

Programming Boot Camp

Advanced Lecture

Tokyo Institute of Technology 2021/11/27

Ryo Imahashi, Naotake Kyogoku

Table of Contents

- Review and confirmation of what we will do
- External Integration
- Tips for collaborating with no-code tools
- Team development exercise
- Summary

Reflection and confirmation of what we will do this time

- In the previous four lectures, we have learned how to develop applications using two no-code tools, Adalo and Bubble.
- In this lecture, we will first learn how to integrate Adalo and Bubble with external services.
- Next, we will show you some tips and cautions about Team collaboration.
- Finally, we will practice and present application development in Development Phase teams.

External Integration

If there is something you cannot achieve with Adalo or Bubble alone, you may be able to achieve it by collaborating with external services.

We will show you how to do that.

External Integration with Adalo

The following three methods of external integration are introduced.

- Marketplace external integration component
- Custom Action
- External Collection

Marketplace external integration components

You can add a component that enables external integration from Marketplace.

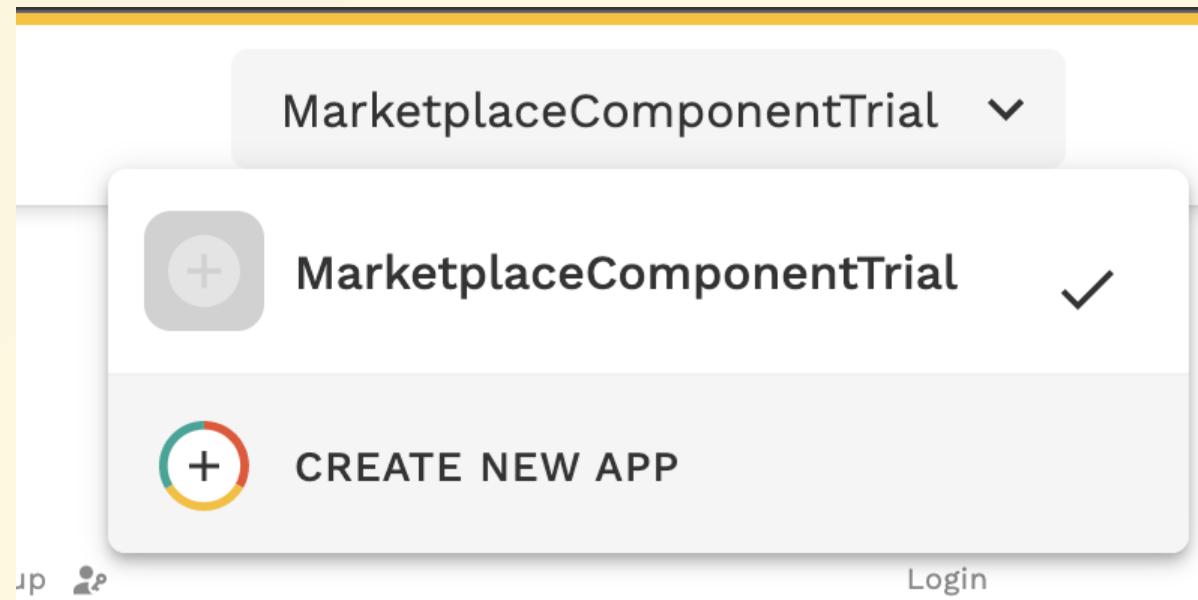
The screenshot shows the Adalo Marketplace interface. At the top left is the 'Marketplace' logo. To its right is a search bar with the placeholder 'Search Marketplace...'. Below the search bar are two main components:

- Map**: A circular icon containing a Google Maps pin. Below it is a brief description: "Display one or multiple markers using Google Maps." An image of a map with a pin is shown. At the bottom are two buttons: "By Adalo" and a green "INSTALLED" button.
- Calendar**: A circular icon containing a calendar icon. Below it is a description: "Display a list of events on a calendar. Also includes option for an agenda view." An image of a calendar grid is shown. At the bottom are two buttons: "By Adalo" and a green "INSTALL" button.

At the bottom of the screen are three additional component cards, partially visible, featuring the Google logo and a document icon. At the very bottom are two buttons: a teal rounded rectangle with the text "CREATE YOUR OWN COMPONENT" and a teal rounded rectangle with the text "DONE".

First, let's create a new application.

- Login to Adalo
Go to
<https://app.adalo.com/login>
- Click CREATE NEW APP



- Select Native Mobile App for Platform Click

1 Platform ————— 2 Template ————— 3 Branding

What kind of app do you want to make?



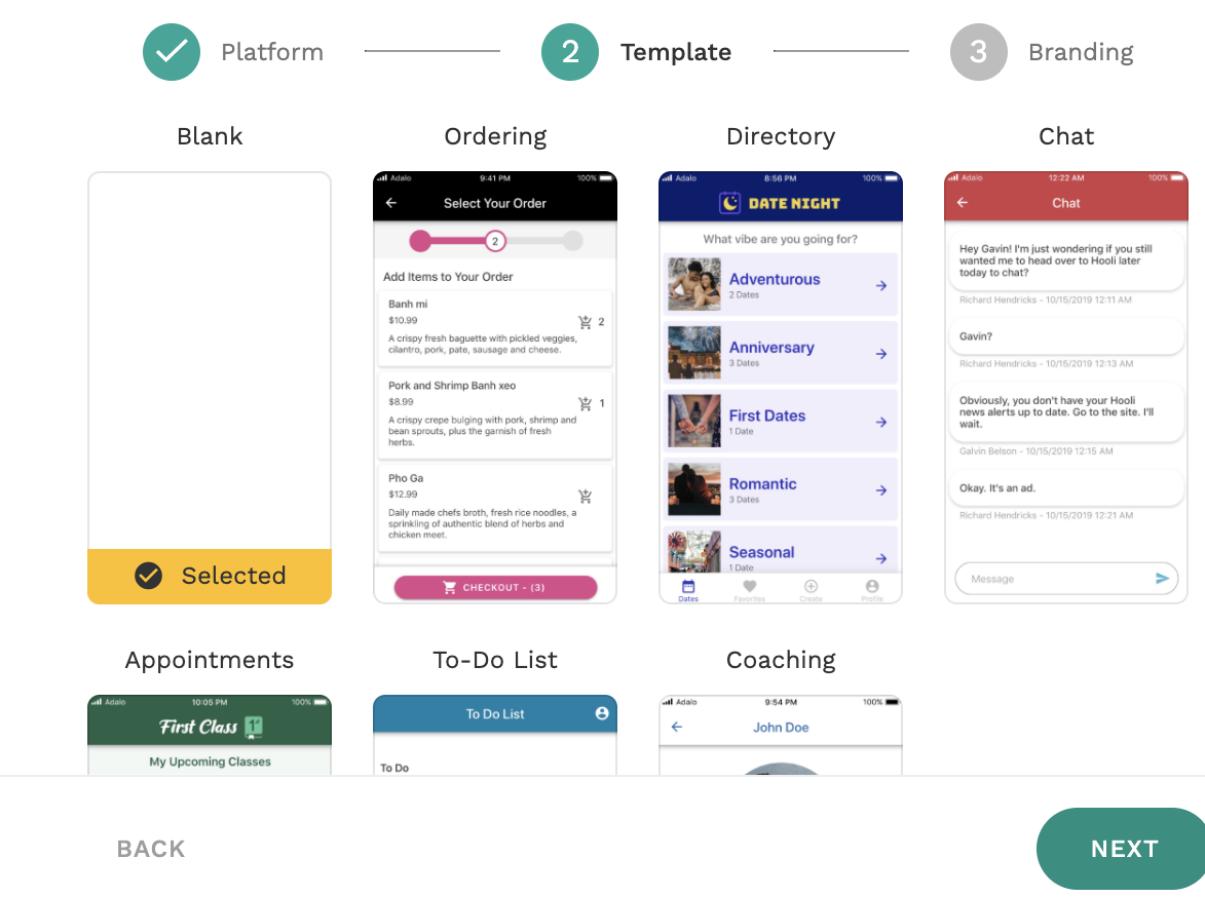
Native Mobile App



Desktop Web App

NEXT

- Select Blank for Template.



- For App Name, enter MarketplaceComponentTrial.
- Please set the Color freely.

The screenshot shows a branding configuration screen. At the top, there are three circular progress indicators: 'Platform' (green with checkmark), 'Template' (green with checkmark), and 'Branding' (green with number '3'). Below these are two dropdown menus for color selection. The first dropdown, labeled 'Primary Color' (#00A898), has a teal background. The second dropdown, labeled 'Secondary Color' (#FFC00E), has a yellow background. At the bottom, there are 'BACK', 'ADVANCED OPTIONS', and 'CREATE' buttons.

