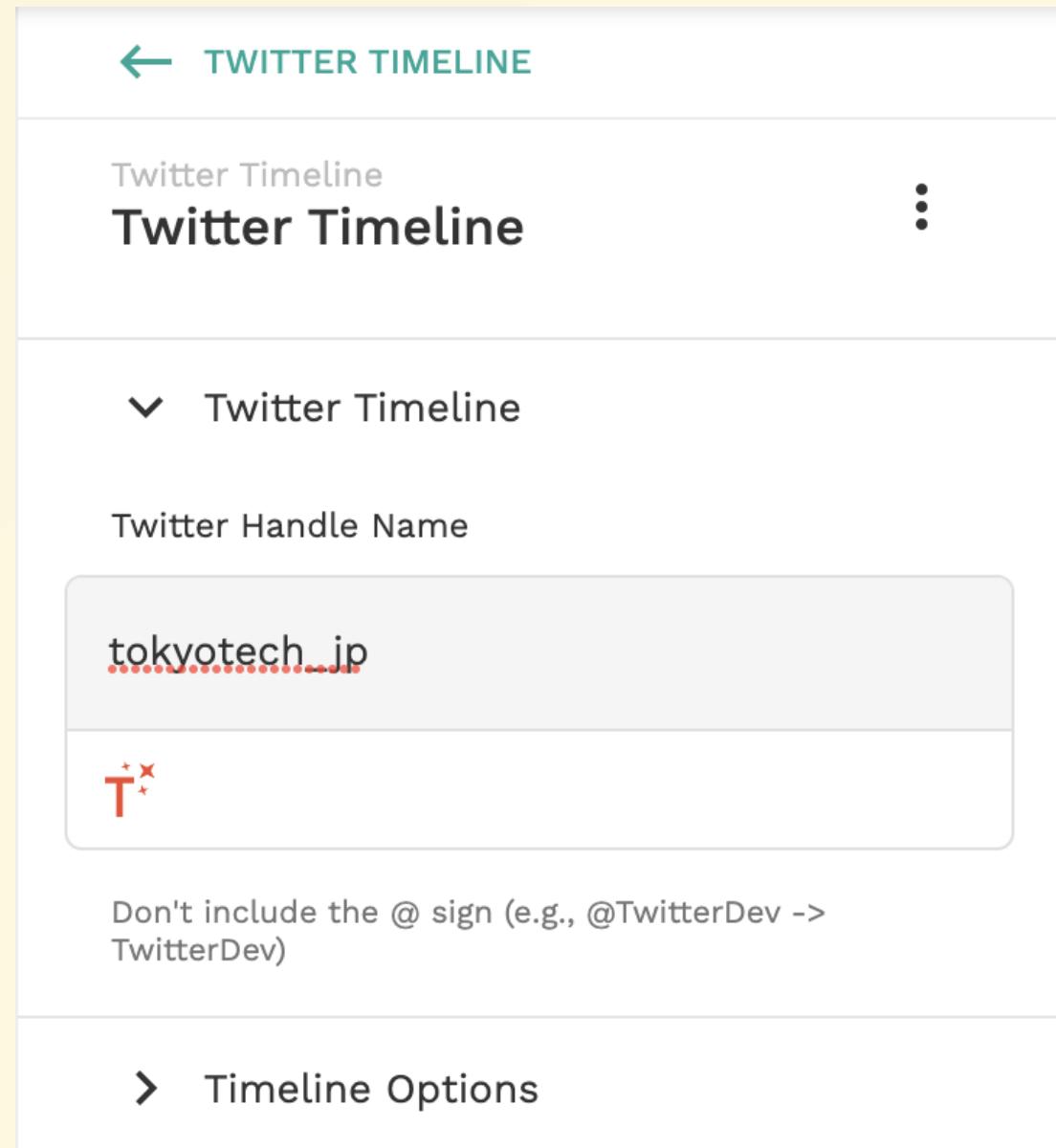
Packages

No packages published

Languages

JavaScript 100.0%

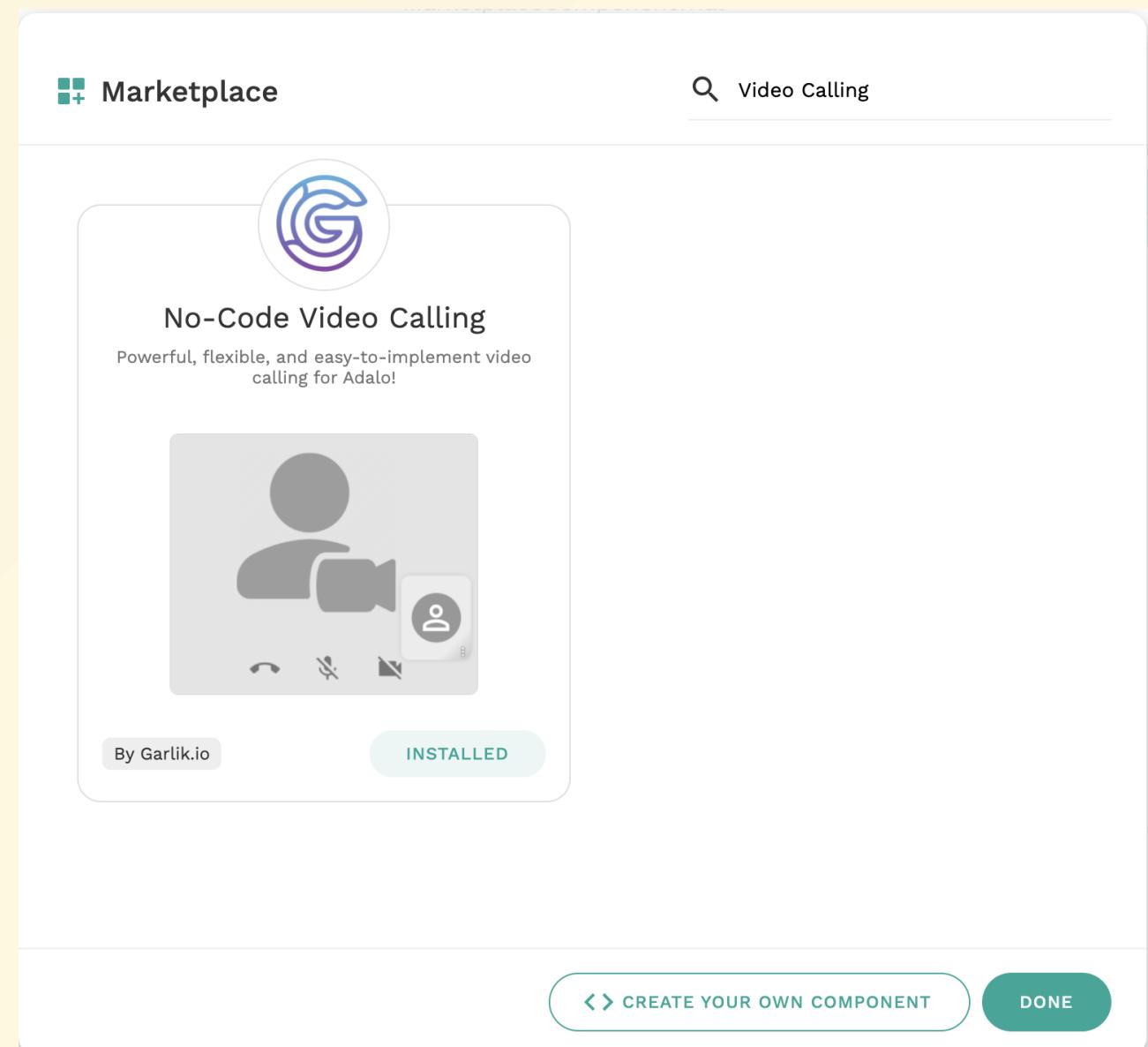
- Twitter TimelineコンポーネントのTwitter Handle Nameは直接値を入力した状態に戻しておきましょう



Video Calling

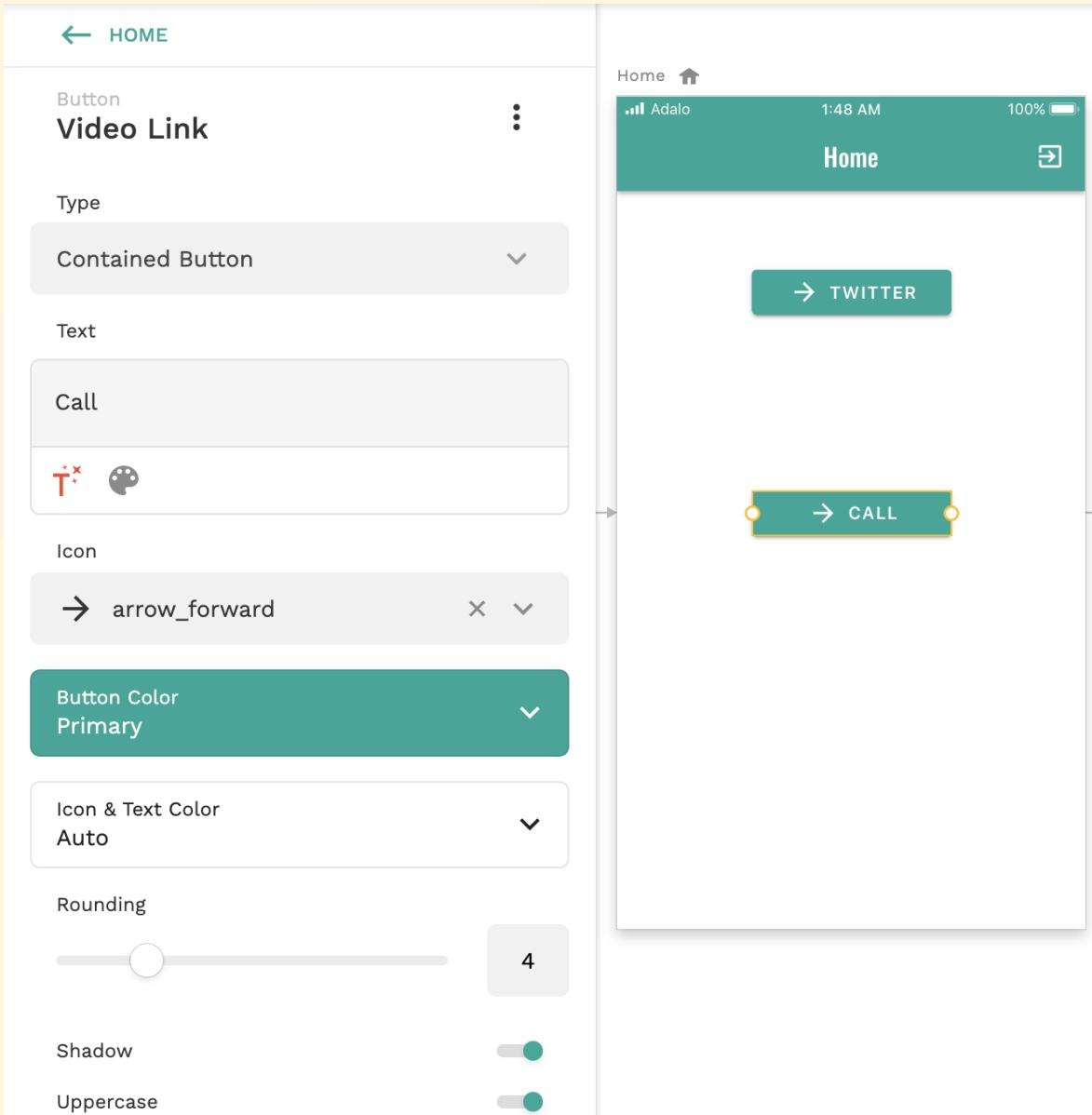
次に、ビデオ通話コンポーネントを紹介します。

- Marketplaceで "No-Code Video Calling"コンポーネントをINSTALL

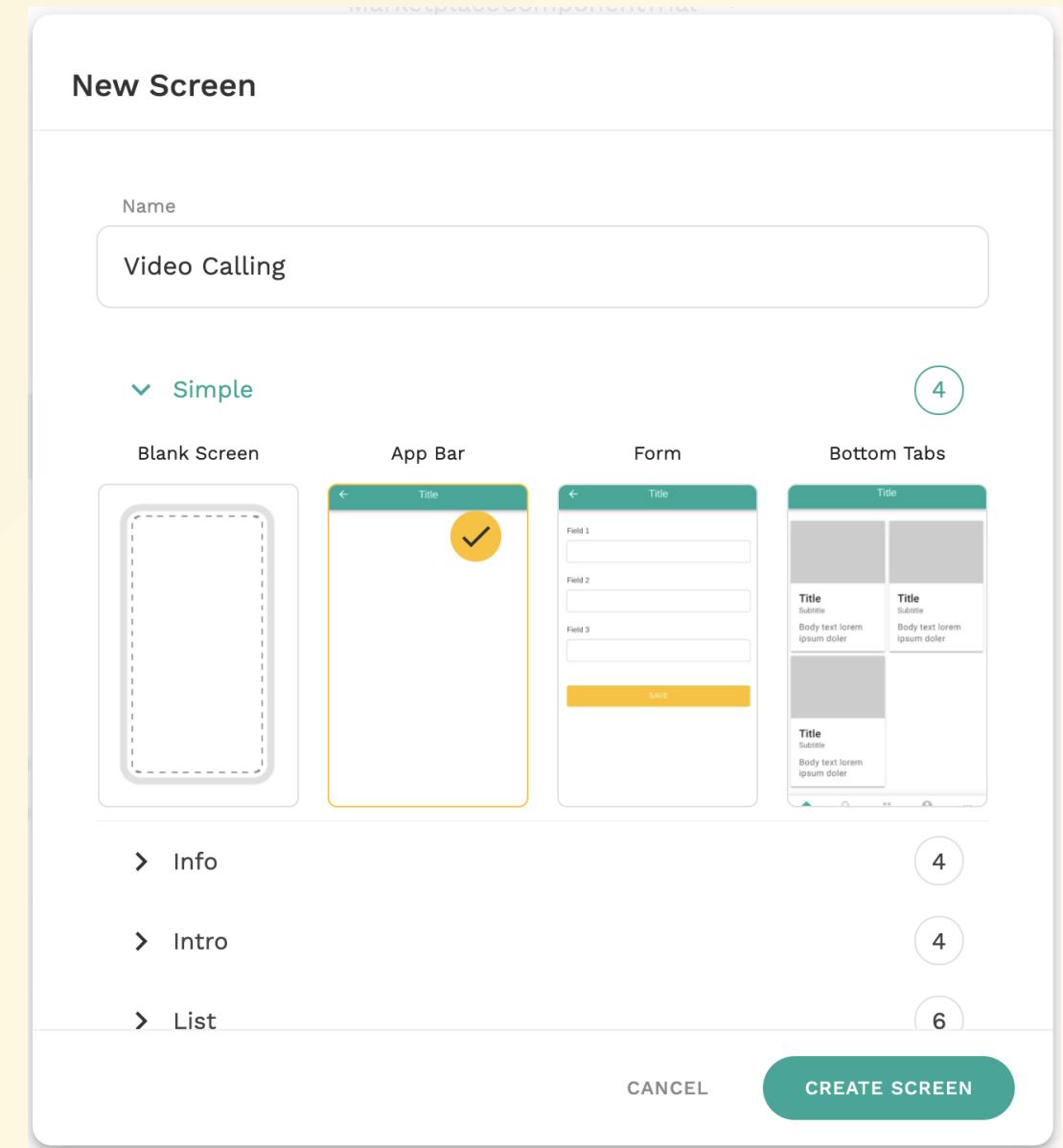


ビデオ通話コンポーネントを配置するための新しい画面を作りましょう。

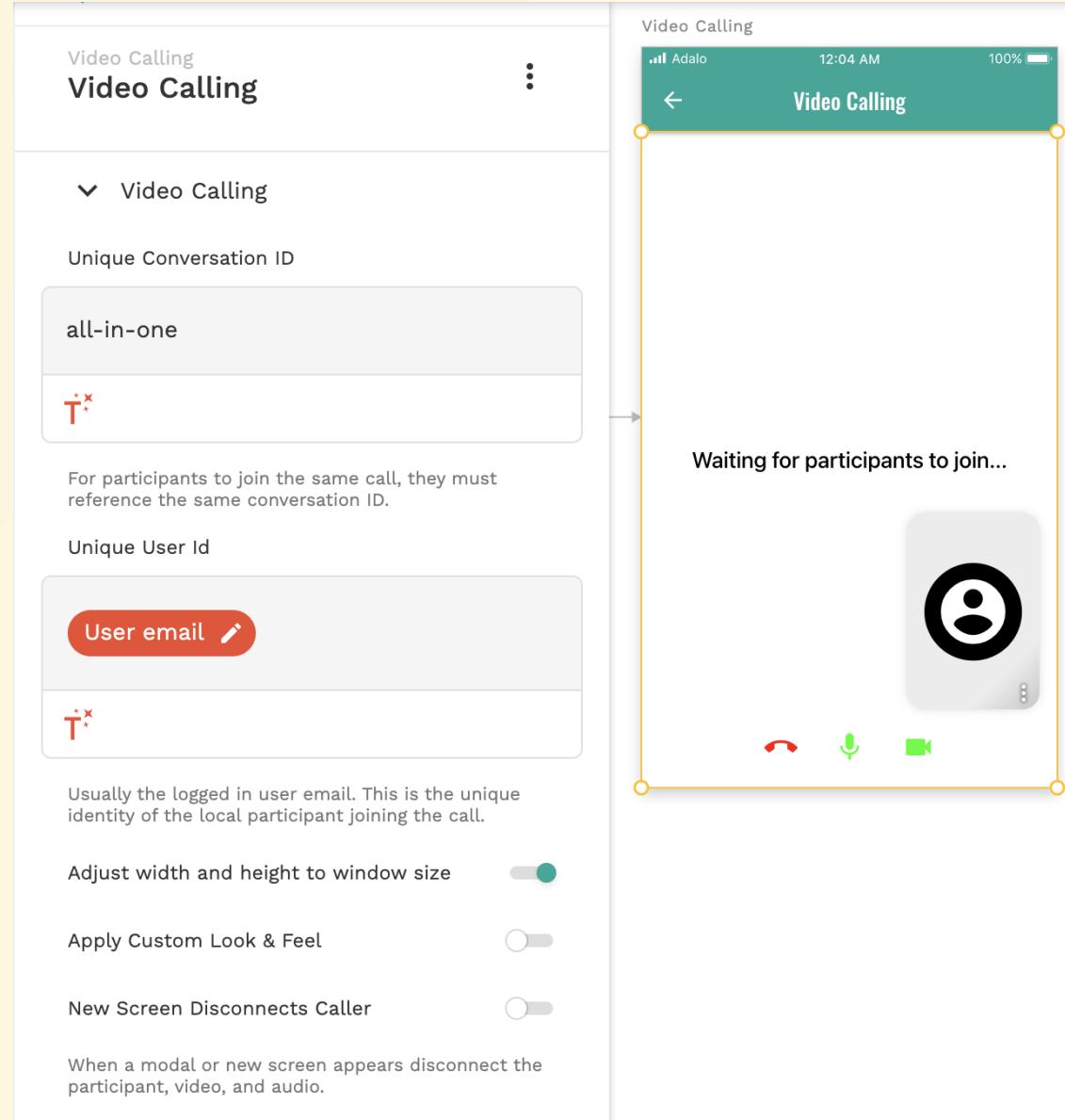
- Home画面にビデオ通話用画面へのLinkボタンを追加
- ADD ACTIONからNew ScreenへのLinkを追加



- Templateで"App Bar"を選択し、"Video Calling" 画面を作成



- Video Callingコンポーネントを画面上に配置
- Unique Conversation IDに "all-in-one" と入力
 - これは、全ユーザーを一つのビデオ通話に参加させる設定です
- Unique User id に "Logged In User's Email" を設定



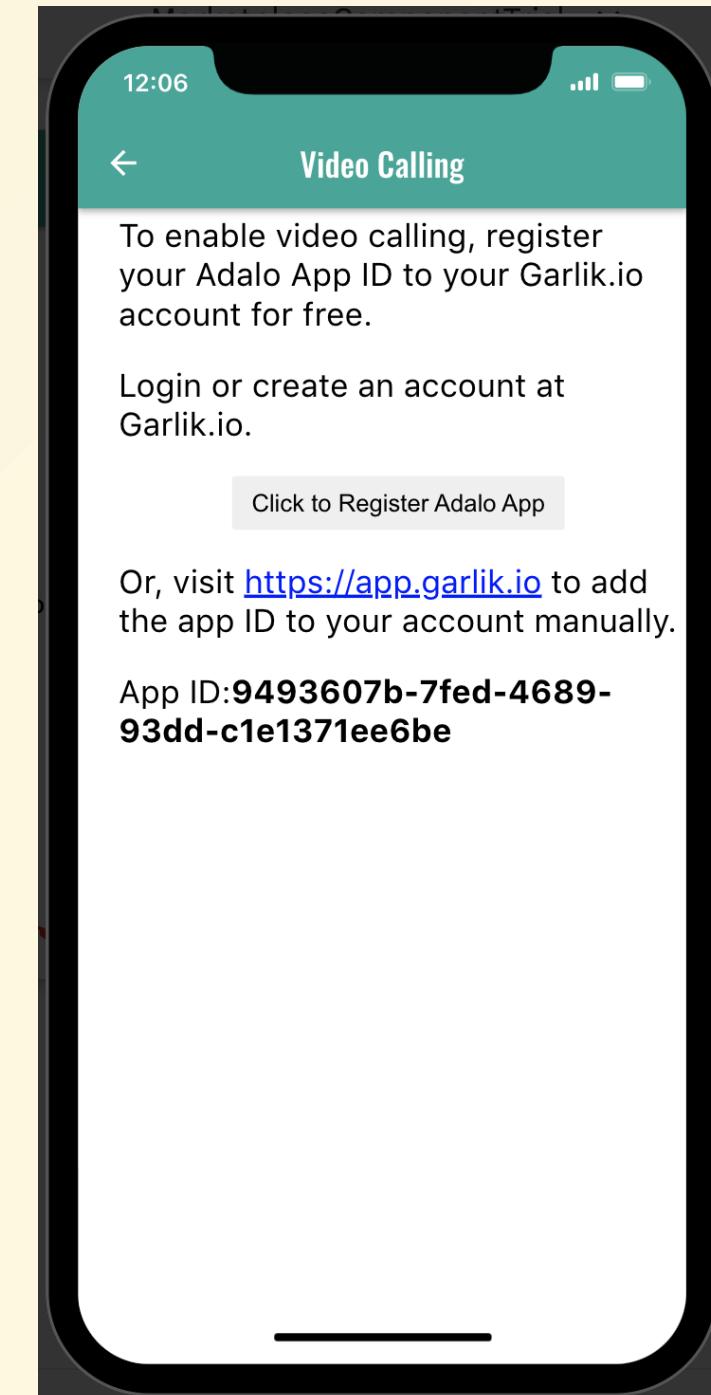
補足

今回は全ユーザーを一つのビデオ通話に参加させる設定にしましたが、複数のユーザーが所属するグループを作ってグループメンバーだけで通話することも可能だと思います。

興味があれば、挑戦してみてください。

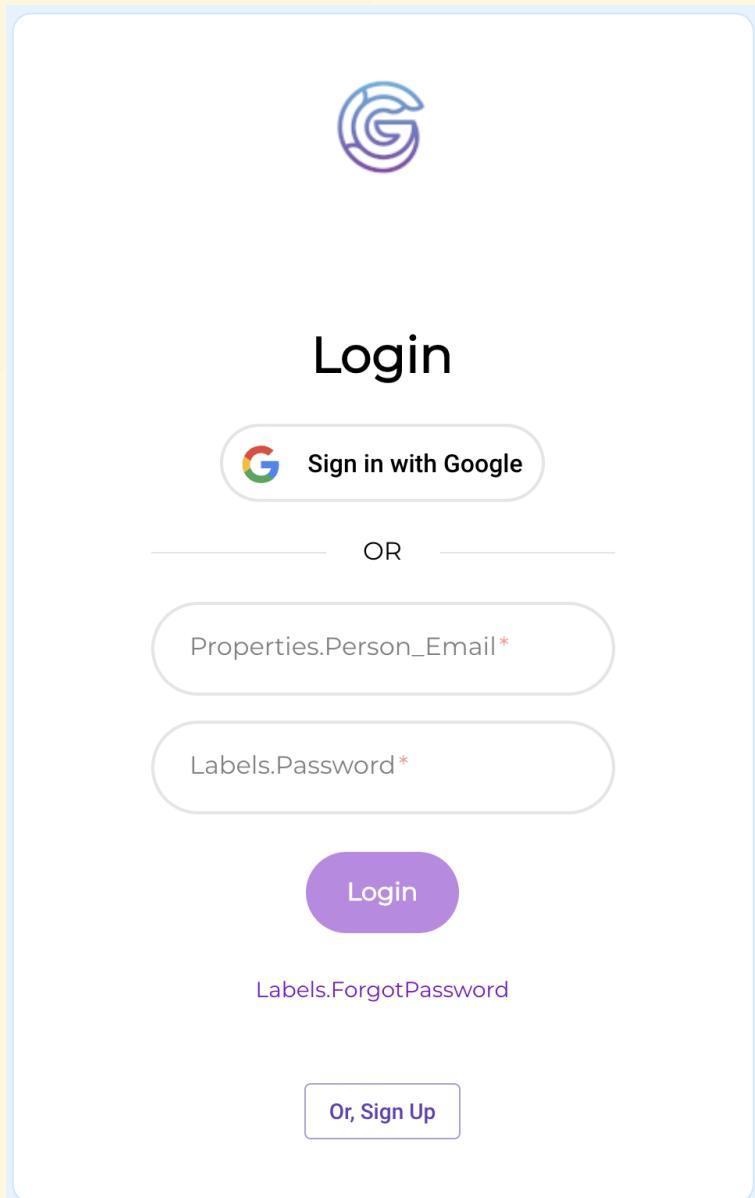
Preview画面で確認してみると、"register your Adalo App ID to your Garlick.io account for free" というメッセージが表示されます。案内に従いましょう。

- "Click to Register Adalo App" をクリック



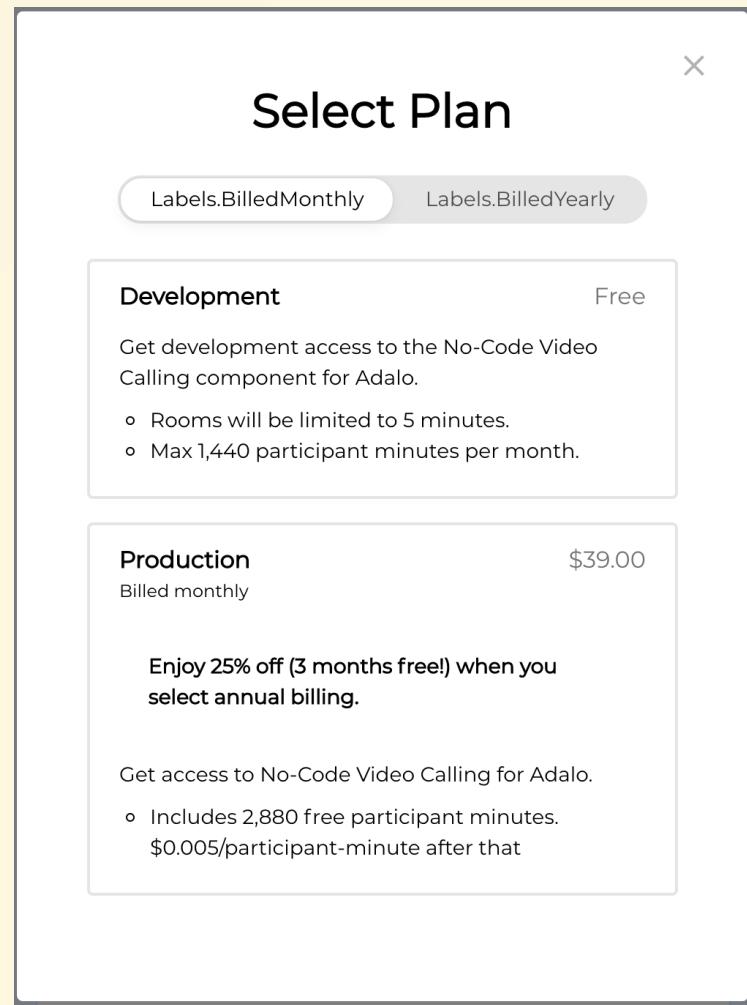
ログイン画面が表示されますが、まだアカウントを作成していません。

- "Or, Sign Up"をクリック

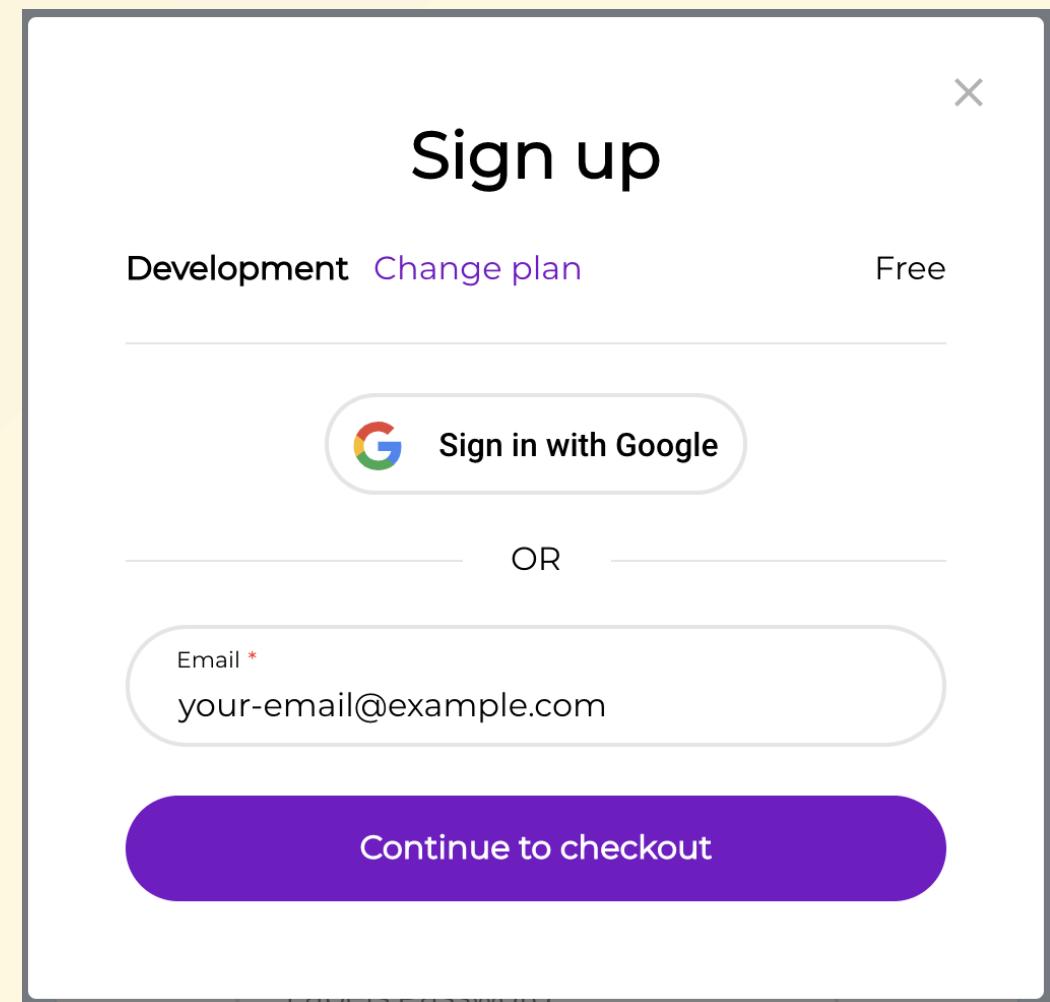


The screenshot shows a login interface with a light blue header containing a purple circular logo with a stylized 'G'. Below the header is the word 'Login' in a large, bold, black font. Underneath 'Login' is a button labeled 'Sign in with Google' featuring the Google 'G' logo. A horizontal line with the word 'OR' in the center separates this from two input fields. The first input field is labeled 'Properties.Person_Email*' and the second is labeled 'Labels.Password*'. Below these fields is a purple rounded rectangular button labeled 'Login'. At the bottom of the form is a link labeled 'Labels.ForgotPassword'. In the bottom right corner of the form area is a small button labeled 'Or, Sign Up'.

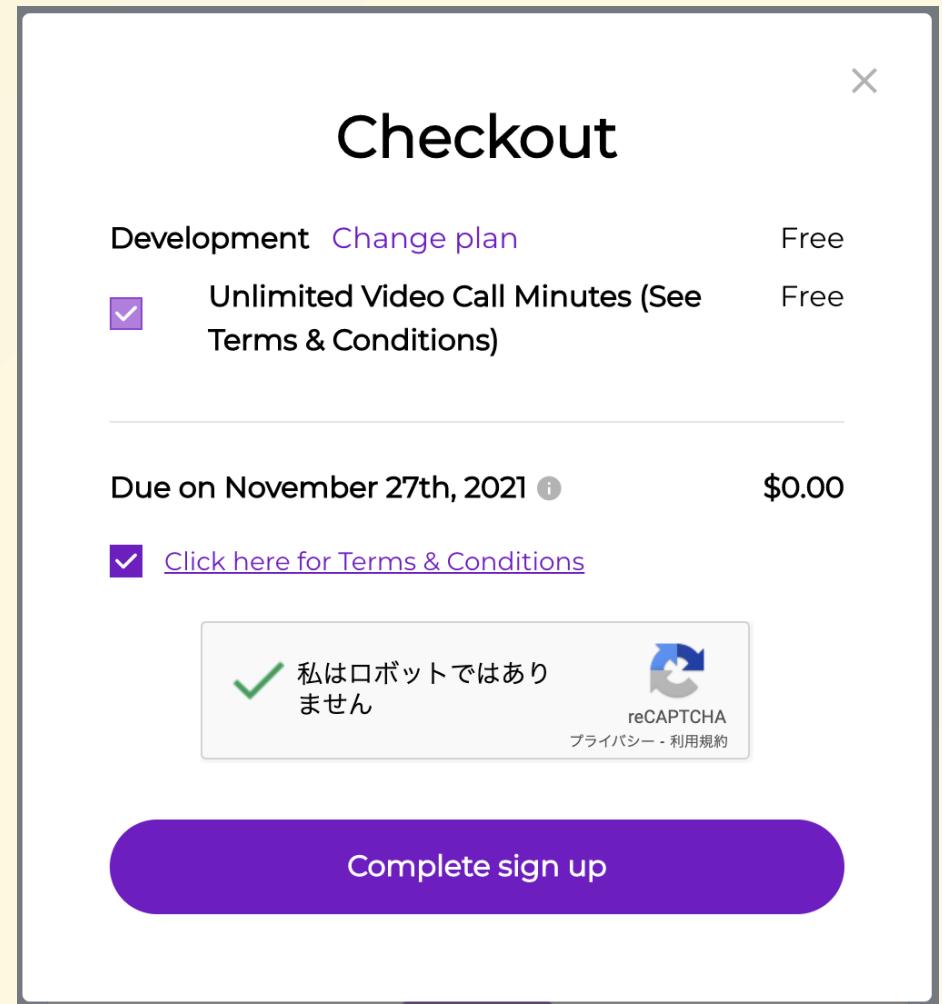
- Development Planを選択(無料です)



- Emailを入力して "Continue to checkout" をクリック
 - Googleアカウントを持っているれば、Sign in with GoogleをクリックしてもOKです。



- 3つのチェックボックスにチェックをつけて、"Complete sign up" をクリック



- 届いたメールを開いて、
"Confirm your account" を
クリック

Confirm your Garlik.io account 外部 受信トレイ ×

 Garlik.io <support@garlik.io>
To imahashi+20211126 ▾

0:11 (0 分前) ☆ ⏪ ⏴



Hello imahashi+20211126@guildworks.jp,

Thanks for signing up for Garlik.io! Please confirm your account to get started.