Platform

Template

3 Branding

App Name

MarketplaceComponentTrial

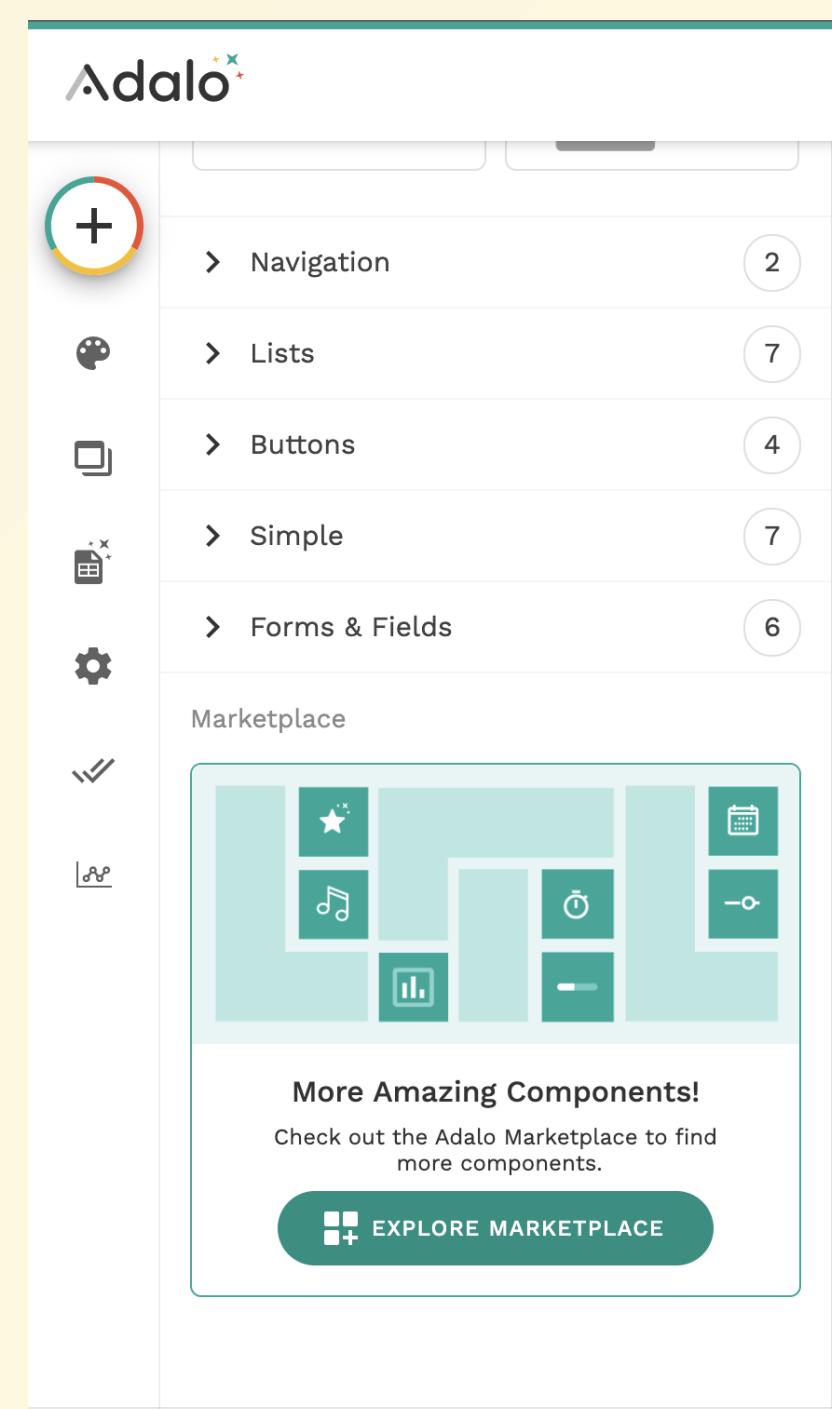
Primary Color
#00A898

Secondary Color
#FFC00E

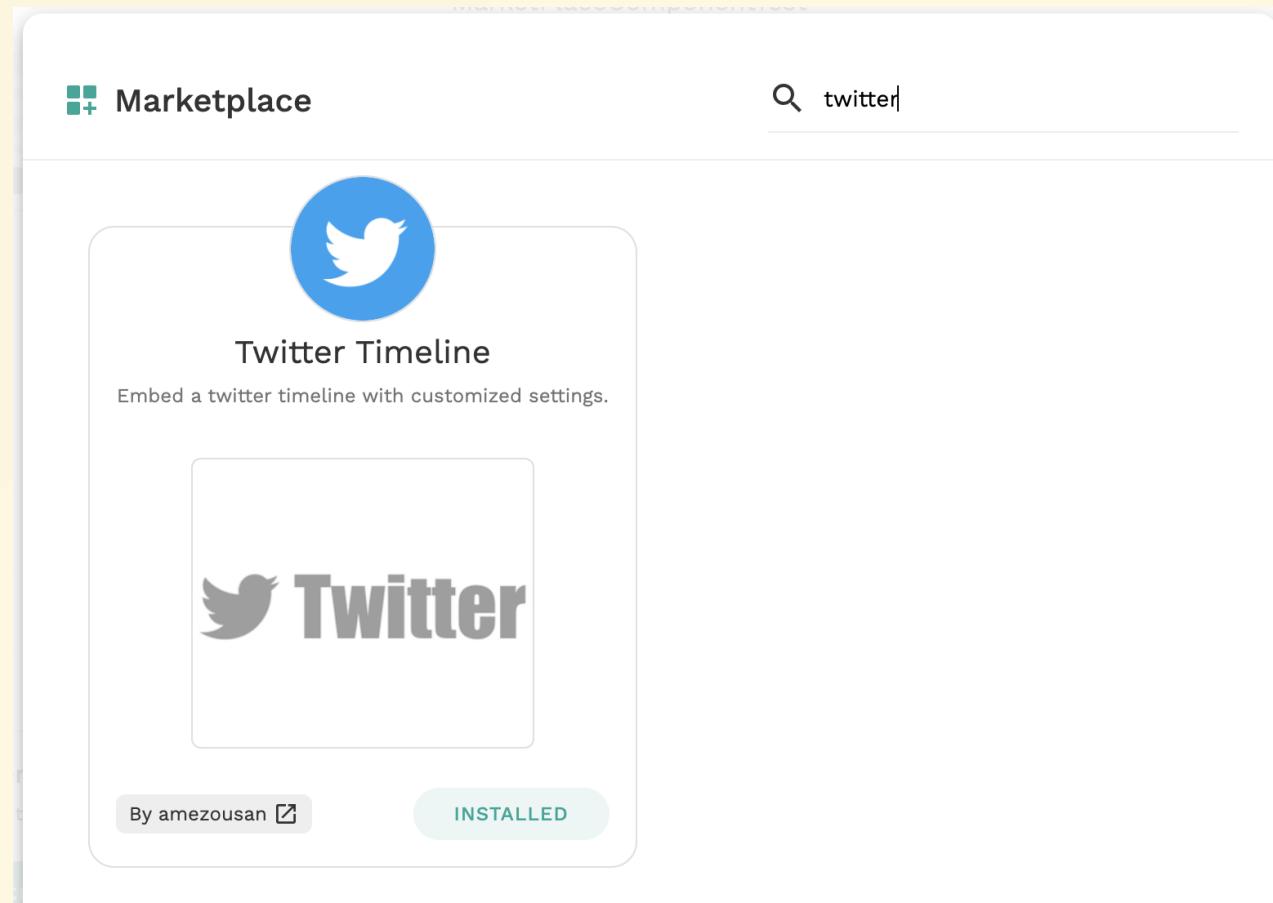
BACK ADVANCED OPTIONS CREATE

Twitter Timeline Component

- Press the + button and click Explore Marketplace in ADD COMPONENT

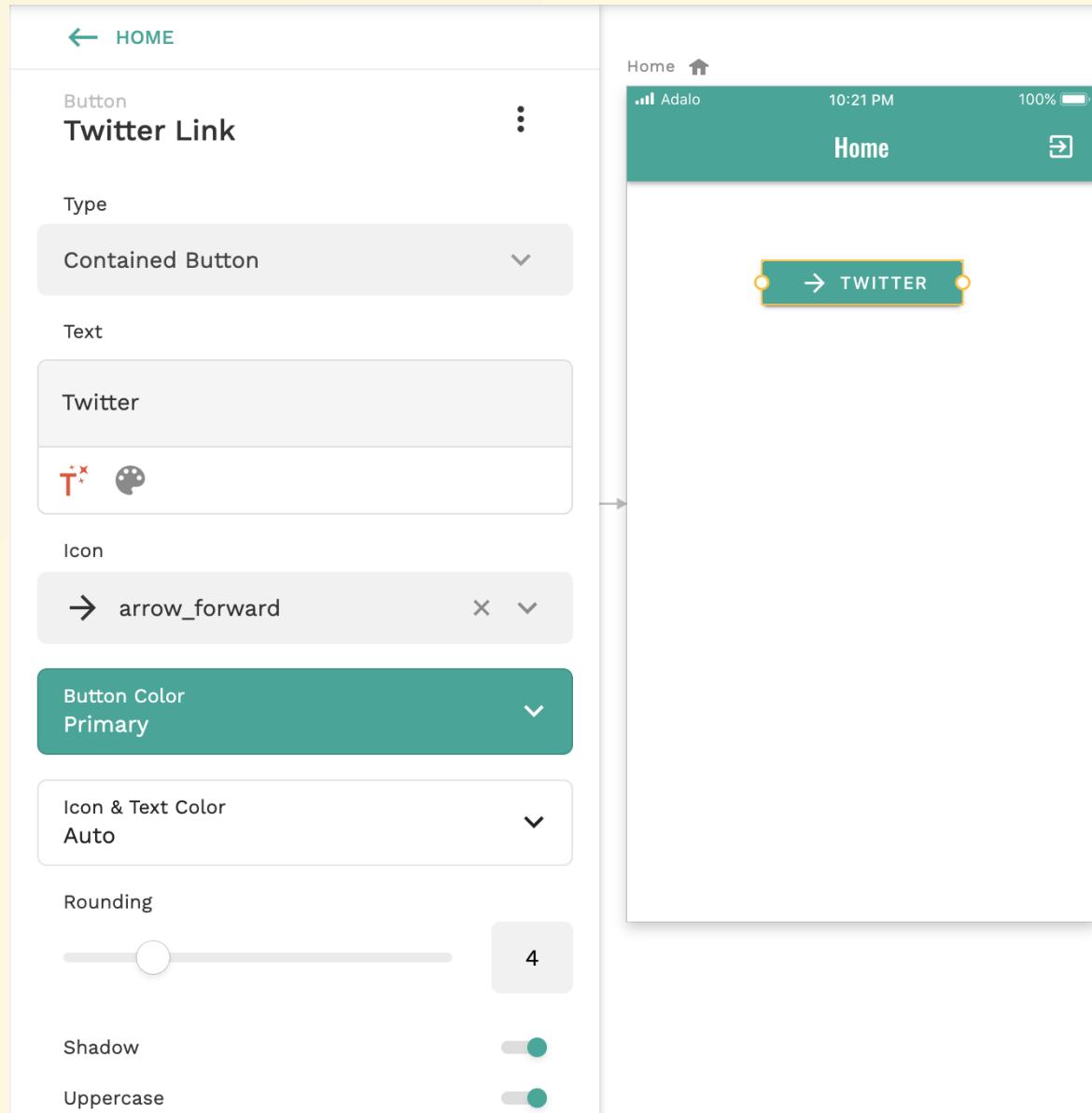


- INSTALL the Twitter Timeline component

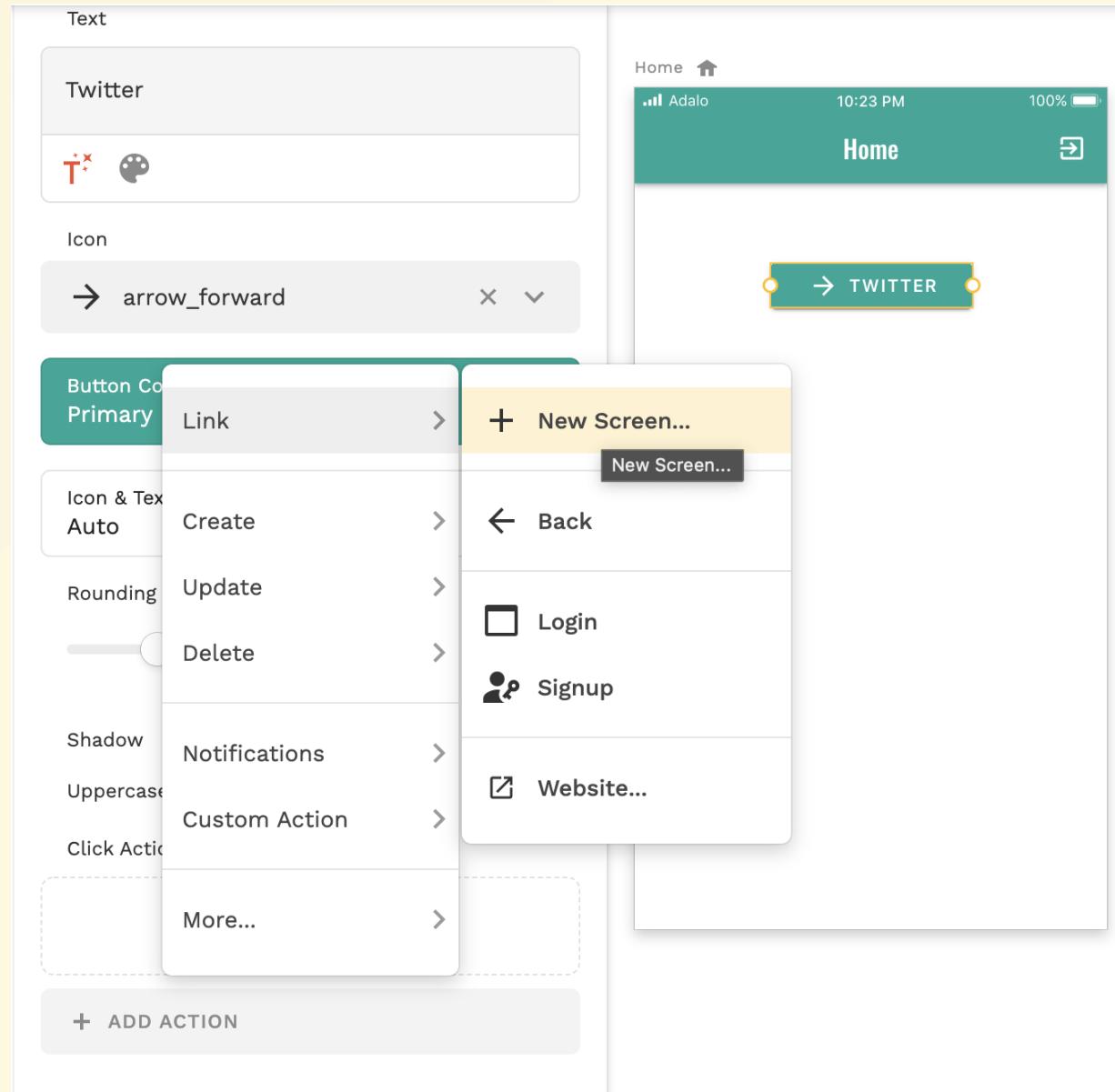


Place the Twitter Timeline component.

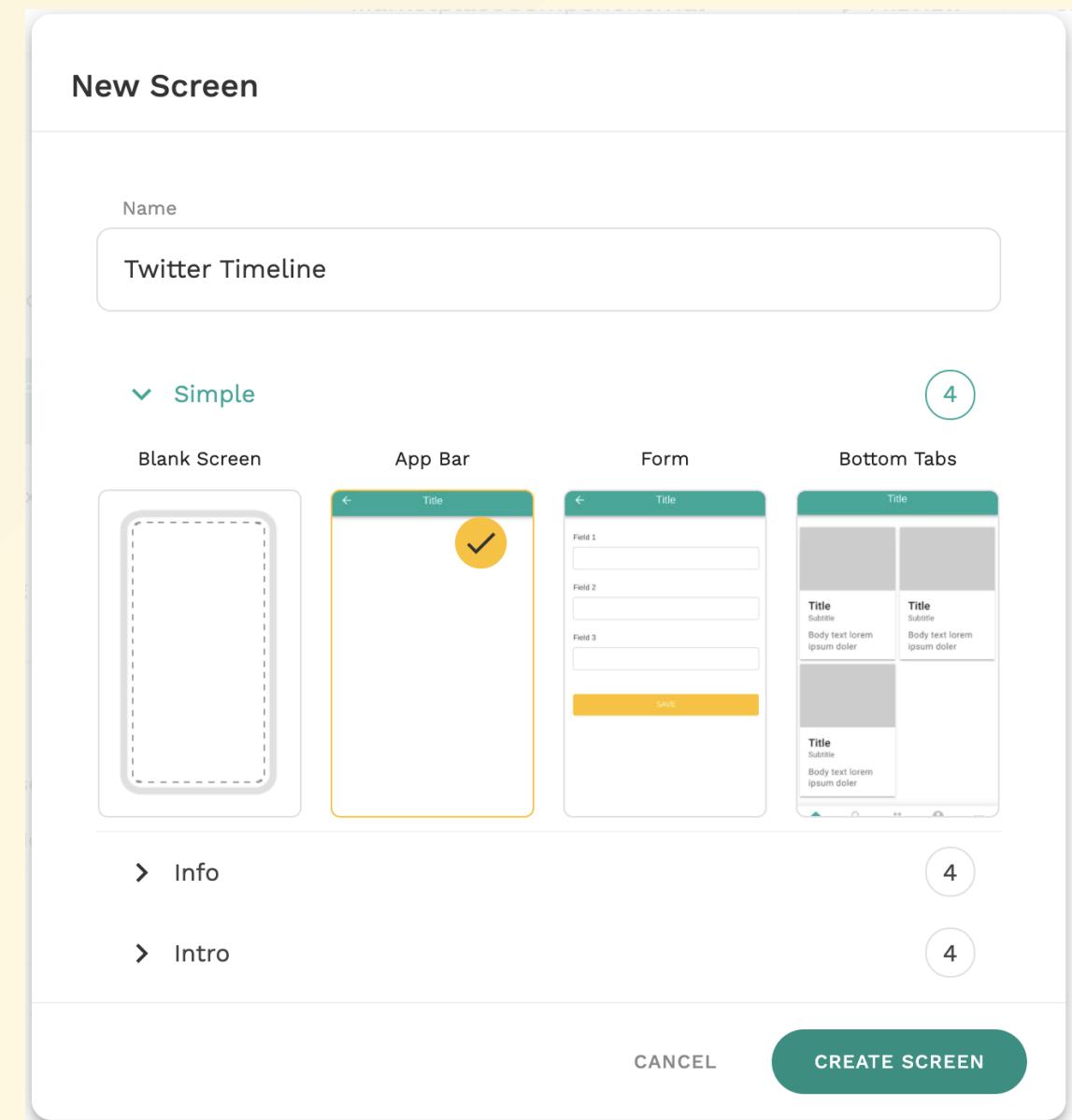
- Add a link button to the Twitter screen to the Home Screen.



- Add Link to New Screen from ADD ACTION



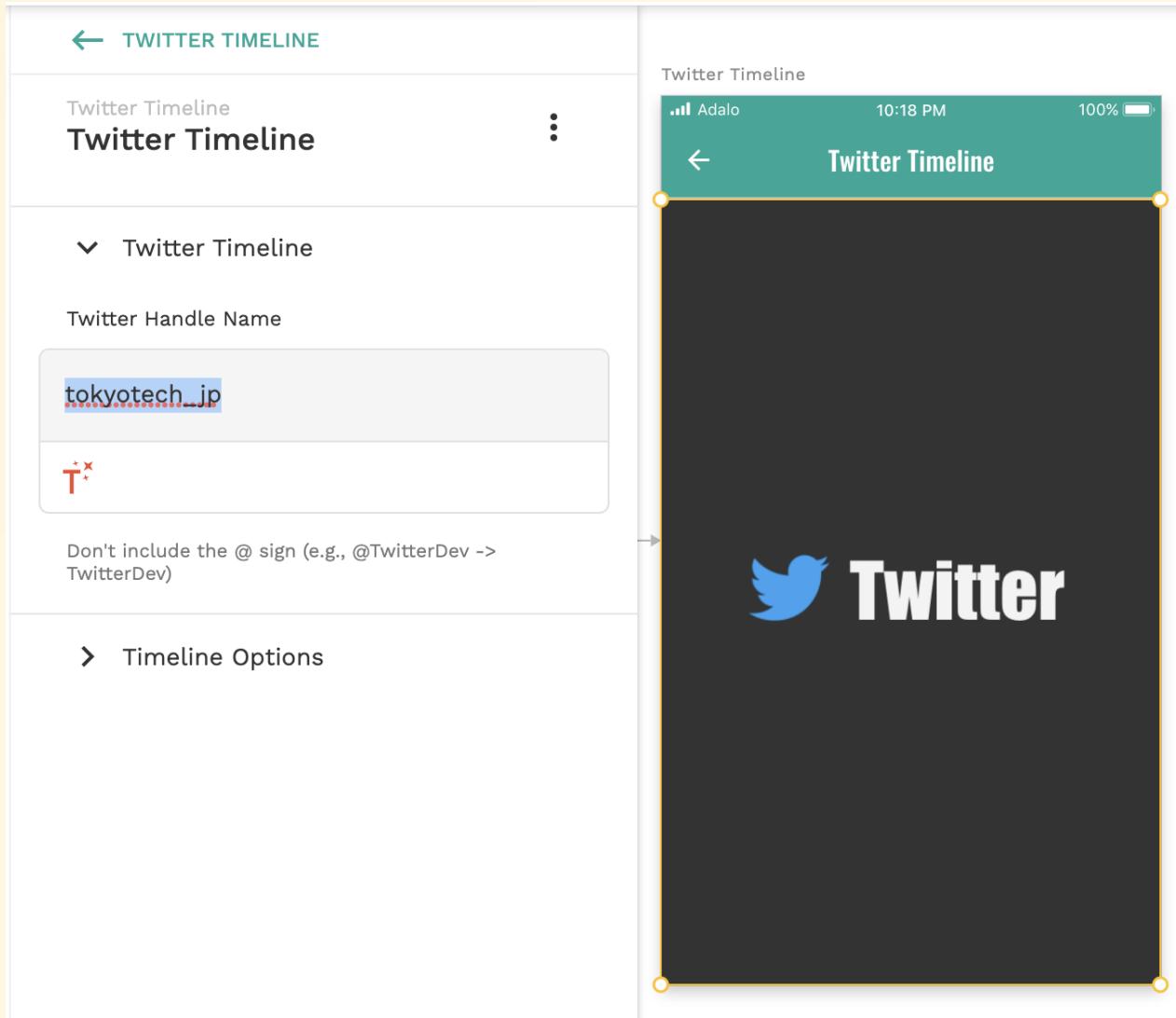
- Select "App Bar" in Template and create a "Twitter Timeline" screen.



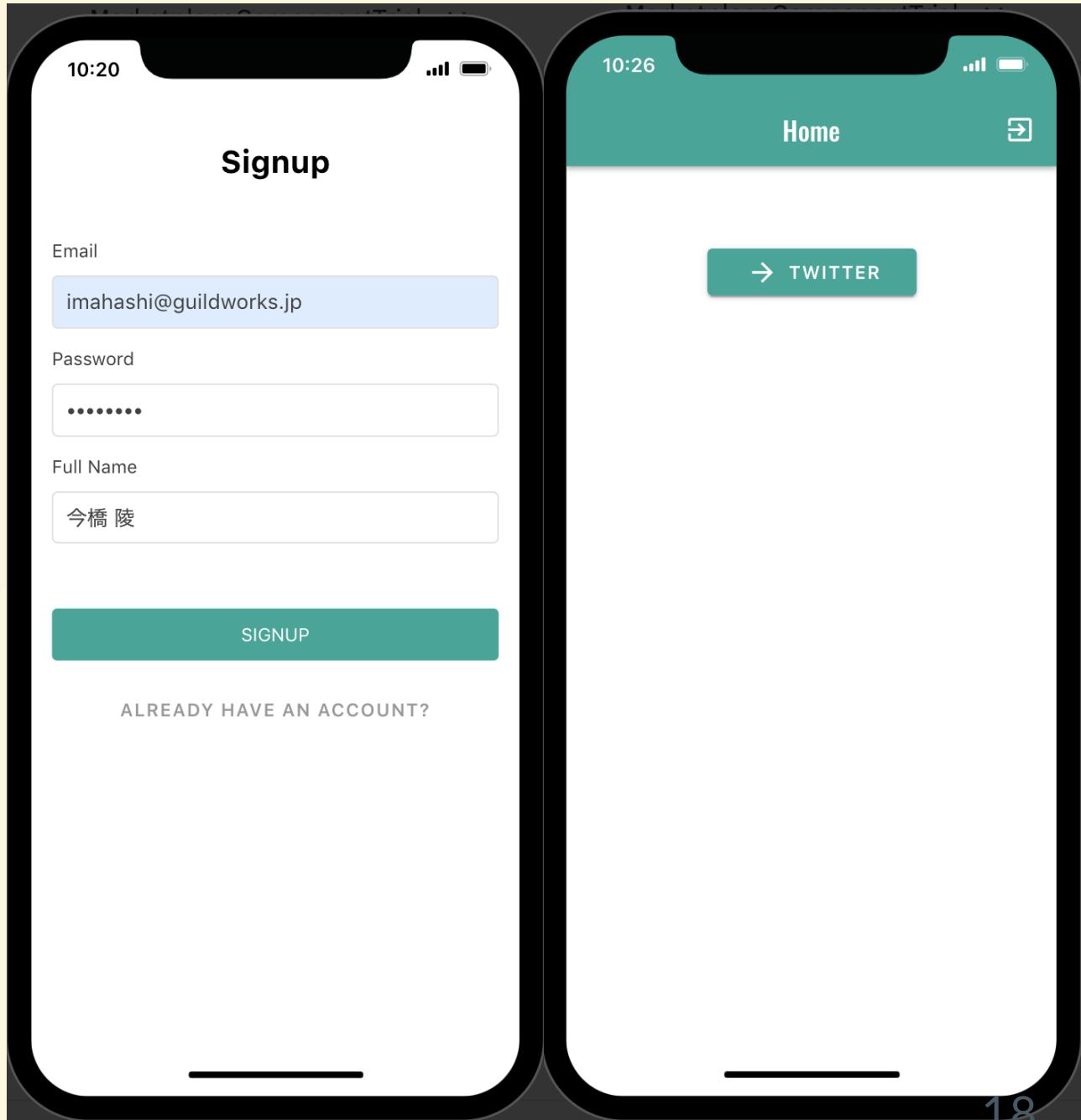
- Place the Twitter Timeline component

The screenshot shows the Adalo component library interface. At the top, there are three navigation items: "Simple" (with a circled '7'), "Forms & Fields" (with a circled '6'), and "Marketplace". Below "Marketplace" is a callout box containing the text "More Amazing Components!" and "Check out the Adalo Marketplace to find more components.", with a "EXPLORE MARKETPLACE" button. A large circular arrow icon is positioned to the right of this callout. Below the callout is a section titled "Installed" (with a circled '1'). Under "Installed" is a component card for "Twitter Timeline". The card features the Twitter logo and the word "Twitter" in a large, bold, sans-serif font. To the right of the card is a preview window showing a mobile device screen with the title "Twitter Timeline" and the Adalo interface.

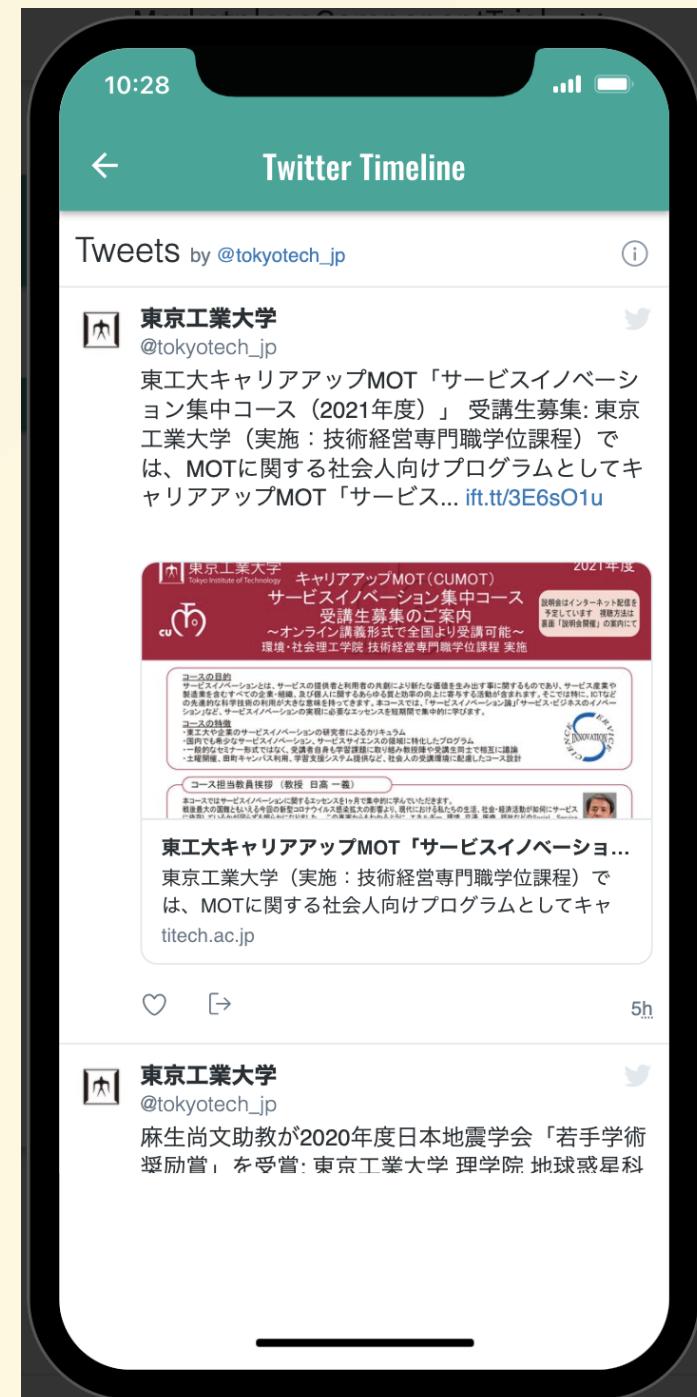
- Enter "tokyotech_jp" as the Twitter Handle Name
 - You can also change it to the Handle Name of your favorite Twitter account!



- Signup in the Preview function and click the Twitter button.



- The posts of the Twitter account with the Handle Name you entered will be listed.



~~Let's modify it so that it lists
the posts of the logged-in
user's own Twitter account.~~

We're going to run out of time
because of a lot of slides, so
we'll skip this part...

- Add a TwitterHandleName Property to the Users Collection.
 - Select Text as the Type

The screenshot shows a database interface with a sidebar of icons. The main area displays a 'Database Collections' section with a 'Users' collection containing 1 Record. The collection view includes fields for Email, Password, Username, and Full Name. A modal window is open to add a new property, with 'Type' set to 'Text' and 'Name' set to 'TwitterHandleName'. The modal has 'CANCEL' and 'SAVE' buttons at the bottom.

Database Collections

Database Collections

Users 1 Record :

Email

Password

Username

Full Name

New Property

Type: Text

Name: TwitterHandleName

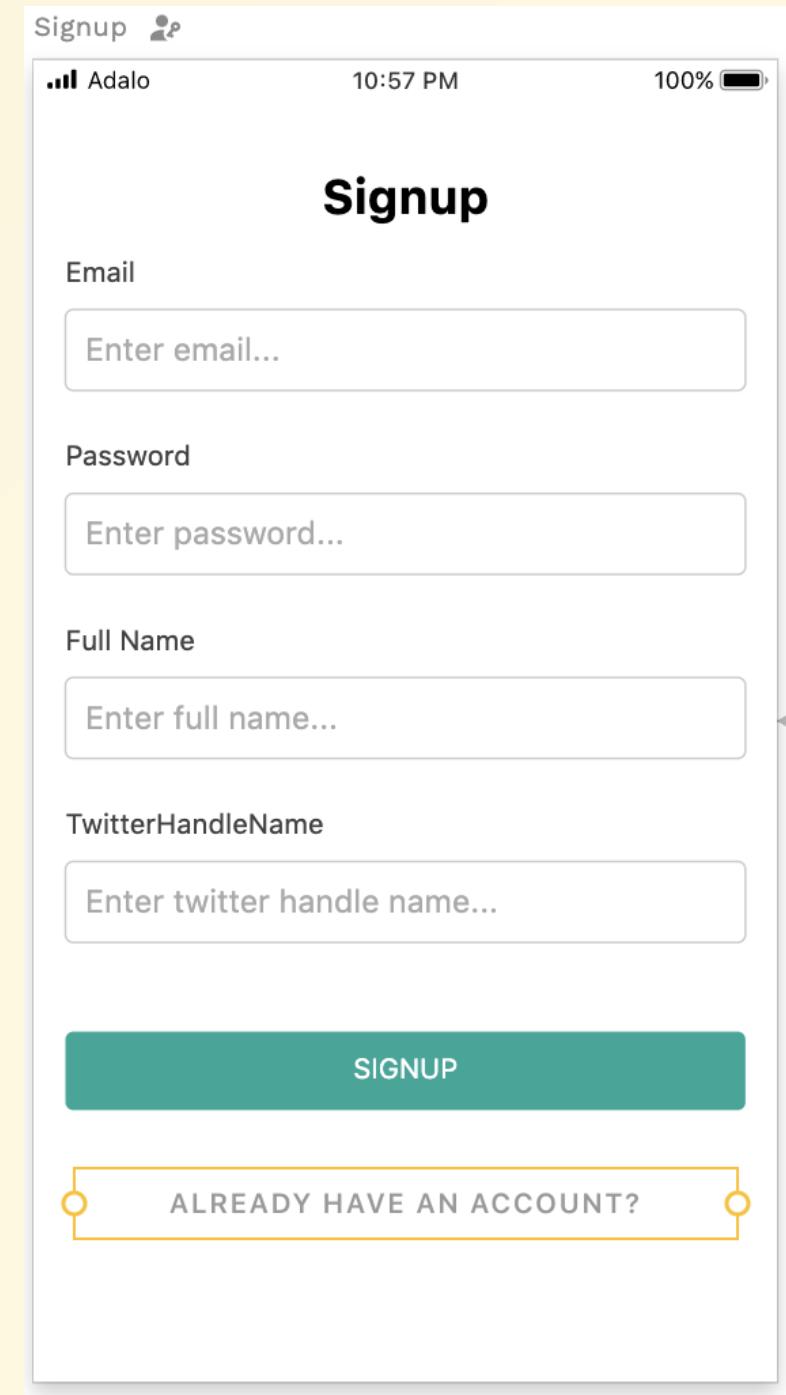
CANCEL SAVE

+ ADD PROPERTY

- Click Form on the Signup screen
- Fields > ADD VISIBLE FIELD > TwitterHandleName
- Change to Not Required

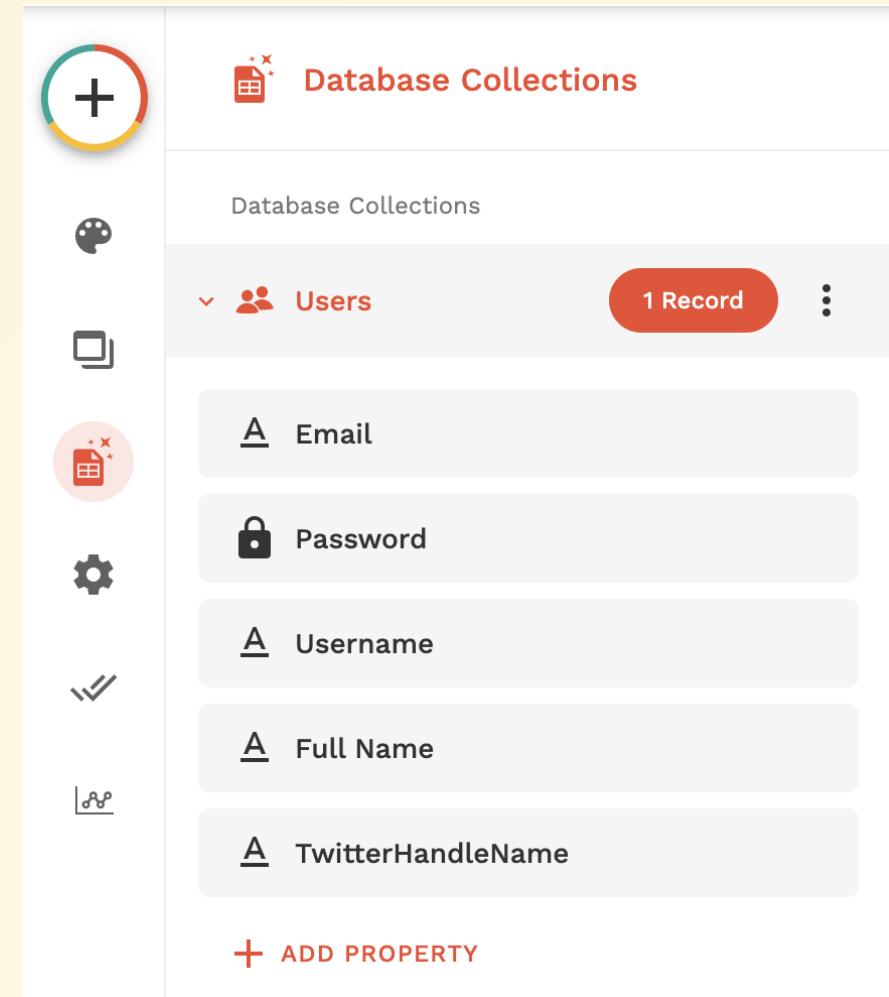
The image displays two side-by-side views of a mobile application interface. On the left is a screenshot of the Adalo app's 'Fields' configuration screen. It shows a list of fields: 'Email', 'Password', 'Full Name', and 'TwitterHandleName'. The 'TwitterHandleName' field is highlighted with a teal background. Below it, detailed settings are shown: Label ('TwitterHandleName'), Placeholder ('Enter twitter handle name...'), Input Type ('Single Line'), and Required Error Text ('Not Required'). A 'DONE' button is at the bottom right. On the right is a screenshot of the 'Signup' screen on a mobile device. The screen has a light gray header with the text 'Signup' and the Adalo logo. The main area contains four input fields: 'Email' (placeholder 'Enter email...'), 'Password' (placeholder 'Enter password...'), 'Full Name' (placeholder 'Enter full name...'), and 'TwitterHandleName' (placeholder 'Enter twitter handle name...'). Below these fields is a teal button labeled 'SIGNUP'. At the bottom of the screen, there is a link 'ALREADY HAVE AN ACCOUNT?'.

- "ALREADY HAVE AN ACCOUNT?" link was overlapping the form, so moved it to the bottom



Let's set the TwitterHandleName for the registered users

- Click the "1 Record" button in the Users Collection



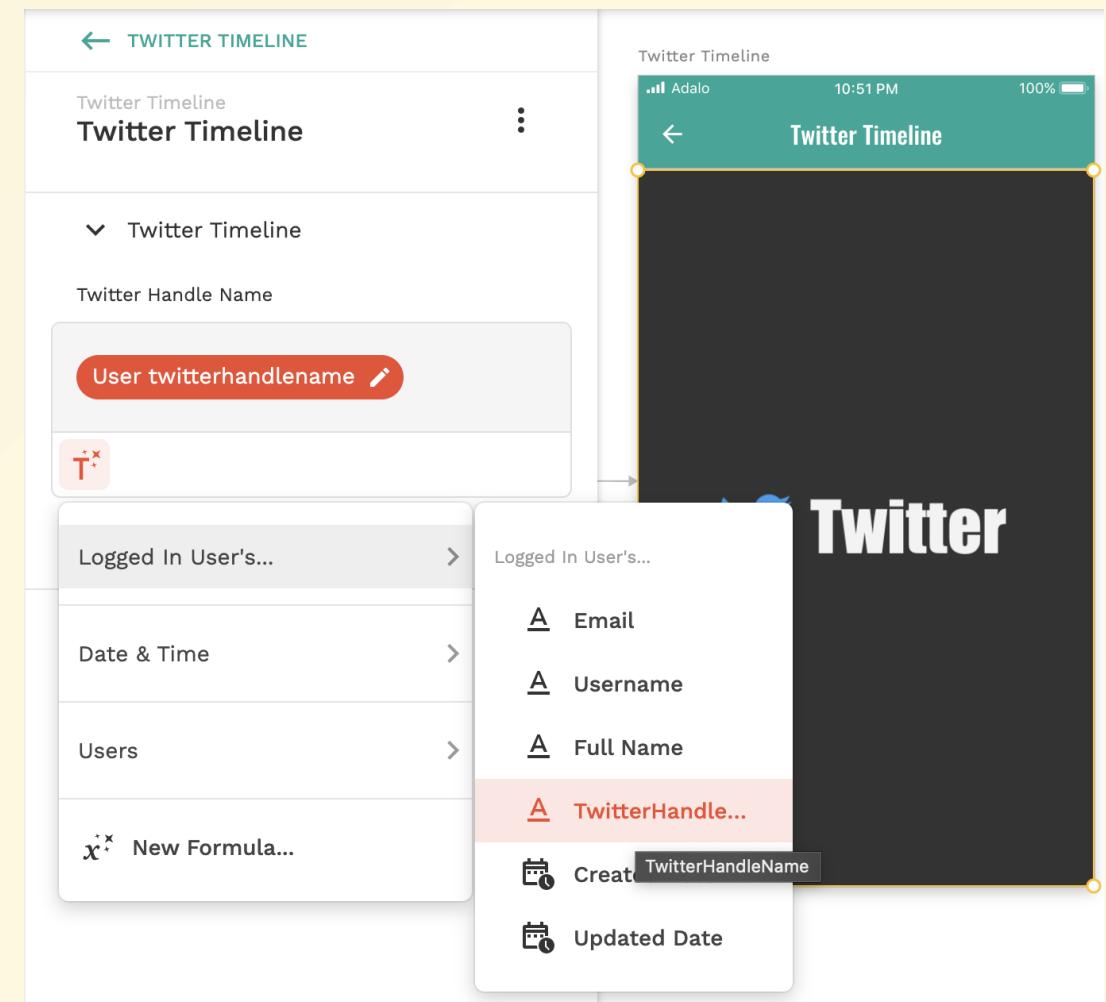
- Click on the record of the registered user.
- Enter TwitterHandleName
(You can use your own account or any account you like.)

Jser

A Email	<input type="text" value="imahashi@guildworks.jp"/>
>Password	<input type="password"/>
A Username	<input type="text"/>
A Full Name	<input type="text" value="今橋 陵"/>
A TwitterHandleName	<input type="text" value="titechcoop"/> S

CANCEL

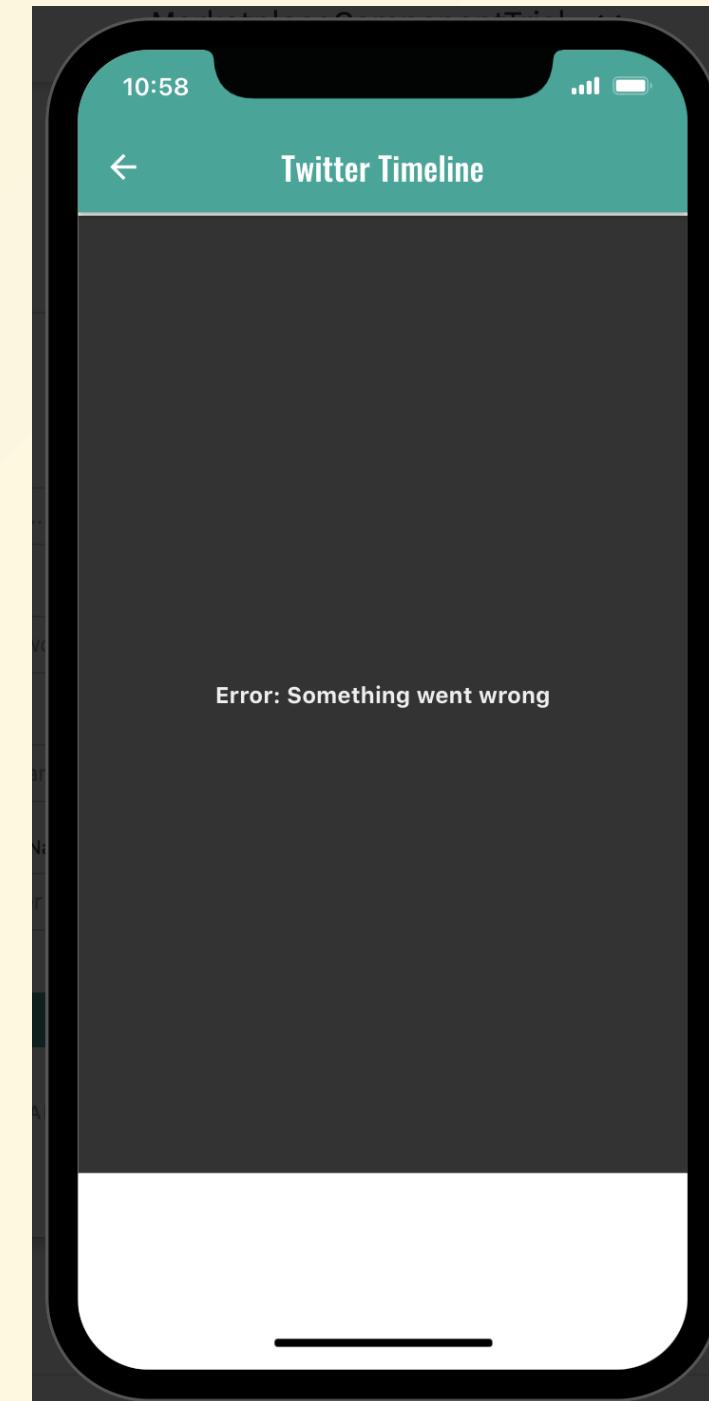
- Change the Twitter Handle Name in the Twitter Timeline component to "Logged In User's TwitterHandleName".