[Confirm your account](#)

-- The Garlik.io Team

If you're having trouble clicking the confirmation button, copy and paste the URL below into your web browser.

<https://garlik.io/email-confirmed?confirmationToken=838d169af792bef2bfc3cdccbea17841>

- Passwordを設定

A screenshot of a password setup dialog box. The title is "Set Password". Below it is a message: "Please set your password." A large input field is labeled "Labels.Password*" with a note below it: "Must be at least 8 characters". At the bottom are two buttons: "Cancel" (gray) and "Set Password" (purple).

Set Password

Please set your password.

Labels.Password*

Must be at least 8 characters

Cancel Set Password

アカウント登録が完了しました。
次は、Adalo App IDを登録します。

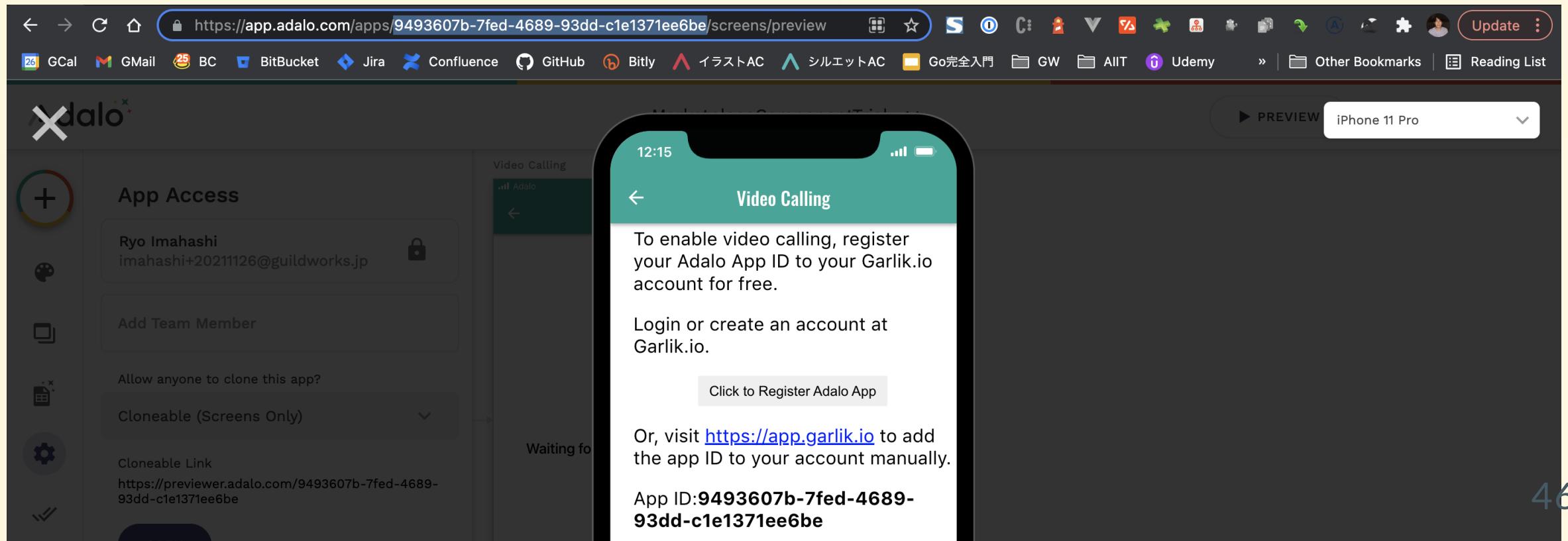
The screenshot shows the Adalo No-Code Components interface. On the left, there's a sidebar with icons for 'No-Code Components' (No-Code Video Calling, Documentation, Feature Requests, Support), 'JA' (Japanese), a message icon, a user icon, and a gear icon.

Main content area:

- Participant Minutes Consumed (Development Plan)**: Shows 0.00/1440 Minutes consumed, with 0% completion. It indicates 0 billable minutes consumed and 0 free participant minutes consumed.
- Register Adalo App**: A form field labeled "Adalo App ID" contains the value "ed7577d9-3a1b-4a07-bd04-de4c5fce29c3". Below the field is the text "Register your Adalo app IDs to enable video calling." and a purple "Register App" button.
- Registered Apps**: A placeholder box for listing registered apps.

- Adaloの管理画面を開いて、URLに含まれるApp IDをコピー
 - 以下のURLの "xxx..." の部分です。

<https://app.adalo.com/apps/xxxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/screens/preview>

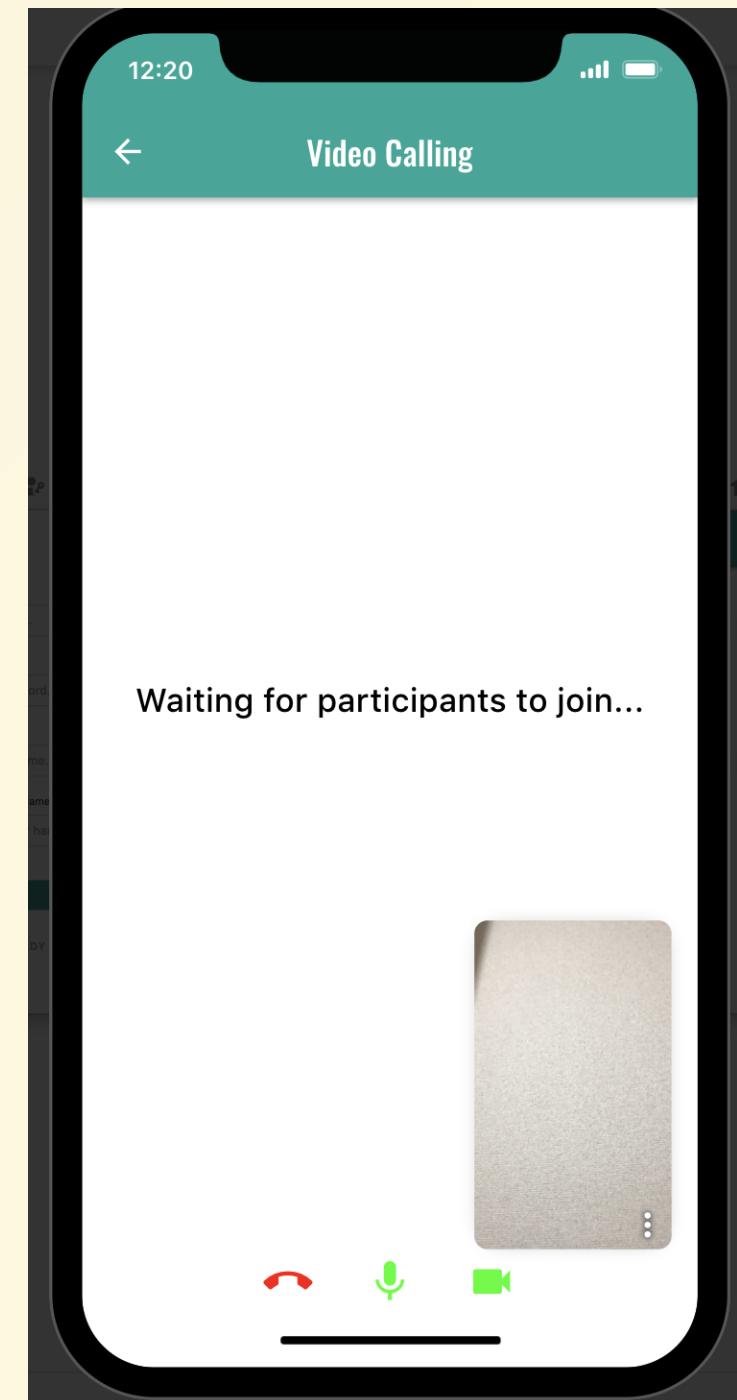


- Garlick.ioの管理画面でAdalo App IDを入力して、"Register App"をクリック

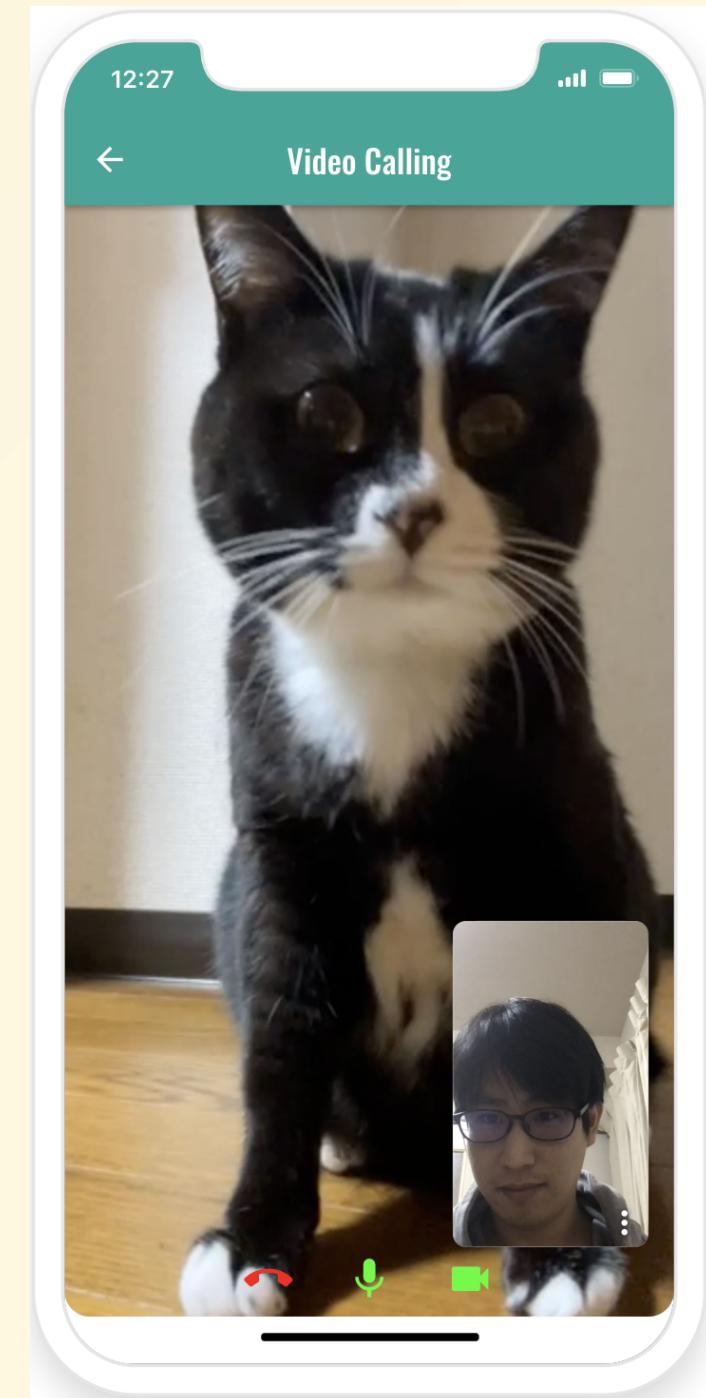
The screenshot shows the Garlick.io management interface with three main sections:

- Participant Minutes Consumed (Development Plan)**: Shows 0.00/1440 Minutes consumed, with 0% completion. It indicates 0 billable minutes consumed and 0 free participant minutes consumed.
- Register Adalo App**: A form where the Adalo App ID **9493607b-7fed-4689-93dd-c1e1371ee6be** has been registered successfully, indicated by a green checkmark. A note below says "Register your Adalo app IDs to enable video calling." A purple "Register App" button is present.
- Registered Apps**: A list containing one item: **9493607b-7fed-4689-93dd-c1e1371ee6be**, accompanied by a small circular icon with an 'A' and a trash can icon.

AdaloのPreview画面をリロードして再度ビデオ通話画面を表示すると、ビデオ通話の参加者待ちの状態になります。



他のユーザーがログインして
ビデオ通話画面を表示する
と、通話ができます。



質問: ビデオ通話を試してみますか?

試してみたいというリアクションがもらえたなら、実験用のアプリのURLをシェアします。

スライドの枚数的に時間が足りなくなりそうなので、割愛します...

無料プランだと合計通話時間が1440分に到達したら通話ができなくなるようですが、開発時のテストには十分だと思います。

ビデオ通話機能が必要になったら、活用してください。

Participant Minutes Consumed (Development Plan)

32.00/1440 Minutes



0 billable minutes consumed

32.00 free participant minutes consumed

この他にも外部サービスと連携するためのコンポーネントがいくつか提供されています。興味があれば、試してみましょう。

例:

- Youtube(無料)
- Google Map(フリートライアルが使える)
- Google Signin(\$25)

Custom Action

外部連携コンポーネントの次は、APIから取得したデータをAdaloの画面上で扱う方法を紹介します。

参考

“ アプリケーションプログラミングインターフェース（API、英: Application Programming Interface）とは、広義ではソフトウェアコンポーネント同士が互いに情報をやりとりするのに使用するインターフェースの仕様である。 ”

<https://ja.wikipedia.org/wiki/アプリケーションプログラミングインターフェース>

まずは、API連携を試してみましょう。
無料で試せるThe Cat APIを使います。以下のURLにアクセスしてください。

<https://thecatapi.com/>

The screenshot shows the homepage of The Cat API. At the top, there's a navigation bar with links for PRICING, ABOUT, and DOCUMENTATION. Below that is a secondary navigation bar with links for VOTE, BREEDS, IMAGES/SEARCH, FAVOURITES, and UPLOAD. A red banner at the bottom states "Because everyday is a Caturday." and includes a link to "Get your license now.". On the right side, there's a large image of a white cat's paws on a wooden floor, with a "LOVE IT" button above it. At the very bottom, a dark banner says "The Cat API has delivered Billions of requests, to over 30K developers". To the left of the main content area, there are icons for React, jQuery, Vue.js, Node.js, and Angular.

参考: 犬派の人のために、The Dog APIもあります。おそらくThe Cat APIと同様のことができると思います。(試せていないので、まずは一緒にThe Cat APIを使うことをおすすめします)

<https://www.thedogapi.com/>

The screenshot shows the homepage of The Dog API. At the top, there's a blue header bar with navigation links: PRICING, ABOUT, and DOCUMENTATION. Below the header, the main title "The Dog API - Dogs as a Service" is displayed, followed by the tagline "Because everyday is a Dog Day." A short description below states: "A public service API all about Dogs, free to use when making your fancy new App, Website or Service. [Get your license now.](#)" To the left of the main content area, there are icons for React, jQuery, Vue.js, Node.js, and Angular. The central part of the page features a large image of a white dog sitting in a grassy field. Above the image is a black navigation bar with links: VOTE, BREEDS, IMAGES/SEARCH, FAVOURITES, and UPLOAD. Below the image are two buttons: a green "LOVE IT" button with a thumbs-up icon and a red "NOPE IT" button with a thumbs-down icon. At the bottom of the page, a blue footer bar contains the text: "The Dog API has delivered Billions of requests, to over 30K developers".

APIを使用する際には、APIの提供者からAPIキーを発行してもらう必要がある場合が多いです。

The Cat APIでもAPIキーが必要になりますので、発行してもらいましょう。

- 下にスクロールしてPricing の欄の "SIGNUP FOR FREE" をクリック

Pricing

Free



\$0
No hidden costs

Free for you to use on your non-monetized app

SIGNUP FOR FREE

- 10,000 requests / month
- Free Code Samples
- [Community forums](#)
- SSL Encryption

- E-mail、App Description、type of projectを入力し、"SIGNUP" をクリック

Signup for an API key

It's completely free, email is only be used to send you an API Key & your stats, and you can use the API as much as you like. No Spam Ever.

E-mail

 your-email-address@example.com

App Description

 Sample App

What type of project will you use the API for?

A personal project

A project for school/college/university

A business project

Opt-in to my newsletter about other APIs i'm building?

SIGNUP

[Privacy Policy](#) | [Terms & Conditions](#)

- メールで送られてきたAPIキーを確認(後で使います)

The screenshot shows an email client interface with a single message. The message is from 'aden@thatapiguy.com' (To: 自分) and was sent on 2021/11/25 17:01 (1日前). The subject of the email is 'Welcome to The Cat API!'. The email body contains the following text:

The Cat API - API Key

Thanks for signing up, welcome to the API!

Your API key:

xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Use it as the 'x-api-key' header when making any request to the API, or by adding as a query string parameter e.g. 'api_key=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx' [More details on authentication.](#)

[API Documentation](#) | [Postman Collection](#) | [Public Forum](#)

Join our new API Slack Group - [That API Slack Group](#)

All the best, Aden & That API Team.

p.s. We build APIs for companies & good causes. If you're interested find out more at <https://thatapicompany.com>

Big love!

次に、APIドキュメントでAPIの使い方を確認しましょう。

- 以下のURLへアクセス(先程のメールにも"API Documentation"というリンクが記載されています)

<https://docs.thecatapi.com/>

トップページに記載されている、ランダムな子猫の画像を取得するAPIを使います。

Exampleと同じように、ボタンを押すと画像が切り替わるようになります。

Quickstart - Get a random Kitty

Using the language / framework of your choice:

1. Load <https://api.thecatapi.com/v1/images/search>
2. Get the first Array object of the JSON response
3. Load it's .url
4. Enjoy the kitty 😺

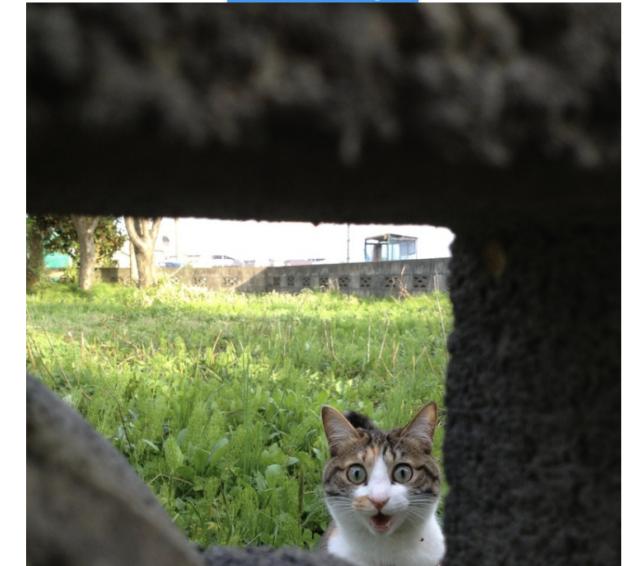
Bonus: Add a button to repeat the request for a new kitty!

Example

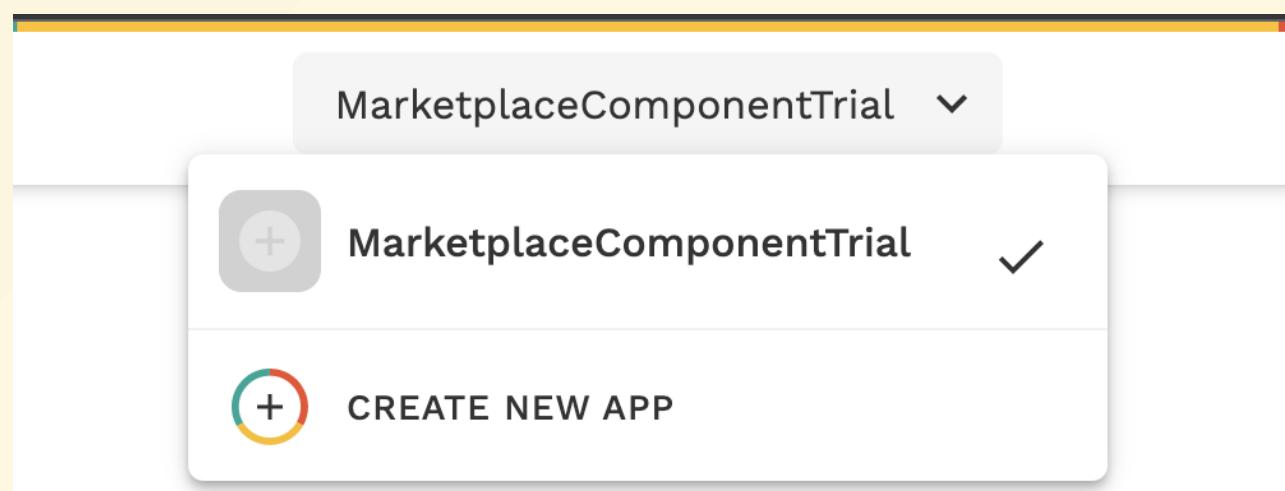
Uses <https://api.thecatapi.com/v1/images/search> to show one random image every time "Another" is clicked.

Example Source Code: VueJS

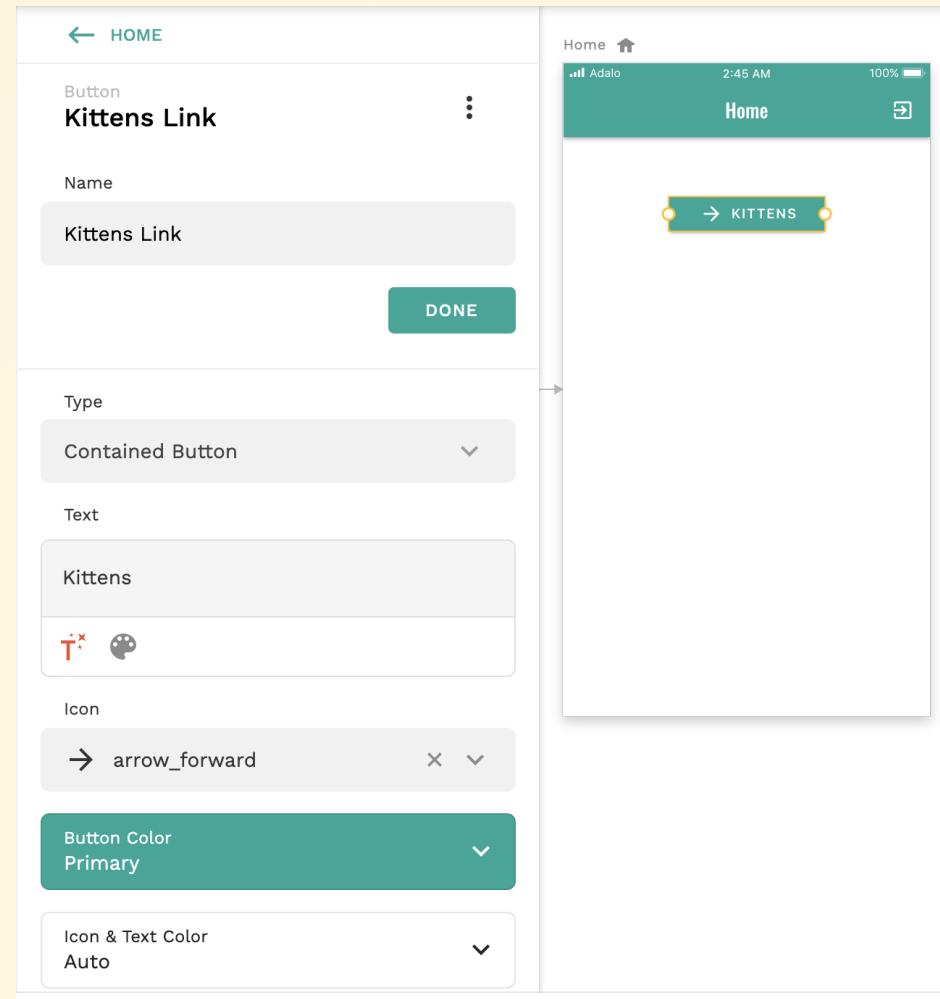
ANOTHER ↗



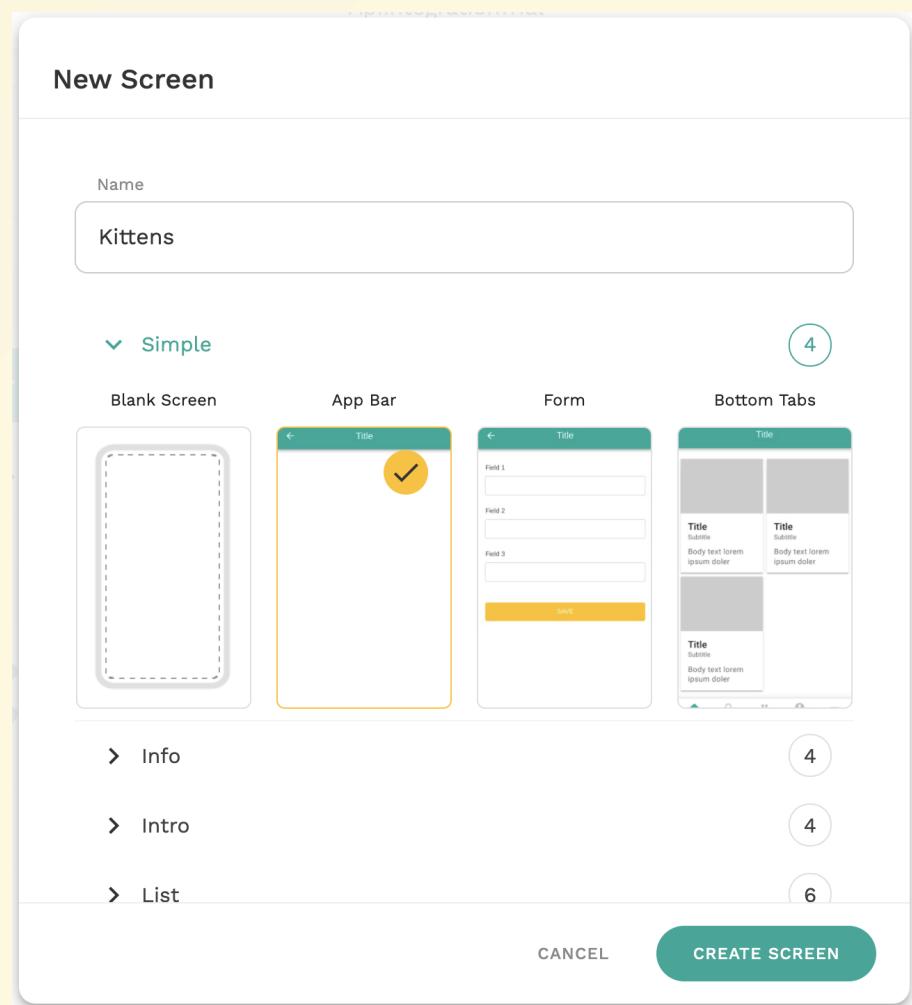
- Adaloの管理画面でCREATE NEW APP
- 設定は以下の通り
 - Platform: Native Mobile App
 - Template: Blank
 - App Name: ApilntegrationTrial



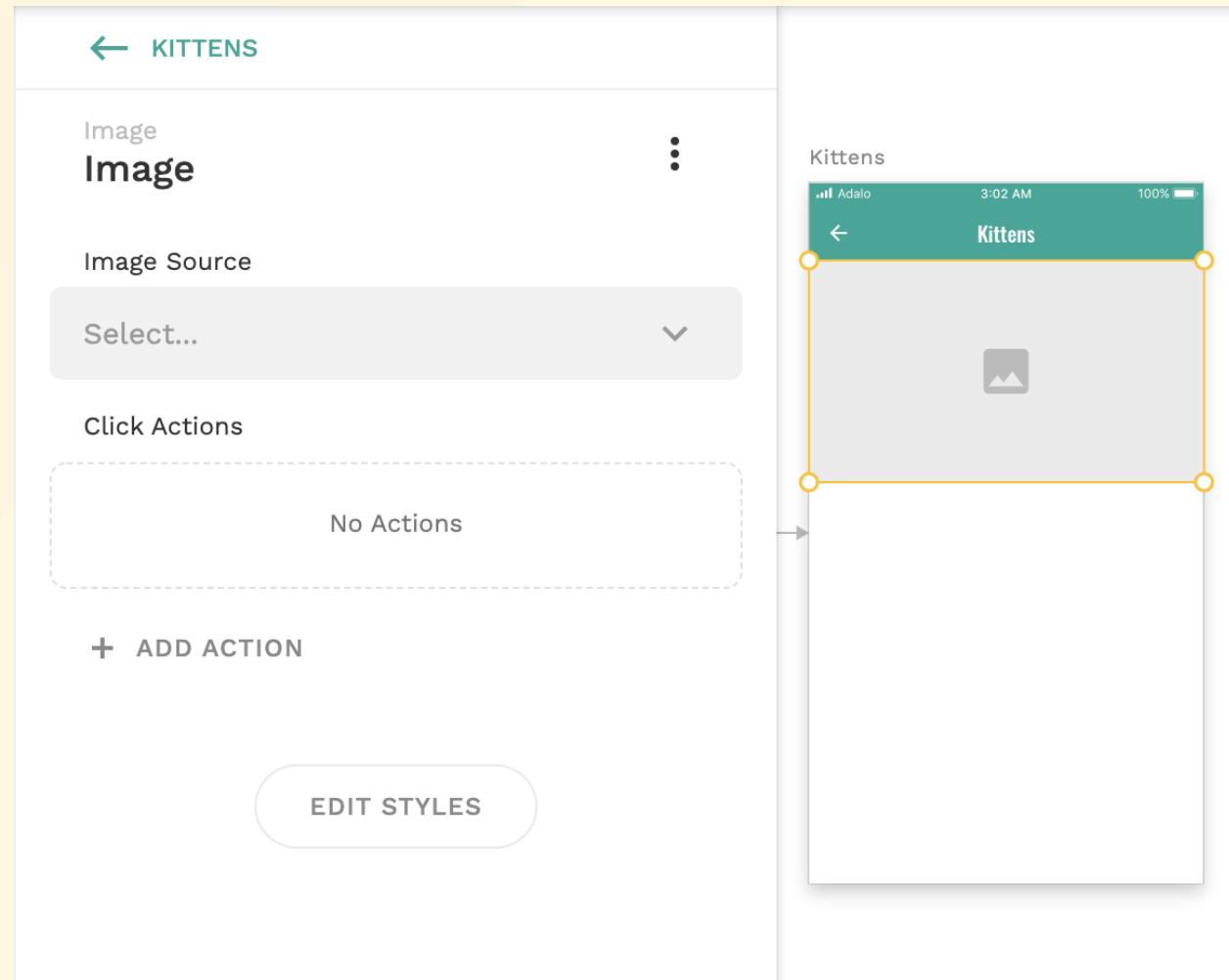
- Home画面に子猫画像表示画面へのリンクボタンを追加
- ADD ACTIONからNew ScreenへのLinkを設定



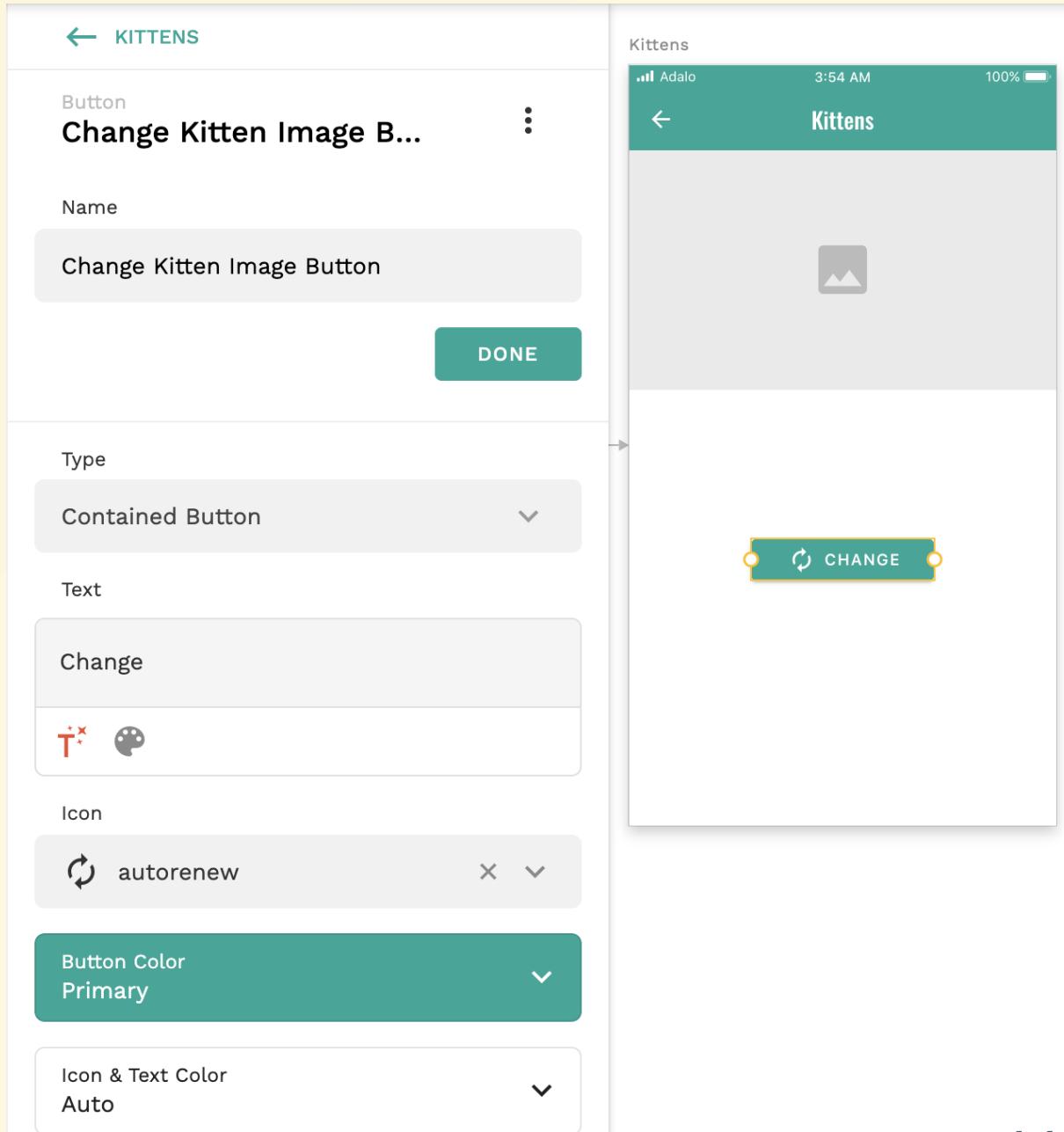
- TemplateにApp Barを選択し、Kittens画面を作成



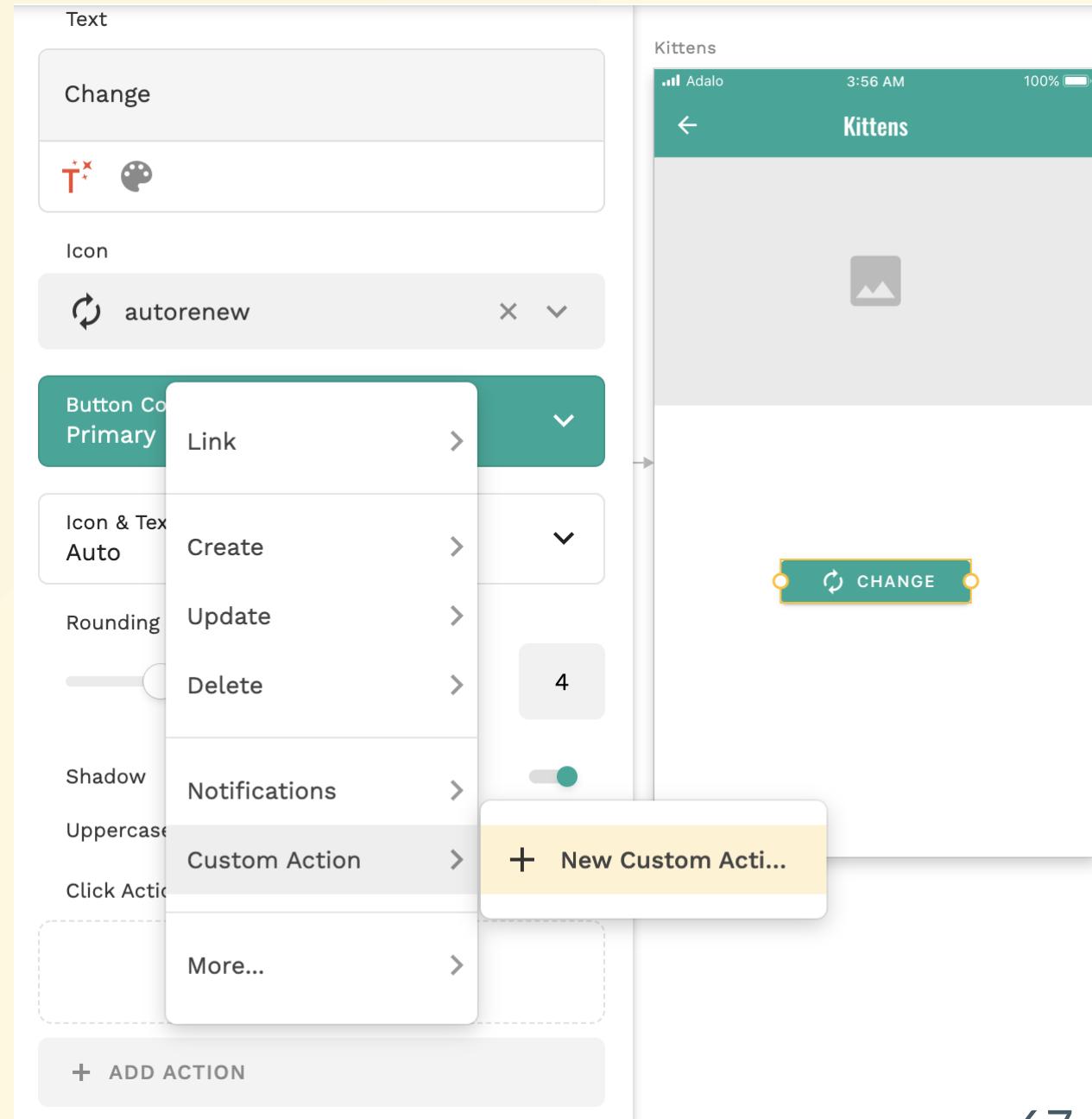
- Imageコンポーネントを画面上に配置
- コンポーネントの設定はそのままにしておく(後で設定します)



- Change Kitten Image
Buttonを追加



- ADD ACTIONからNew Custom Actionを選択



14日間のIntegration Pack Trial(無料)の開始を促されます。

- "START INTEGRATION PACK TRIAL" をクリック

Interested In Trying Out Custom Actions?