Theoretically, this should display the list of posts of the Twitter account set by the logged-in user himself.

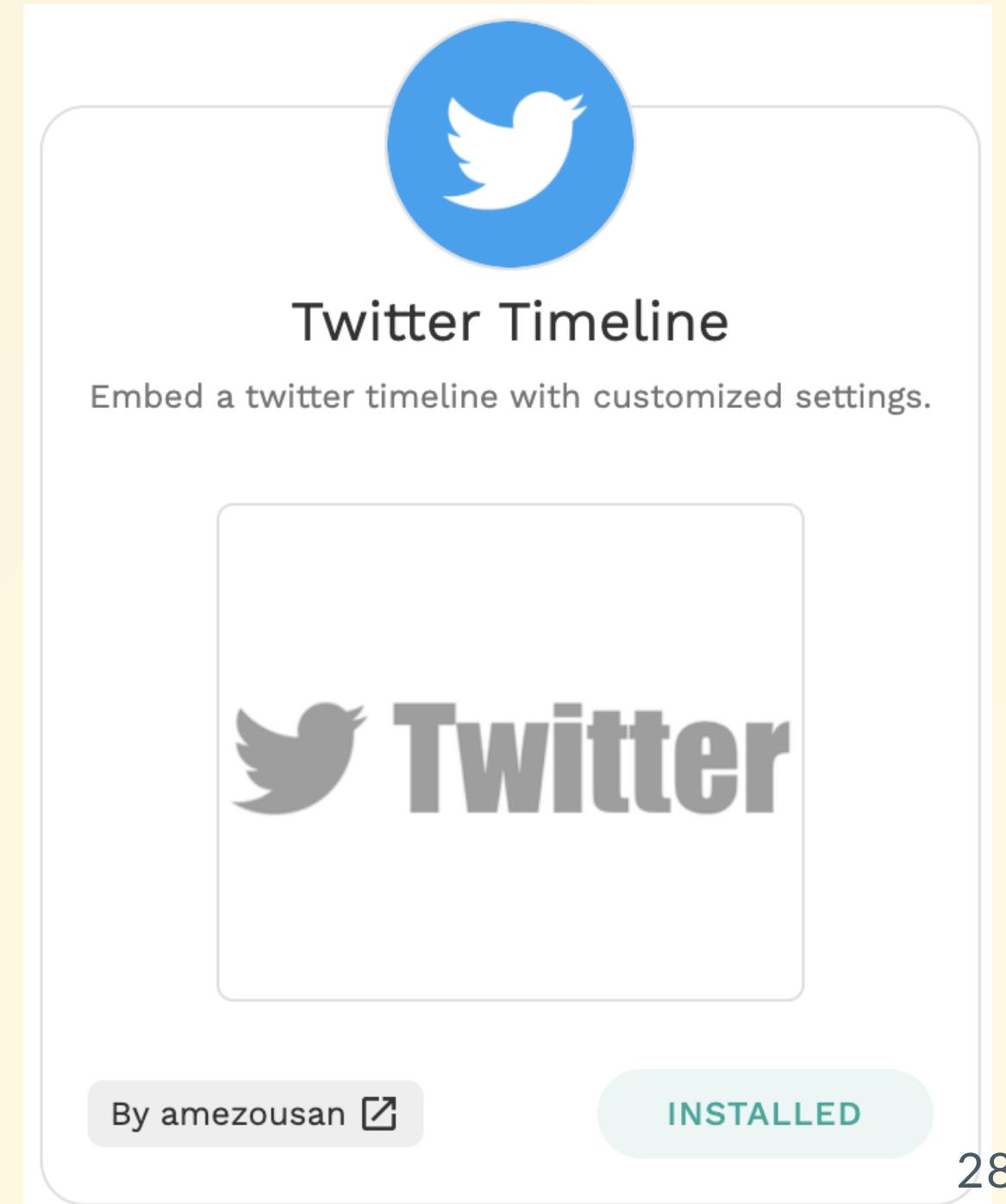
Let's check it out with the Preview function.

I think an error will occur.



Twitter Timeline is a component that has been made available for free by volunteers.

What someone else makes doesn't always work the way you want it to.



The image shows a screenshot of a mobile application interface for the "Twitter Timeline" component. At the top is a large blue circular icon containing the white Twitter bird logo. Below the icon is the text "Twitter Timeline". Underneath that is a subtitle: "Embed a twitter timeline with customized settings." A large rectangular placeholder area is shown below, featuring the Twitter logo and the word "Twitter" in a large, bold, dark font. At the bottom left is a button labeled "By amezousan" with a small edit icon. At the bottom right is a green rounded rectangle button labeled "INSTALLED".

Twitter Timeline

Embed a twitter timeline with customized settings.

By amezousan

INSTALLED

Some of you may be thinking, "If you can't do it, don't introduce it."
Sorry...

After some trial and error, "I found a way to avoid the error!" I thought I had found a way to avoid the error by trial and error, so I wrote about it in the document, but later I realized that it was a misunderstanding. (Browsers have a mechanism called caching, which sometimes shows us something like an illusion.)

I hope you'll keep in mind that it can happen.

Note: The code for the Twitter Timeline component is publicly available, so you may be able to modify it to use it. (I won't do it this time).

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

The screenshot shows the GitHub repository page for 'adalo-twitter-timeline'. The repository is public and has 13 commits. The 'Code' tab is selected. The repository contains files like README.md, index.js, logo.png, package.json, and yarn.lock. The 'About' section describes it as an adalo plugin for embedding a Twitter timeline. It also lists 'adalo-plugin' as a dependency and shows no releases or packages published. The 'Languages' section indicates 100.0% JavaScript.

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.

About

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

adalo-plugin

Readme

Releases

No releases published

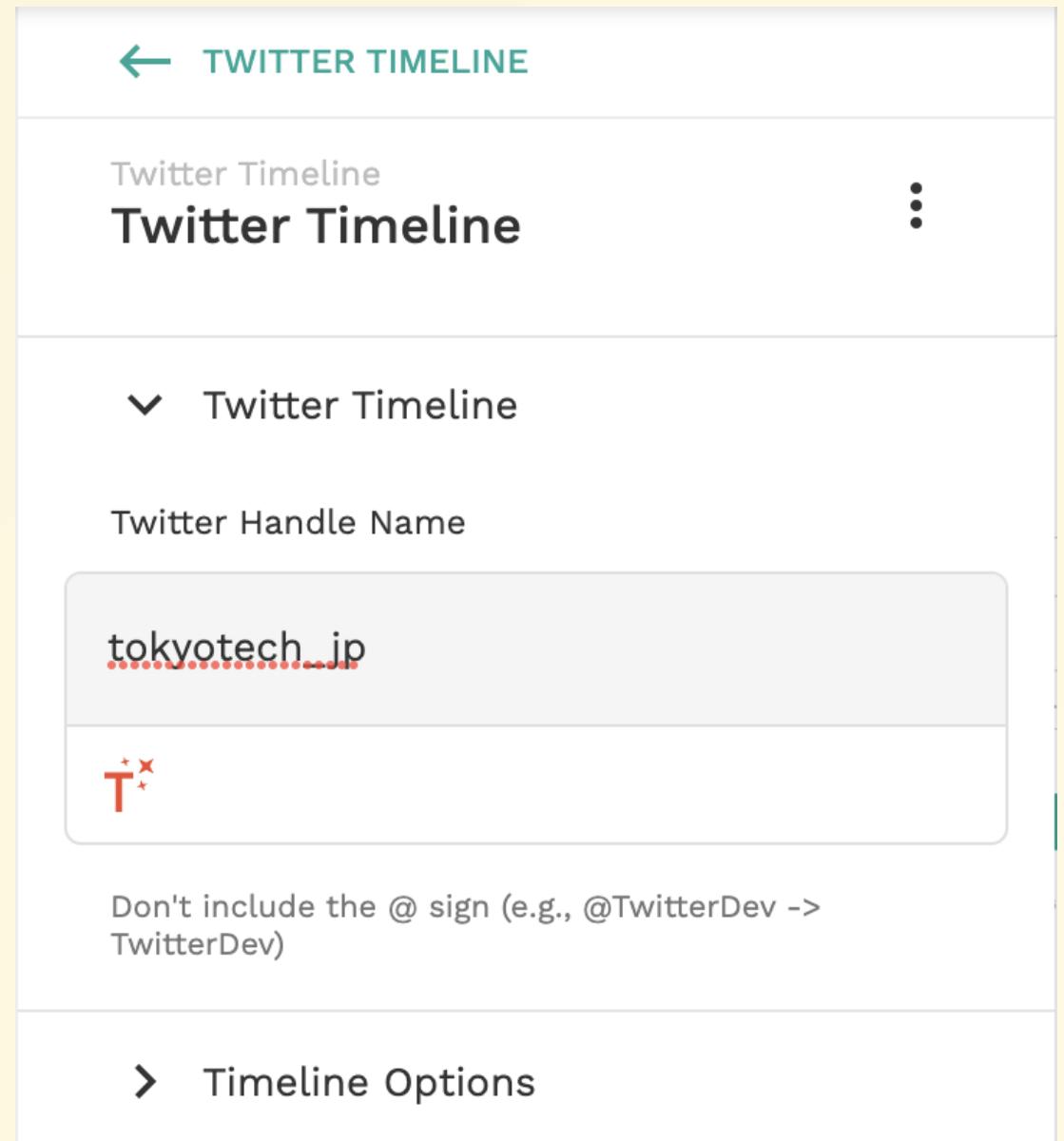
Packages

No packages published

Languages

JavaScript 100.0%

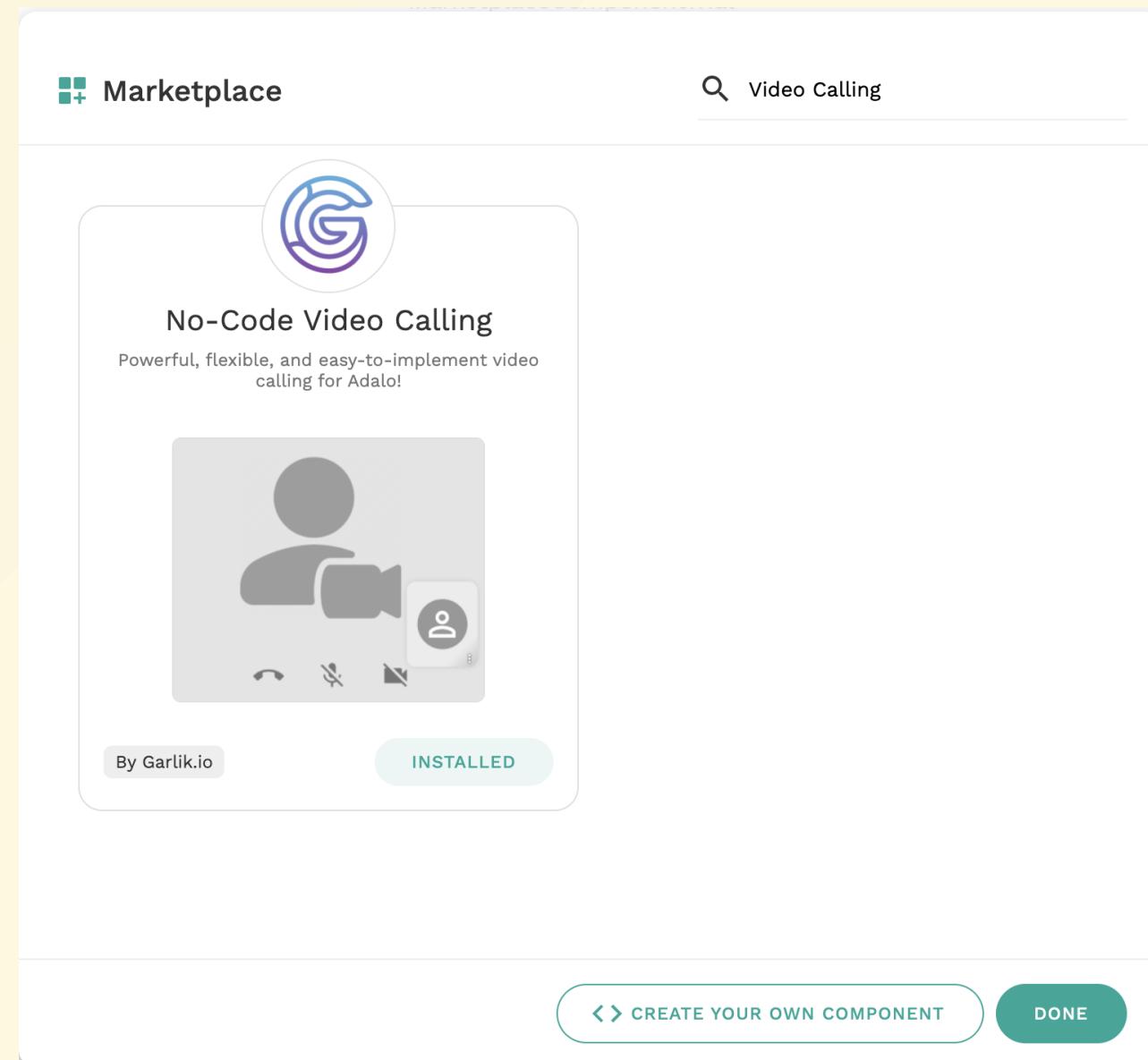
- The Twitter Handle Name in the Twitter Timeline component should be set back to the value we entered directly.



Video Calling

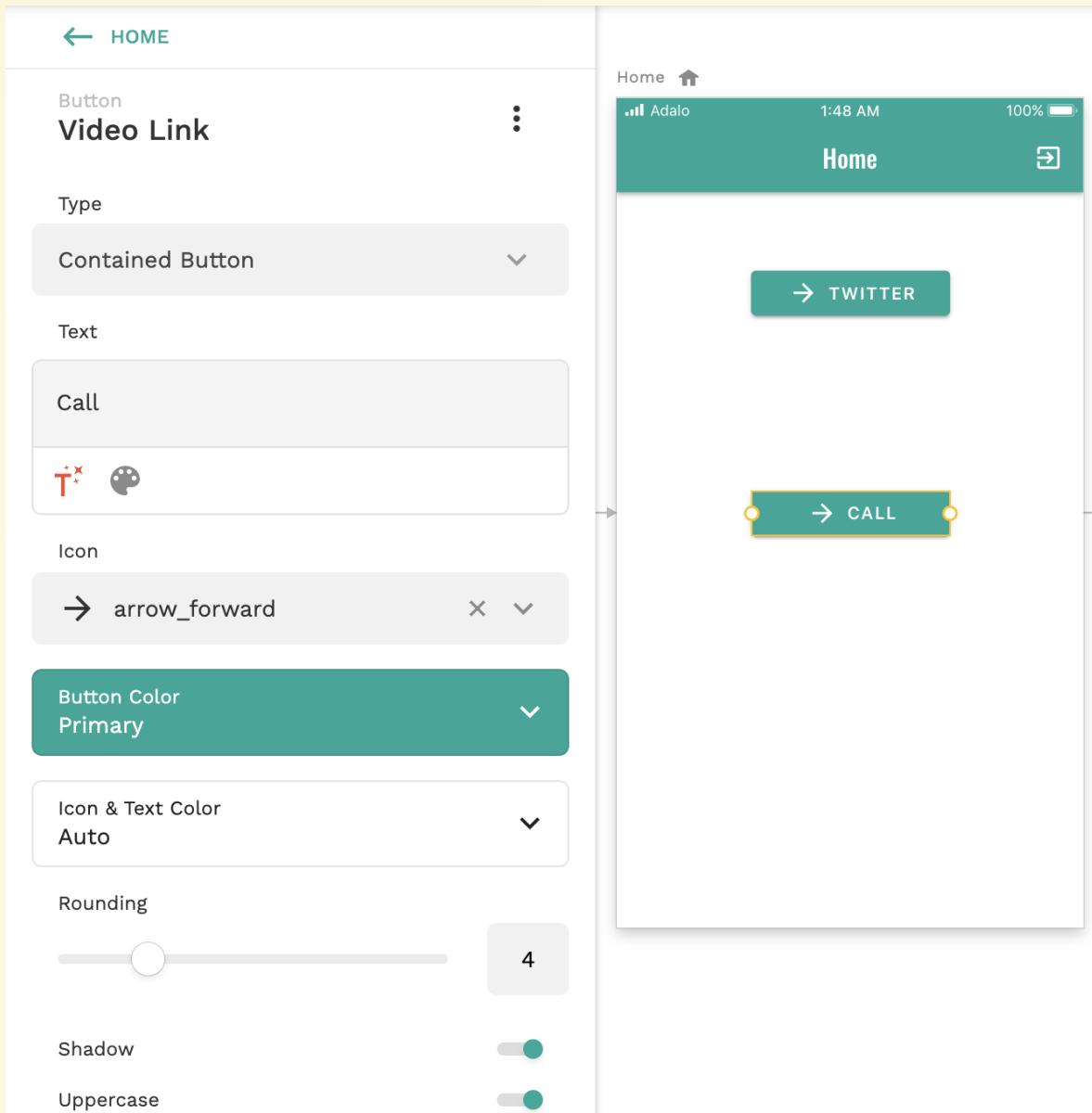
Next, let's take a look at the video calling component.

- INSTALL the "No-Code Video Calling" component in the Marketplace

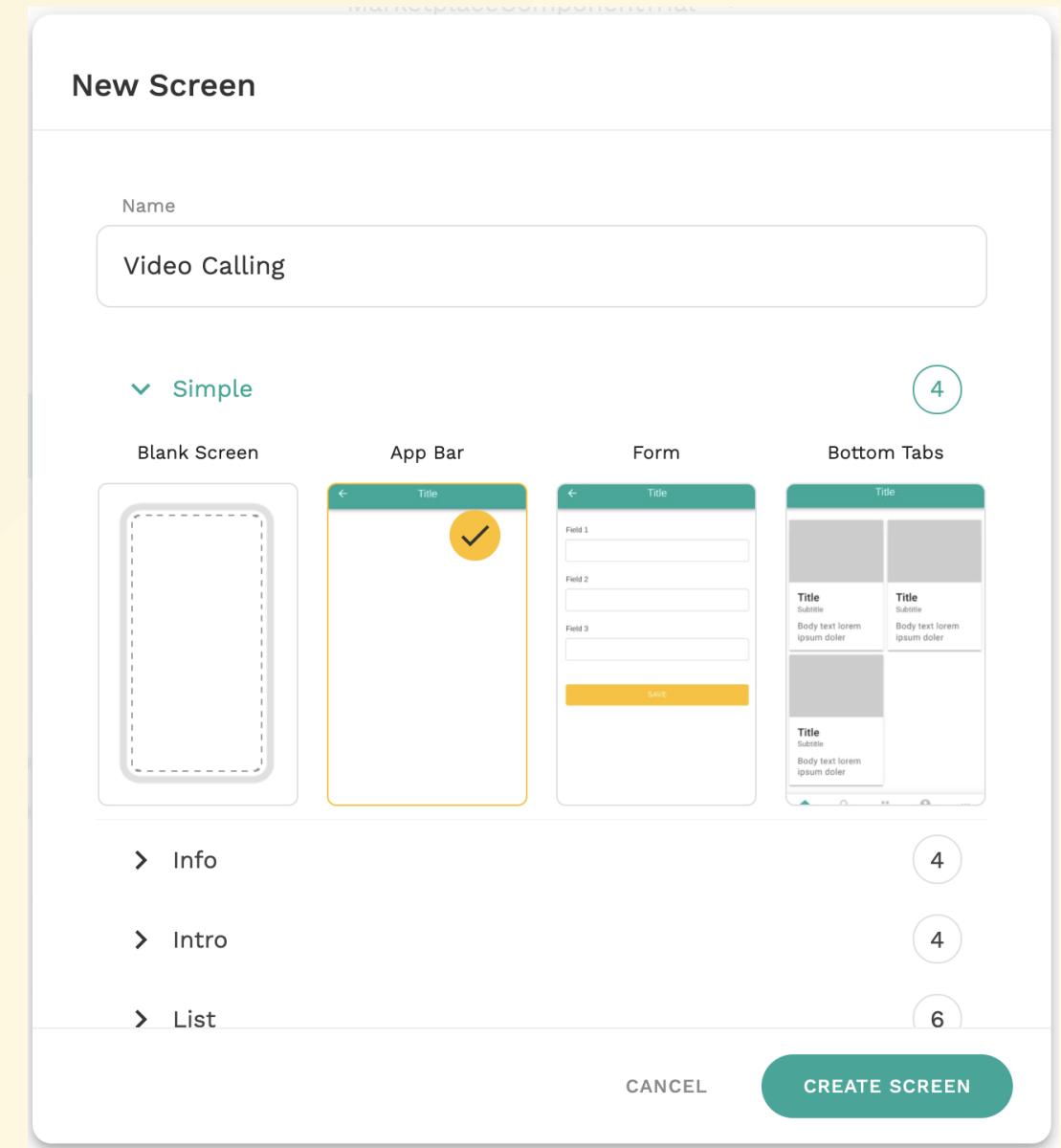


Let's create a new screen to place the video calling component.

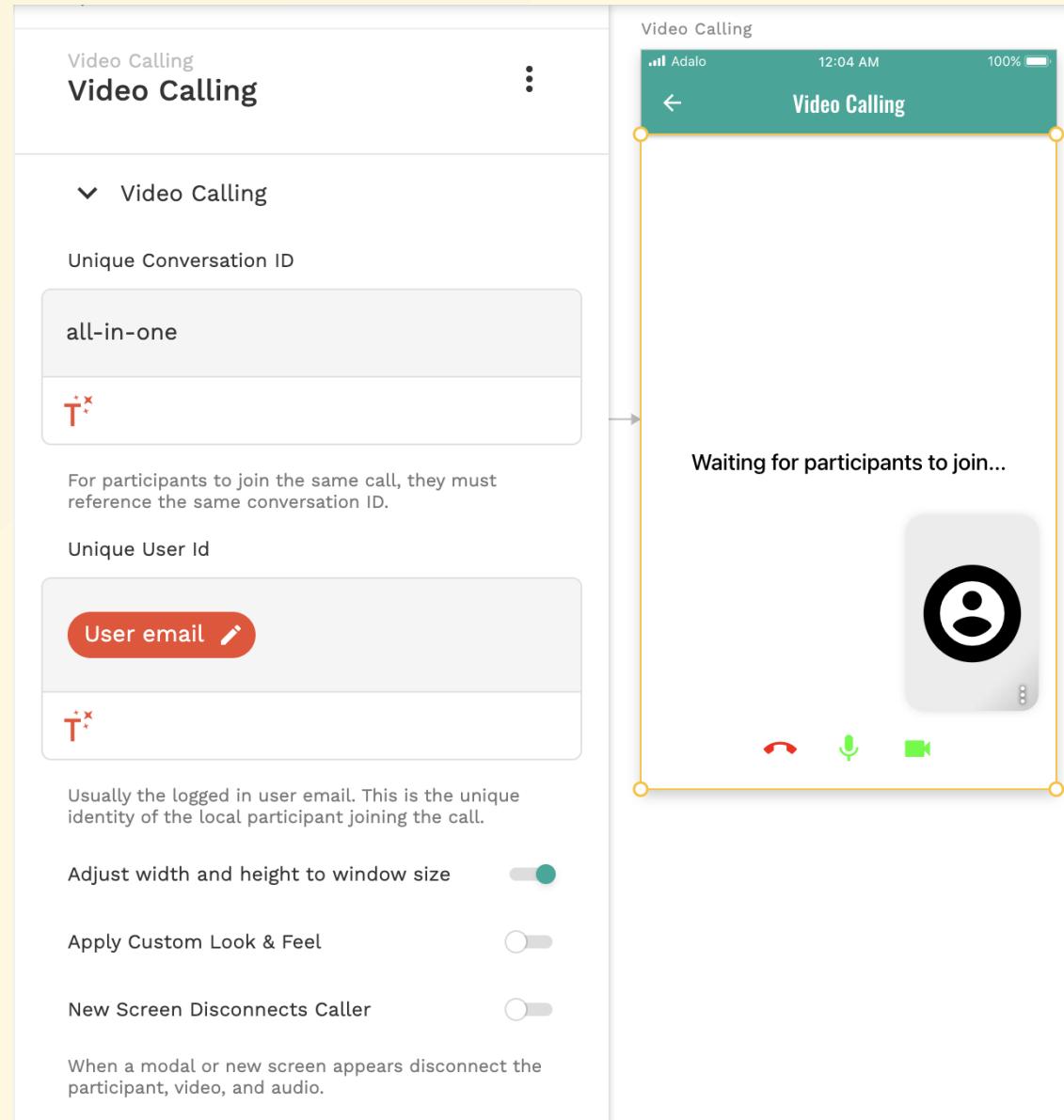
- Add a Link button to the Home screen to the screen for the video call.
- Add a Link to New Screen from ADD ACTION



- Create a "Video Calling" screen by selecting "App Bar" in Template.



- Place the Video Calling component on the screen
- Enter "all-in-one" in Unique Conversation ID
 - This will allow all users to participate in a single video call
- Set Unique User id to "Logged In User's Email"



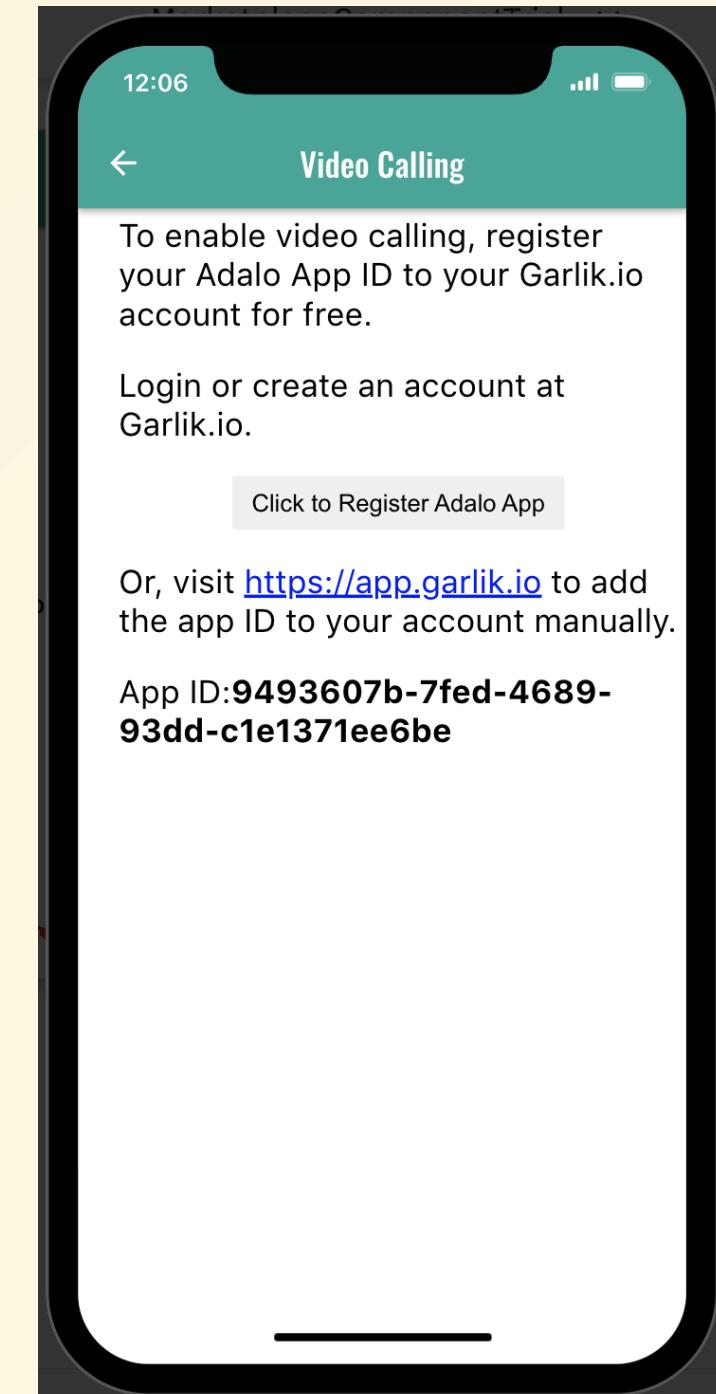
Additional information

This time, I set up all users to participate in one video call, but I think it is also possible to create a group with multiple users and call only the group members.

If you are interested, please give it a try.

If you check the Preview screen, you will see the message "register your Adalo App ID to your [Garlick.io](#) account for free". Follow the directions.

- Click on "Click to Register Adalo App



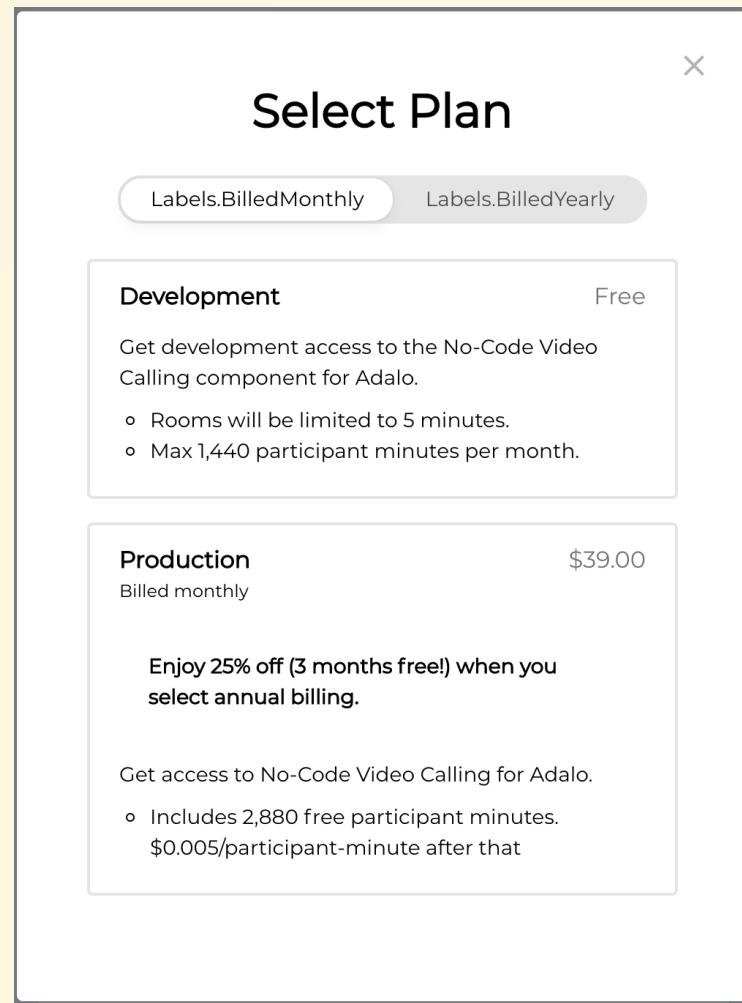
You will see a login screen, but you have not created an account yet.

- Click on "Or, Sign Up."

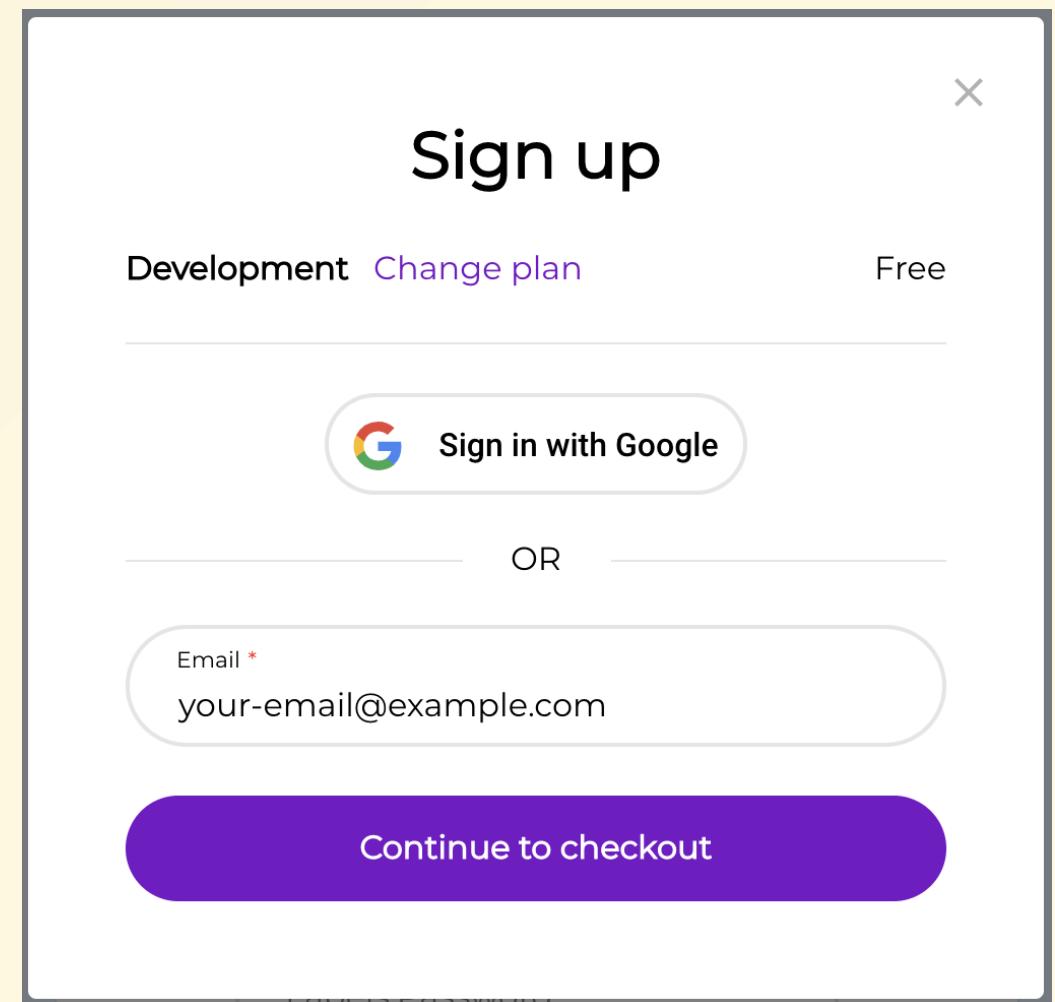
The image shows a login interface with the following elements:

- A purple stylized 'G' logo at the top left.
- The word "Login" centered above the sign-in options.
- A "Sign in with Google" button featuring the Google logo.
- A horizontal line with the word "OR" in the center.
- An input field labeled "Properties.Person_Email*".
- An input field labeled "Labels.Password*".
- A large purple "Login" button.
- A link labeled "Labels.ForgotPassword" below the login button.
- A white button with a purple border labeled "Or, Sign Up" at the bottom.