Start your **14-day free trial** of the Integration Pack to get full use of External Collections, Custom Actions, & Collections API. No credit card needed!



External Collections

- ➔ Integration data from external sources like Airtable & Google
- ➔ Works with custom APIs

[LEARN MORE](#)

Custom Actions

- ➔ Connect & run actions in external apps (i.e. trigger an email, text, or Slack message)
- ➔ Works with Zapier, Integromat, in addition to any custom API

[LEARN MORE](#)

Collections API

- ➔ Full access to your Adalo App's API
- ➔ Send and receive custom API requests

[LEARN MORE](#)[NOT RIGHT NOW](#)[START INTEGRATIONS PACK TRIAL](#)

トライアルが開始しました。

- "CREATE NEW CUSTOM ACTION" をクリック



Congrats! You now have 14 days to try out integrations on Adalo!

The Integrations Pack Trial includes access to External Collections, Custom Actions and use of the Collections API.

[DONE](#) [CREATE NEW CUSTOM ACTION](#)

- 以下を入力してNEXTをクリック
 - Name: GetRandomKitten
 - Type: Create

The screenshot shows a user interface for creating a new API endpoint. It consists of three numbered steps at the top: 1. Name & Type (highlighted in yellow), 2. API Request, and 3. Outputs. The current step, 'Name & Type', contains two input fields: 'Name' (containing 'GetRandomKitten') and 'Type' (containing 'Create', with a dropdown arrow indicating more options). At the bottom right of the screen are two buttons: 'CANCEL' and a large yellow 'NEXT' button.

次に、送信するAPI Requestを設定していきます。

The screenshot shows a user interface for configuring an API request. The process is divided into three steps:

- Name & Type (Completed)
- API Request (Selected)
- Outputs

API Request Step:

- API Base URL:** https://api.example.com/v1/action
- Method:** Select a Method (dropdown menu)
- Headers & Queries:** No Auth Parameters
- Body:** Enter JSON here

Outputs Step:

- Inputs:** No Inputs
- Add Item:** + ADD ITEM

Buttons at the bottom:

- BACK
- RUN TEST REQUEST (highlighted in yellow)

以下のURLにアクセスして、
使用するAPIのドキュメントか
ら設定項目を確認します。

<https://docs.thecatapi.com/api-reference/images/images-search>

- API Base URLは
<https://api.thecatapi.com/v1/images/search>
- MethodはGET
- Headerに x-api-key という
NameでAPI keyを設定

GET /images/search

Search & Iterate through all public images. <https://api.thecatapi.com/v1/images/search>

Authorization

▼ apiKey - x-api-key

Just use the API Key emailed to you when you signed up.

- Name: x-api-key
- Location: header

- 確認した結果を踏まえてAPI Requestを設定
 - API Base URLは
<https://api.thecatapi.com/v1/images/search>
 - MethodはGET
 - Headerに x-api-key というNameでAPI keyを設定
- "RUN TEST REQUEST" をクリック

The screenshot shows a step-by-step API configuration process:

- Step 1: Name & Type** (Completed)
- Step 2: API Request** (Currently Selected)
- Step 3: Outputs** (Not Yet Started)

Under Step 2: API Request, the configuration includes:

- API Base URL:** https://api.thecatapi.com/v1/images/search
- Method:** GET
- Headers & Queries:** A red-highlighted section containing a single header entry:
 - Header:** x-api-key
 - Name:** x-api-key
 - Value:** 250736df-9bd0-40c6-9090-5f16b5b8ef05

On the right side, there is an **Inputs** section labeled "No Inputs" with a "+ ADD ITEM" button.

At the bottom right of the main form area are buttons for **BACK** and **RUN TEST REQUEST**.

Testが成功すると、APIから取得したデータ(Magic Text Output Properties)が表示されます。これらは、後続のアクションで使用できます。

- "SAVE CUSTOM ACTION" をクリック

The screenshot shows a user interface for creating a custom action. At the top, there are three tabs: "Name & Type" (checked), "API Request" (checked), and "Outputs" (3 items). Below the tabs, a message says "Test Successful!". Under "URL Called", the URL <https://api.thecatapi.com/v1/images/search> is listed. A note below says "Magic Text Output Properties - Learn how to use these in subsequent actions [here!](#)". There are four items listed under "Outputs":

- A **id**: CzXQVvf23
- A **url**: <https://cdn2.thecatapi.com/images/CzXQVvf23.jpg>
- W **width**: 976
- W **height**: 549

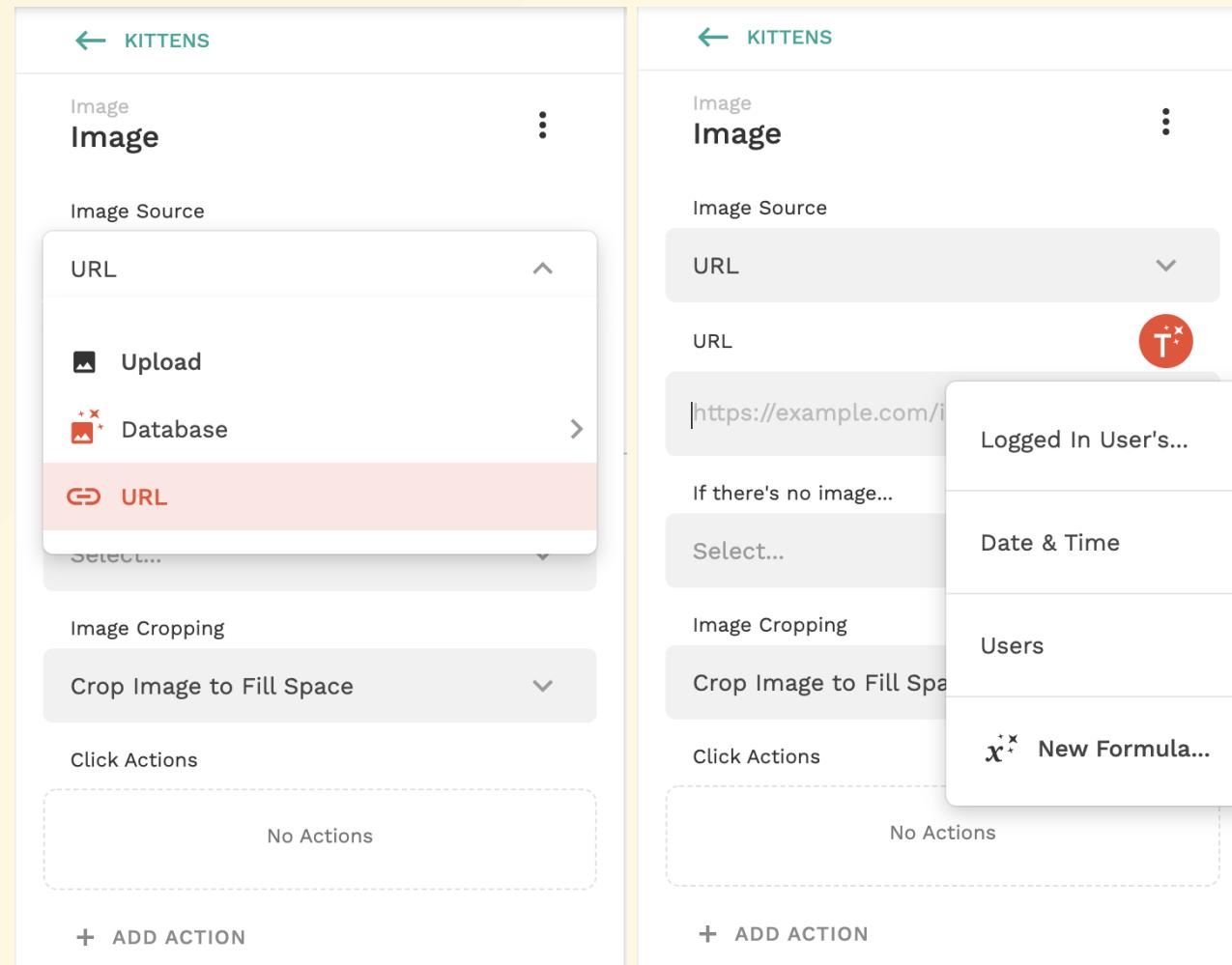
At the bottom left, there is a red "+ ADD ITEM" button. At the very bottom, there is a "SHOW FULL RESPONSE" link.

次に、APIから取得した子猫の画像のURLをImageコンポーネントのImage Sourceに設定します。

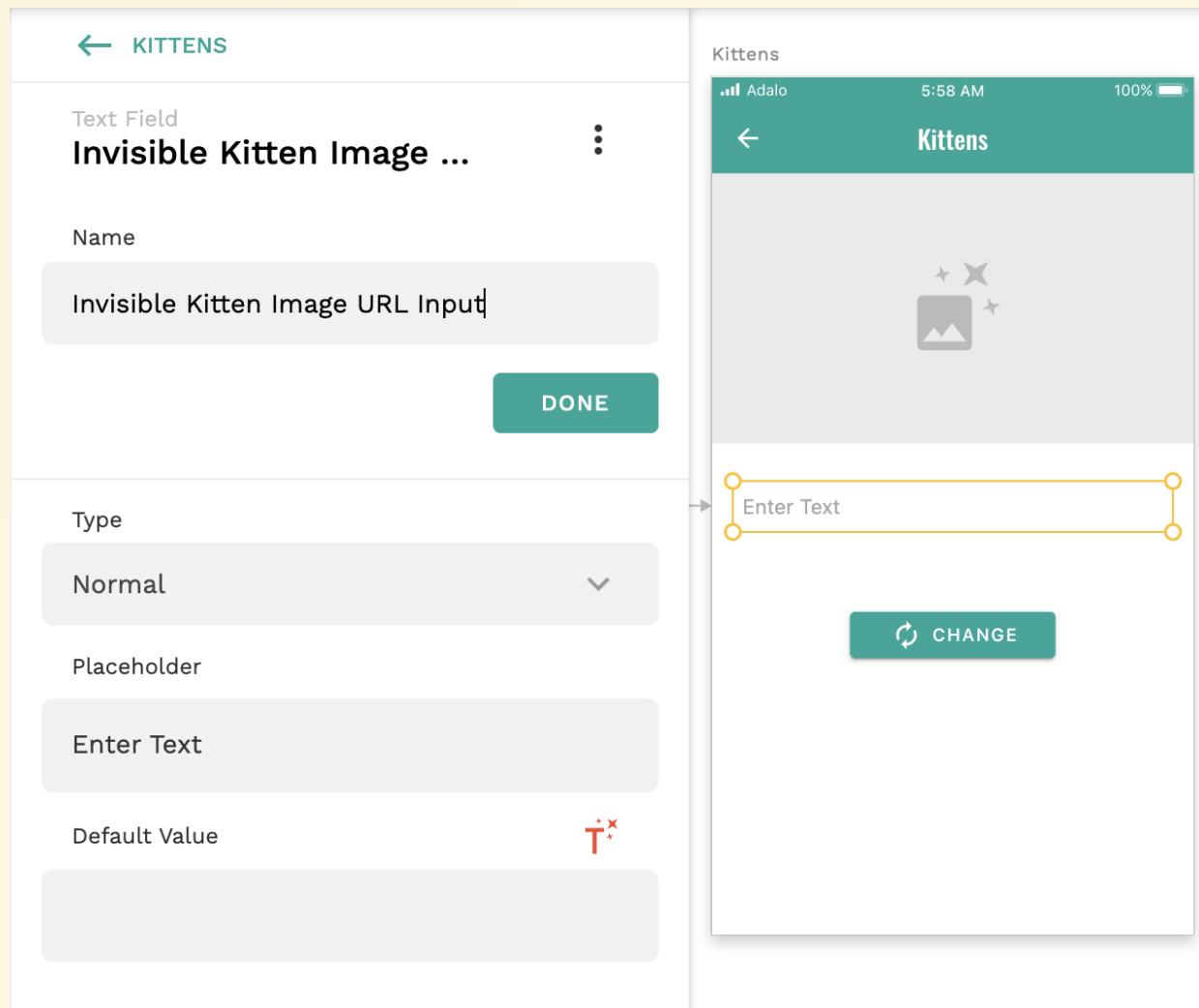
そのままでは、選択肢の中にAPIから取得したデータは出てきません。

A url

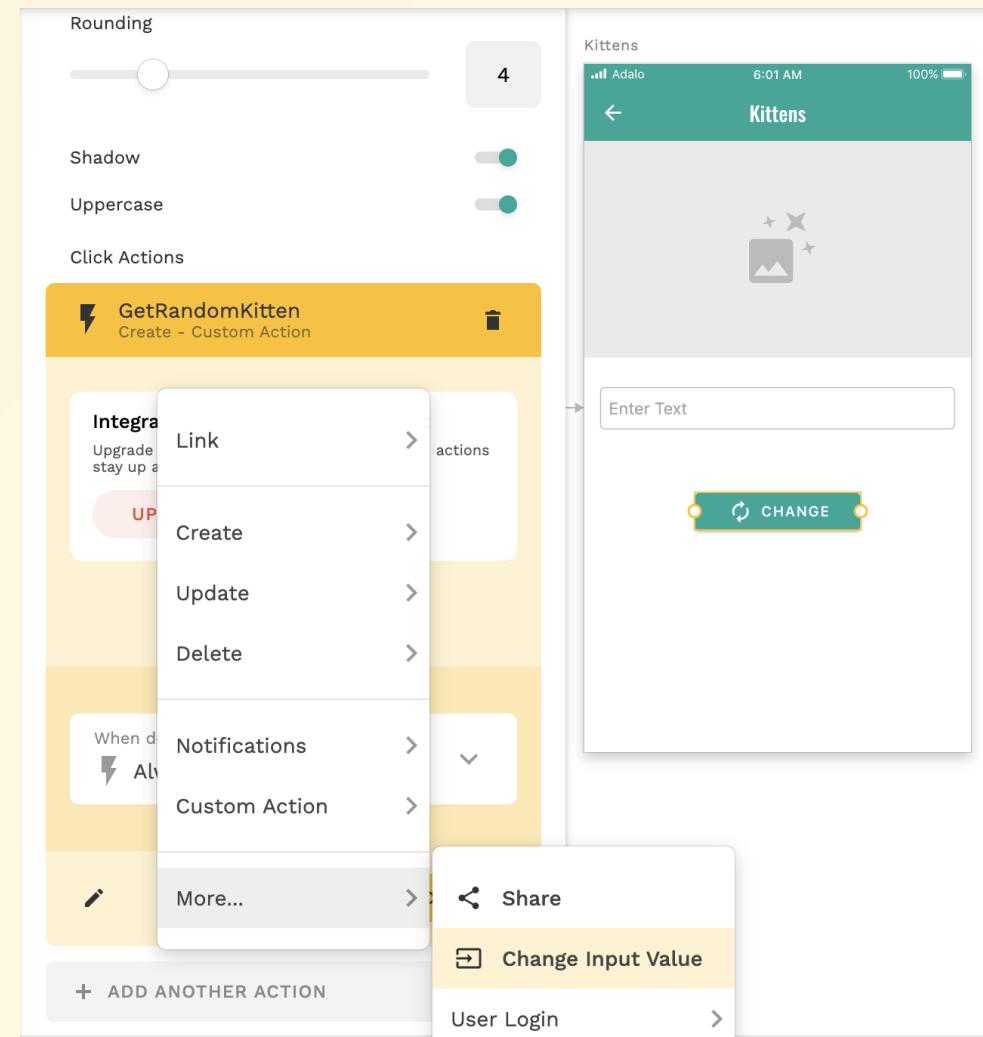
<https://cdn2.thecatapi.com/images/6k7.jpg>



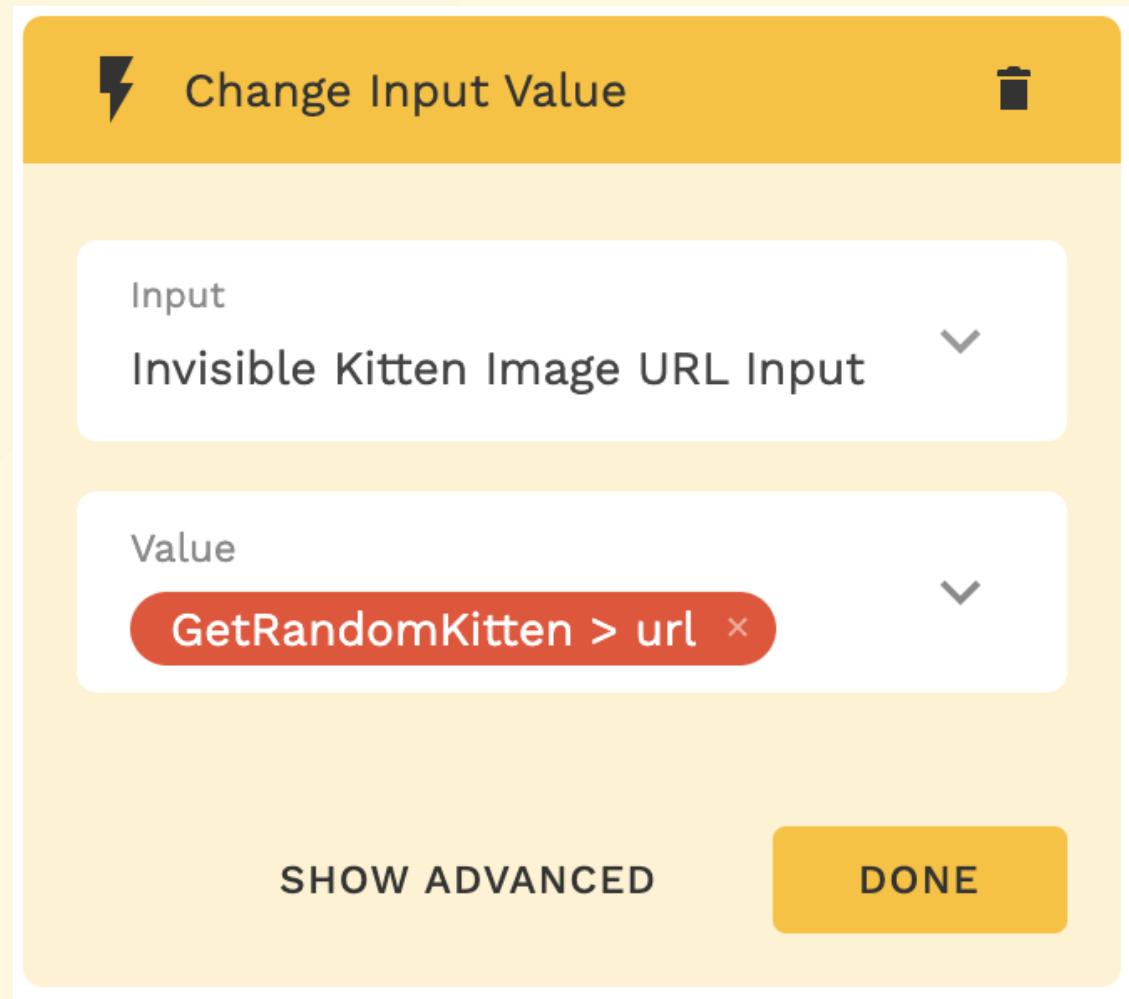
- TextInputコンポーネントを画面上に追加
- Nameを "Invisible Kitten Image URL Input" に変更



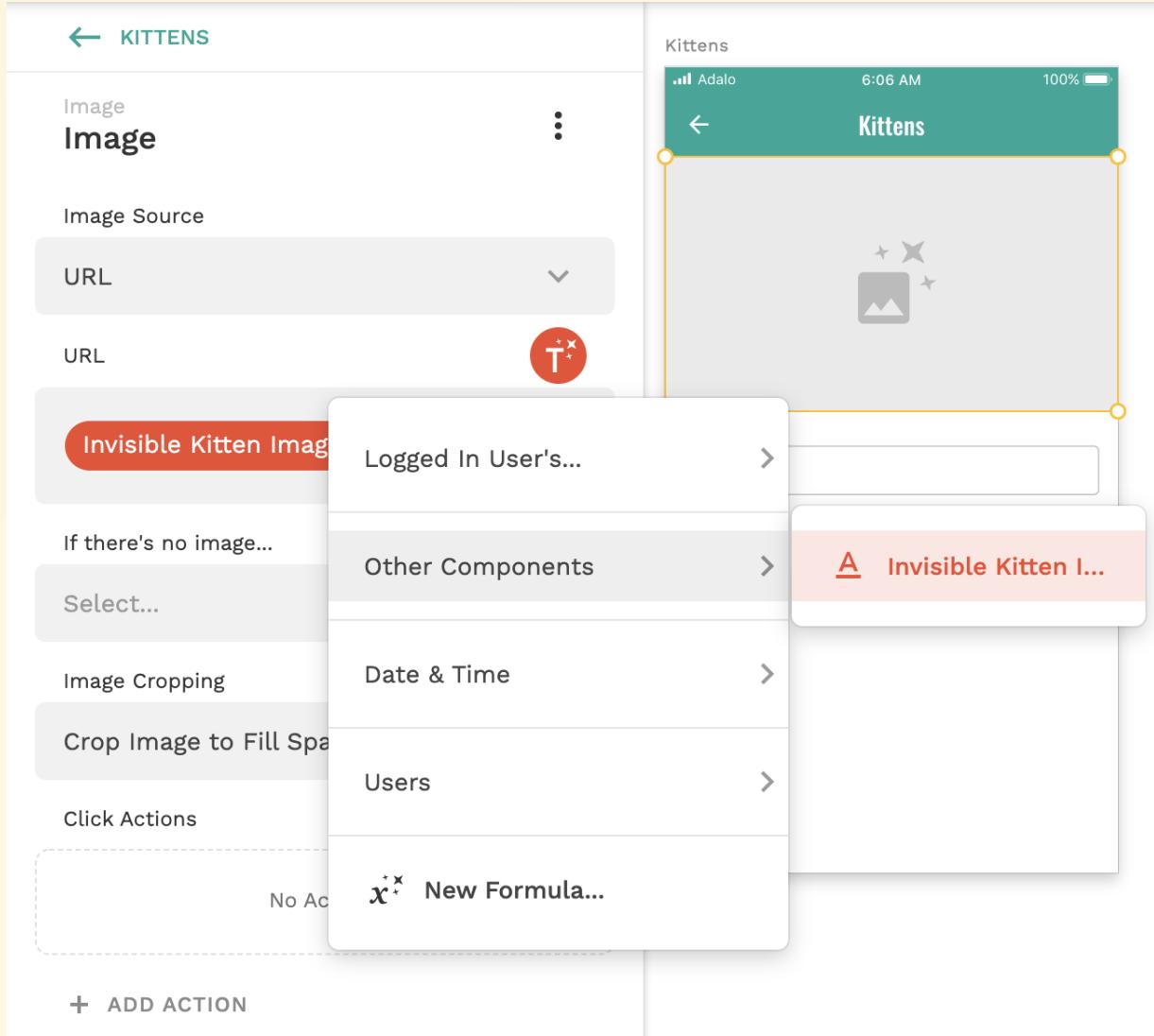
- "Change Kitten Image Button" をクリック
- ADD ANOTHER ACTION から Change Input Value を選択



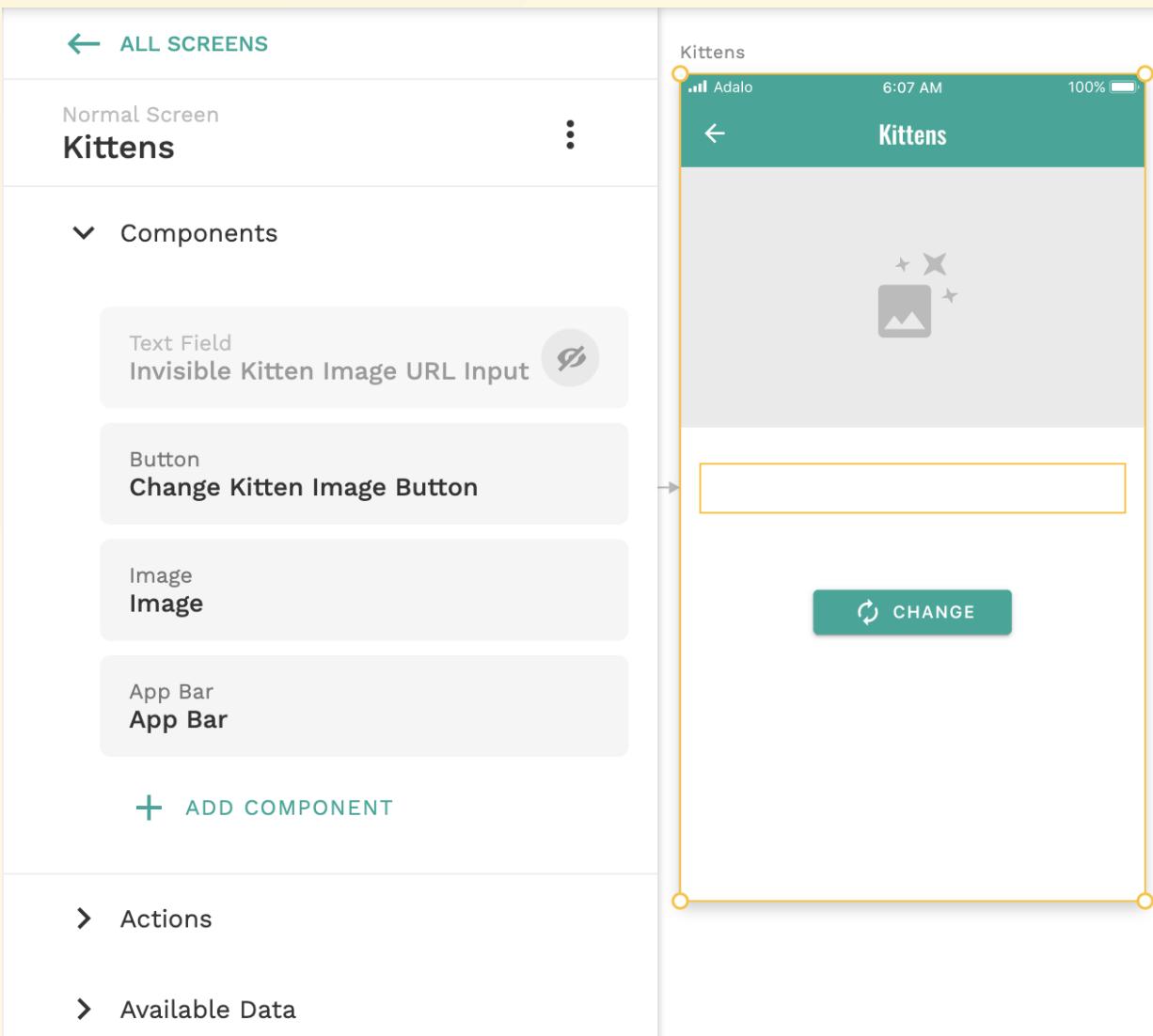
- Inputに "Invisible Kitten Image URL Input" を設定
- Valueに "GetRandomKitten > url" を設定
- "DONE" をクリック



- Imageコンポーネントをクリック
- URLに "Invisible Kitten Image URL Input" を設定

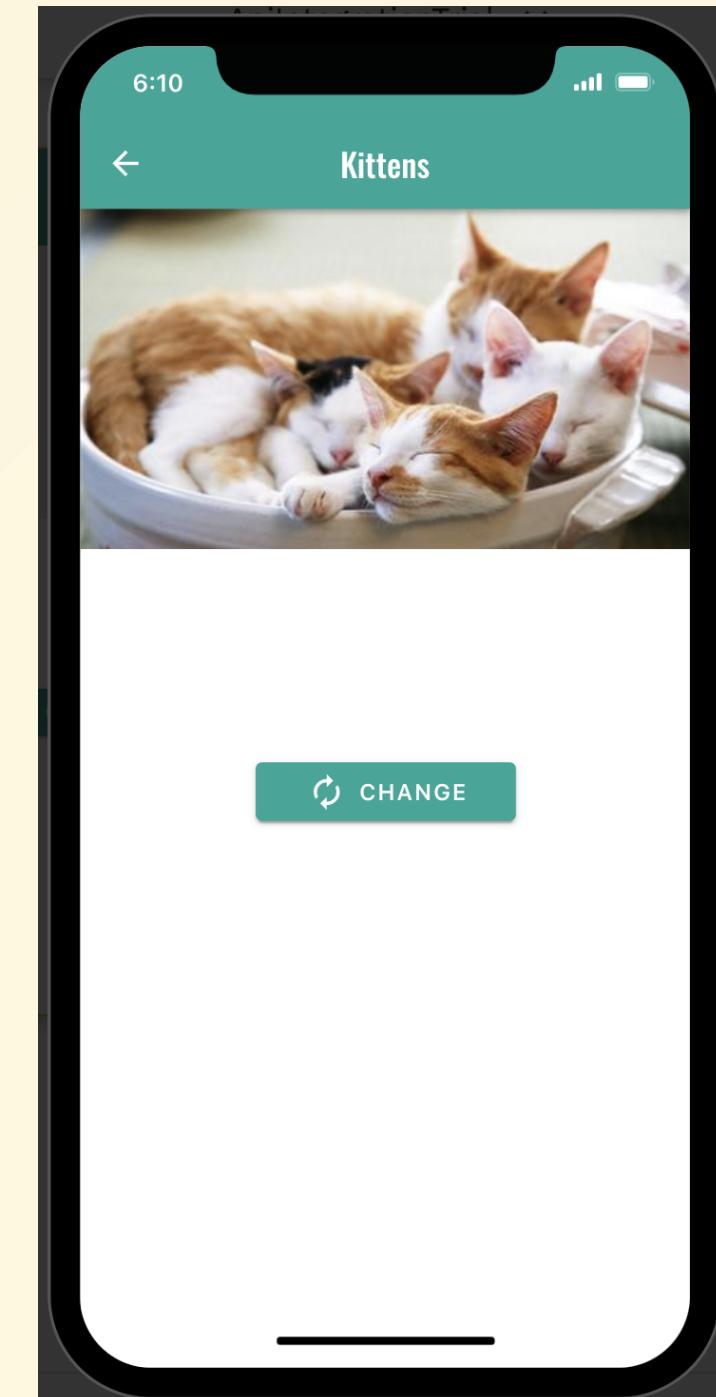


- 画面名 "Kittens" をクリック
- "Invisible Kitten Image URL Input" の右側の目のアイコンをクリックして非表示にする



Preview機能で確認します。

CHANGEボタンをクリックすると、子猫の画像が表示されました。



補足

- Custom ActionでAPIから取得したデータは、後続のActionで使用できます。そのデータをコンポーネントで使いたい場合は、以下のいずれかの方法を使いましょう。
 - 今回のように、後続のActionのChange Input Valueで同一画面上のText InputのValueにデータを設定し、それを読み込みましょう。
 - あるいは、そのデータを後続のActionでデータベースに保存して、それを他のコンポーネントから読み込むことも可能です。この場合、画面遷移後にもアクセスできます。
 - 例: <https://help.adalo.com/integrations/custom-actions>

注意事項

現状、Custom Actionにはいくつかの制限があります。

- Custom ActionはFormコンポーネントのSubmitボタンでは動作しません。
- Custom Actionが画面全体のActionとして使用されている場合、APIのレスポンスとして取得したデータは皇族のActionで使用できません。
- アプリをCloneしてもCustom Actionはコピーされません。Custom Actionを含むアプリをCloneしたら、手動で作成し直してください。

>アプリをCloneしてもCustom Actionはコピーされません。Custom Actionを含むアプリをCloneしたら、手動で作成し直してください。

14日間のIntegration Pack Trialは、Development Phaseの真っ只中で終了すると思われます。

Custom Actionを多用する可能性がある場合、Development Phaseの作業開始前に新しくAdaloのアカウントを作成し、Integration Pack Trialを新たに開始することをおすすめします。

External Collection

APIから取得したデータをAdaloのCollectionとして扱う方法を紹介します。

複数のデータを一括取得してそれらを画面上に一覧表示するような場合はCustom Actionではなく、External Collectionを使います。

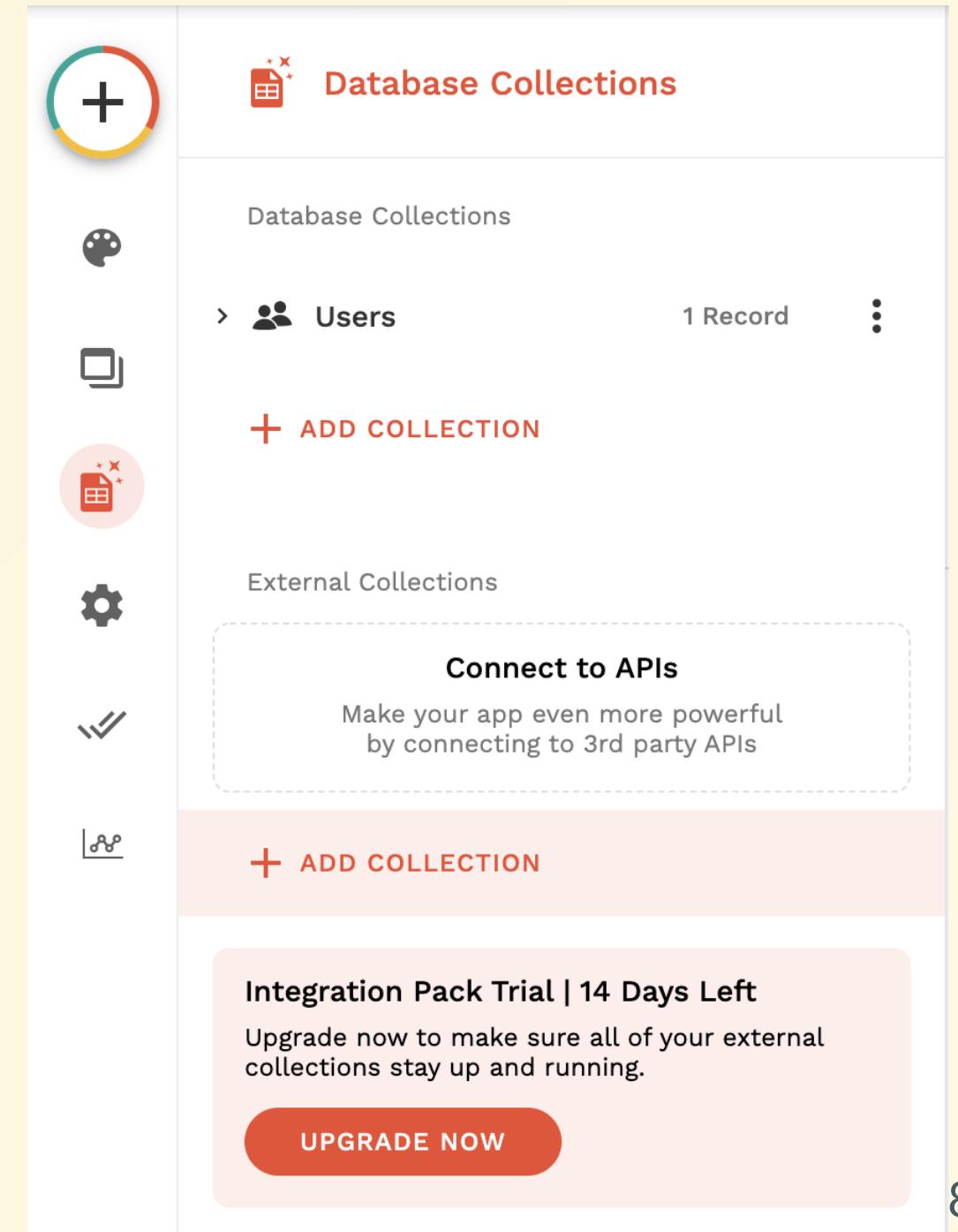
このAPIを使って、猫の品種の一覧を取得&表示しましょう。

<https://docs.thecatapi.com/api-reference/breeds/breeds-list#send-a-test-request>

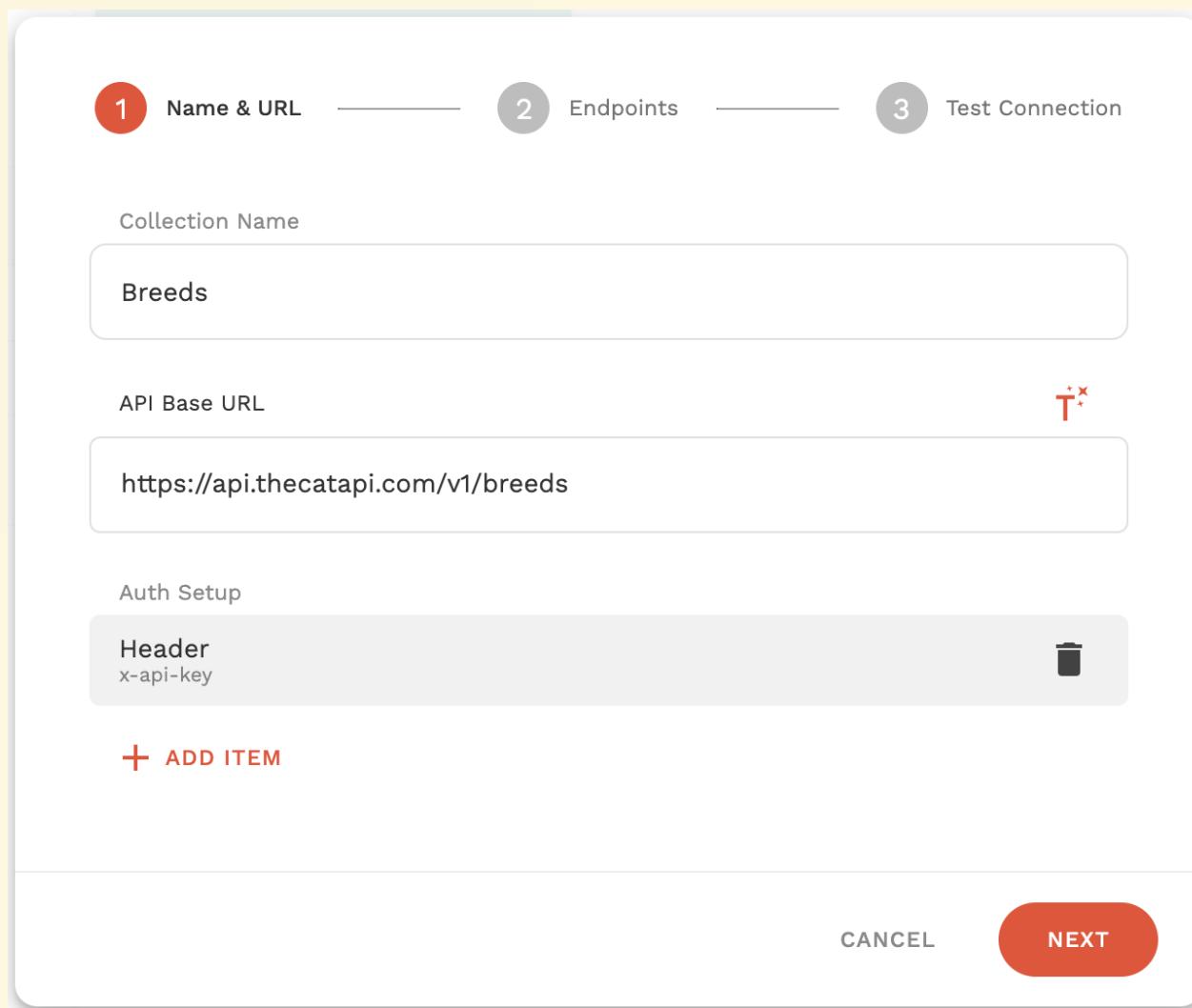
⌚ Send a Test Request

The screenshot shows the Postman interface for sending a test request. At the top, there is an orange 'Send' button, a 'GET' method indicator, and the URL `https://api.thecatapi.com/v1/breeds`. Below the URL, there are tabs for 'Settings', 'Headers [1]', 'Query [3]', and 'Code Generation'. The 'Headers [1]' tab is selected. Under 'Headers', there is a button 'Add Header' and a key-value pair 'x-api-key : {\$\$.env.x-api-key}'. The entire interface is set against a yellow background.

- DatabaseのExternal Collectionsで "ADD COLLECTION" をクリック



- Collection Name: Breeds
- Base URL:
<https://api.thecatapi.com/v1/breeds>
- Method: GET
- Auth Setup
 - Header x-api-key: 発行したAPI Key



The screenshot shows a user interface for setting up a new API collection. At the top, there are three tabs: 1. Name & URL (highlighted in red), 2. Endpoints, and 3. Test Connection. The 'Collection Name' field contains 'Breeds'. The 'API Base URL' field contains 'https://api.thecatapi.com/v1/breeds'. Under 'Auth Setup', there is a single item: 'Header x-api-key'. A red '+ ADD ITEM' button is visible below the auth setup section. At the bottom right are 'CANCEL' and 'NEXT' buttons.

1 Name & URL 2 Endpoints 3 Test Connection

Collection Name
Breeds

API Base URL
https://api.thecatapi.com/v1/breeds

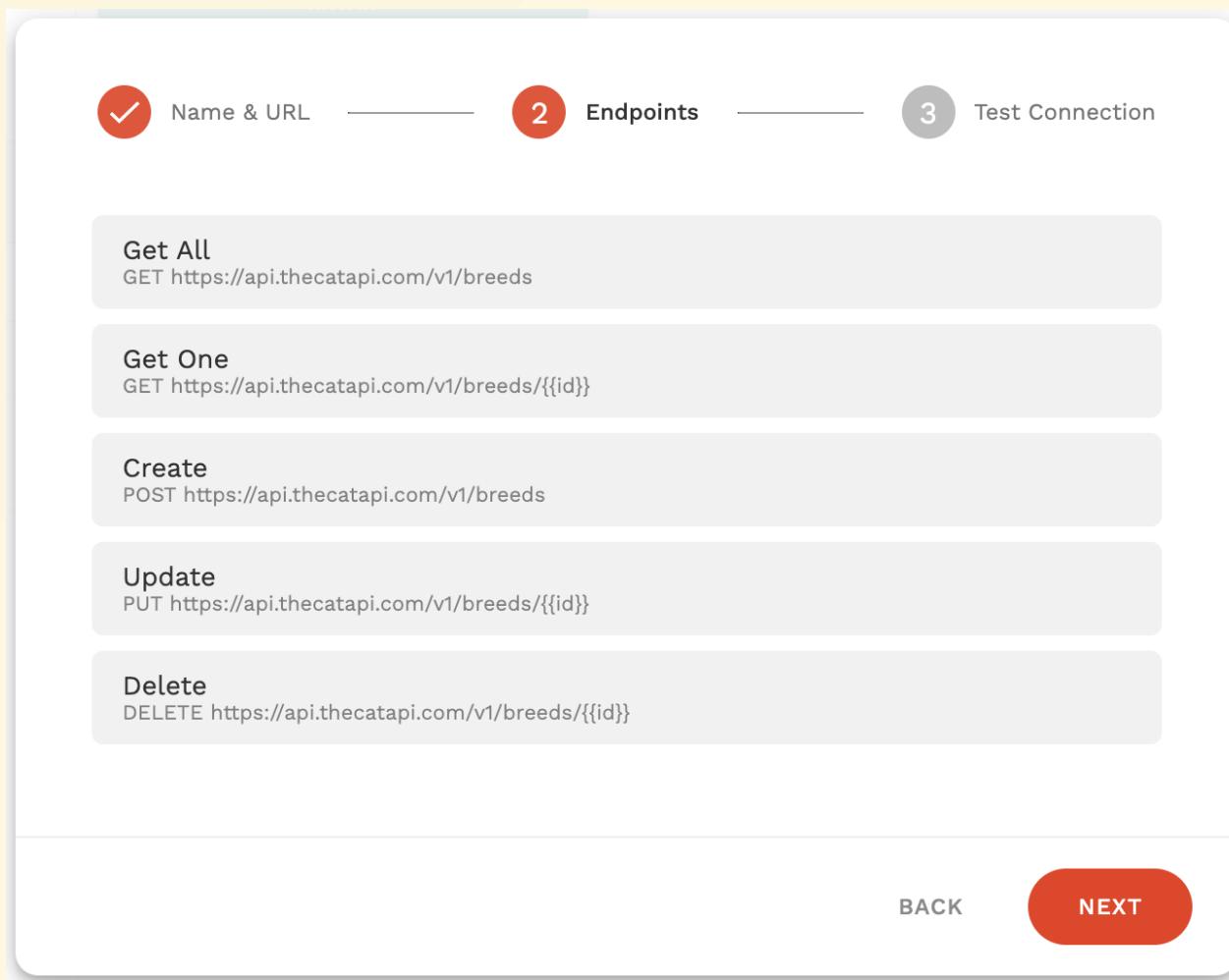
Auth Setup
Header x-api-key

+ ADD ITEM

CANCEL NEXT

Adaloでは、APIでアクセスするリソース(この例ではbreeds)毎に5つのEndpoints(アクセス方法)が設定できます。

APIの仕様によってはそれに合わせるための修正が必要になりますが、今回はそのままNEXTをクリックしてOKです。



- テストを実行して成功したら、"CREATE COLLECTION" をクリック

The screenshot shows a user interface for managing a MongoDB collection. At the top, there are three status indicators: "Name & URL" (green checkmark), "Endpoints" (green checkmark), and "Test Connection" (red circle with the number 3). Below these, the text "Test Successful!" is displayed in bold, followed by the message "Here's the full response we got back:". A JSON document is shown, representing a single document from the collection. The document contains fields such as "weight", "id", "name", "cfa_url", "vetstreet_url", "vcahospitals_url", "temperament", "origin", "country_codes", "country_code", "description", "life_span", "indoor", "lap", "alt_names", "adaptability", "affection_level", "child_friendly", "dog_friendly", "energy_level", "grooming", "health_issues", "intelligence", "shedding_level", "social_needs", "stranger_friendly", "vocalisation", and "experimental". The "name" field is set to "Abyssinian". The "cfa_url" field contains the URL "http://cfa.org/Breeds/BreedsAB/Abyssinian.aspx". The "vetstreet_url" field contains "http://www.vetstreet.com/cats/abyssinian". The "vcahospitals_url" field contains "https://vcahospitals.com/know-your-pet/cat-breeds". The "temperament" field is a comma-separated list of traits: "Active, Energetic, Independent, Intelligent, Gentle". The "origin" field is "Egypt". The "country_codes" and "country_code" fields are both "EG". The "description" field is a detailed sentence about the Abyssinian cat breed. The "life_span" field is "14 - 15". The "indoor" field is 0. The "lap" field is 1. The "alt_names" field is an empty string. The "adaptability" field is 5. The "affection_level" field is 5. The "child_friendly" field is 3. The "dog_friendly" field is 4. The "energy_level" field is 5. The "grooming" field is 1. The "health_issues" field is 2. The "intelligence" field is 5. The "shedding_level" field is 2. The "social_needs" field is 5. The "stranger_friendly" field is 5. The "vocalisation" field is 1. The "experimental" field is 0.

BACK CREATE COLLECTION

External Collectionが作成されました。

APIから取得するデータが全てプロパティとして設定されています。

External Collections

▼  Breeds 

A id

A weight > imperial

A weight > metric

A name

A temperament

A origin

A country_codes

A country_code

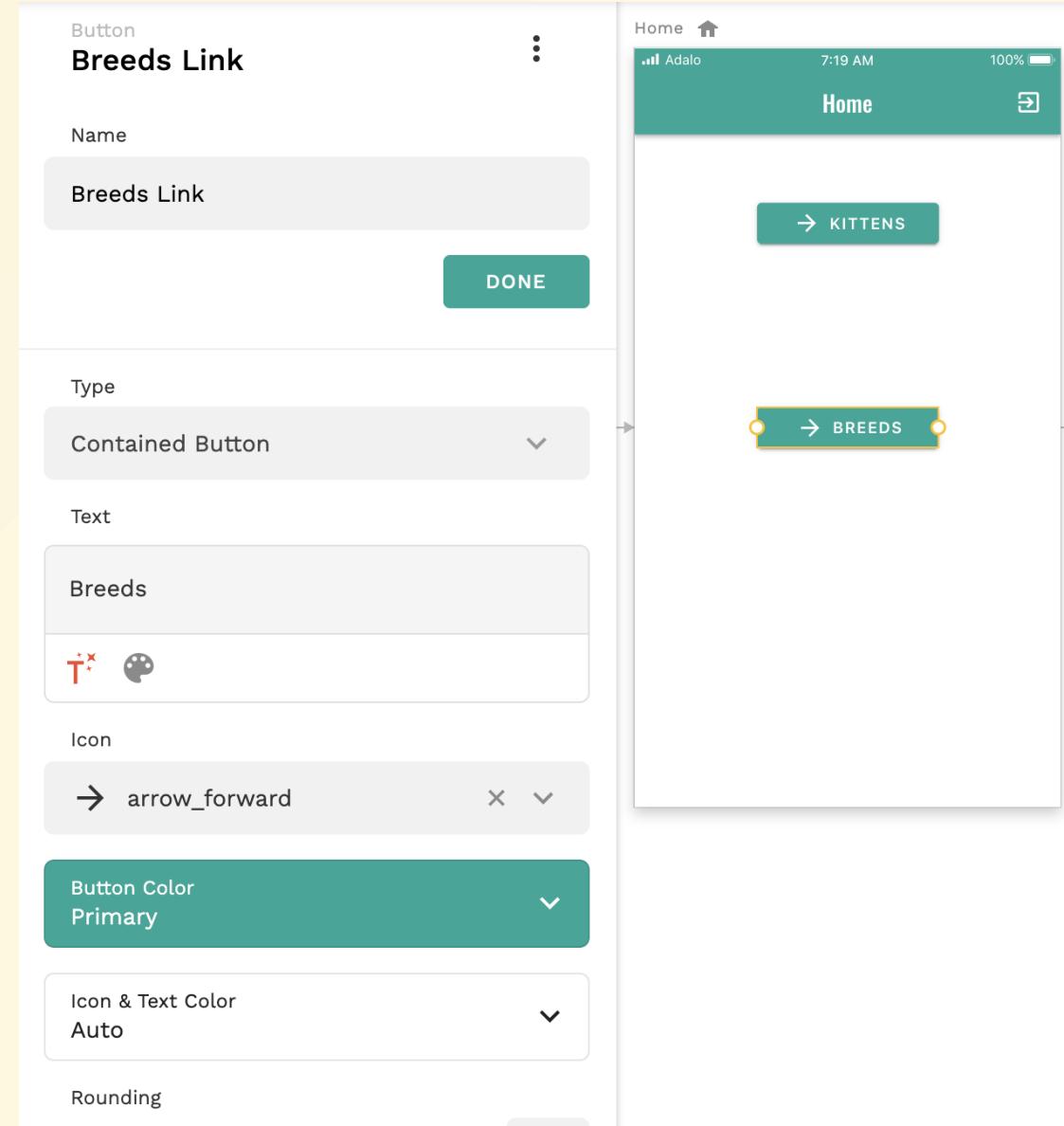
A description

A life_span

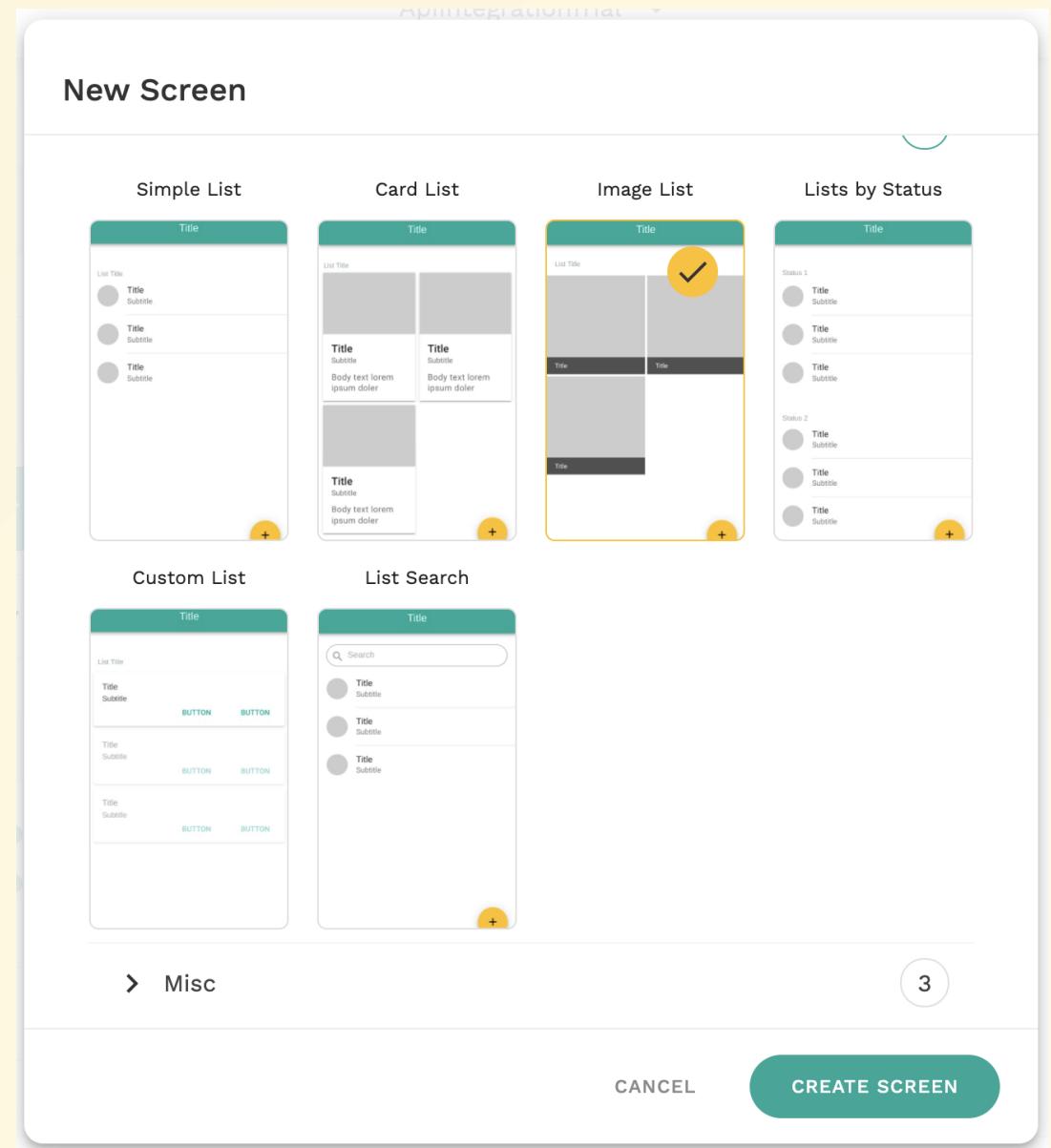
 indoor

取得したデータを一覧表示しましょう。

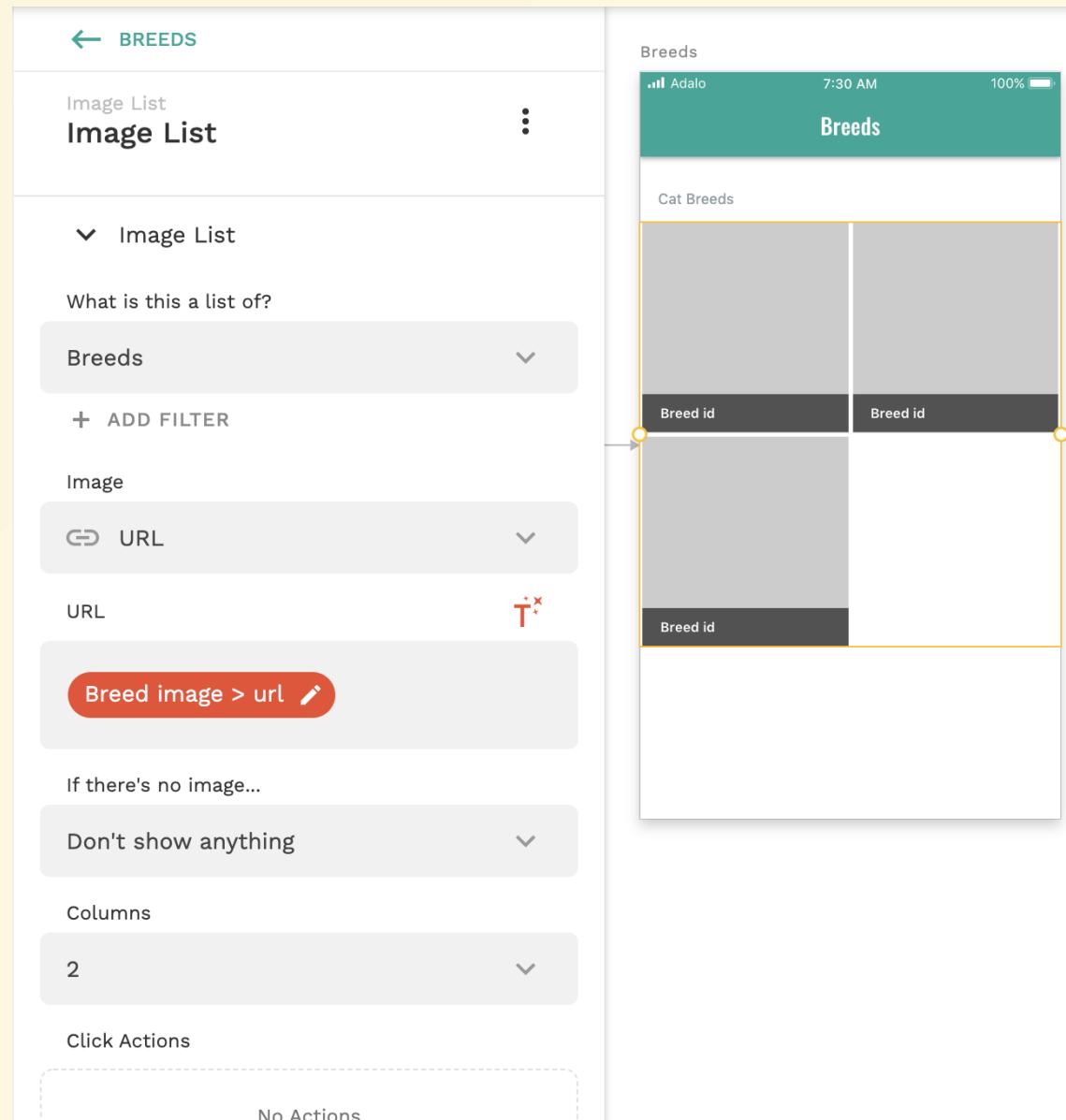
- Home画面に "Breeds Link" ボタンを追加
- ADD ACTIONからNEW SCREENへのLINKを追加



- Nameに "Breeds"と入力
- TemplateでImage Listを選択
- CREATE SCREEN をクリック



- List TitleのTextを"Cat Breeds"に変更
- Image ListをBreeds Collectionのリストとして設定
- ImageのURLに "Breed Image > url" を設定
- "If there's no image..." に "Don't show anything" を設定
- 右下の+ボタンは不要なので削除



- Textを "Breed name" に変更

The image shows a mobile application interface with two main sections: a configuration screen on the left and a preview screen on the right.

Configuration Screen (Left):

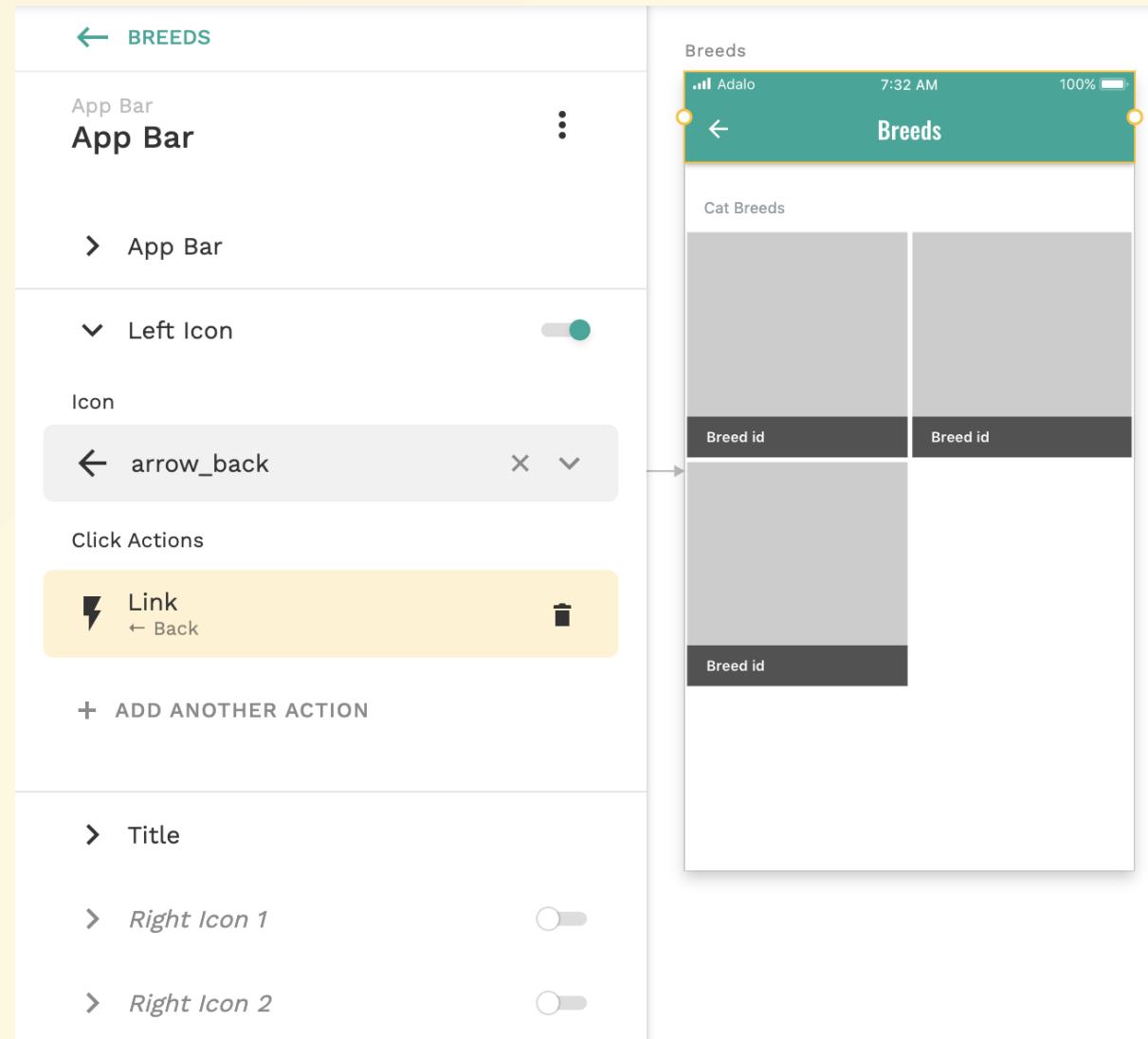
- List Header:** A toggle switch is turned off.
- Text:** A toggle switch is turned on. Below it is a **Title** field containing the text "Breed name" with a red edit icon. To the right of the title are icons for bold, italic, and underline.
- Subtitle:** A text input field with the placeholder "Enter Text..." and a red edit icon below it.
- Subtitle Position:** A dropdown menu with the placeholder "Select..." and a downward arrow icon.
- Text Position:** A dropdown menu with the placeholder "Select..." and a downward arrow icon.
- Background Effect:** A dropdown menu with the placeholder "Select..." and a downward arrow icon.
- Image Style:** A section indicated by a chevron icon.

Preview Screen (Right):

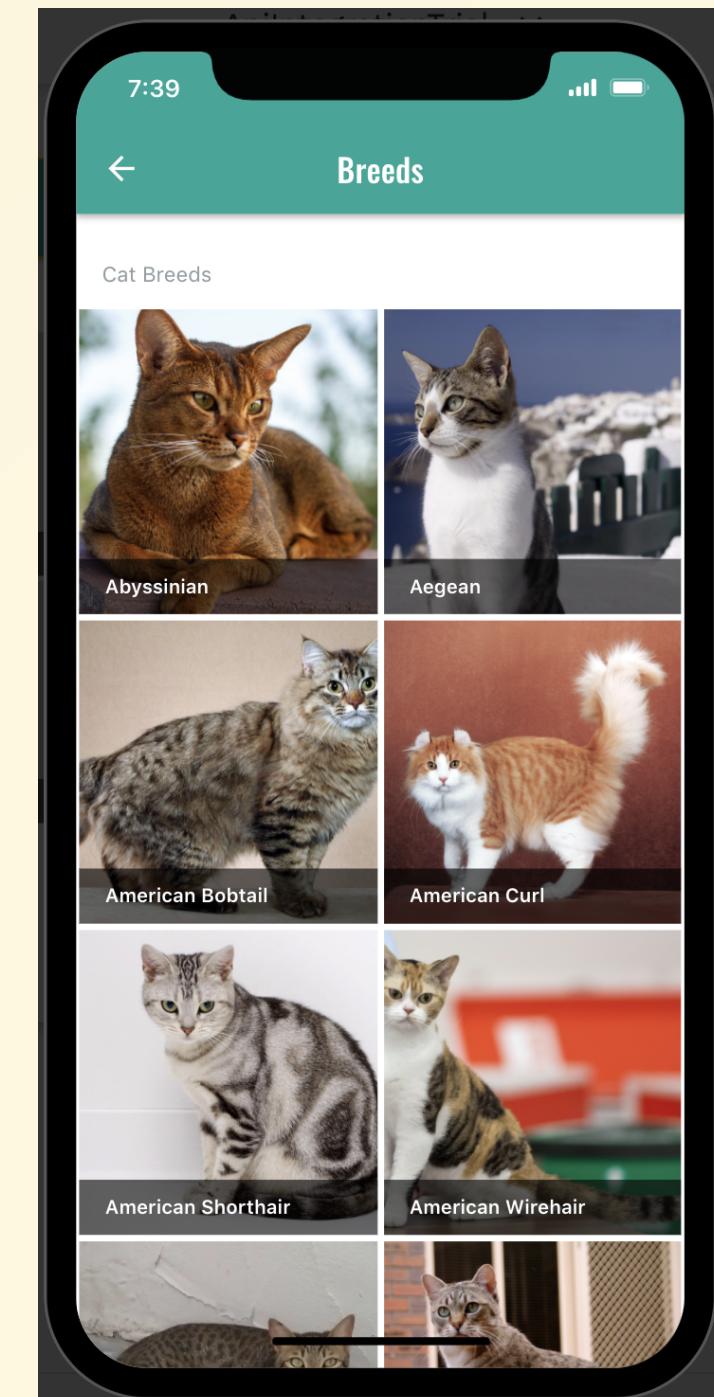
- Header:** Shows "Breed" and "Breeds" in a teal header bar.
- Section:** "Cat Breeds" is displayed.
- Content:** A grid of three cards, each labeled "Breed name". The first card is highlighted with a yellow border and a yellow selection handle at the bottom-right corner.