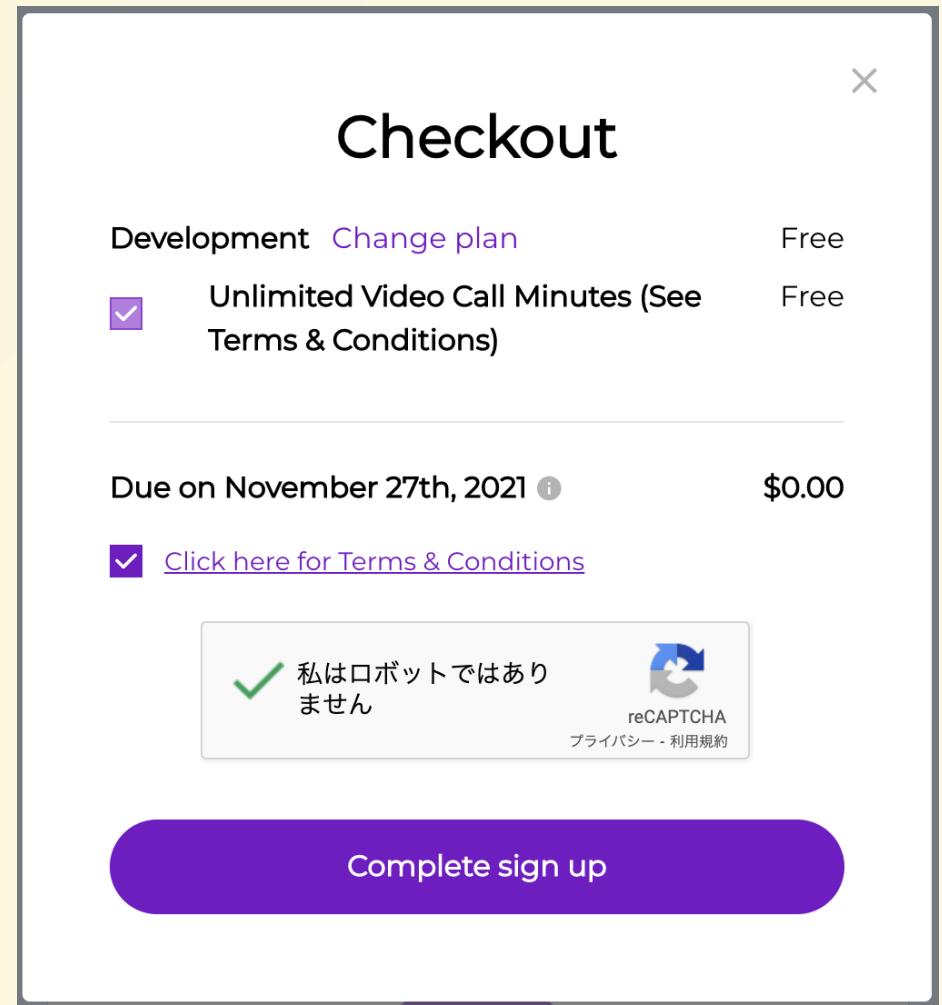
- Select Development Plan
(it's free)



- Enter your Email and click "Continue to checkout
 - If you have a Google account, you can also click Sign in with Google.



- Tick the three checkboxes and click "Complete sign up"



- Open the email you received and click "Confirm your account" !

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>

- Set your Password

The screenshot shows a mobile-style password setup screen. At the top right is a close button (X). The title 'Set Password' is centered at the top. Below it is a message 'Please set your password.' A large input field is labeled 'Labels.Password*' and has a note below it stating '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*

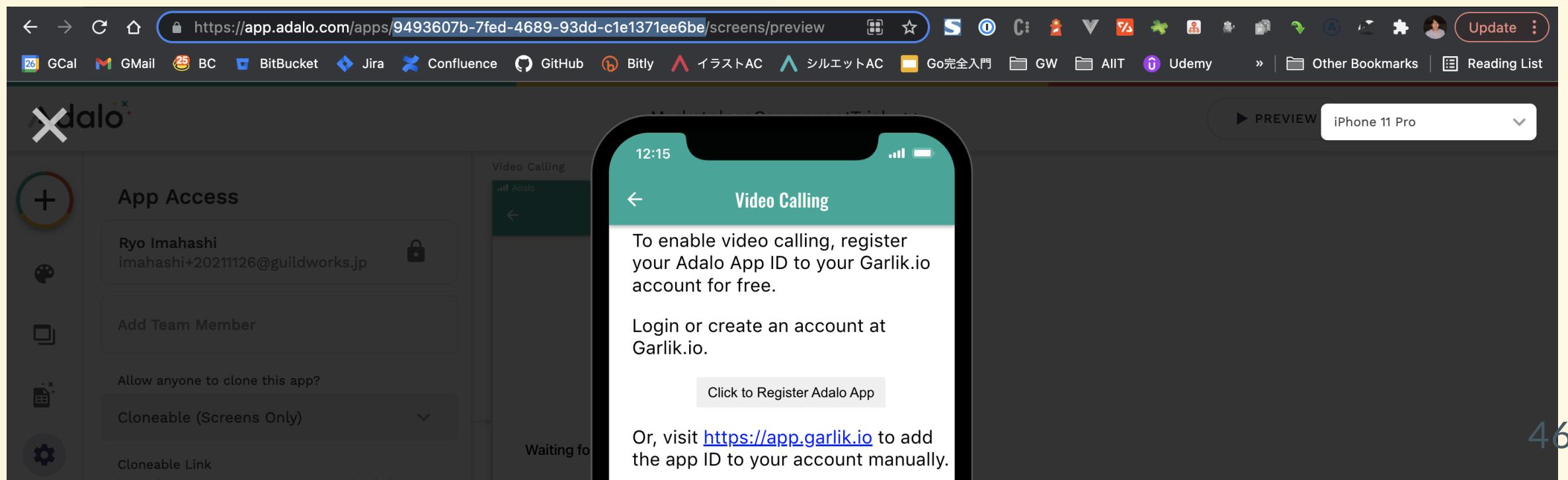
Must be at least 8 characters

Cancel Set Password

Account registration is now complete.
The next step is to register your Adalo App ID.

The screenshot shows the Adalo No-Code Components dashboard. 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. The main area has a light blue background. It features two main sections: 'Participant Minutes Consumed (Development Plan)' and 'Register Adalo App'. The first section shows '0.00/1440 Minutes' consumed, with '0 billable minutes consumed' and '0 free participant minutes consumed'. The second section contains a text input field labeled 'Adalo App ID' containing the value 'ed7577d9-3a1b-4a07-bd04-de4c5fce29c3', a note 'Register your Adalo app IDs to enable video calling.', and a purple 'Register App' button. Below these sections is a third one labeled 'Registered Apps' which is currently empty.

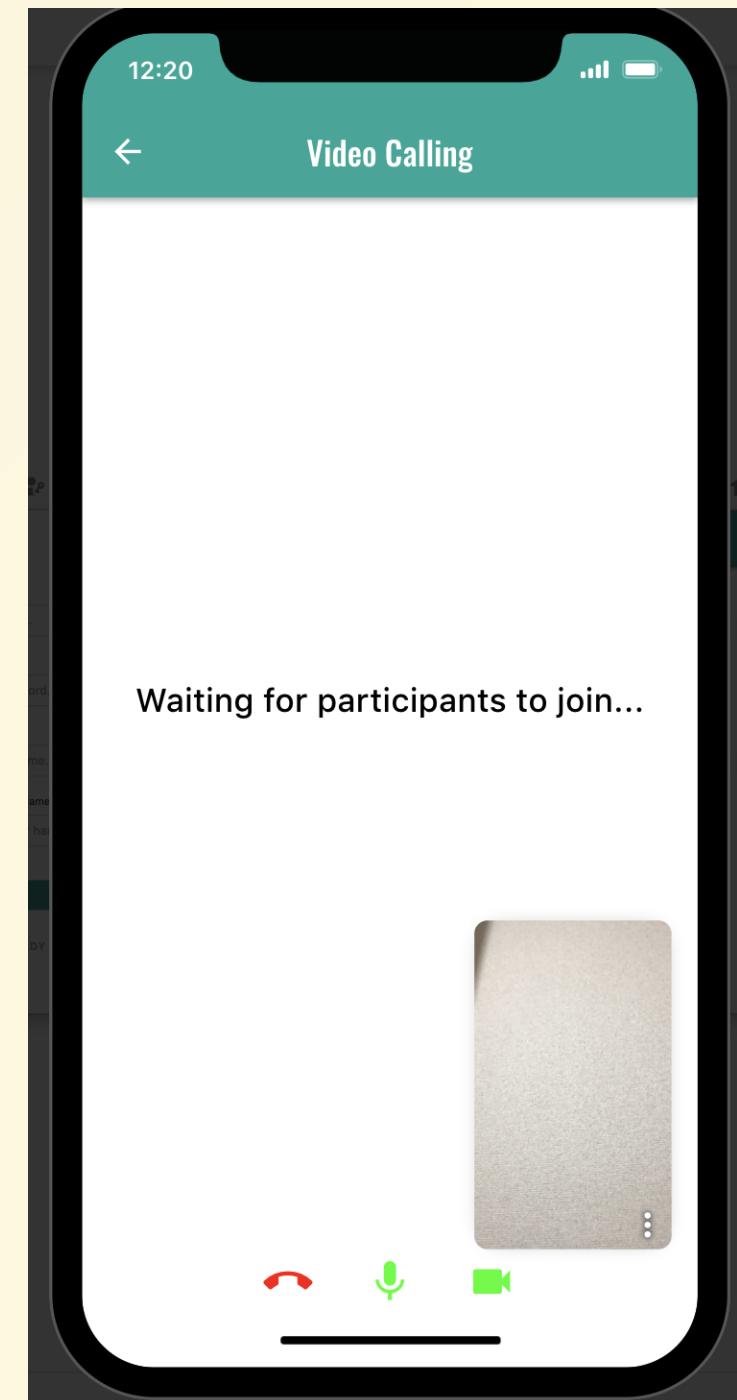
- Open the Adalo admin panel and copy the App ID contained in the URL
 - "xxx..." in the following URL part of the following URL.
<https://app.adalo.com/apps/xxxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/screens/preview>



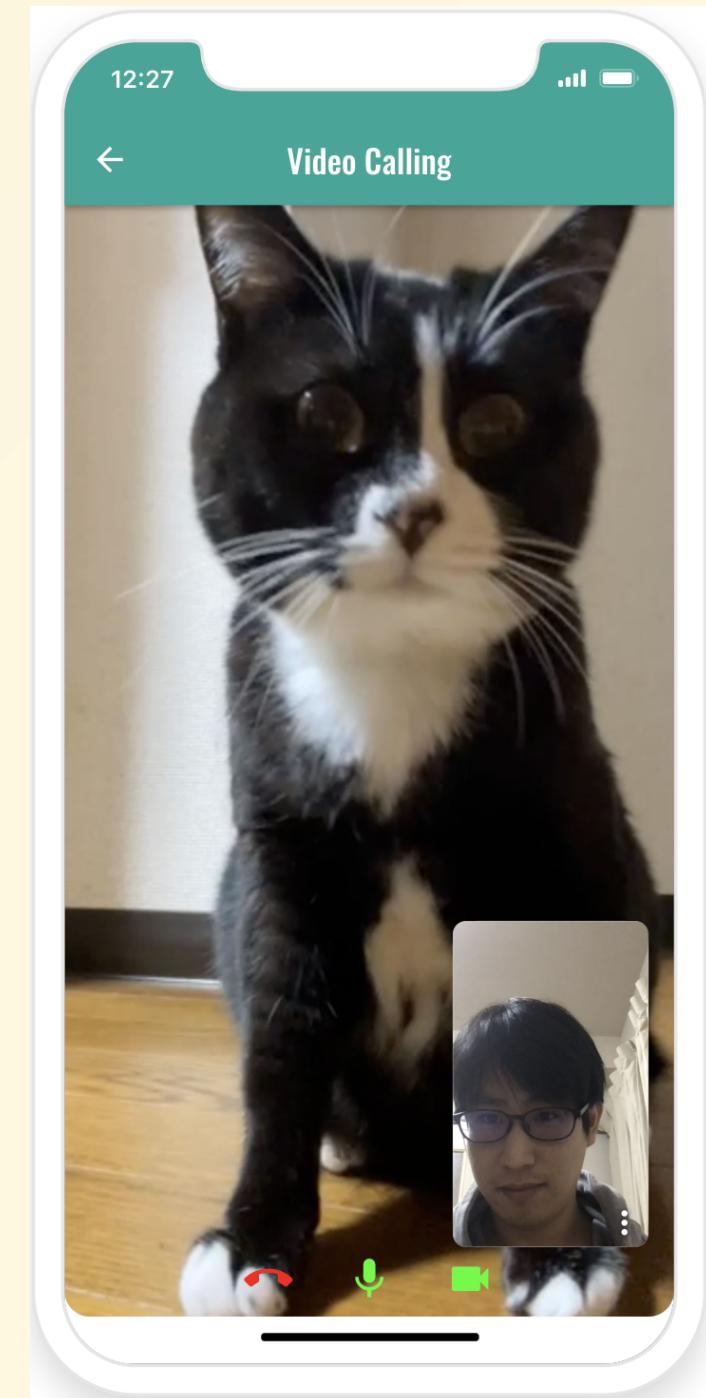
- In the [Garlick.io](#) admin panel, enter your Adalo App ID and click "Register App" !

The screenshot shows the Garlick.io admin panel interface. On the left, there's a section titled "Participant Minutes Consumed (Development Plan)" showing "0.00/1440 Minutes" with a progress bar at 0%. Below the bar are two buttons: "0 billable minutes consumed" and "0 free participant minutes consumed". On the right, there's a "Register Adalo App" section. It contains a green success message: "App 9493607b-7fed-4689-93dd-c1e1371ee6be has been registered". Below it is a field labeled "Adalo App ID" containing the value "9493607b-7fed-4689-93dd-c1e1371ee6be". A note below the field says "Register your Adalo app IDs to enable video calling." At the bottom right of this section is a purple "Register App" button. At the very bottom left, there's a "Registered Apps" section showing a single entry: "9493607b-7fed-4689-93dd-c1e1371ee6be" next to a small icon and a trash can icon.

Reload Adalo's Preview screen and display the video call screen again, and you will be waiting for participants to join the video call.



When other users log in and display the video call screen, they can make a call.



~~Question: Would you like to try a video call?~~

~~If I get a reaction that you want to try it, I'll share the URL of the experimental app.~~

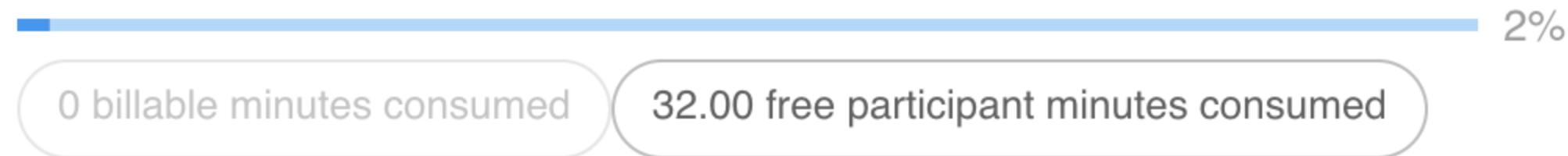
We are running out of time because of a lot of slides, so skip this part...

With the free plan, it seems that calls are disabled after the total call time reaches 1440 minutes, but I think it's enough for testing during development.

If you need the video calling feature, please take advantage of it.

Participant Minutes Consumed (Development Plan)

32.00/1440 Minutes



There are several other components provided to integrate with external services. If you are interested, you can try them.

Example:

- Youtube (free)
- Google Map(free trial available)
- Google Signin(\$25)

Custom Action

Next to the external integration component, we will show you how to handle the data retrieved from the API on the Adalo screen.

Reference

“ In a broad sense, an Application Programming Interface (API) is a specification of an interface used by software components to exchange information with each other. ”

<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>

First, let's try API integration.
We will use The Cat API, which is free to try. Please access the following URL.

<https://thecatapi.com/>

The screenshot shows the homepage of The Cat API. At the top, there is a navigation bar with links for PRICING, ABOUT, and DOCUMENTATION. Below the navigation bar, the main heading reads "The Cat API - Cats as a Service." A sub-headline says "Because everyday is a Caturday." Below this, a text block states: "A public service API all about Cats, free to use when making your fancy new App, Website or Service. [Get your license now.](#)" To the right of the text, there is a video player displaying a cat walking on a wooden floor. Above the video player are two buttons: "LOVE IT" (green) and "NOPE IT" (red). At the bottom left, there is a footer banner with the text "The Cat API has delivered Billions of requests, to over 30K developers". On the left side of the page, there are icons for React, jQuery, Vue.js, Node.js, and Angular.

Note: For the dog people, there is also The Dog API. It probably does the same thing as The Cat API. (I have not tried it, so I recommend using The Cat API with it first)

<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 brief description 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, there are icons for React, jQuery, Vue.js, Node.js, and Angular. In the center, there's a large image of a white dog sitting in a grassy field. Below the image are two buttons: "LOVE IT" (green) and "NOPE IT" (red). 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".

When using an API, it is often necessary to have an API key issued by the API provider. The Cat API also requires an API key, so have it issued to you.

- Scroll down to the bottom and click "SIGNUP FOR FREE" in the Pricing section.

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

- Enter your email, App Description, type of project and click "SIGNUP" !

Signup for an API key

It's completely free, email is only 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](#)

- Check the API key that was sent to you by email (you will use it later).

The image shows an email client interface with a single message. The message is from 'The Cat API' (represented by a yellow cat icon) to 'aden@thatapiguy.com'. The subject of the email is 'Welcome to The Cat API!'. The body of the email contains the following text:

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>

Next, let's check the API documentation to see how to use the API.

- Go to the following URL (you can also find the link "API Documentation" in the email you received earlier)

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

Use the API to get a random kitten image as described in the top page.

As in the example, make sure that the image changes when the button is pressed.

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 😺

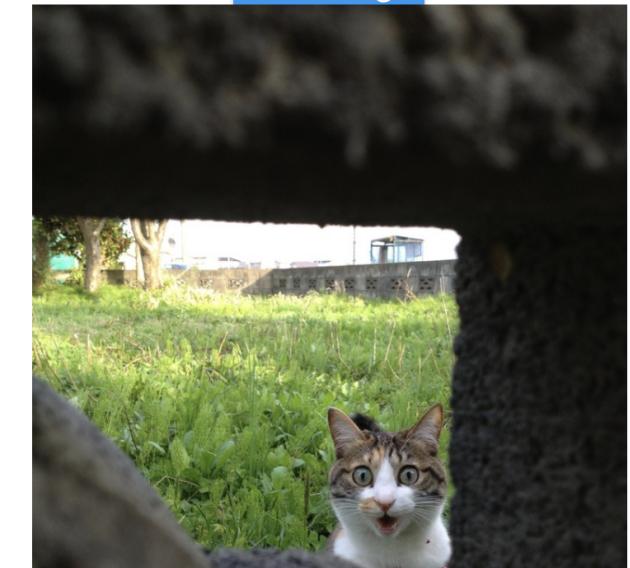
Bonus: Add a button to repeat the request for a new kitty!

Example

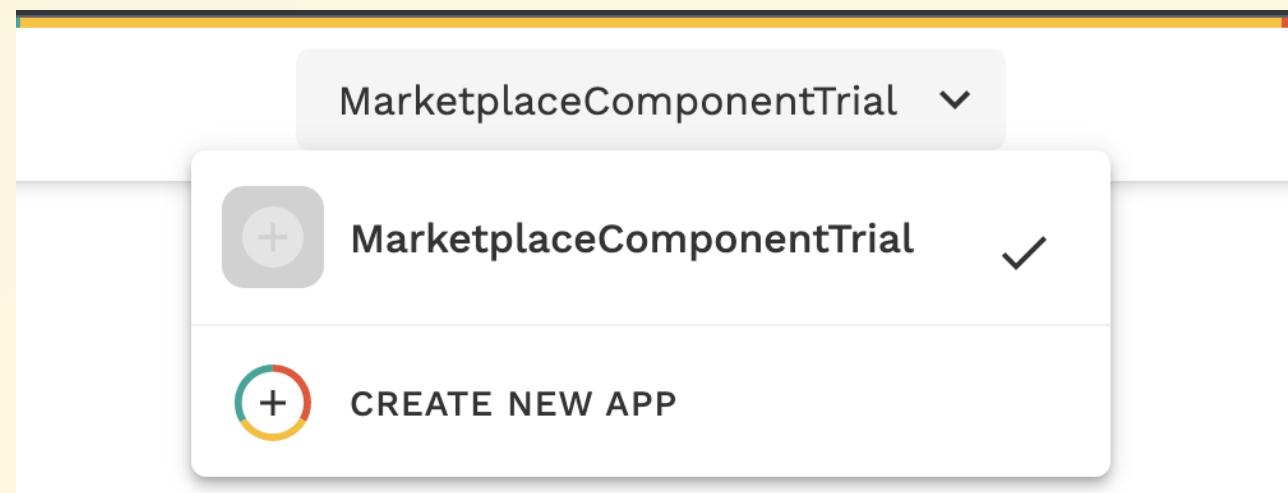
Uses <https://api.thecatapi.com/v1/images/search> to show one random image every time "Another" is clicked.

Example Source Code: VueJS

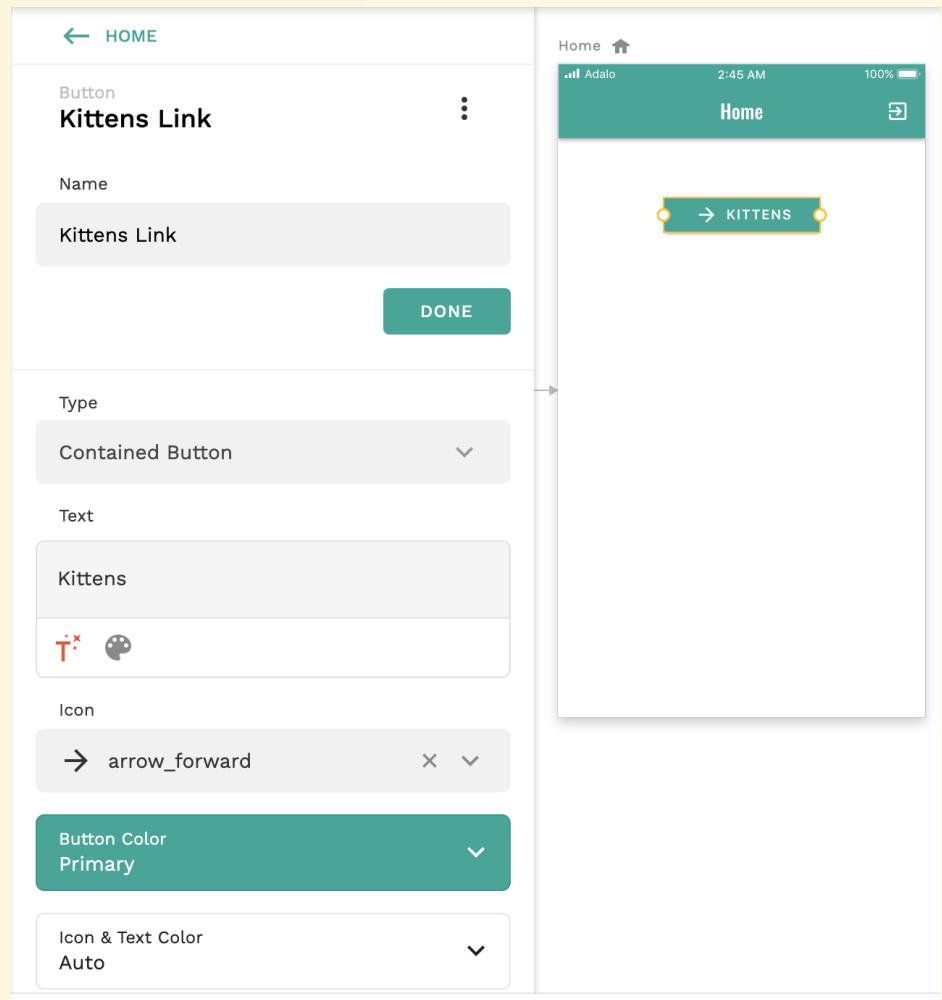
ANOTHER 



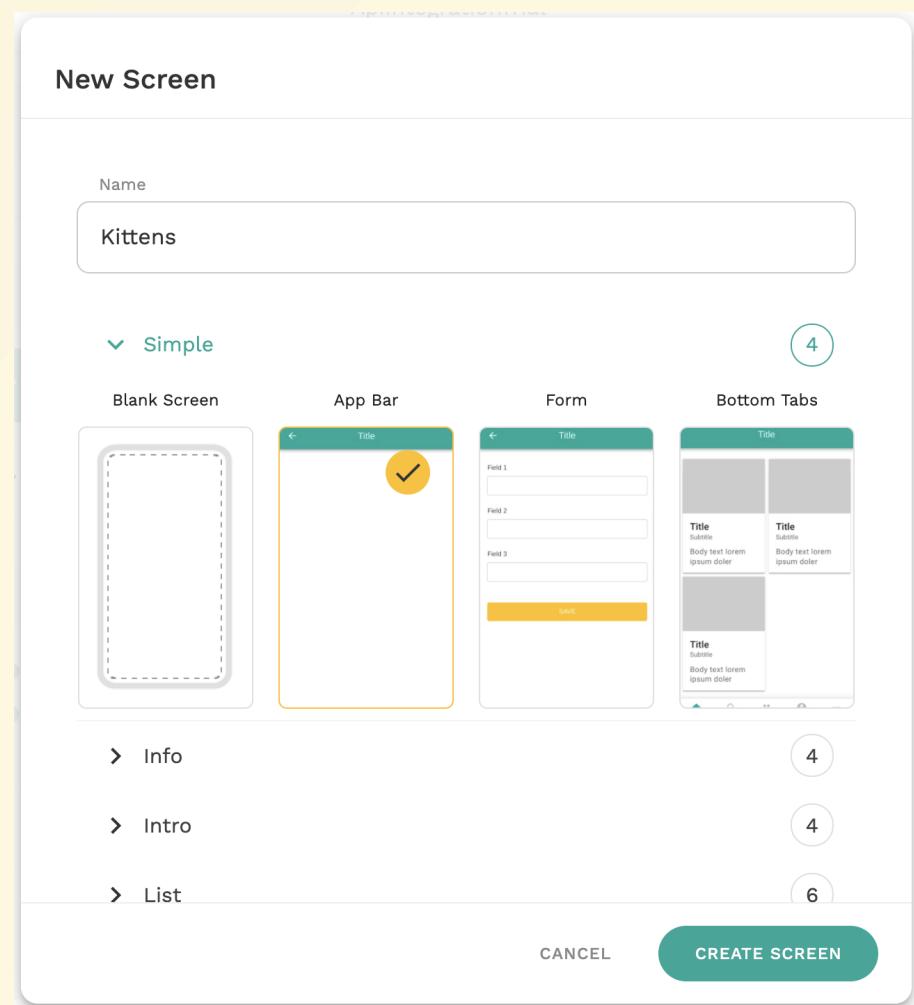
- CREATE NEW APP in Adalo's admin panel
- The settings are as follows
 - Platform: Native Mobile App
 - Template: Blank
 - App Name: ApilntegrationTrial



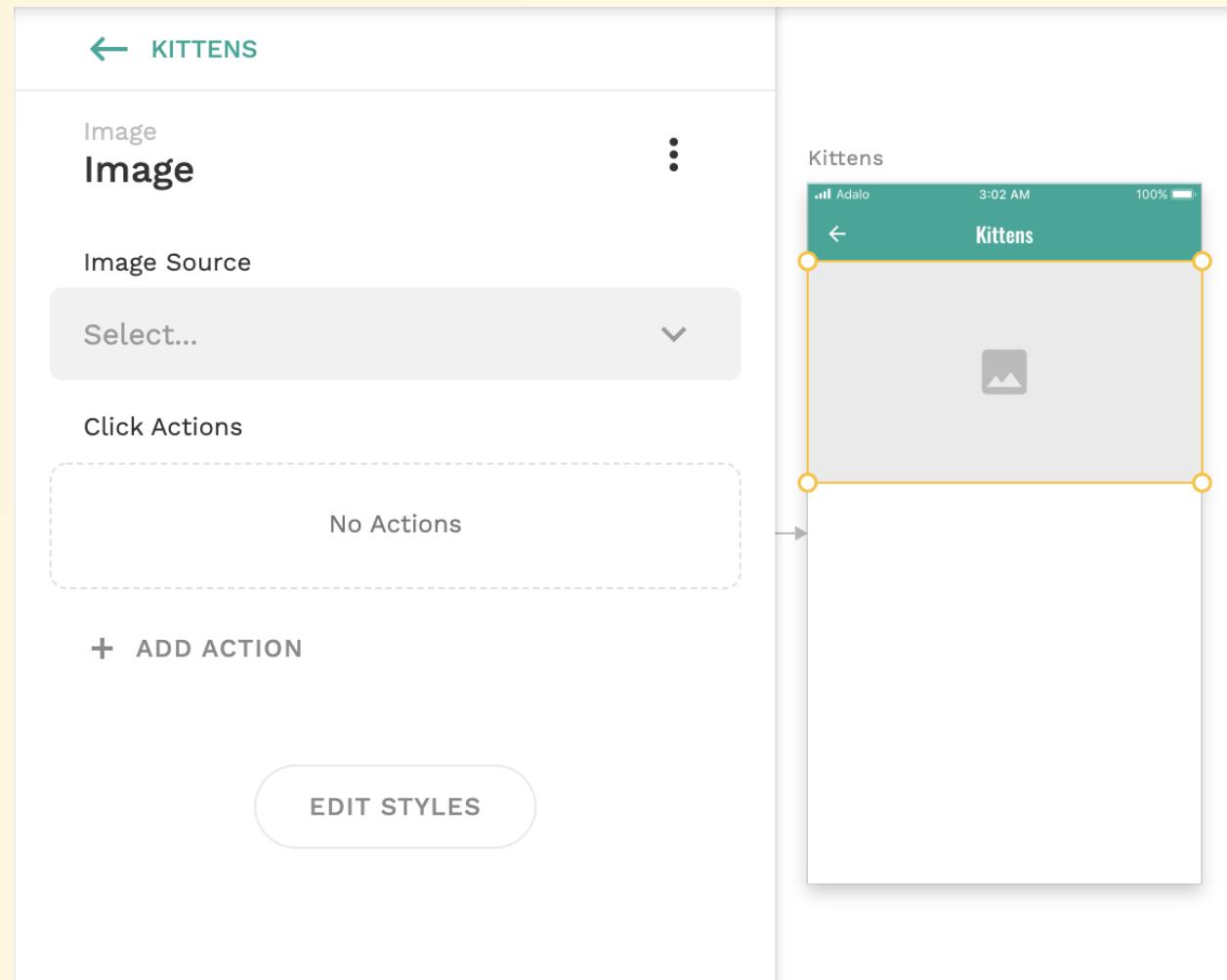
- Add a link button to the kitten image display screen on the Home screen.
- Set Link to New Screen from ADD ACTION



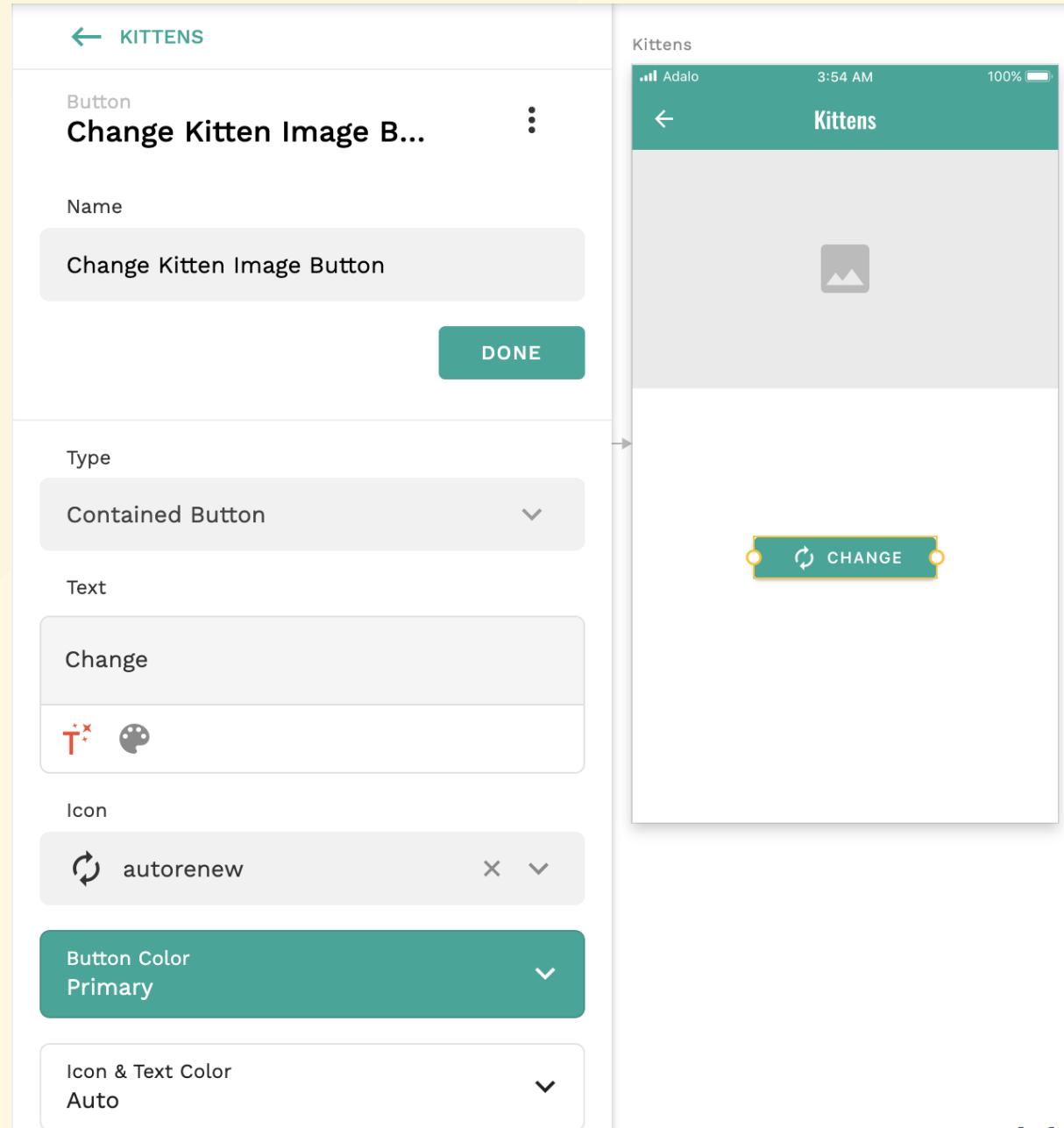
- Select App Bar as Template and create Kittens screen.



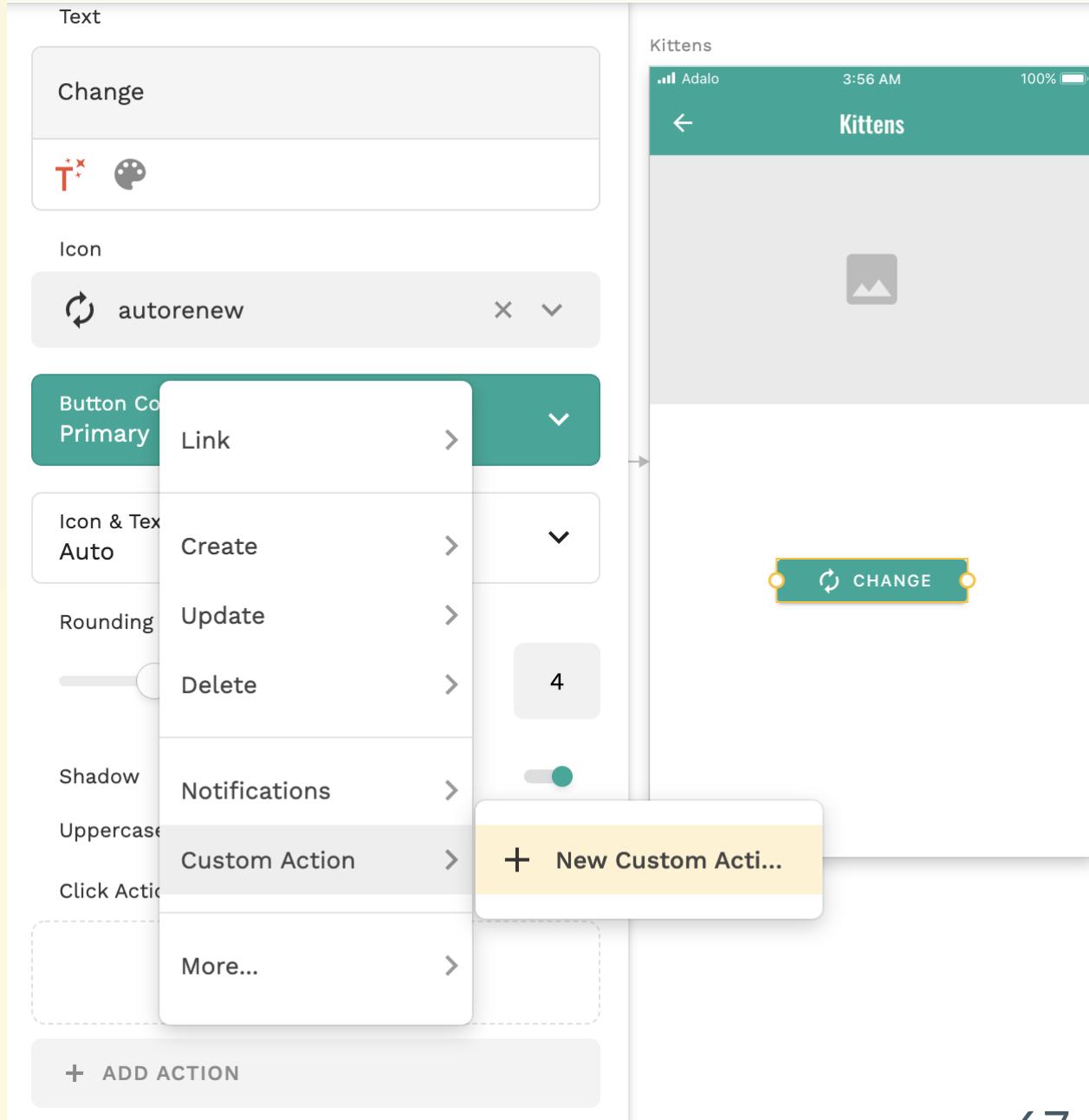
- Place the Image component on the screen
- Leave the component settings as they are (we'll set them later)



- Add a Change Kitten Image Button



- Select New Custom Action from ADD ACTION



You will be prompted to start
a 14-day Integration Pack
Trial (free).