戻るアイコンが非表示になつ
ているので、表示します。

- AppBarをクリック
- Left IconのトグルをONに変更



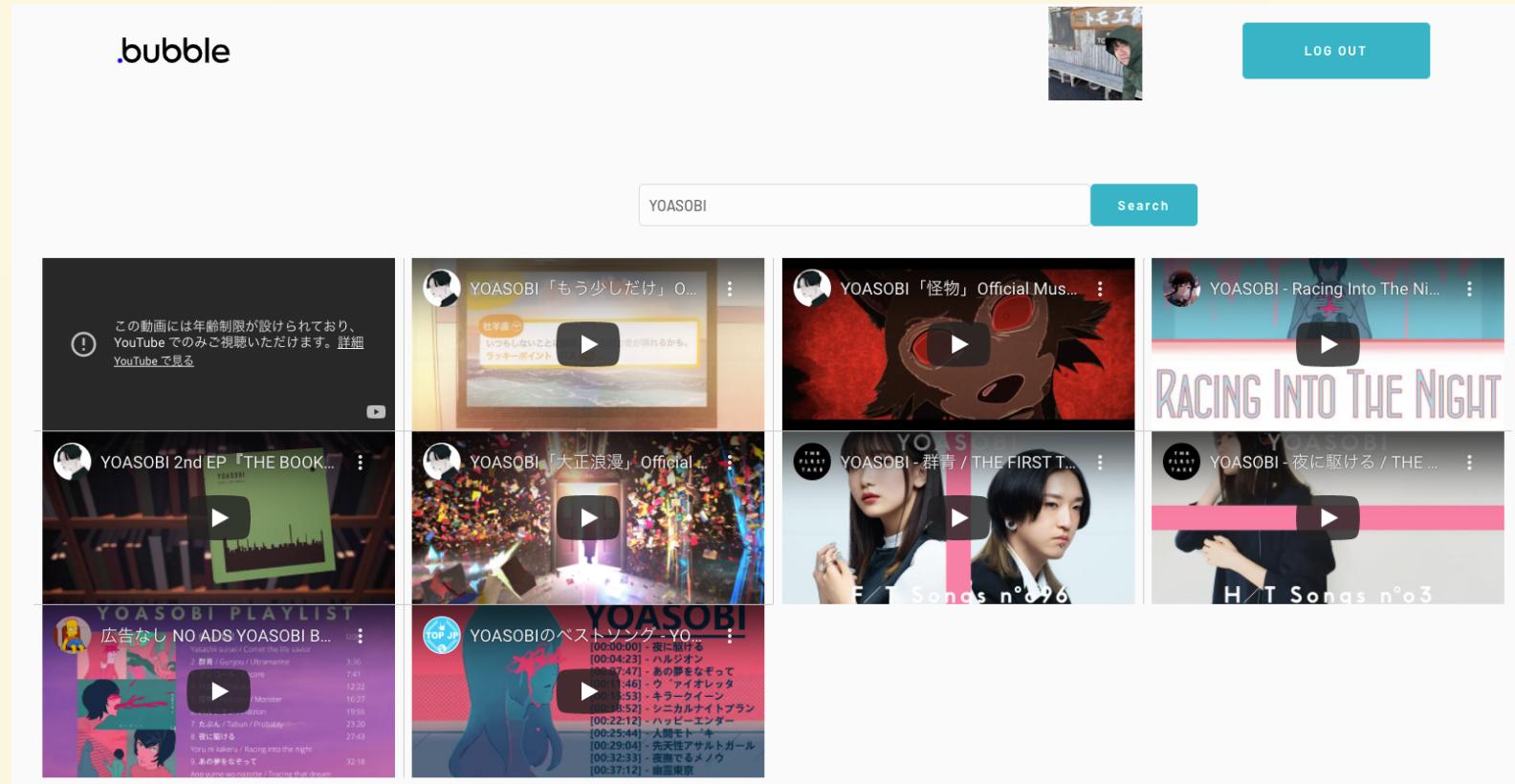
Preview機能で確認すると、猫の品種の一覧が表示されました。



Adaloでの外部連携についての紹介は以上です。

Bubble での外部連携

つづいて Bubble での外部連携を試していきます。
今回作る画面はこんな感じです。



機能としては下記を盛り込んだ画面となります

- YouTube の動画一覧を表示
- Google アカウントを使ったソーシャルログイン
- YouTube の動画一覧を動的に検索

まずは土台となるアプリケーションを新たに用意

- Advanced-bubble-{ご自身の名前} というアプリを新たに作成

Create a new app

X

Name of this new app

Advanced-bubble-kyogoku

Start from a template (optional)

You can add a domain name later.

What kind of app are you building?

Other

Details of what you're building

Advanced bubble

Is it customer-facing or internal?

Internal or employee-facing

Try to be specific so that we can help!

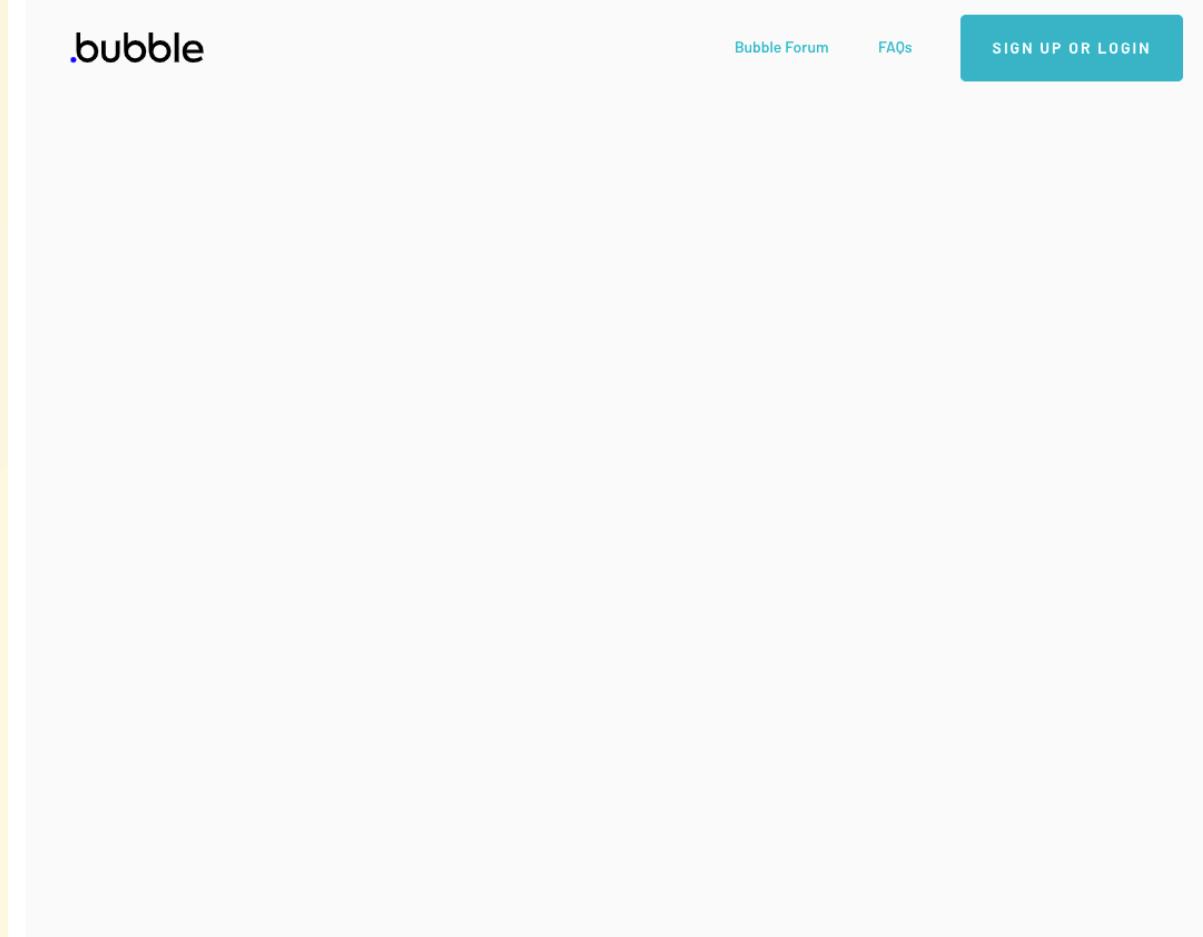
What's your goal with this application?

Other

Need help getting started? [Hire someone](#) to build your app for you or ask other users on the [forum](#)!

Create a new app

- 第3回の時と同様、ヘッダーを除いてすべての要素を削除してください
 - 一度すべて削除した後にヘッダーコンポーネントを再配置でもOK

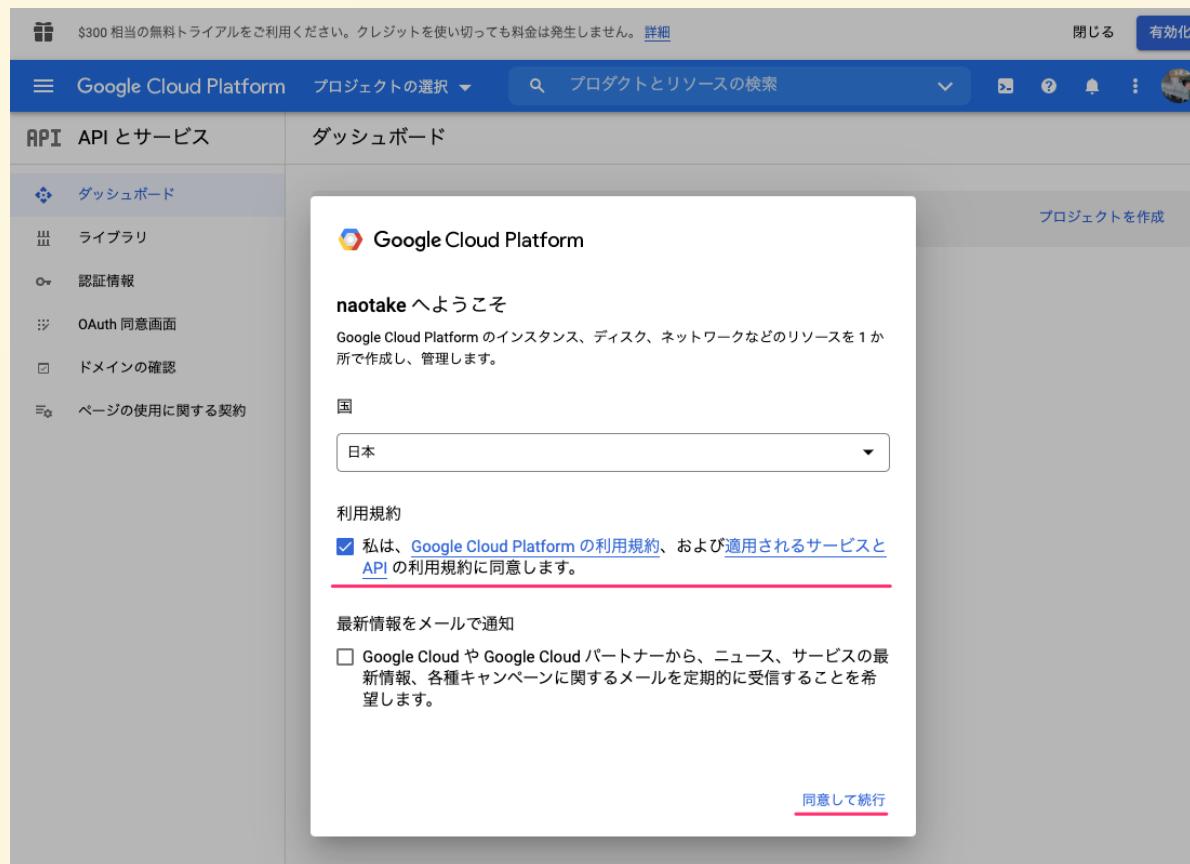


続いて Google アカウント側の事前準備

- YouTube の動画一覧を表示するために必要な下準備を行います
- Google アカウントにログインしていただき、下記 URL にアクセスします

<https://console.developers.google.com/>

- はじめて Google Cloud Platform(GCP) の画面にアクセスするところのようなダイアログが表示されますので、国を選択し、利用規約にチェックを入れて「同意して続行」



- 同意すると、API とサービスのダッシュボードが表示されると思いますので、プロジェクトを作成のリンクを押下して今回の講義用のプロジェクト（箱）を作成します

The screenshot shows the Google Cloud Platform API & Services dashboard. On the left, there is a sidebar with icons for Dashboard, Library, Authentication, OAuth Consent Screen, Domain Verification, and Terms of Use. The main area is titled "ダッシュボード" (Dashboard). A message at the top states: "\$300相当の無料トライアルをご利用ください。クレジットを使い切っても料金は発生しません。 詳細" (A \$300 equivalent free trial is available. Even if you run out of credit, no fees will be charged.) with "閉じる" (Close) and "有効化" (Enable) buttons. Below the message, there is a search bar and a navigation bar with icons for Home, Help, Notifications, and Profile. A prominent red box highlights the "プロジェクトを作成" (Create Project) button, which is located in a callout box that says: "このページを表示するには、プロジェクトを選択してください。" (To view this page, please select a project.)

- 新しいプロジェクトの作成
画面が表示されるのでプロジェクト名に適当な名前を指定して作成

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 12 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 * [?](#)

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

場所 * [参照](#)

親組織またはフォルダ

[作成](#) [キャンセル](#)



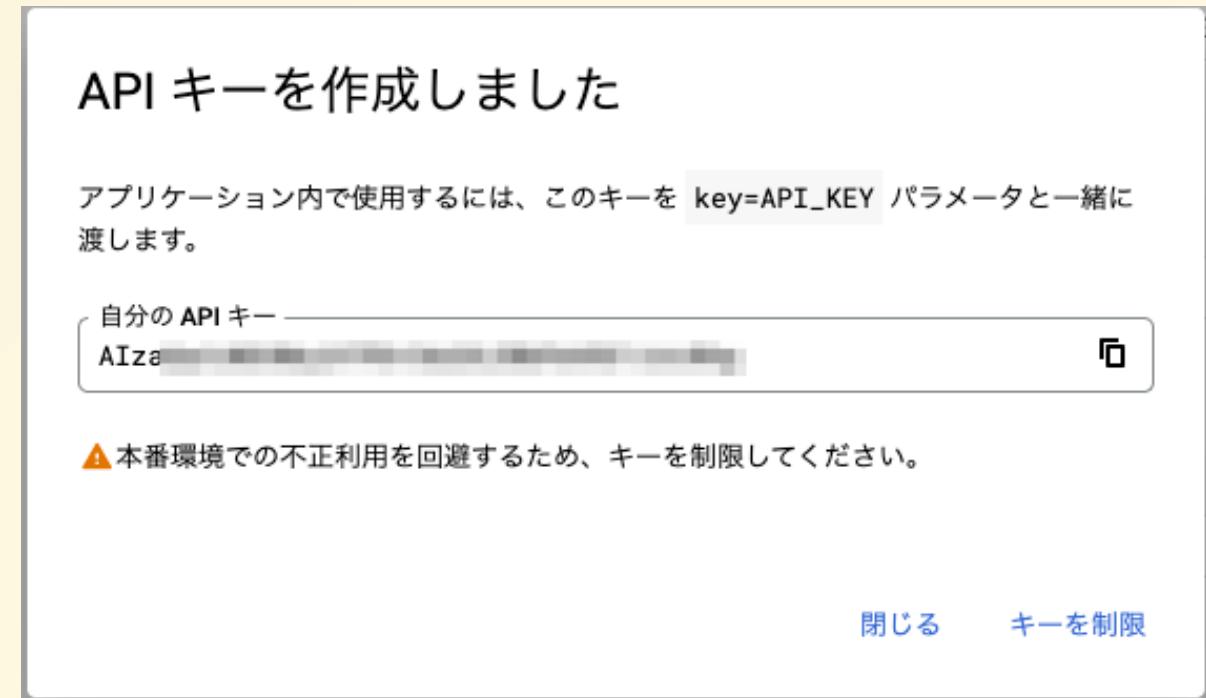
- 作成が完了すると、作成したプロジェクトのダッシュボードが表示されます
 - 画面上部の Google Cloud Platform の左隣に作成したプロジェクト名が表示されていれば OK

The screenshot shows the Google Cloud Platform API & Services dashboard. The left sidebar lists several options: ダッシュボード, ライブラリ, 認証情報 (which is selected and highlighted in blue), OAuth 同意画面, ドメインの確認, and ページの使用に関する契約. The main content area has tabs for 認証情報 (Authentication) and API キー (API Keys). The Authentication tab displays a message: '有効な API にアクセスするための認証情報を作成します。 [詳細]'. Below this, there's a table for API Keys with one entry: a key labeled 'bubble advance' with a creation date of '2024-01-11'. The API Keys table includes columns for '名前' (Name), '作成日' (Created At), '制限' (Limits), and 'キー' (Key). The OAuth 2.0 Client ID tab shows two entries: 'bubble advance' and 'bubble advance'. The Services Accounts tab shows a single entry for 'メール' (Email) with the message '表示するサービス アカウントがありません' (No service accounts are displayed).

- 続いて YouTube の動画一覧を表示するために必要な認証キーを発行します
- 左メニューから「認証情報」を選択
- 右パネルが認証情報に変わるので、画面上部の「+ 認証情報を作成」をクリック
- ポップアップから「API キー」を選択

The screenshot shows the Google Cloud Platform interface. The top navigation bar includes the Google Cloud logo, a search bar, and various icons. The main menu on the left has 'API & Services' selected. Under 'API & Services', '認証情報' (Authentication) is highlighted. A modal window is open over the main content area. The modal title is '+ 認証情報を作成' (Create Authentication Information). It contains two tabs: 'API キー' (API Key) which is selected, and 'OAuth クライアント ID'. The 'API キー' tab has a warning message: '必ず、API キーを生成してください。' (Please generate an API key.) Below it are sections for 'API キー' (API Key), 'OAuth クライアント ID' (OAuth Client ID), and 'サービス アカウント' (Service Account). At the bottom of the modal is a '同意画面を設定' (Set Consent Screen) button. The main content area below the modal shows a table for 'OAuth 2.0 クライアント ID' (OAuth 2.0 Client ID) with one row: '表示する OAuth クライアントがありません' (No OAuth client is displayed).

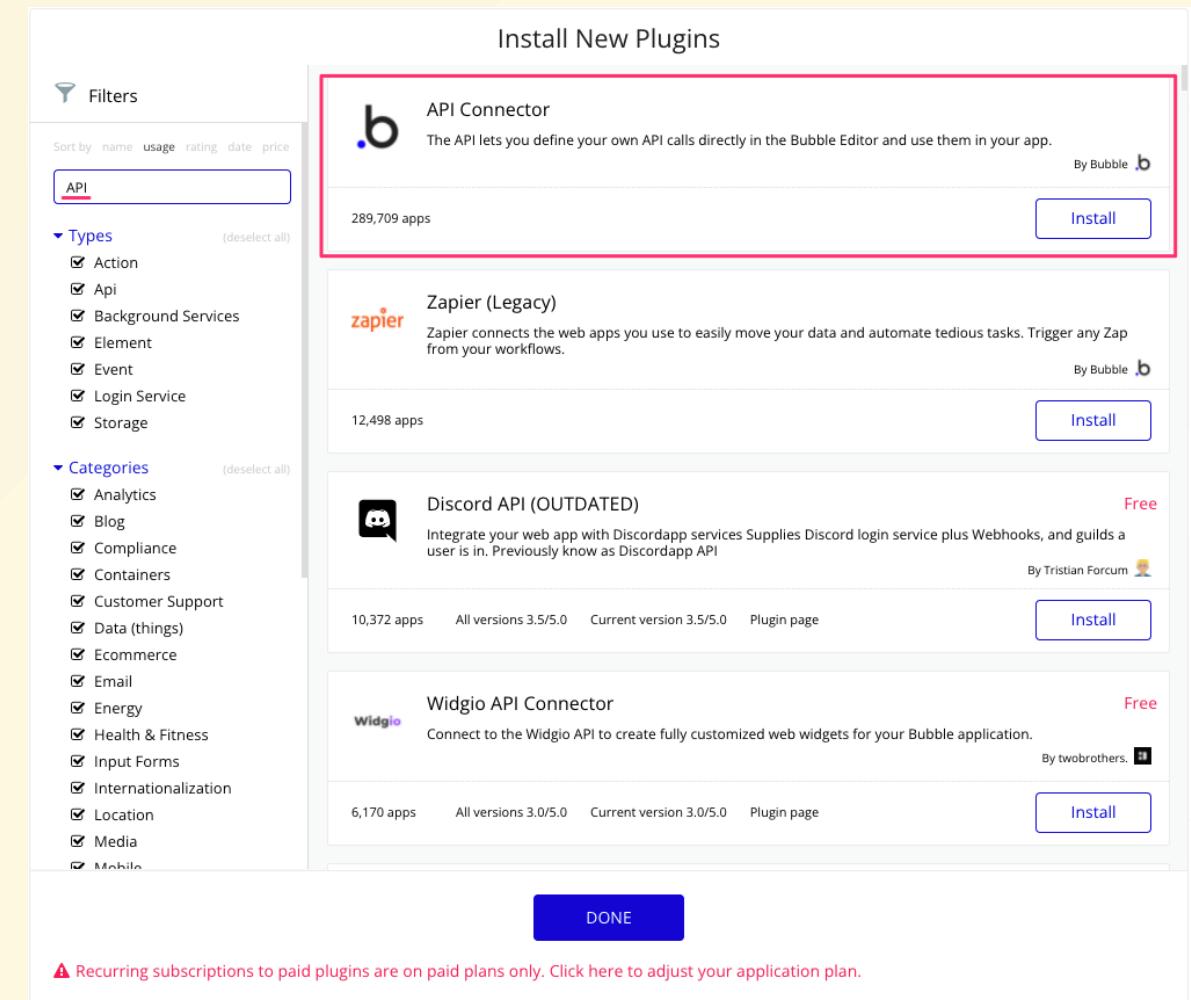
- すると API キー作成完了の ポップアップが表示される ので、その値を控えておき ます
- キーを制限しろと言われま すが、講義の最後にこれら のキーを削除しますので一 旦このまま行きます



- これで Bubble から YouTube の一覧を取得するためのキーが発行できたので Bubble 側の設定を行っていきます
- まずは Bubble から YouTube の一覧を API を介して取得するための準備を行います

Bubble の API 連携の準備

- 左メニューから Plugins を選択し、Add plugins ボタンをクリック
- API で検索し、**API Connector** というプラグインをインストールします



- このプラグインは Bubble のアプリから、世の中で公開されている API を使って、API の向こう側の情報と Bubble を繋げるためのプラグインとなります
- いくつかの Bubble のプラグインの中には、特定の API に特化させたものもあり、それらのプラグインはこの API Connector よりも設定が単純なものもあります
- 今回は一番の基本形となる API Connector プラグインを介して YouTube API を使い、動画の一覧を表示してみましょう

- それでは YouTube API を使うための設定を行っていきます
- まず "Add another API" をクリックし、新しい API 連携の設定を開始します

The screenshot shows the "API Connector" settings page. At the top left is the ".b" logo and the text "API Connector". To the right is a "reference →" link. Below the title, there is descriptive text: "The API lets you define your own API calls directly in the Bubble Editor and use them in your app. A good practice is to get the calls to work in Postman first, and then add them here. You can also convert your connections into plugins that can be used in more than one app, and even shared with other Bubblers! See [your personal plugins page](#) for more details." At the bottom left is a blue button labeled "Add another API" with a red rectangular border around it. At the bottom right is a blue button labeled "Uninstall this plugin".

.b API Connector

reference →

The API lets you define your own API calls directly in the Bubble Editor and use them in your app.

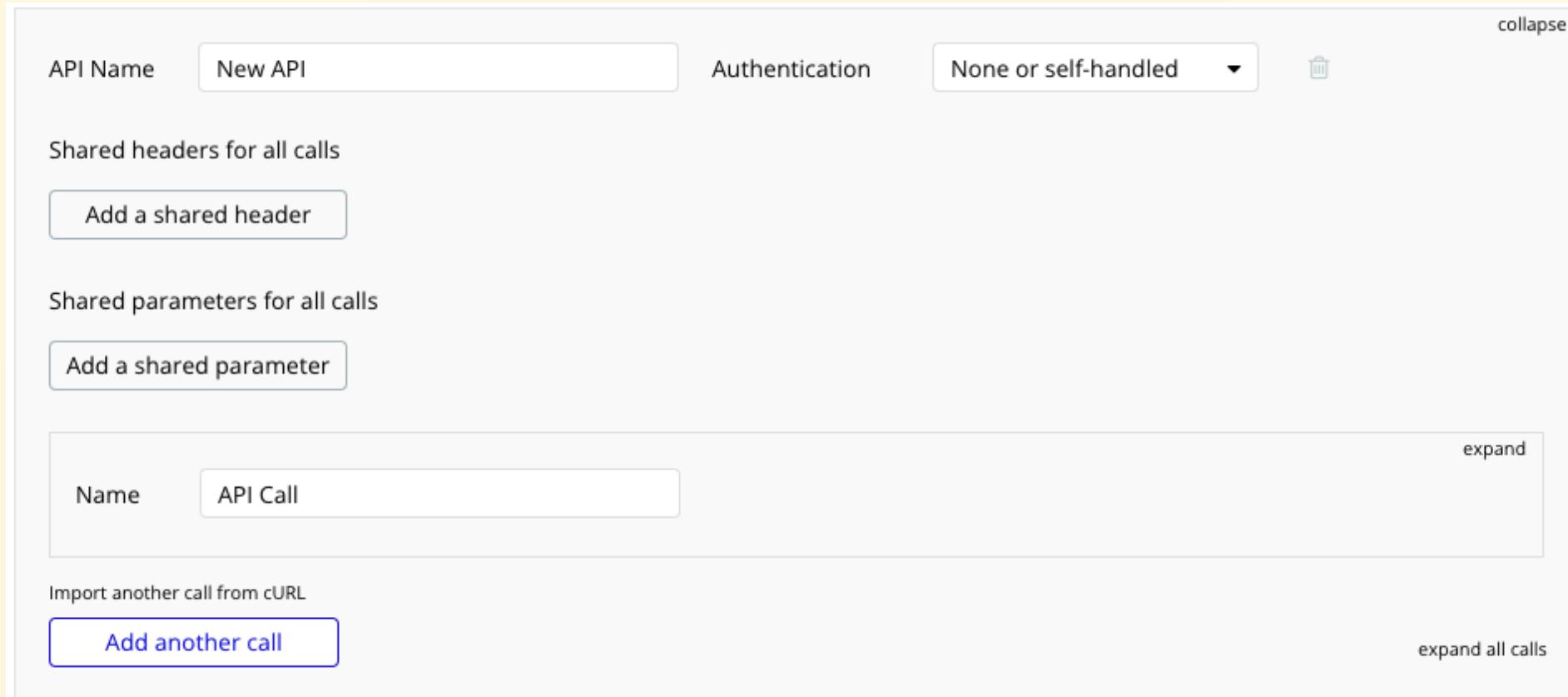
A good practice is to get the calls to work in Postman first, and then add them here.

You can also convert your connections into plugins that can be used in more than one app, and even shared with other Bubblers! See [your personal plugins page](#) for more details.

Add another API

Uninstall this plugin

- "Add another API" をクリックすると、このような API 設定部品が表示されると思います



- 各項目を簡単に説明します
- 1と2については合わせて設定を行ってください

1. API Name: この API 連携のグループ名

- 今回は YouTube

The screenshot shows the Zapier API configuration interface. At the top, there are fields for 'API Name' (set to 'New API'), 'Authentication' (set to 'None or self-handled'), and a 'collapse' button. Below these are sections for 'Shared headers for all calls' (with a 'Add a shared header' button) and 'Shared parameters for all calls' (with a 'Add a shared parameter' button). A large section at the bottom is titled 'Name' and contains a 'API Call' field, with an 'expand' button to its right. At the very bottom, there's a link to 'Import another call from cURL' and a blue 'Add another call' button, along with an 'expand all calls' button.

① API Name New API

② Authentication None or self-handled

③ Shared headers for all calls

Add a shared header

④ Shared parameters for all calls

Add a shared parameter

Name API Call

Import another call from cURL

Add another call

expand

expand all calls

2. Authentication: このグループ配下の API で行う共通の認証方式

- 今回は "None or self-handled"

The screenshot shows the configuration interface for a new API. At the top, there are fields for 'API Name' (set to 'New API') and 'Authentication' (set to 'None or self-handled'). Below these, there are sections for 'Shared headers for all calls' and 'Shared parameters for all calls'. Each section has a 'Add a shared header' or 'Add a shared parameter' button respectively. A large 'expand' button is located at the bottom right of the main configuration area. At the very bottom, there is a link to import a call from cURL and a blue 'Add another call' button.

① API Name New API

② Authentication None or self-handled

③ Shared headers for all calls

Add a shared header

④ Shared parameters for all calls

Add a shared parameter

Name API Call

Import another call from cURL

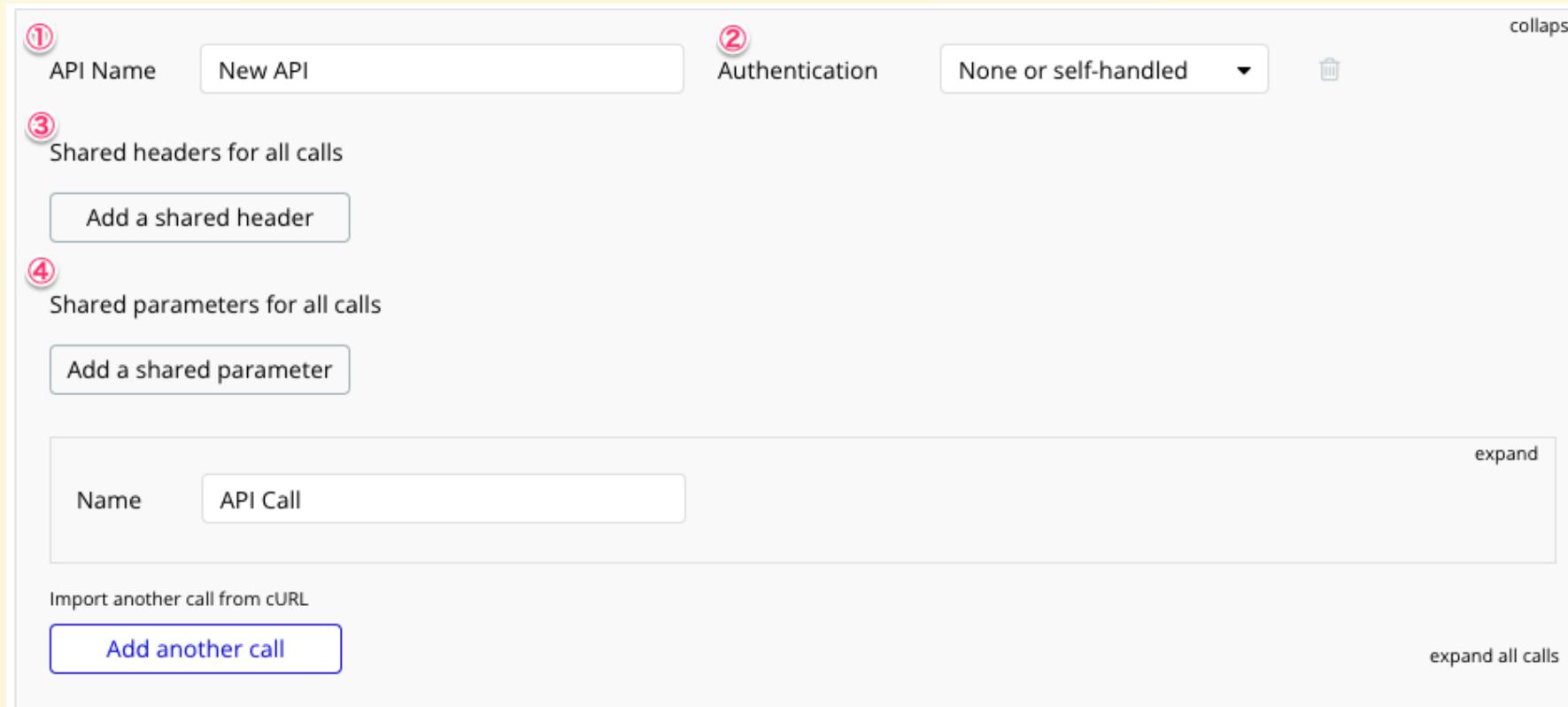
Add another call

expand

expand all calls

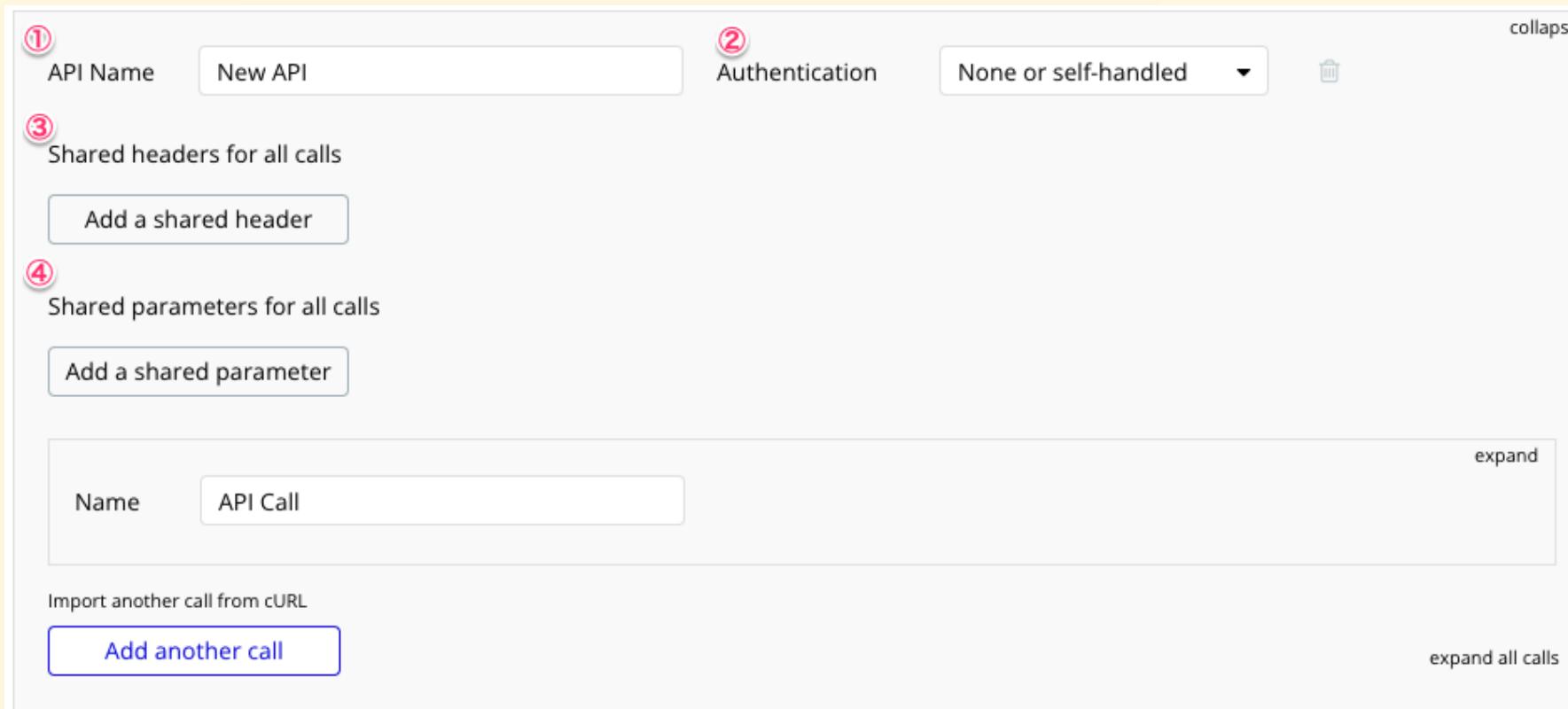
3. Shared headers for all calls: このグループ配下の API すべてで指定する共通のヘッダー情報

- 今回は特に指定しません

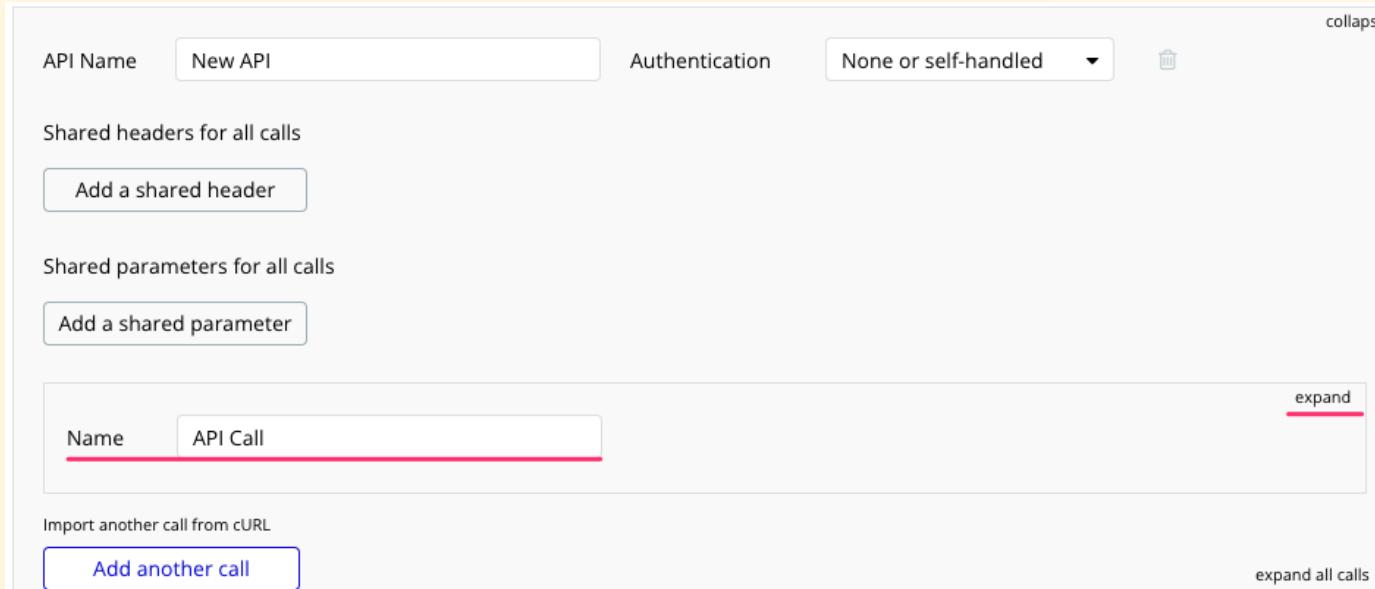


4. Shared parameters for all calls: このグループ配下の API すべてで指定する共通のパラメータ情報

- 今回は特に指定しません



- 続いて、このグループに対して具体的な API の内容を設定していきます
- 先ほどの画面で "API Call" となっているブロックの右側に expand というリンクがあるのでそれをクリック
- すると、具体的な API の設定項目が表示されますので、簡単に説明します



1. Name: 具体的な API の名前

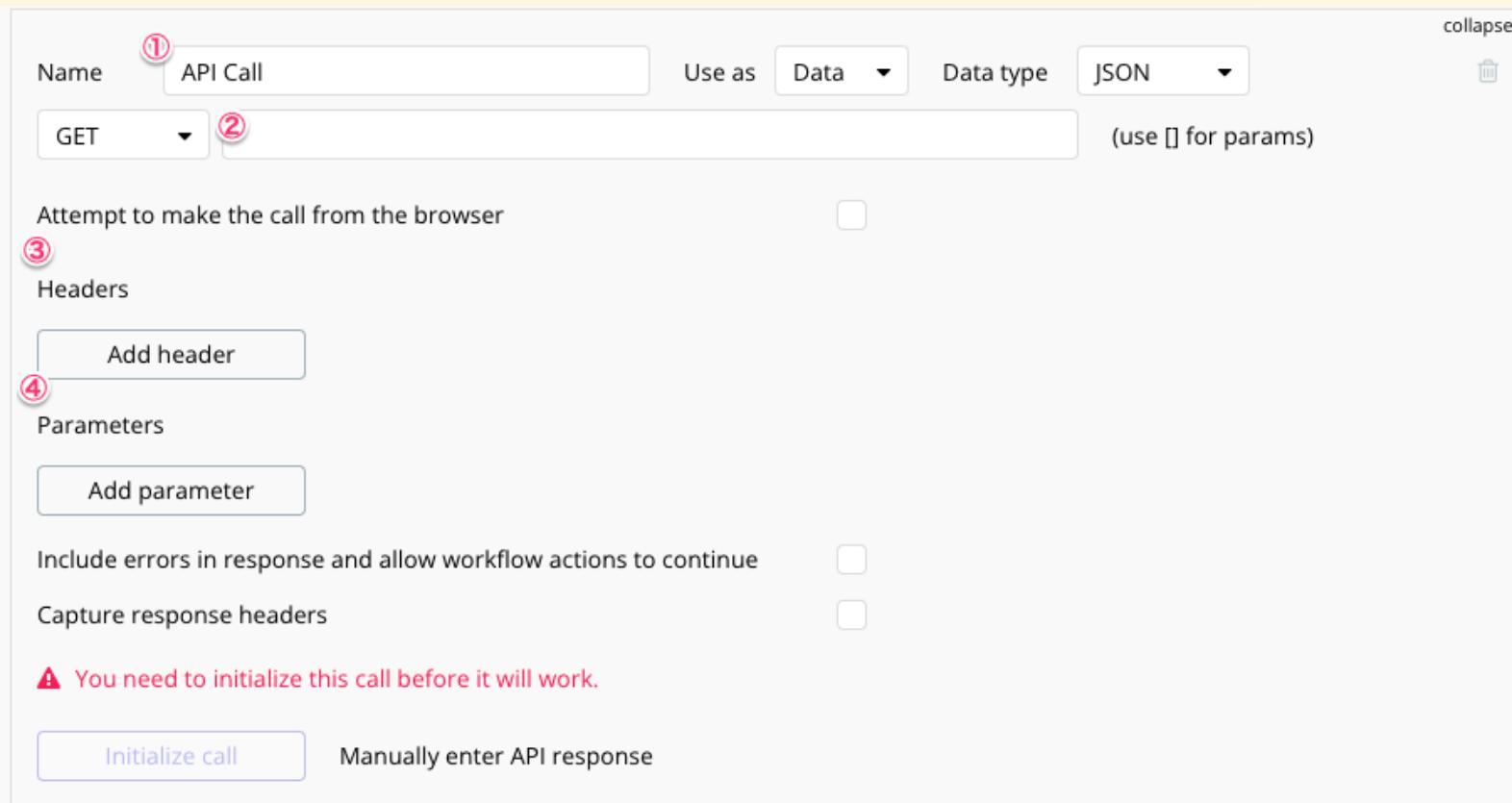
- 今回は "search" と入力します

The screenshot shows a configuration interface for an API call. At the top, there's a 'Name' field containing 'API Call' with a circled '1' above it. To its right are dropdowns for 'Use as' (set to 'Data') and 'Data type' (set to 'JSON'). A 'collapse' button and a trash bin icon are also present. Below this, a 'Method' dropdown is set to 'GET' with a circled '2' above it, and a note '(use [] for params)' is shown. The main area contains sections for 'Headers' (with a circled '3' above an 'Add header' button) and 'Parameters' (with a circled '4' above an 'Add parameter' button). There are also checkboxes for 'Attempt to make the call from the browser', 'Include errors in response and allow workflow actions to continue', and 'Capture response headers'. A warning message at the bottom left states '⚠ You need to initialize this call before it will work.' with buttons for 'Initialize call' and 'Manually enter API response'.

Name API Call
Use as Data Data type JSON
collapse
GET (use [] for params)
Attempt to make the call from the browser
Headers
Parameters
Include errors in response and allow workflow actions to continue
Capture response headers
⚠ You need to initialize this call before it will work.
Initialize call Manually enter API response

2. Path: 具体的な API の URL

- 今回は `https://www.googleapis.com/youtube/v3/search` を入力します



3. Headers: この API 固有のヘッダー情報

- 今回は特に指定しません

The screenshot shows the configuration of an API Call node in a workflow editor. The node has the following settings:

- Name:** API Call (marked with ①)
- Method:** GET (marked with ②)
- Use as:** Data
- Data type:** JSON
- Headers:** An "Add header" button (marked with ③).
- Parameters:** An "Add parameter" button (marked with ④).
- Attempt to make the call from the browser:** A checkbox that is unchecked.
- Include errors in response and allow workflow actions to continue:** A checkbox that is unchecked.
- Capture response headers:** A checkbox that is unchecked.

A red warning message at the bottom left states: **⚠ You need to initialize this call before it will work.**

At the bottom, there are two buttons: **Initialize call** and **Manually enter API response**.

4. Parameters: この API 固有のパラメータ情報

- 今回は下記 2 つのパラメータを指定します。Private / Allow blank / Optional は添付の通り
 - 1. Key: **key**、 Value: 先ほど Google アカウント側で発行した API キー
 - 2. Key: **q**、 Value: YouTube で動作検索をする際のキーワード（ご自由に指定ください）

- すべて設定するとこんな感じです

collapse

Name Use as Data type

GET (use [] for params)

Headers

Parameters

Key <input type="text" value="key"/>	Value <input type="text" value="AlzaSyCiN"/>	Private <input checked="" type="checkbox"/>	Allow blank <input type="checkbox"/>	Optional <input type="checkbox"/>	<input type="button" value="Delete"/>
Key <input type="text" value="q"/>	Value <input type="text" value="おしゃべり唐揚げ"/>	Private <input checked="" type="checkbox"/>	Allow blank <input type="checkbox"/>	Optional <input type="checkbox"/>	<input type="button" value="Delete"/>

Include errors in response and allow workflow actions to continue

Capture response headers

A You need to initialize this call before it will work.

Manually enter API response

- 設定が終わったら、設定内容が正しいかを確認します
- 下部にある "Initialize call" ボタンをクリック

- おそらく Status code 403 のエラーメッセージが返ってきたはずです
- 内容は書いてある通りですが、要約するとこんな感じです

YouTube Data API v3 はあなたのプロジェクトで無効になっています。
この URL にアクセスして有効にしてから再試行してください。

API Connector

There was an issue setting up your call.

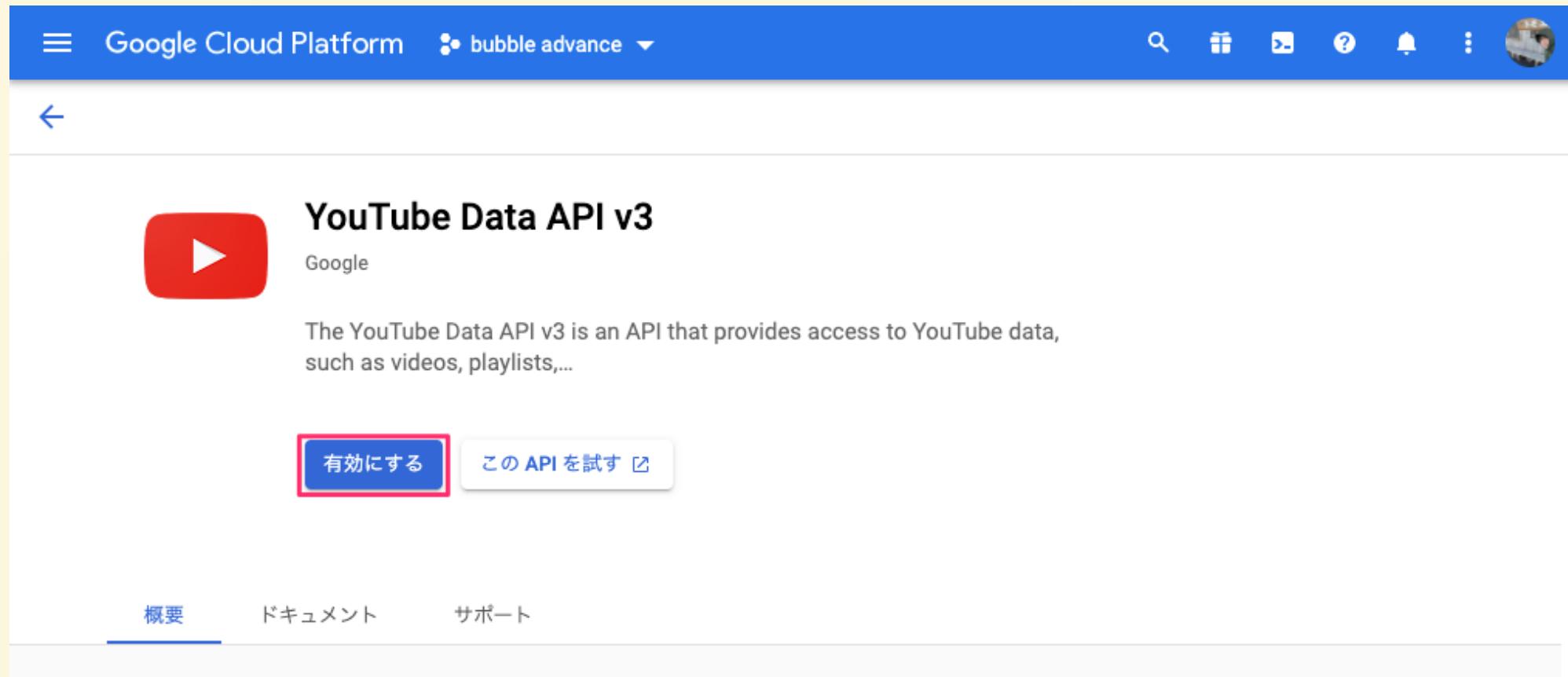
Raw response for the API

```
Status code 403
{
  "error": {
    "code": 403,
    "message": "YouTube Data API v3 has not been used in project 858502125299 before or it is disabled. Enable it by visiting https://console.developers.google.com/apis/api/youtube.googleapis.com/overview?project=858502125299 then retry. If you enabled this API recently, wait a few minutes for the action to propagate to our systems and retry.",
    "errors": [
      {
        "message": "YouTube Data API v3 has not been used in project 858502125299 before or it is disabled. Enable it by visiting https://console.developers.google.com/apis/api/youtube.googleapis.com/overview?project=858502125299 then retry. If you enabled this API recently, wait a few minutes for the action to propagate to our systems and retry."
      }
    ],
    "status": "PERMISSION_DENIED",
    "details": [
      {
        "@type": "type.googleapis.com/google.rpc.Help",
        "links": [
          {
            "description": "Google developers console API activation",
            "url": "https://console.developers.google.com/apis/api/youtube.googleapis.com/overview?project=858502125299"
          }
        ]
      },
      {
        "@type": "type.googleapis.com/google.rpc.ErrorInfo",
        "reason": "SERVICE_DISABLED"
      }
    ]
  }
}
```

- どうやら今回利用した YouTube Data API が各人の Google アカウント上で有効になっていないようです
- これは Google さんが全員が使わないであろう機能は最初から OFF にしていて、必要ある人だけ自分で ON にしてね、という状態になっています
- なので今回は YouTube Data API を使いたいので、メッセージに記載されている URL にアクセスして有効化しましょう

<https://console.developers.google.com/apis/api/youtube.googleapis.com/overview?project={各人のプロジェクトID}>

- すると親切に今回利用したい "YouTube Data API v3" の画面へ遷移したはずです
- 画面にあるとおり「有効にする」をクリックして有効にします



- しばらくするとこんな画面に遷移するはずです
- 画面上部の赤線部分が「API を無効にする」となっていれば有効化は成功です

The screenshot shows the Google Cloud Platform interface for managing APIs. The top navigation bar includes the Google Cloud logo, the project name 'bubble advance', a search bar, and various navigation icons. The main content area is for the 'YouTube Data API v3' service. On the left, there's a sidebar with tabs: '概要' (selected), '指標', '割り当て', and '認証情報'. The main panel has two sections: '詳細' (Details) and 'レスポンス コード別のトラフィック' (Traffic by Response Code). The '詳細' section shows the API name as 'YouTube Data API v3', provider as 'Google', service name as 'youtube.googleapis.com', and a detailed description of the API. It also displays the status as '有効' (Enabled). The 'レスポンス コード別のトラフィック' section shows a chart with no data available for the selected time frame, indicating 0 requests per second from October 31 to November 21.

- YouTube Data API が有効化できたので、再度 Bubble の画面に戻り "Initialize call" ボタンをクリック
- 成功すると "Returned values - search" というポップアップが表示されるはずです

Returned values - search

You can modify the data types that are returned by the call. This affects how you can use the data in Bubble. If you chose 'Ignore field', the fields won't be shown in the dropdowns.

kind youtube#searchListResponse	<input type="text"/>
etag bMKu0aiX56rekNwT5RVHZLOYR-o	<input type="text"/>
nextPageToken CAUQAA	<input type="text"/>
regionCode US	<input type="text"/>
pageInfo totalResults 1000000	<input type="number"/>
pageInfo resultsPerPage 5	<input type="number"/>
items (list) (see fields below)	<input type="text"/> search item
kind youtube#searchResult	<input type="text"/>
etag WahfcvNgYipX-i3rjRIPUrHi10	<input type="text"/>
id kind youtube#channel	<input type="text"/>
id channelId UCcRigzI_jBAZh_UySjv8xuw	<input type="text"/>
id videoId L2SaFPQI4_g	<input type="text"/>

Cancel

- これは今回設定した `https://www.googleapis.com/youtube/v3/search` の API を実行した結果の情報を表しています

詳しくは割愛しますが、ポイントとしては

- `items(list)` となっている部分が検索結果の動画一覧が格納されている一覧になります
- その型となるのが `search item` という型となり、検索結果に含まれる複数系のデータであることを表しています
- そして、この後の設定で特に重要なのが `id videoId` となっている項目です

- YouTube を見たことがある人なら分かると思いますが、この Videoid というのが、それぞれの動画を一意に識別する動画となっており、画面表示するときにもこの Id が必要となります
- その他、特に参照しない項目については紛らわしいのでプルダウンの中から **Ignore field** を選択しておきます
- こうすることで、Dynamic data として扱う時に、選択肢の中に表示しないようになり、設定項目を選ぶ際に悩まずに済みます

- すべて設定するとこんな感じです
- 設定が完了したら "SAVE" をクリックして、設定内容を保存しておきます

Returned values - search

You can modify the data types that are returned by the call. This affects how you can use the data in Bubble. If you chose 'Ignore field', the fields won't be shown in the dropdowns.

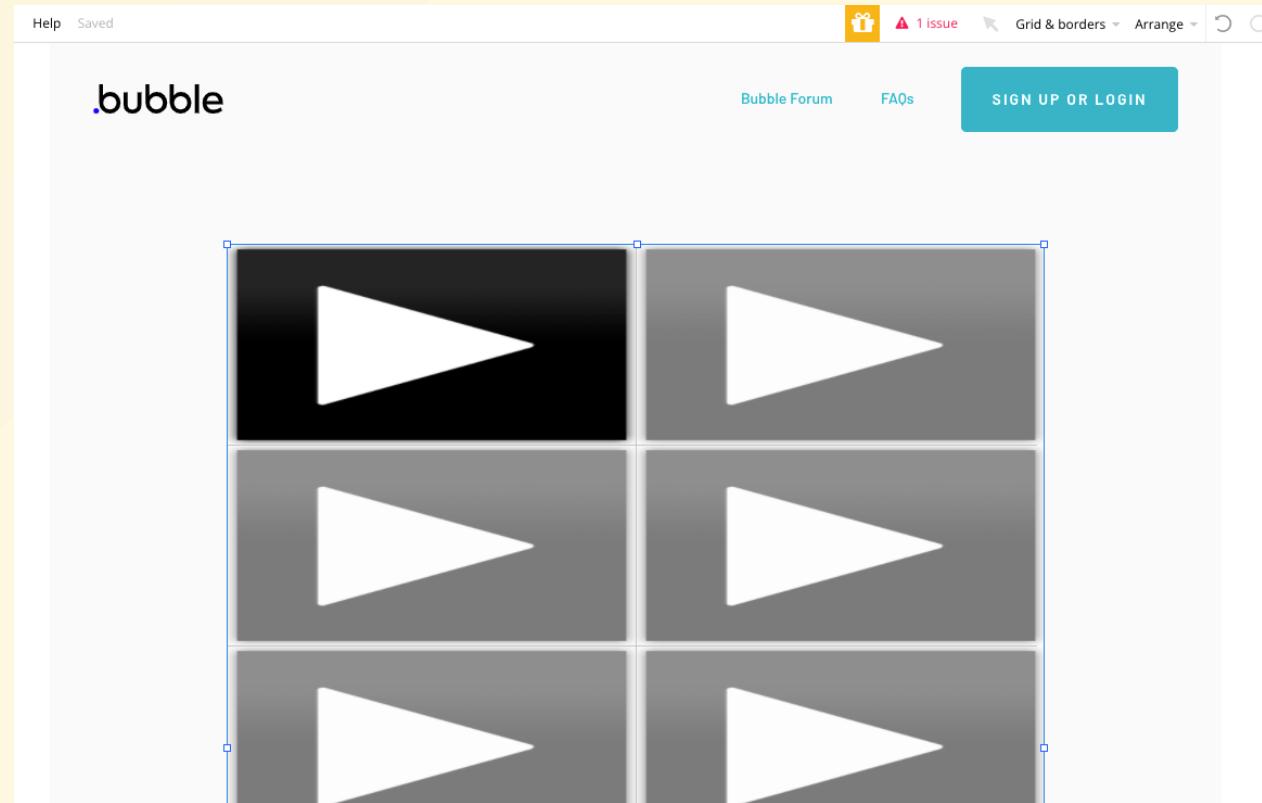
kind youtube#searchListResponse	Ignore field ▾
etag bMKu0aiXS6rekNwT5RVHZLOYR-o	Ignore field ▾
nextPageToken CAUQAA	Ignore field ▾
regionCode US	Ignore field ▾
pageInfo totalResults 1000000	Ignore field ▾
pageInfo resultsPerPage 5	Ignore field ▾
<u>items (list)</u> (see fields below)	search item ▾
kind youtube#searchResult	Ignore field ▾
etag WahfIicVNgYipX-i3rjRiPUrHi10	Ignore field ▾
id kind youtube#channel	Ignore field ▾
id channelId UCcRigzl_jBAZh_UySjv8xuw	Ignore field ▾
id videoId L2saHPPQH_E	text ▾

SAVE Cancel

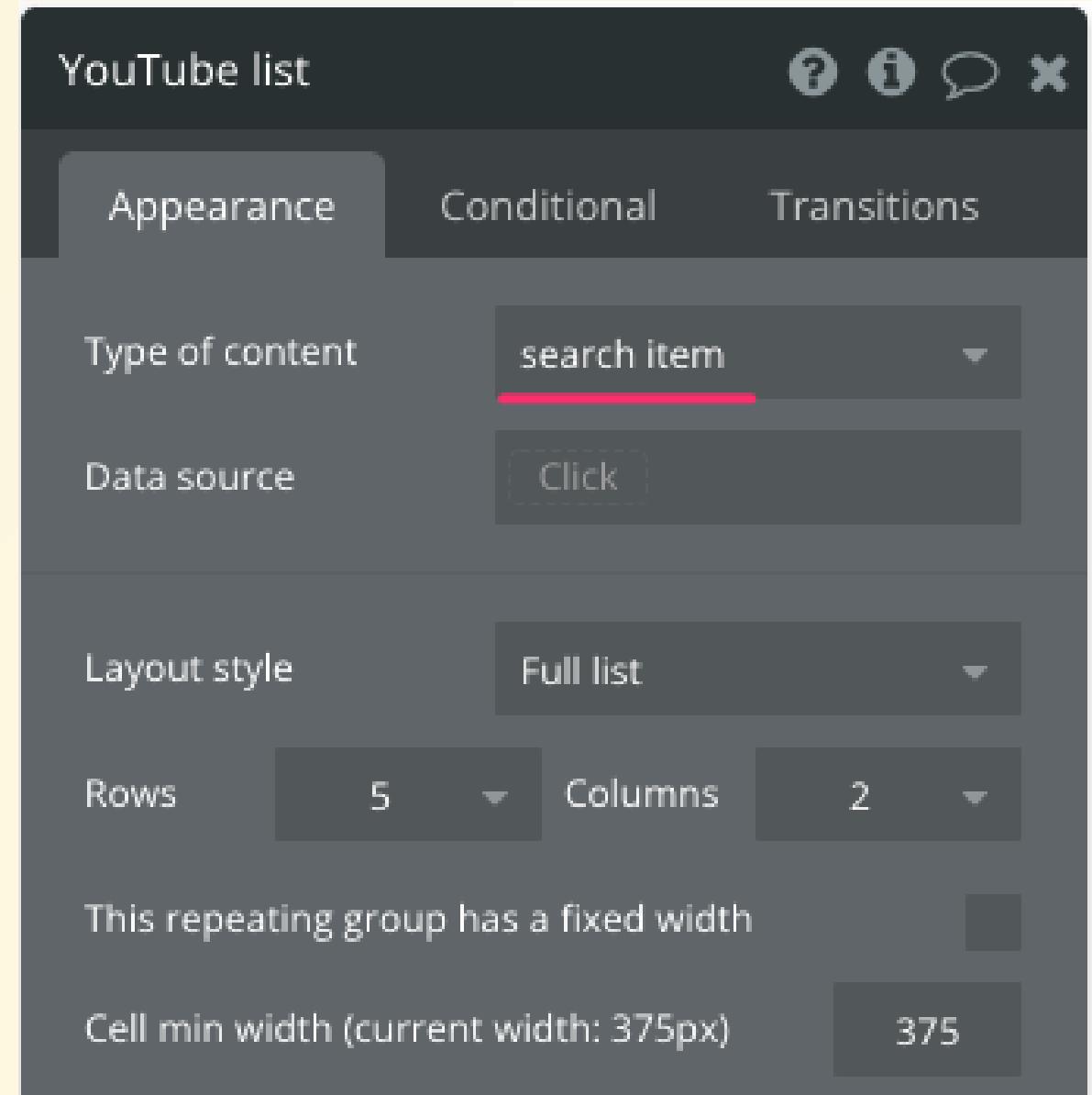
画面表示の準備

- Bubble の API 連携の準備が出来たので、いよいよ画面の設定をしていきます
- 画面の設定は第 3 回、4 回の講義でマスターしていると思いますので、概要だけお伝えして画面レイアウトを組み立ててみてください

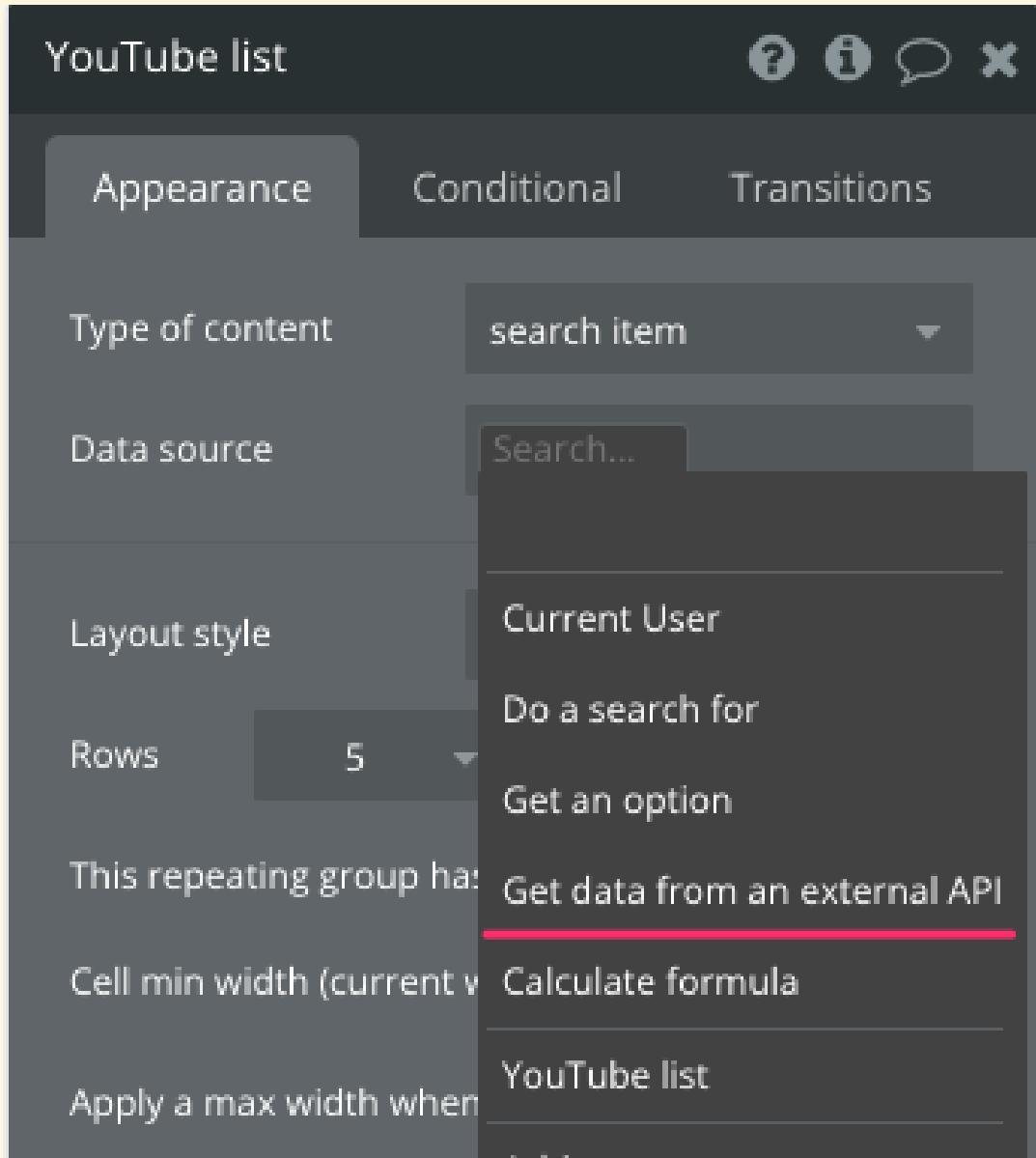
- index ページで Repeating Group を使って一覧を組み立てます
- Repeating Group の各セルの中には Video を配置
- こんな感じですね



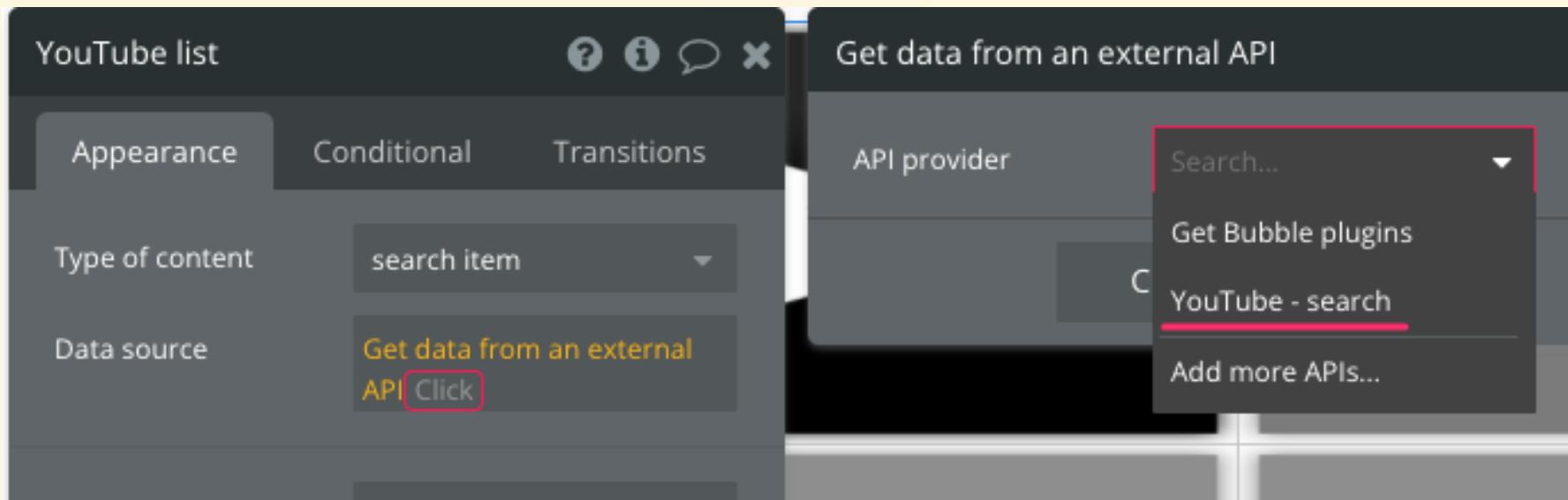
- レイアウトが出来たので実際に YouTube の一覧表示を設定していきます
- まず Repeating Group の Type of content は、先ほど API 実行時に確認した search item 型となります



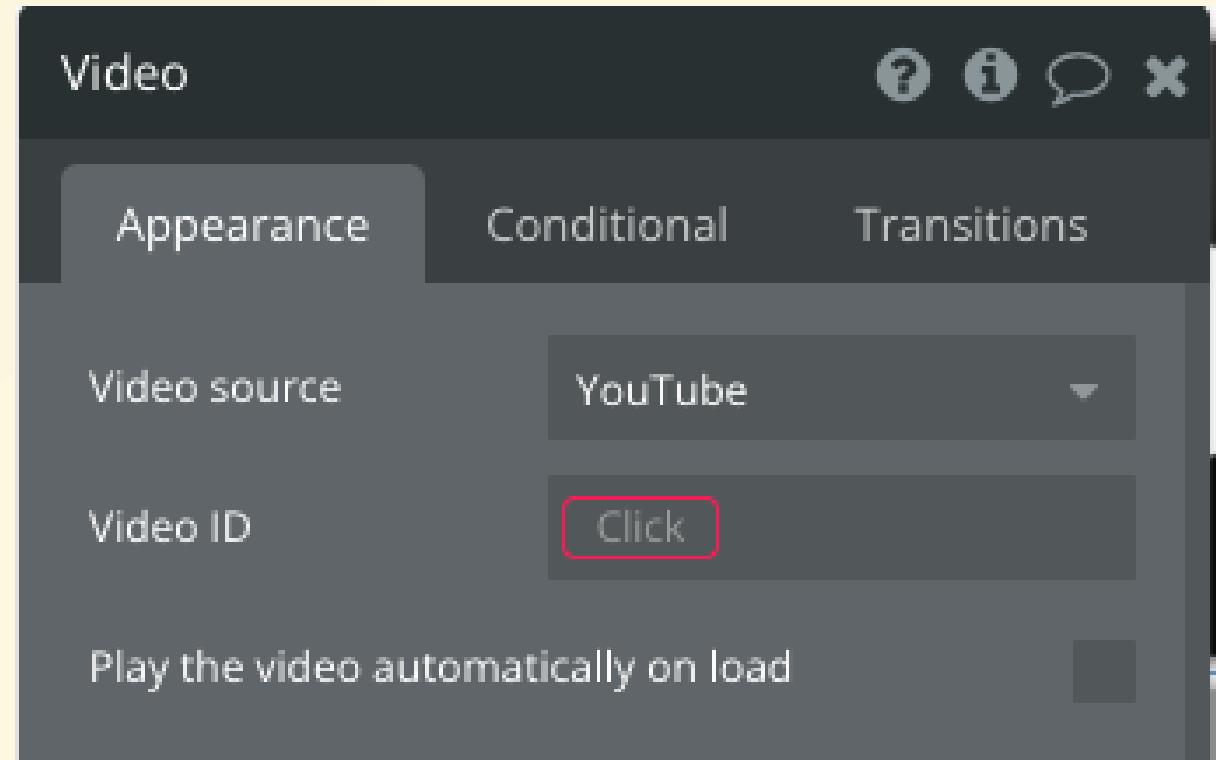
- 次に Data source ですが、
今日は API から取得した
YouTube の動画一覧を対象
とします
- Click から候補を見てみると
Get data from an external
API
という項目があると思うの
でそちらを選択



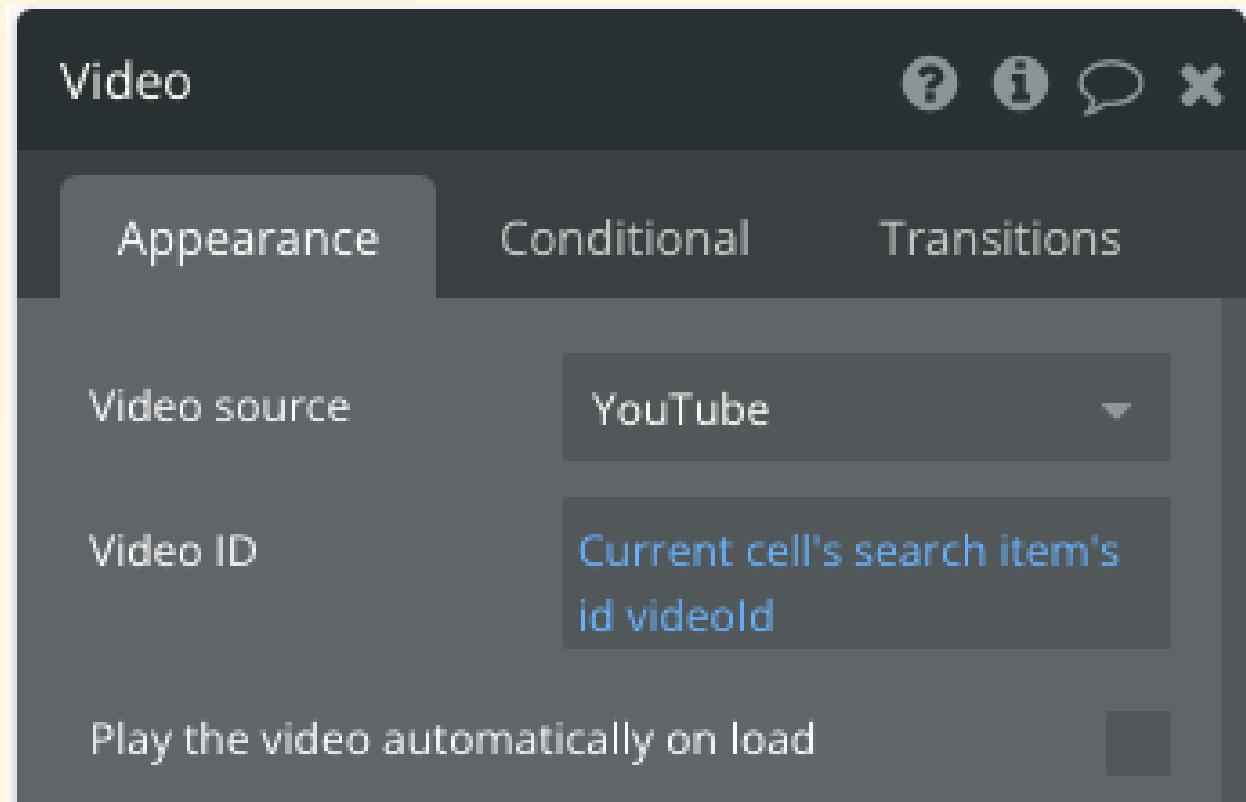
- 隣に Get data From an external API のポップアップが表示されるので、API provider のプルダウンを選んでください
- すると、プルダウンの中に先ほど設定した YouTube - search の候補があると思いますのでそれを選択してください
 - ハイフンの前が API のグループ名で、後ろが具体的な API の名前です



- 次に各セルの設定をしていきますが、まずは皆さん設定してみましょう
- Video source は YouTube
- すると Video ID という項目が表示されるので、先ほど API 設定をしたときにみた Video Id を参照すればよさそうですね



- こんな感じですね

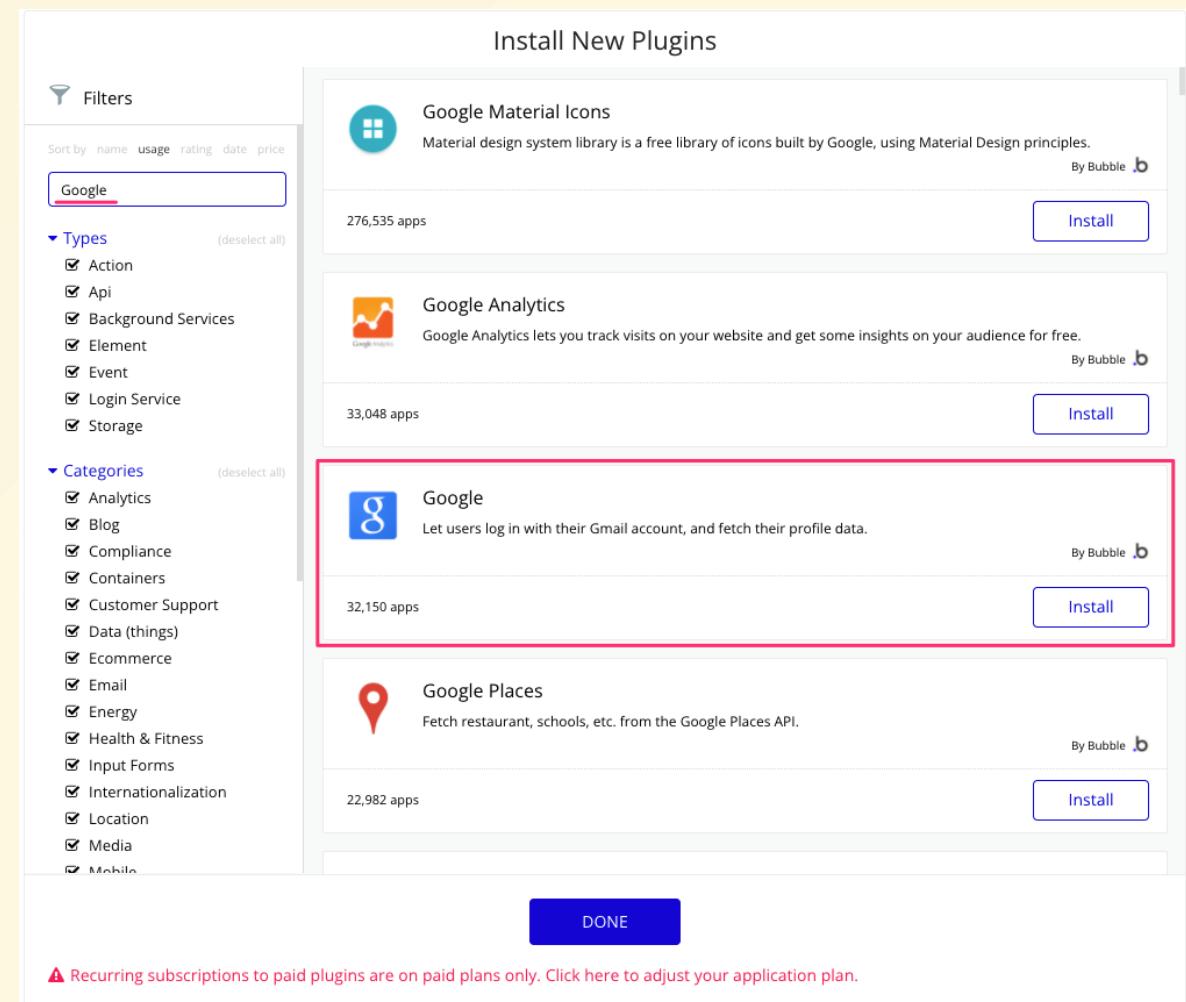


- これで設定完了です！さっそくプレビューしてみましょう！
- 事前に API で指定したキーワードの動画一覧が表示されましたか？



Google ログインを試してみよう

- 続いて Bubble アプリへのログインを Google アカウントを使ってログインしてみましょう！
- まずは Google のプラグインをインストール



- インストールした Google プラグインを選択
- 右パネルに表示されている `Use a generic redirect URL` の欄にチェックを入れる
 - この URL もメモしておくのですが、画面上で選択ができないため、下記 URL のアプリ名部分だけご自身のアプリに読み替えてメモしておいてください

`https://advanced-bubble-{Your`

- `Name}.bubbleapps.io/api/1.1/oauth_redirect`



Google

Service page →

Let users log in with their Gmail account, and fetch their profile data.

App Secret

App ID/API Key

Use a generic redirect URL (https://advanced-bubble-kyogoku.bubbleapps.io/api/1.1/oauth_redirect)



App Secret - dev.

(optional)

App ID/API Key - dev.

(optional)

Plugin content

This plugin is an authentication provider.

Googleへのログイン後にアプリに戻ってくるための認証準備

- Google Cloud Platform 画面にアクセス

<https://console.developers.google.com/>

- プロジェクトを選択し、左メニューから OAuth 同意画面を選択

The screenshot shows the Google Cloud Platform interface. In the top navigation bar, 'Google Cloud Platform' and 'Bubble test' are visible. A search bar at the top right contains the text 'プロダクトとリソースの検索'.

The left sidebar menu is titled 'API & サービス' and includes the following items:

- ダッシュボード
- ライブラリ
- 認証情報
- OAuth 同意画面** (highlighted with a red underline)
- ドメインの確認
- ページの使用に関する契約

The main content area is titled 'OAuth 同意画面'. It contains the following information:

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

Google の OAuth に関する [ご意見やご要望をお聞かせください。](#)

- User Type に "外部" を選択して「作成」
- 必要事項を入力したら、保存して次へ

The screenshot shows the Google Cloud Platform interface for configuring an OAuth consent screen. The left sidebar lists several options: ダッシュボード, ライブラリ, 認証情報, OAuth 同意画面 (which is selected and highlighted with a red underline), ドメインの確認, and ページの使用に関する契約. The main content area is titled 'OAuth 同意画面' and contains instructions about how the app is configured and registered. It shows two radio button options for 'User Type': '内部' (selected) and '外部'. The '外部' option is described as allowing Google accounts to use the app in test mode. At the bottom is a large blue '作成' (Create) button.

API API とサービス OAuth 同意画面

ダッシュボード ライブラリ 認証情報 OAuth 同意画面 ドメインの確認 ページの使用に関する契約

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

Google の OAuth に関する [ご意見やご要望をお聞かせください。](#)

- アプリ情報
 - アプリ名: bubble で作成したアプリ名
 - ユーザーサポートメール: ご自身の Google アカウントのメールアドレス

アプリ登録の編集

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

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*
advanced-bubble-kyogoku

同意を求めるアプリの名前

ユーザー サポートメール*
kyougoku2bubble@gmail.com

▼

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 参照

ユーザーがアプリを認識できるように、同意画面に 1 MB 以下の画像をアップロードします。使用できる画像形式は、JPG、PNG、BMP です。最適な結果を得るには、ロゴを 120 x 120 ピクセルの正方形にすることをおすすめします。



- 承認済みドメイン
 - bubbleapps.io
- デベロッパーの連絡先情報
 - ご自身の Google アカウントのメールアドレス

承認済みドメイン [?](#)

同意画面または OAuth クライアントの構成でドメインが使用されている場合は、ここで事前登録する必要があります。アプリの検証が必要な場合は、[Google Search Console](#) にアクセスして、ドメインが承認済みであるかどうかを確認してください。承認済みドメインの上限の[詳細](#)をご覧ください。

bubbleapps.io

+ ドメインの追加

デベロッパーの連絡先情報

メールアドレス * [×](#)

これらのメールアドレスは、プロジェクトの変更について Google からお知らせするために使用します。

[保存して次へ](#) [キャンセル](#)

- スコープは特に設定変更せず、保存して次へ

- テストユーザにはご自身の Google メールアドレスを入力しておく
- 設定したら保存して次へ

アプリ登録の編集

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

テストユーザー

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](#)

+ ADD USERS

≡ フィルタ プロパティ名または値を入力 ?

ユーザー情報	
kyougoku[REDACTED]gmail.com	刪除

保存して次へ キャンセル

- これで登録は完了したので続いて認証情報を発行します
- 左メニューから認証情報を選択して、「+ 認証情報を作成」をクリック
- サブメニューから OAuth クライアント ID を選択

The screenshot shows the Google Cloud Platform interface for creating an authentication configuration. The left sidebar is titled 'API & Services' and has several options: Dashboard, Libraries, Authentication (which is highlighted with a red underline), OAuth Consent Screen, Domain Verification, and Page View Metrics. The main content area is titled 'Authentication' and shows a sub-section for 'OAuth Client ID'. A button '+ Create authentication configuration' is underlined in red. Below it, there are three options: 'API Key' (disabled), 'OAuth Client ID' (selected and underlined in red), and 'Service Account'. A tooltip for 'OAuth Client ID' explains that it allows access to user data with user consent. At the bottom, there's a section for 'Wizard Selection' with the note 'Select the type of authentication information to be used by answering a few questions'. A message at the very bottom says 'No OAuth Client ID is displayed'.

- OAuth クライアント ID の作成画面が表示されるので下記の通り入力し「保存」
- アプリケーションの種類: ウェブアプリケーション
- 名前: Bubble OAuth

[← OAuth クライアント ID の作成](#)

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 *

名前 *

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。

ⓘ 下で追加する URI のドメインは、[OAuth 同意画面に承認済みドメイン](#)として自動で追加されます。

承認済みの JavaScript 生成元 [?](#)

ブラウザからのリクエストに使用します

[+ URI を追加](#)

承認済みのリダイレクト URI [?](#)

ウェブサーバーからのリクエストに使用します

URI *

[+ URI を追加](#)

[作成](#) [キャンセル](#)

- 承認済みのリダイレクト
URI: Bubble 側で Google プラグインインストール時に控えた URL

`https://advanced-bubble-{Your Name}.bubbleapps.io/api/1.1/oauth_redirect`

← OAuth クライアント ID の作成

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 *
ウェブアプリケーション

名前 *
Bubble OAuth

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。

下で追加する URI のドメインは、[OAuth 同意画面に承認済みドメイン](#)として自動で追加されます。

承認済みの JavaScript 生成元 ②

ブラウザからのリクエストに使用します

+ URI を追加

承認済みのリダイレクト URI ②

ウェブサーバーからのリクエストに使用します

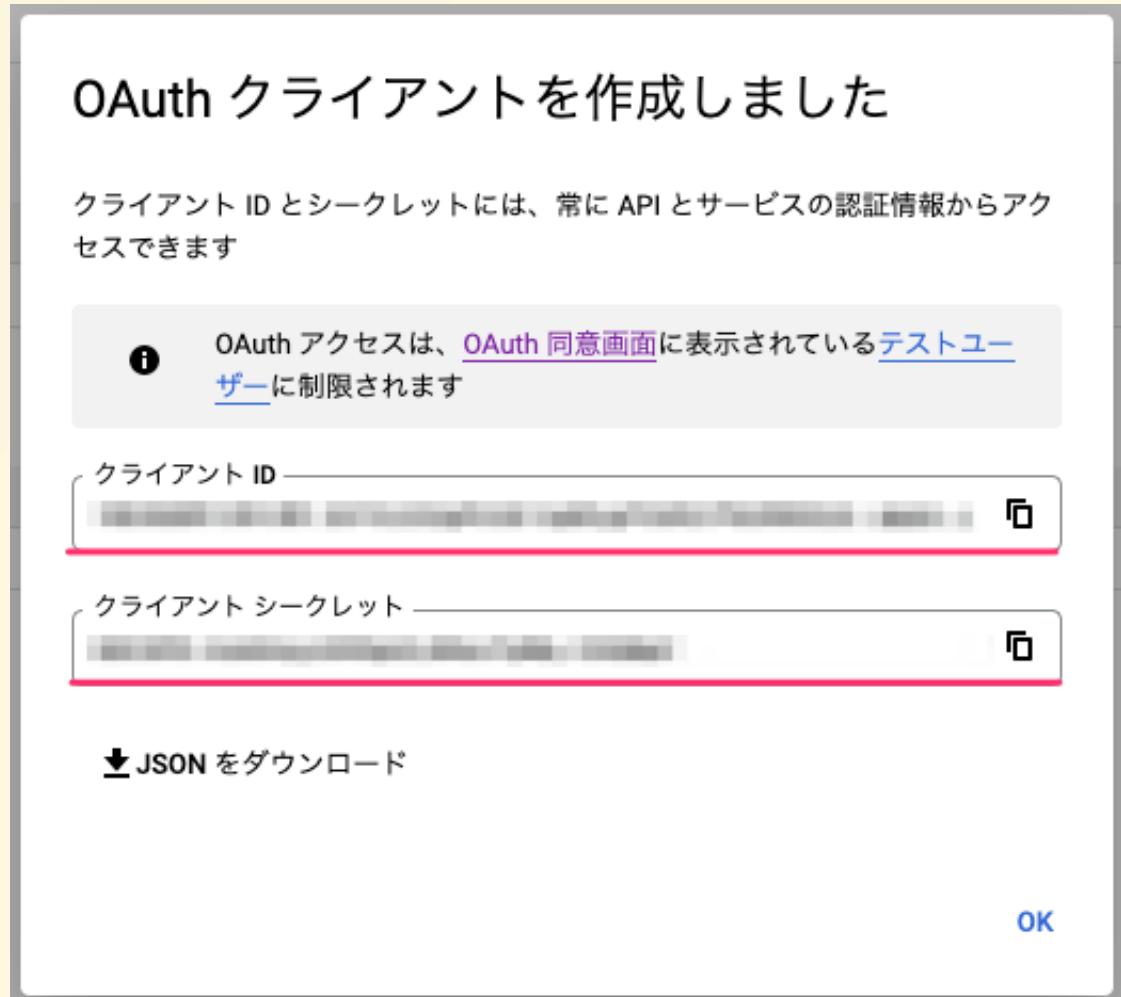
URI *

`https://advanced-bubble-kyogoku.bubbleapps.io/api/1.1/oauth_redirect`

+ URI を追加

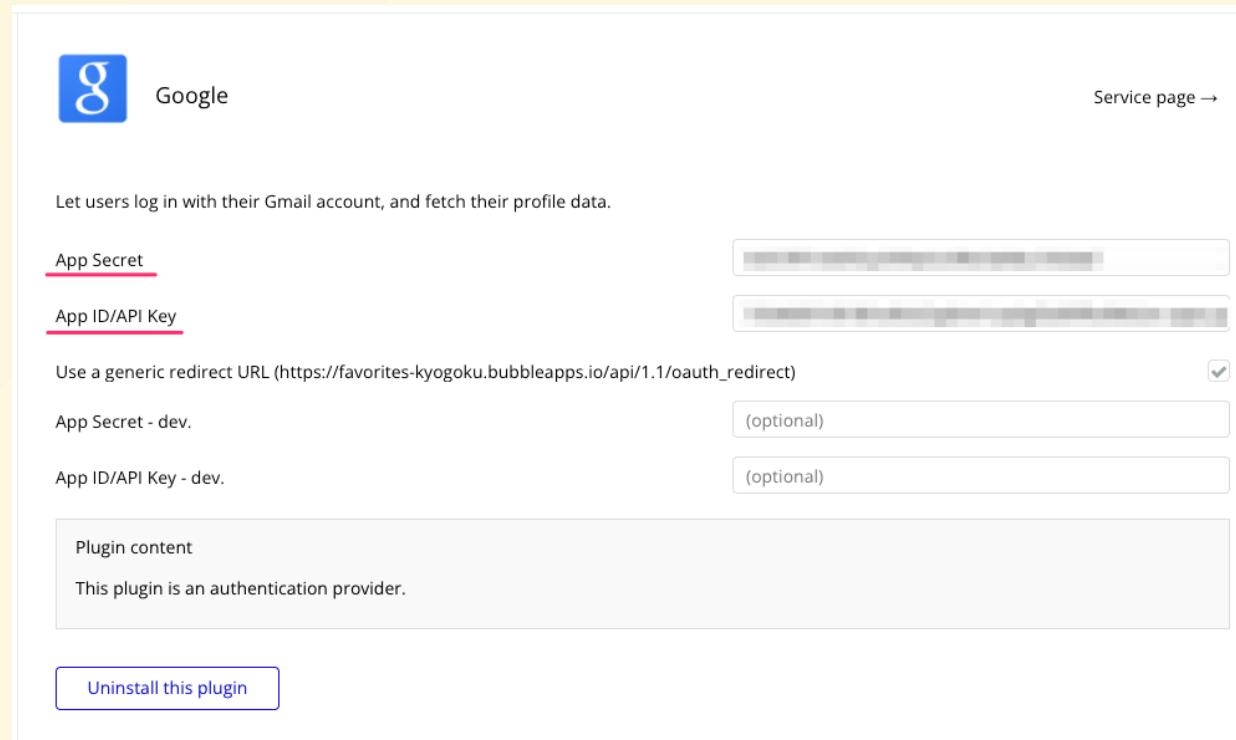
作成 キャンセル

- すると「OAuth クライアントを作成しました」というポップアップが表示され、そこにクライアント ID とクライアントシークレットが表示されているので、それをメモしておく
 - メモだと忘れるかもしれない場合は JSON をダウンロードしても OK
- メモしたら OK でクローズ



発行した認証情報を Bubble 側に設定

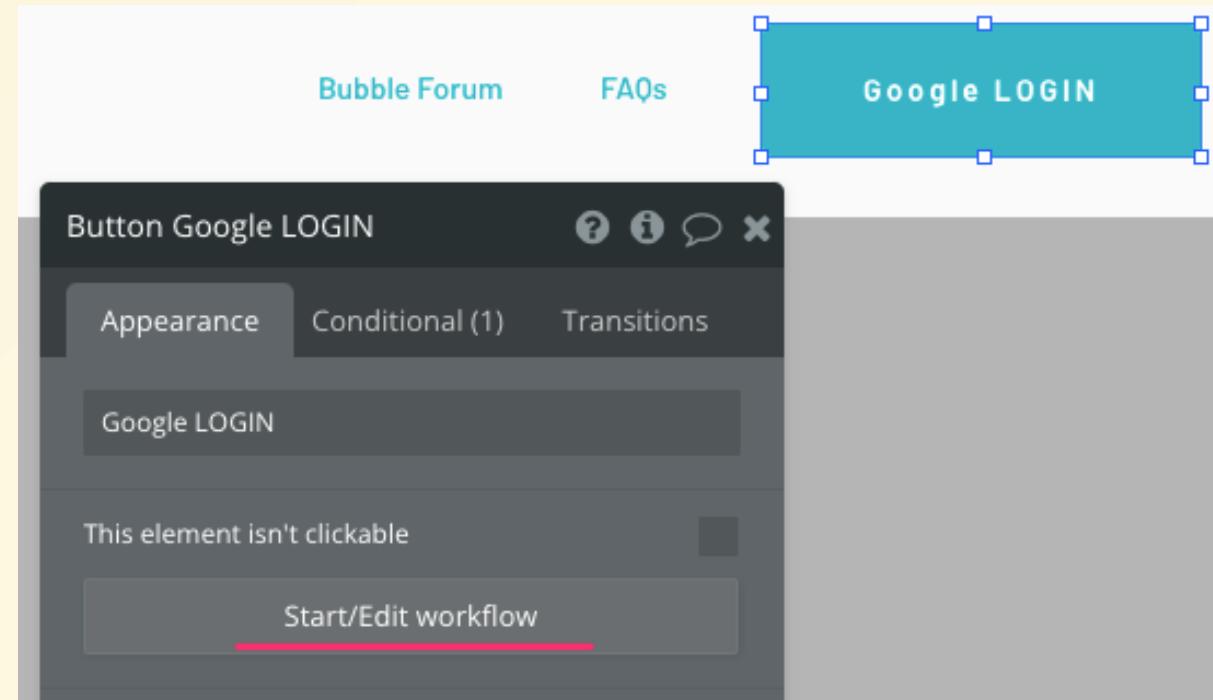
- Bubble の Google プラグイン画面に戻り、先ほど取得したキーをそれぞれの項目へ入力する
 - クライアントID → App ID/API Key
 - クライアントシークレット → App Secret



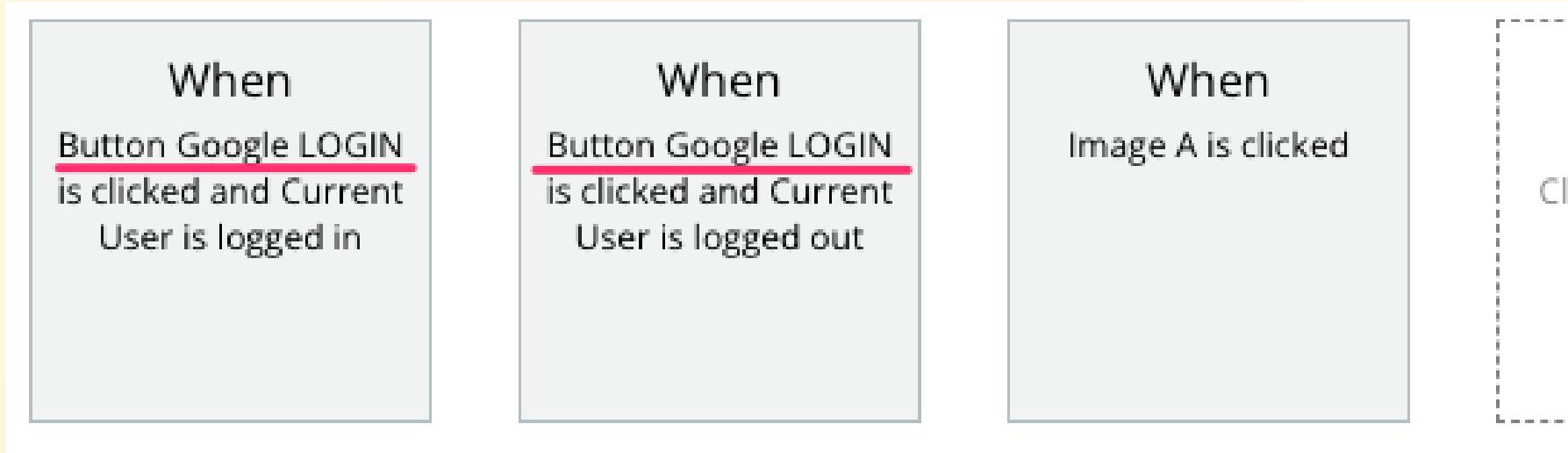
- これで事前設定は完了
- 続いてログイン機能を作り込んでいきます

ログイン機構を Google に置き換える

- ヘッダーコンポーネントを開き、SIGN UP OR LOGIN ボタンを選択してワークフロー画面へ遷移
 - ついでにボタンのラベルを "Google LOGIN" に変更



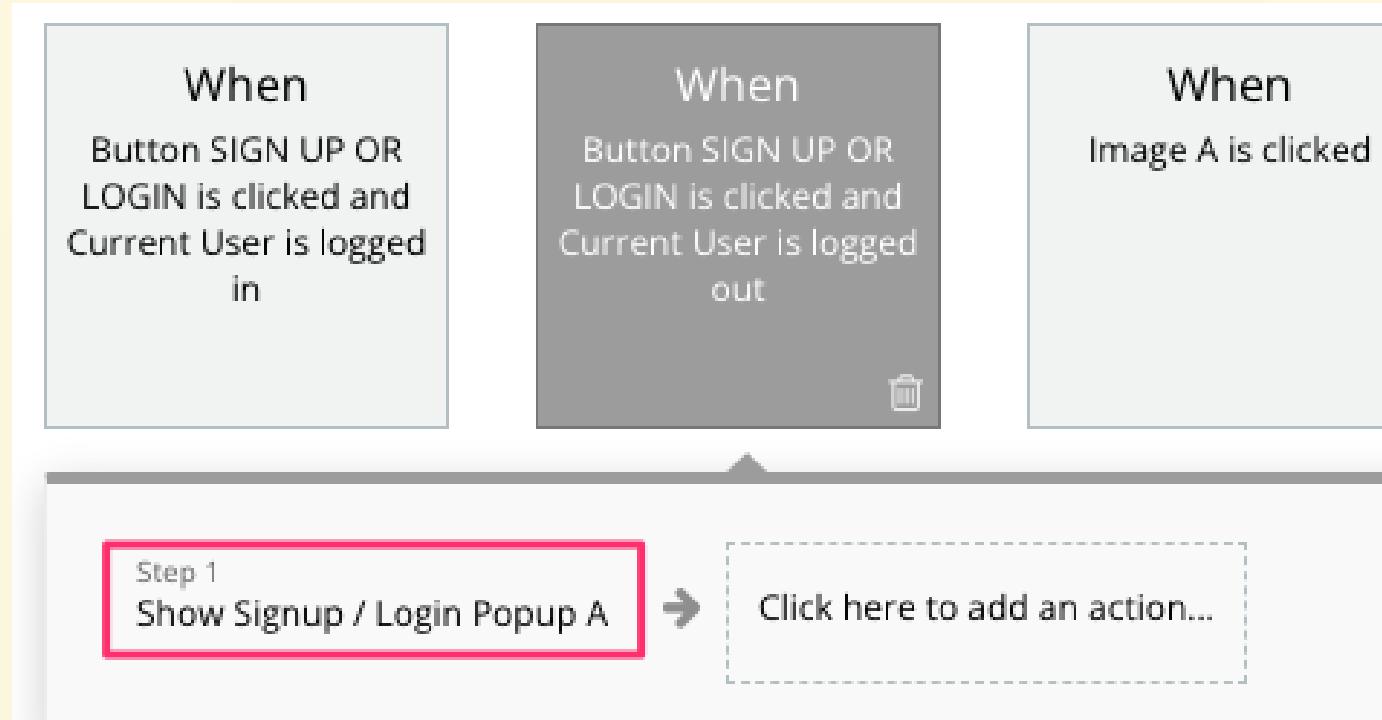
- ワークフロー画面へ遷移すると Button Google LOGIN に対して 2 つの When に対する振る舞いが設定されていると思います



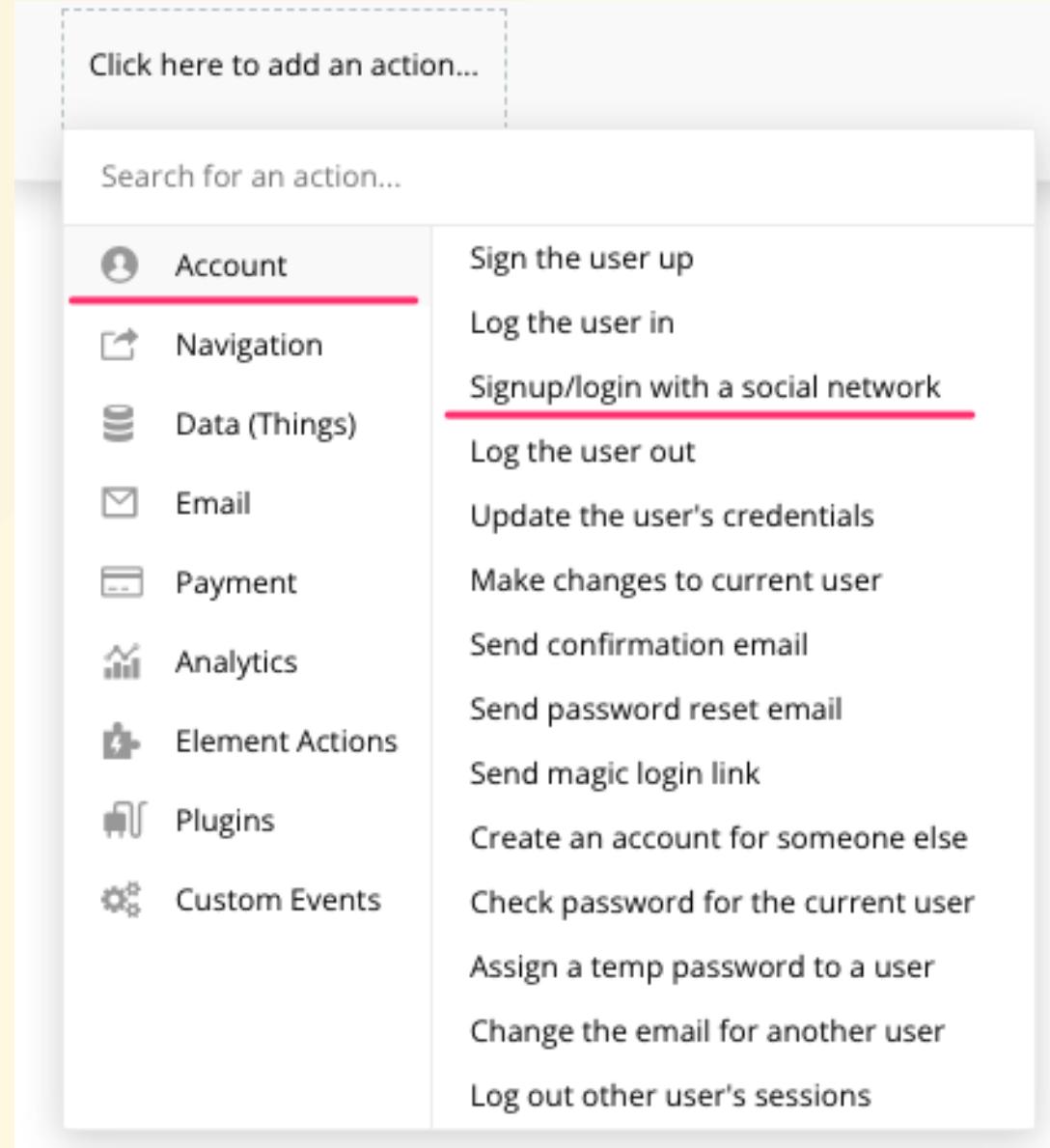
- When: Button Google LOGIN is clicked and Current User is logged in
- これは文字通り、カレントユーザがログイン済み状態で「Google LOGIN」ボタンを押された場合、のワークフローとなります

- When: Button Google LOGIN is clicked and Current User is logged out
- これも文字取り、カレントユーザがログアウト済み状態で「Google LOGIN」ボタンを押された場合、のワークフローとなります

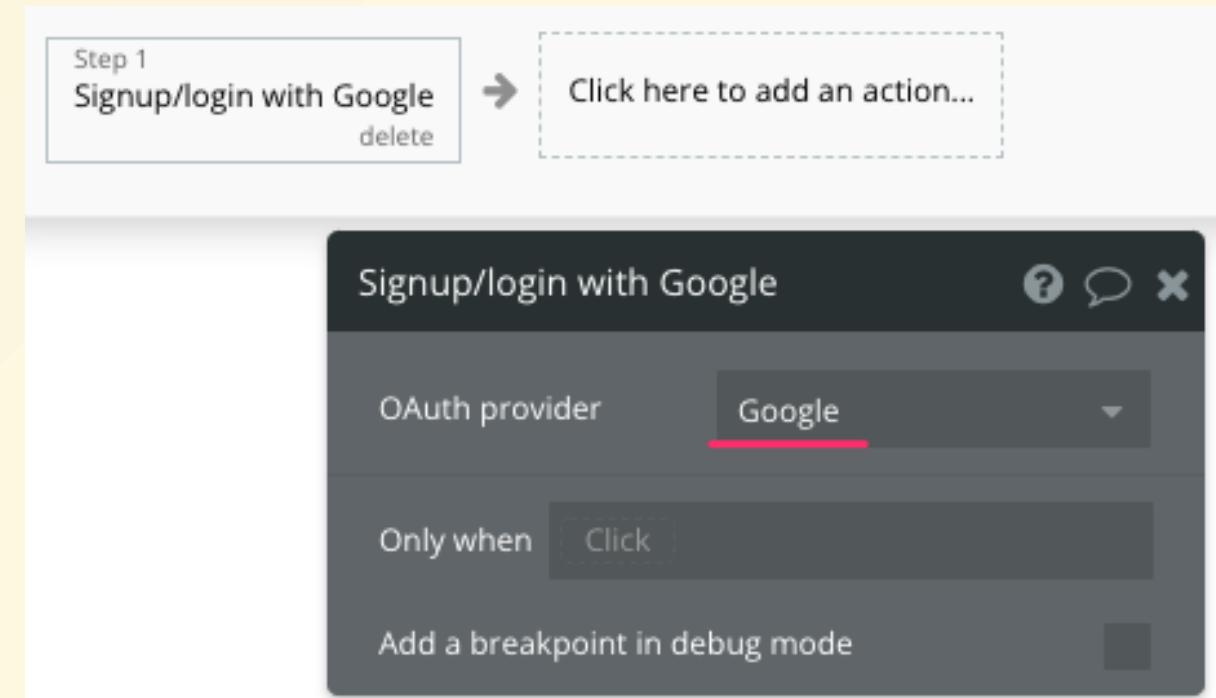
- まずはログイン時のワークフローを設定していきたいので、
Current User is logged out のワークフローを設定していきます
- 既存すでに「Show Sign up / Login Popup A」のステップが定義されているので、そちらを delete します



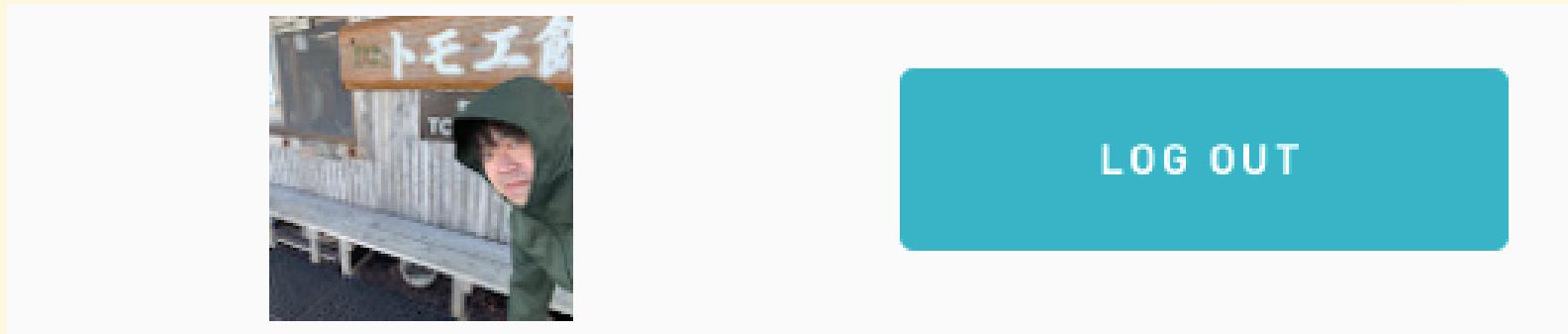
- 続いて Google ログインのワークフローを設定していきます
- Click here to add an action... を選択
- Account から Signup/login with a social network を選択



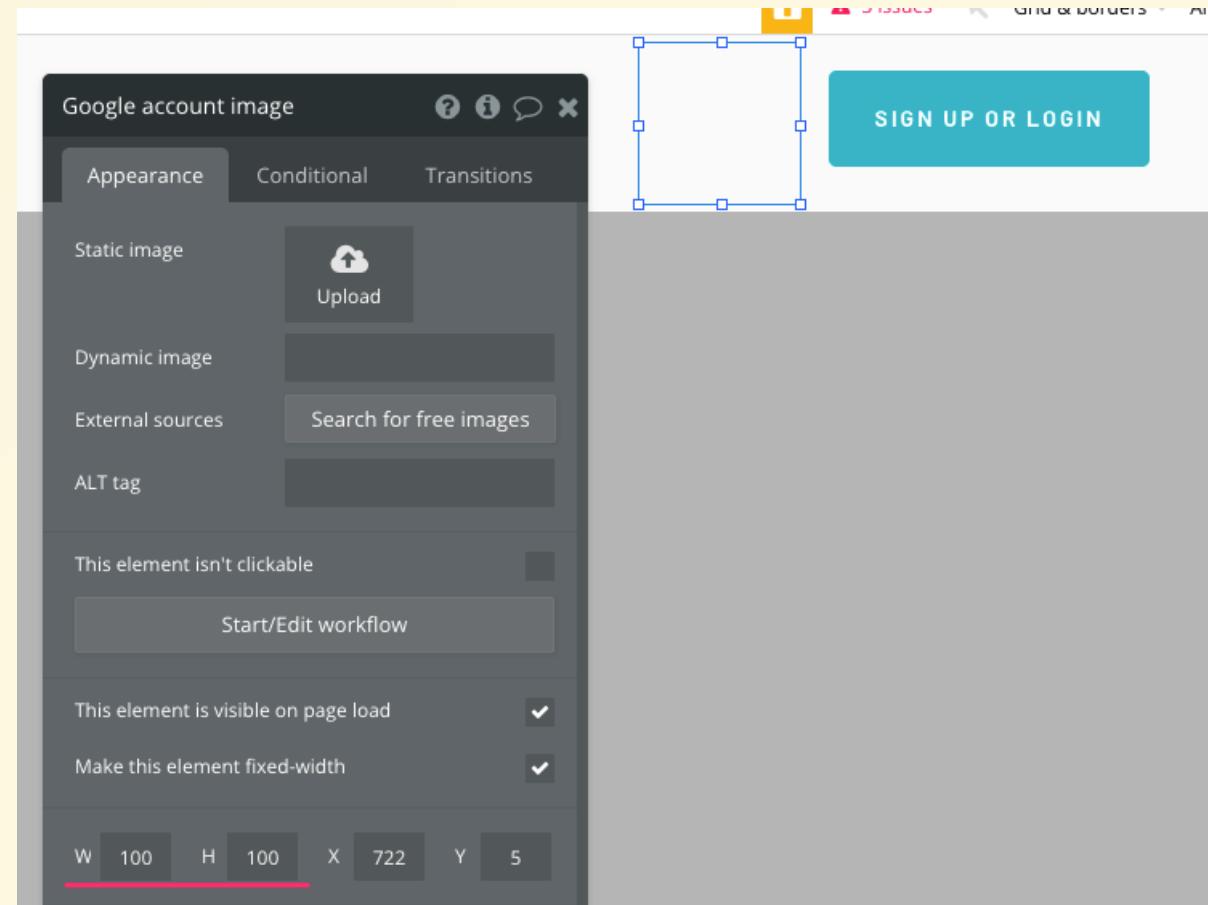
- Signup/login with Google のポップアップが表示されます
- OAuth provider に Google を選択
- 事前に Google と Bubble のプラグインに設定を行ったので、これだけで Google 認証が行えるようになります



- 折角なのでログイン成功したらヘッダー上部にアカウント名と画像を表示してみましょう

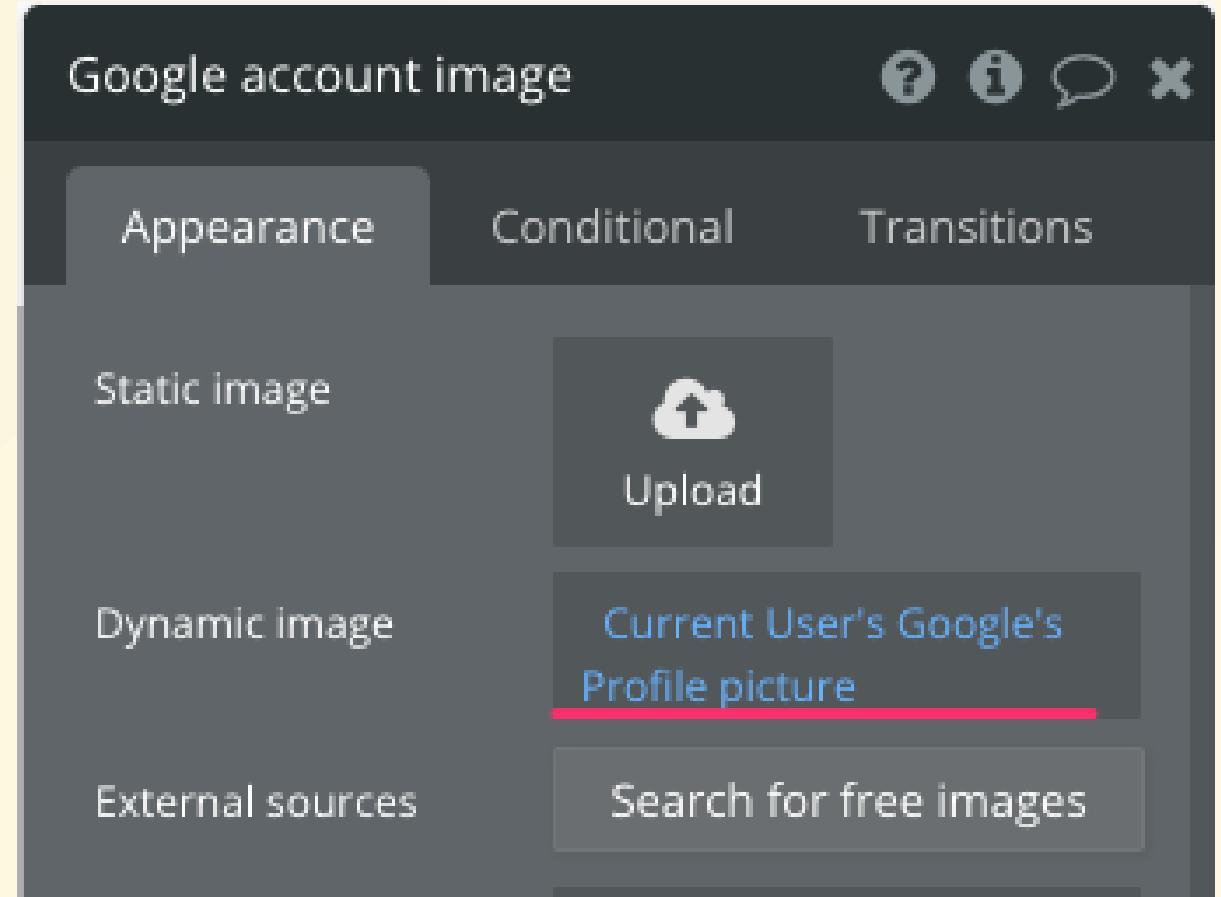


- ヘッダーコンポーネントを開き、LOGIN ボタン左の要素を削除します
- Visual elements から Image を選択し、LOGIN ボタンの左側に配置します
 - 縦横 100×100 の正方形にしておきましょう



- そして、画像の参照元を設定します
- Bubbler の皆さんなら設定方法はイメージつきますよね
- 動的な表示設定は Dynamic image
- 表示したいのは現在ログインしているユーザの Google アカウントのプロフィール画像

- こんな感じですね



- ではプレビューしてみましょう！
 - 事前に Data の User にこれから動作確認する際に使用する Google メールアドレスと同じメールアドレスのデータがある場合は削除してください
- ログインボタンを押すと Google のログイン画面に遷移し、そこでログインをすると Bubble 側に戻ってきましたよね？
- ちなみに、ログインすると Bubble 側の User data にログインした Google アカウントのメールアドレス情報が登録されます

Google LOGIN



LOG OUT

最後に YouTube API を使った検索機能を組み込んでみましょう

The screenshot displays a user interface for searching YOASOBI music videos. At the top, there is a navigation bar with the 'bubble' logo on the left and a 'LOG OUT' button on the right. Below the navigation bar is a search bar containing the text 'YOASOBI'. To the right of the search bar is a blue 'Search' button. The main content area shows a grid of nine video thumbnails, each featuring the YOASOBI logo and a play button. The thumbnails are arranged in three rows of three. The first row includes a thumbnail for a video with a warning about age restrictions. The second row includes thumbnails for 'YOASOBI 2nd EP『THE BOOK』' and 'YOASOBI「大正浪漫」Official Music Video'. The third row includes thumbnails for 'YOASOBI「怪物」Official Music Video', 'YOASOBI - Racing Into The Night', and 'YOASOBI - 夜に駆ける / THE FIRST TAKE'. Below the grid, there are two additional video thumbnails: 'YOASOBI PLAYLIST' and 'YOASOBIのベストソング-YOASOBI BEST SONGS'. The 'YOASOBI PLAYLIST' thumbnail shows a purple background with a play button and a list of songs. The 'YOASOBIのベストソング-YOASOBI BEST SONGS' thumbnail shows a red background with a play button and a list of songs.

- 現状では YouTube API の認証キーと検索キーワードは固定になっています
- なので、これを下記のように変えてみましょう
 - 認証キーはログインしているユーザ情報から取得する（新たに field を追加する）
 - 検索キーワードは画面から入力・検索できるようにする

次のページにヒントを書いておきますのでまずはトライしてみましょ
う！

- まずは User に対して `key` という field を追加し、事前に設定しておく
 - 本来ここもユーザ自身に入力させるなどの導線が自然ですが今回は時間の都合上、API キーの設定画面は割愛します
- API のパラメータを処理の中で動的に設定する場合、まずは固定値を無くし `Private` のチェックを外すことで、DataSource などで API 呼び出しする際にパラメータを指定できます
- index ページに新たに検索用の要素（テキストボックスと検索ボタン）を配置し、検索ボタンがクリックされた時に、YouTube API 経由で検索を行い、結果を Repeating Group にセットしてあげます



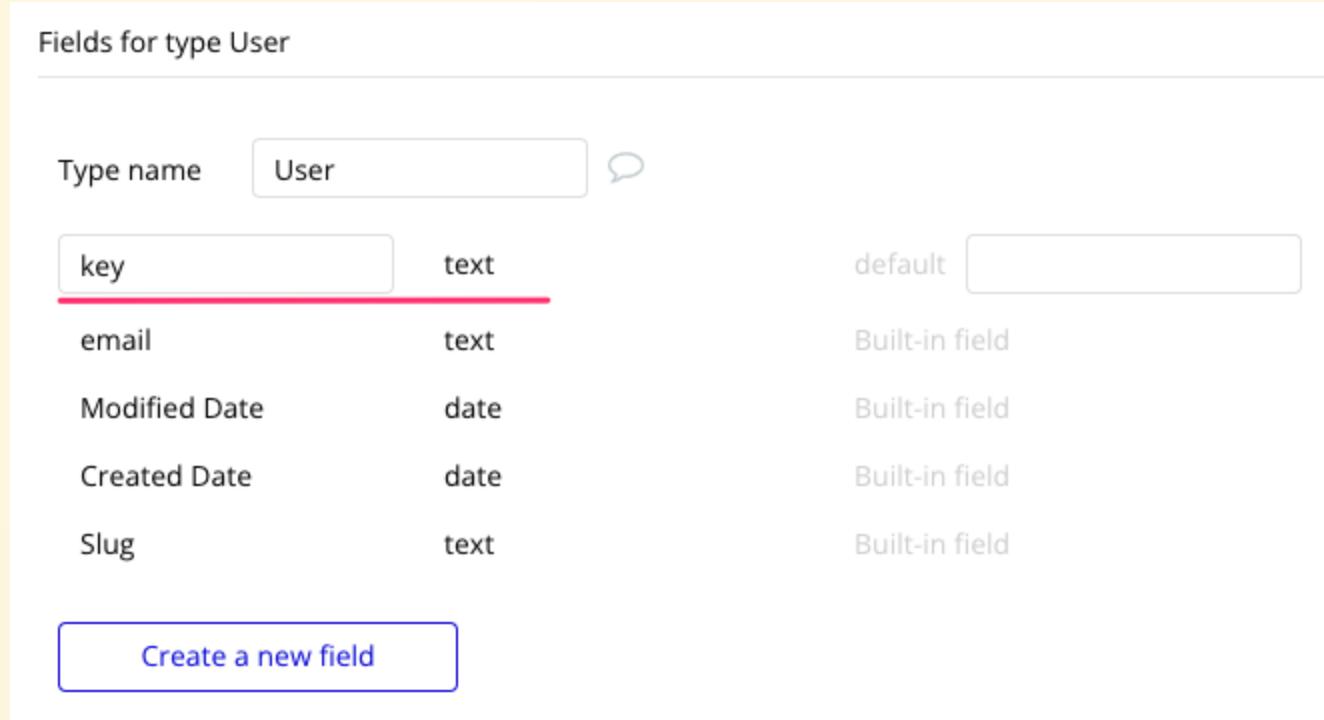
- まずは User の型に対して `key` という field を追加します

Fields for type User

Type name 💬

<input type="text" value="key"/>	text	<input type="text" value=""/>	default
<input type="text" value="email"/>	text	Built-in field	
<input type="text" value="Modified Date"/>	date	Built-in field	
<input type="text" value="Created Date"/>	date	Built-in field	
<input type="text" value="Slug"/>	text	Built-in field	

[Create a new field](#)



- そして、今回取得した認証キー（API キー）を登録済みの User に事前にセットしておきます

The screenshot shows the 'App data' section of the Bubble platform. The top navigation bar includes 'Data types', 'Privacy', 'App data' (which is selected), 'Option sets', and 'File manager'. Below the navigation, the title is 'Database views' and 'Application data - All Users - Development version'. There are buttons for 'New view' and 'Primary fields', and a search bar with the placeholder 'Search view names or data types...'. A status message indicates '1 entries (displaying 1)'. The main area displays a single user entry in a table format:

		Email	key
<input type="checkbox"/>	<input type="checkbox"/>	Run as → kyougoku2bubble@gmail.com	AlzaSy [REDACTED]

At the bottom left, there is a link 'All Users' and a small icon with a pencil and a square.

- 次に YouTube API 側の設定はこんな感じですね

collapse

Name: search Use as: Data Data type: JSON

Method: GET URL: https://www.googleapis.com/youtube/v3/search (use [] for params)

Headers

Add header

Parameters

Key: key	Value:	Private <input type="checkbox"/>	Allow blank <input type="checkbox"/>	Optional <input type="checkbox"/>
Key: q	Value:	Private <input type="checkbox"/>	Allow blank <input type="checkbox"/>	Optional <input type="checkbox"/>
Key: maxResults	Value: 10	Private <input checked="" type="checkbox"/>	Allow blank <input type="checkbox"/>	Optional <input type="checkbox"/>
Key: type	Value: video	Private <input checked="" type="checkbox"/>	Allow blank <input type="checkbox"/>	Optional <input type="checkbox"/>

Add parameter

Include errors in response and allow workflow actions to continue

Capture response headers

This call has been modified since you last initialized; consider reinitializing.

[Reinitialize call](#) [Manually enter API response](#)

trash icons

- ポイントとしては、動的に指定したい項目については Value の値を削除し、Private のチェックを外しました
- これにより、API を使う場面でこれらのパラメータの値を指定することができるようになります

The screenshot shows a configuration interface for an API call named "search". The "Method" is set to "GET" and the "URL" is "https://www.googleapis.com/youtube/v3/search". The "Data type" is set to "JSON".

Headers:

- Add header button

Parameters:

Key	Value	Private	Allow blank	Optional
key	(empty)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
q	(empty)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
maxResults	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
type	video	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Add parameter button

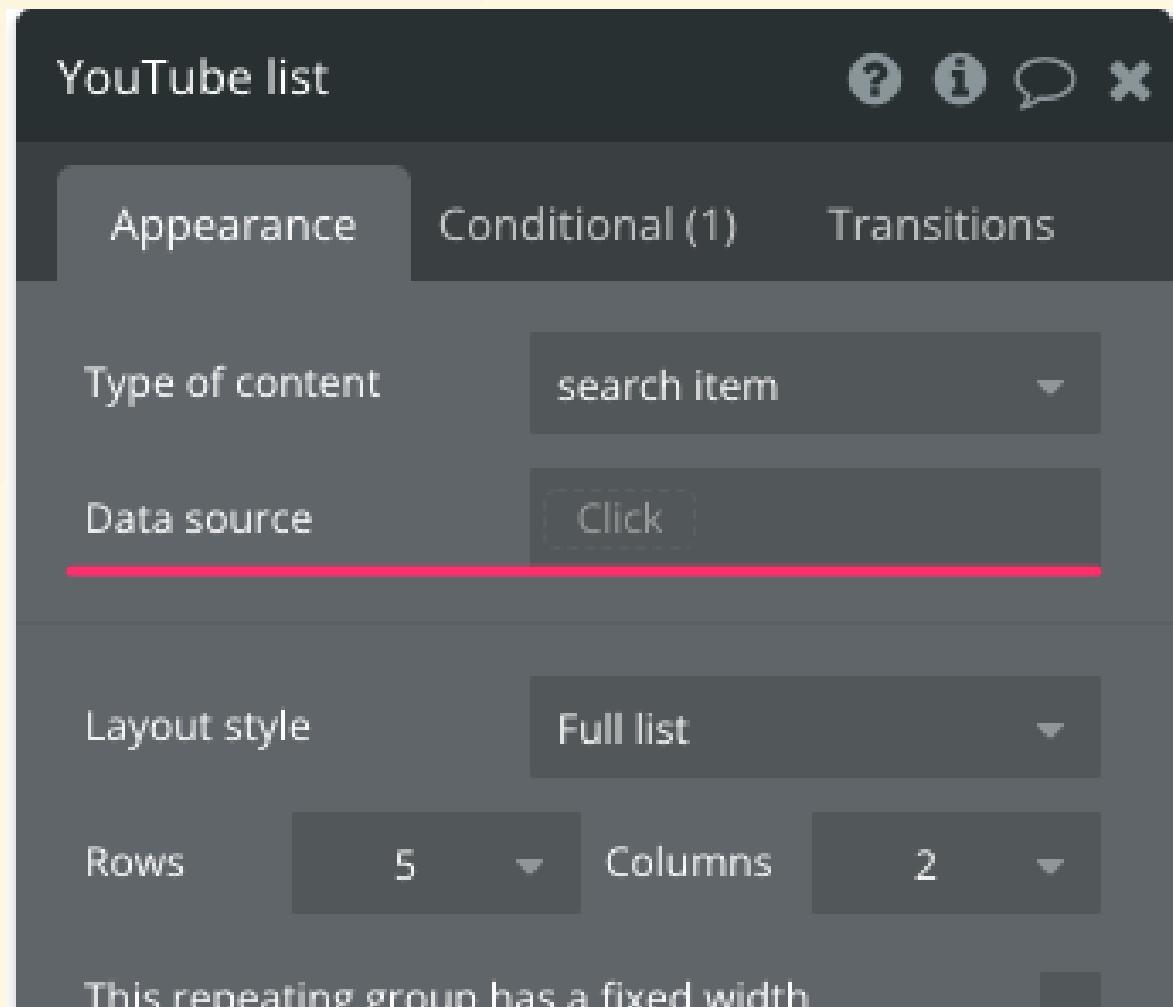
Include errors in response and allow workflow actions to continue

Capture response headers

This call has been modified since you last initialized; consider reinitializing.

Reinitialize call Manually enter API response

- 続いて Repeating Group の Data source を空にします
- これは、新たに設ける検索機能のワークフローの中で、ここで表示するデータを指定するためです

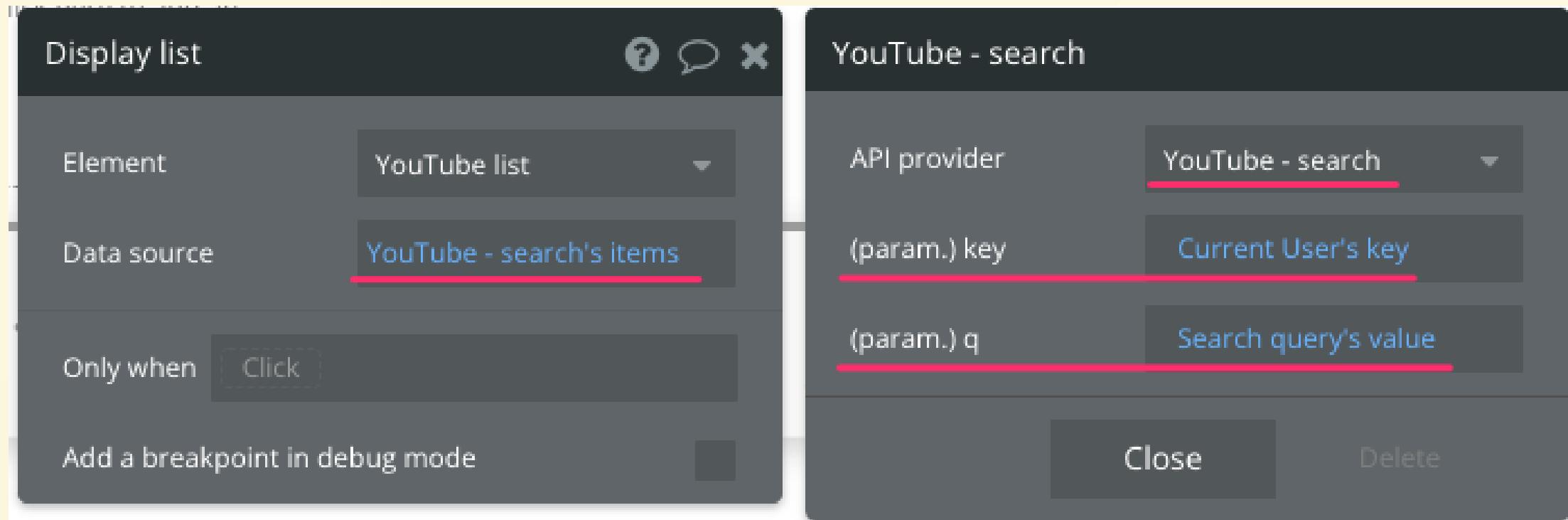


- 次にキーワード検索が行われた時のワークフローの設定です
- 今回は要素に対するアクションとなるため "Element Actions" の中にある "Display list" を選択します

Search for an action...		
	All elements	
👤 Account	Show	Hide
↗ Navigation	Toggle	Animate
💾 Data (Things)	Scroll to	Set state
✉ Email	<i>Input</i>	
💳 Payment	Set focus	Reset inputs
📊 Analytics	<i>Group</i>	
⚡ Element Actions	Display data	Reset data
<i>Repeating Group</i>		
🛒 Plugins	Display list	Show previous
⚙ Custom Events	Show next	Clear list
	Go to page	Scroll to entry

- Data source には Get data from an external API を選択し、API 経由のデータ表示とします
- そして API provider に YouTube - search を選択すると、先ほどはなかった 2 つの param が表示されるはずです
- これは先ほど YouTube API 側の設定で動的に指定したいパラメータについて Private のチェックを外したためとなります
- ここで指定する値はいつもの Dynamic data ですね

- 設定するとこんな感じです



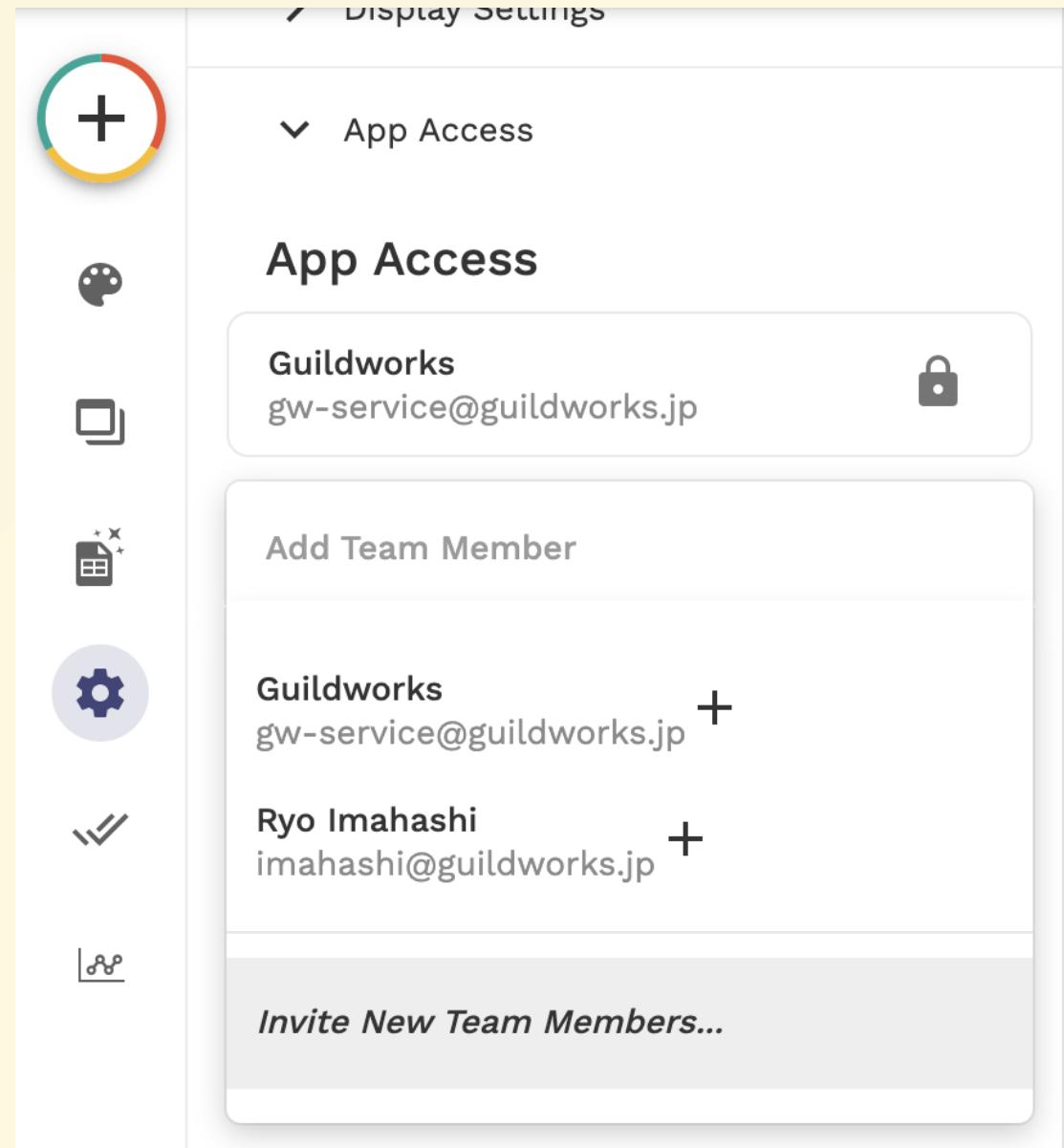
- いかがでしたでしょうか？
- API 連携を行うことで、Bubble で出来る幅がさらに広がったと思います！
- 合宿でも外部の API を使うチームがあるかもしれません、そのときは今日学んだことを活かして、API 連携を実践してみましょう！

ノーコードツールでの共同作業のTips

チームでノーコードツールを使って共同作業をする時に役立つ情報を
お伝えします。

(再掲) Adaloでのチームメンバーの招待方法

- チームメンバーと一つのアプリを共同編集するには、
Settings > AppAccess > Add Team Member > Invite New Team Memberを選択してチームメンバーのメールアドレスを入力しましょう。



Bubbleでのチームメンバーの招待方法

- チームメンバーと一つのアプリを共同編集するには...

	Free	Personal	Professional	Production
Access to live version	—	✓	✓	✓
Custom domain	—	✓	✓	✓
Server capacity	Basic	Basic	3 reserved units	10 reserved units
Ability to add capacity	—	—	Up to 7 units	Up to 30 units
Branding & SEO	Bubble branding	White labeling	White labeling	White labeling
Scheduled workflows	—	Run monthly	Run daily	Run daily
API	—	✓	✓	✓
Password protection	Editor-only	Editor & Run-mode	Editor & Run-mode	Editor & Run-mode
Multiple dev. versions	—	—	2 versions	20 versions
Application editors	1	1	2	15
Import / Export CSV	—	✓	✓	✓
Data backup & restore	—	Last 7 days	Last 30 days	Last 365 days
Logs	30 minutes	24 hours	14 days	14 days
Main app + sub apps	—	—	—	✓
File storage	0.5 GBs	10 GBs	20 GBs	50 GBs
Support	Community	Email	Email	Priority email
Monthly price	Free	\$25 ¹ / \$29	\$115 ¹ / \$129	\$475 ¹ / \$529

Adaloでの同時編集の注意点

- 他の人が行った編集は自分の画面にリアルタイムで反映されません。反映させるためにはリロードが必要です。
- 同じ画面を同時編集すると、先に行われた編集が後から行われた編集に上書きされて、最後に編集した人の画面の状態になります。
 - 編集したコンポーネントは別でも、先に行われた編集は取り消されます。
- Actionも、同じ画面に同時に追加すると、後の方だけが保存されます。
- Databaseも、同時編集した場合は最後に編集した人の状態で上書きされます。

- 別画面の同時編集なら、どちらも保存されるので大丈夫です。編集する時は、どの画面かをチームメンバーに共有しましょう。画面ごとに担当者を決めるのもおすすめです。
- 編集する画面を切り替える前には、画面をリロードして最新の状態を反映しましょう(そうすることで、他の人が編集した画面を自分が古い状態に戻してしまうことを防げます)
- 他の人の担当画面をうっかり触ってしまわないように、Canvas上で画面同士の距離を離してから開発するのもおすすめです。
- Databaseは、誰か1人担当者を決めてその人が更新するようにしましょう。

Bubbleでの同時編集の注意点

- 同じエレメントに対する同時編集は避けましょう(後から行われた編集で上書きされます)
 - そのためには、画面ごとに担当者を決めて同じ画面の同時編集を避けるのがおすすめです。

それ以外の注意点は特にありません。

- 編集は他の人の画面にリアルタイムで反映されます。
 - 画面の編集だけでなく、データベースやWorkflowの編集もリアルタイムで反映されます。
- 画面ごとに作業分担すれば、難なく開発できそうです。

チーム開発演習

- DevelopmentPhaseのチームで、新たに一つアプリを開発してみてください。
 - DevelopmentPhaseで開発予定のアプリとは異なるアプリにしましょう。
- AdaloかBubbleのどちらを使うかは自由です。
 - DevelopmentPhaseで使う予定の方を選んでも良いですし、両方を選んでも良いです。
- 全員が手を動かすようにしましょう。

作業分担のやり方の例

- 画面や機能の洗い出しとデータベース設計は最初にみんなで一緒にやりましょう。
- 画面ごとに担当者を決めましょう(登録画面担当、一覧画面担当、詳細画面担当、更新画面担当など)
- 複数人で一つのPCを使って作業をするというやり方もあります。その場合、PC操作をする人と指示をする人は定期的に交代しましょう。



演習結果の発表

チームごとに演習で作ったアプリについて発表してください。

まとめ

- Learning Phaseのレクチャーは以上になります。
 - AdaloやBubbleをマスターできましたか？
 - チーム開発のやり方のイメージがつきましたか？
- 次回からはDevelopment Phaseとして、各チームがプレゼンをしたアイデアのブラッシュアップや開発をしていきます。
 - それに向けて、自分たちのアイデアの再確認や、開発に使用するツール(Adalo?Bubble?それ以外?)の選定をしておきましょう。

以上です！

お疲れさまでした！