- Click on "START
INTEGRATION PACK
TRIAL"

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](#)

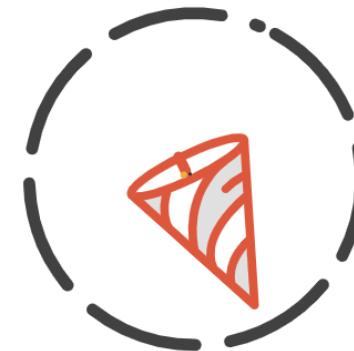
Collections API

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

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

The trial has started.

- Click on "CREATE NEW CUSTOM ACTION" !



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

- Enter the following and click NEXT
 - Name: GetRandomKitten
 - Type: Create

The screenshot shows a user interface for creating a new item. It consists of three numbered steps: 1. Name & Type, 2. API Request, and 3. Outputs. Step 1 is highlighted with a yellow circle and labeled 'Name & Type'. Step 2 is labeled 'API Request' and Step 3 is labeled 'Outputs'. Below the steps, there are two input fields. The first field is labeled 'Name' and contains the value 'GetRandomKitten'. The second field is labeled 'Type' and contains the value 'Create', with a dropdown arrow indicating more options are available. At the bottom right of the screen are two buttons: 'CANCEL' and a large yellow 'NEXT' button.

1 Name & Type ————— 2 API Request ————— 3 Outputs

Name
GetRandomKitten

Type
Create

CANCEL NEXT

Next, we will configure the API Request to be sent.

The screenshot shows a configuration interface for an API request. At the top, there are three tabs: **Name & Type** (done), **API Request** (selected), and **Outputs**. The **API Request** tab contains the following fields:

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

Each field has a red **+ ADD ITEM** button. To the right, under the **Outputs** tab, it says **Inputs** and **No Inputs**, with a red **+ ADD ITEM** button. At the bottom, there are **BACK** and **RUN TEST REQUEST** buttons.

Go to the following URL to check the configuration items.

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

- API Base URL is
<https://api.thecatapi.com/v1/images/search>
- Method is GET
- Set the API key in the Header with the name x-api-key

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

- Set the API Request based on the results of the check.
 - API Base URL is <https://api.thecatapi.com/v1/images/search>
 - Method is GET
 - Set the API key in the Header with the name x-api-key
- Click "RUN TEST REQUEST".

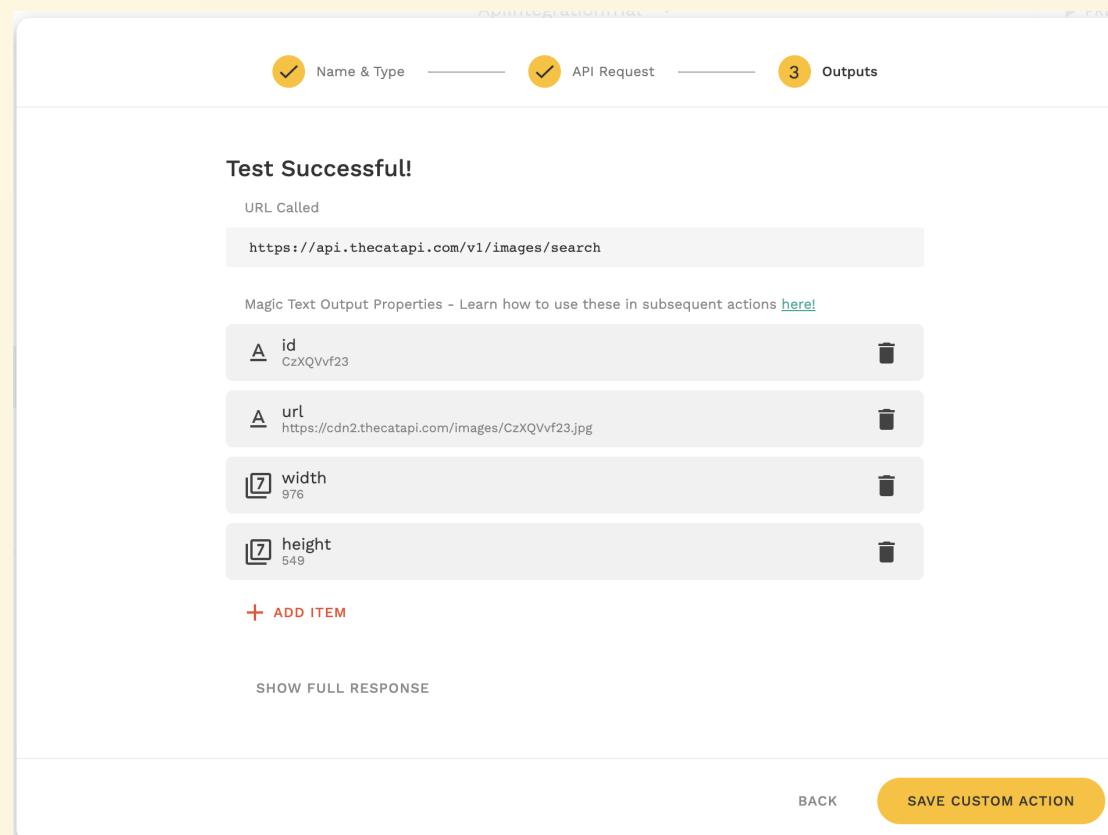
The screenshot shows a user interface for configuring an API request. At the top, there are three tabs: 'Name & Type' (with a checkmark), '2 API Request' (highlighted in yellow), and '3 Outputs'. The 'API Request' tab has three sections:

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

On the right side, there's a 'Inputs' section with a dashed box labeled 'No Inputs' and a '+ ADD ITEM' button. At the bottom right are 'BACK' and 'RUN TEST REQUEST' buttons.

If the Test is successful, the data retrieved from the API (Magic Text Output Properties) will be displayed. These can be used in subsequent actions.

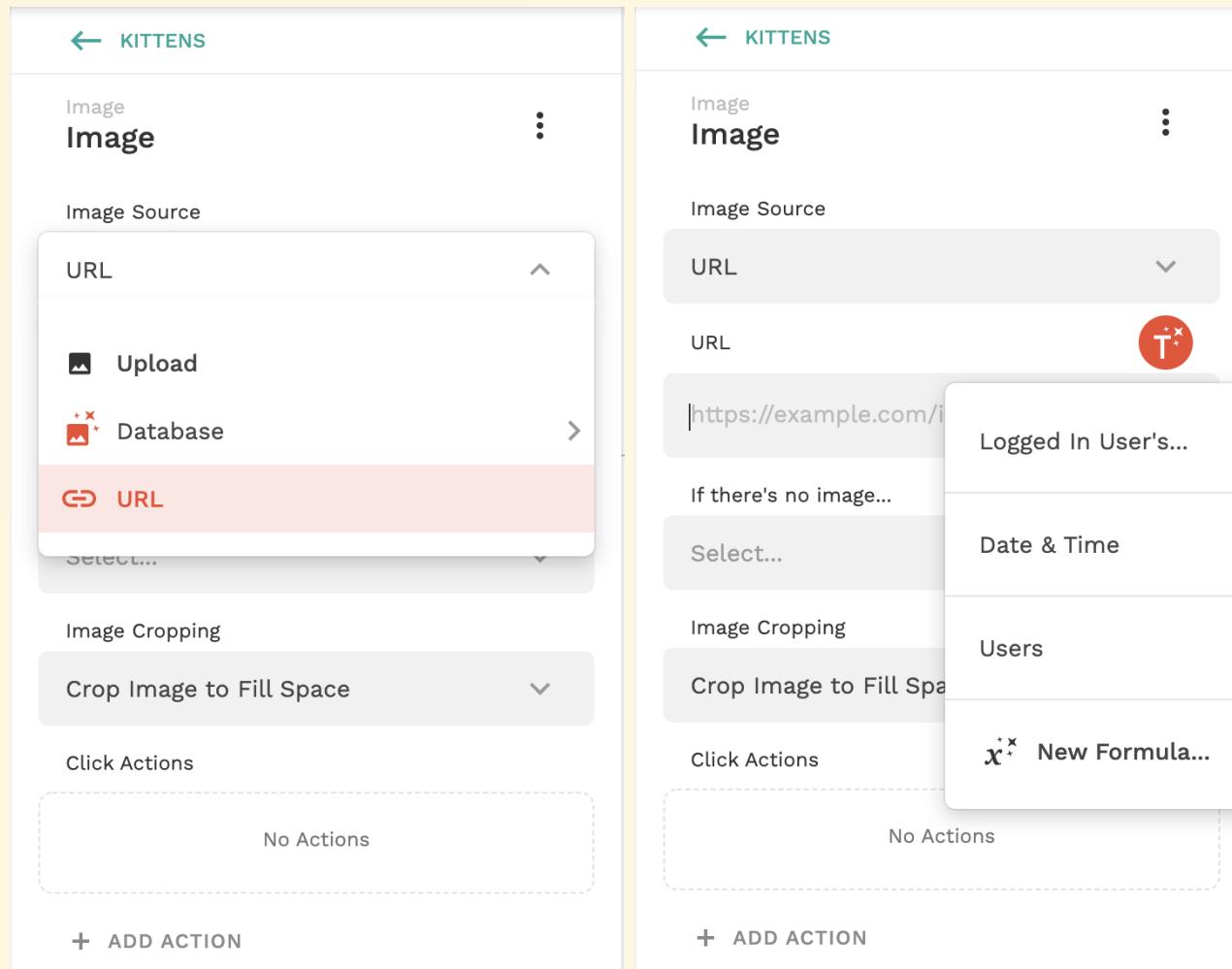
- Click on "SAVE CUSTOM ACTION"



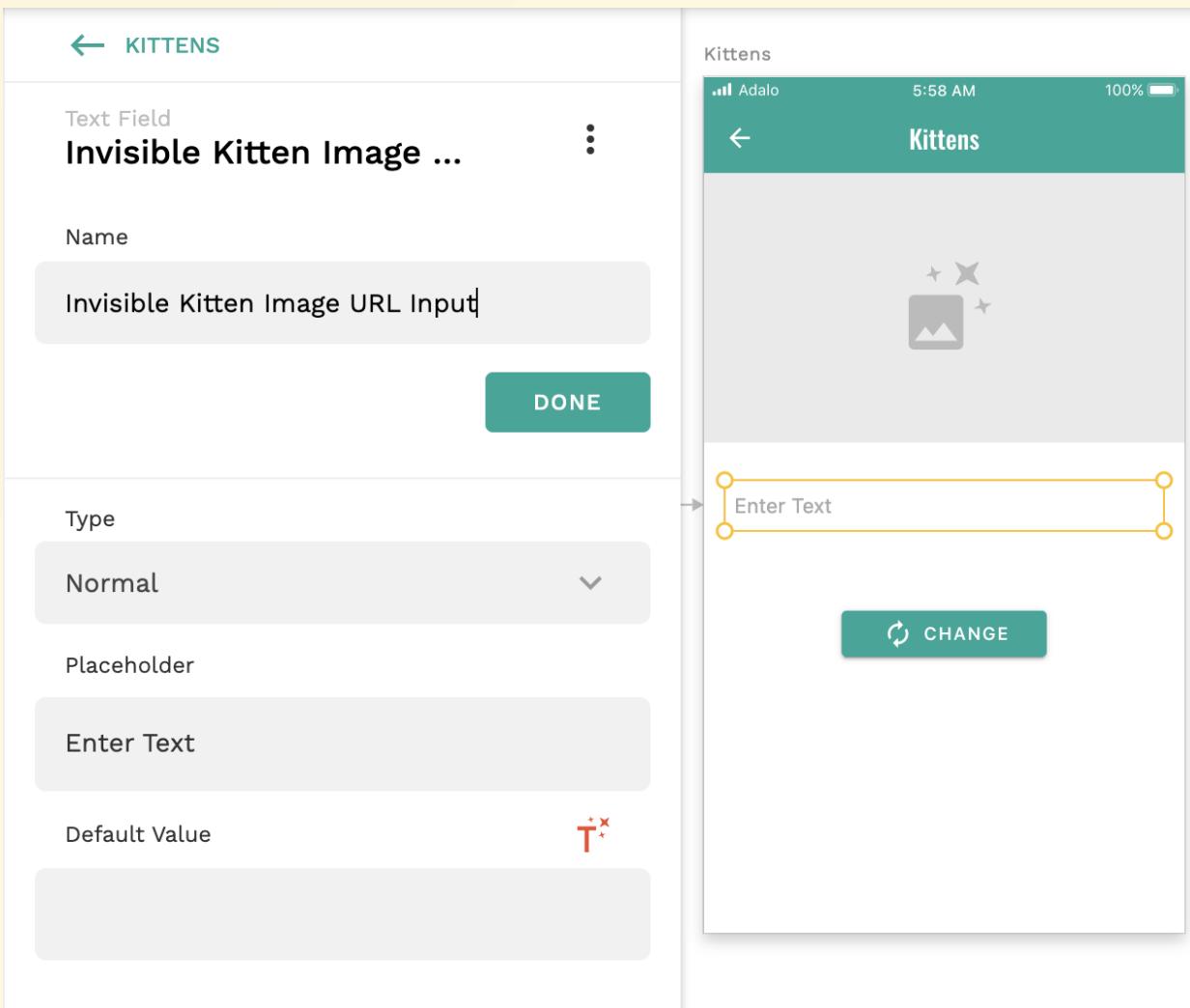
Next, set the URL of the kitten image retrieved from the API to the Image Source of the Image component.

As it is, the data retrieved from the API will not appear in the choices.

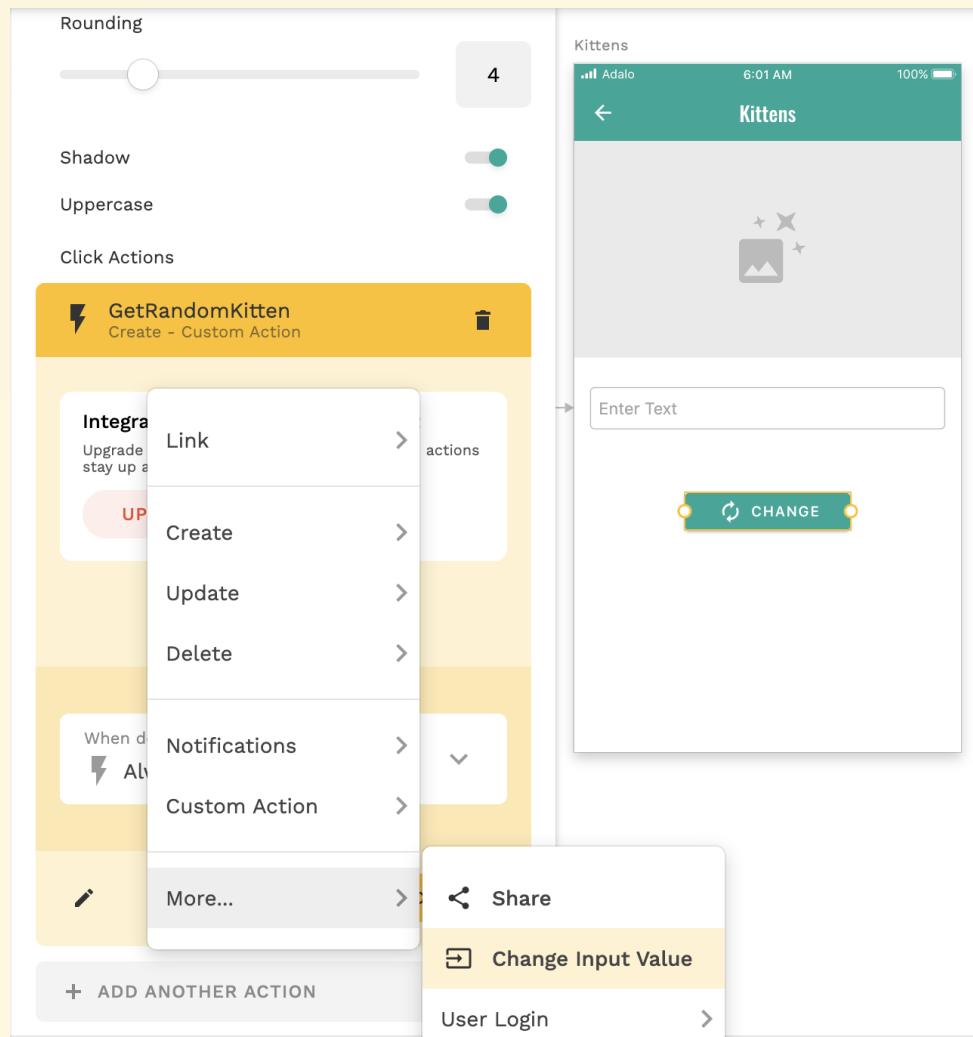
A url
<https://cdn2.thecatapi.com/images/6k7.jpg>



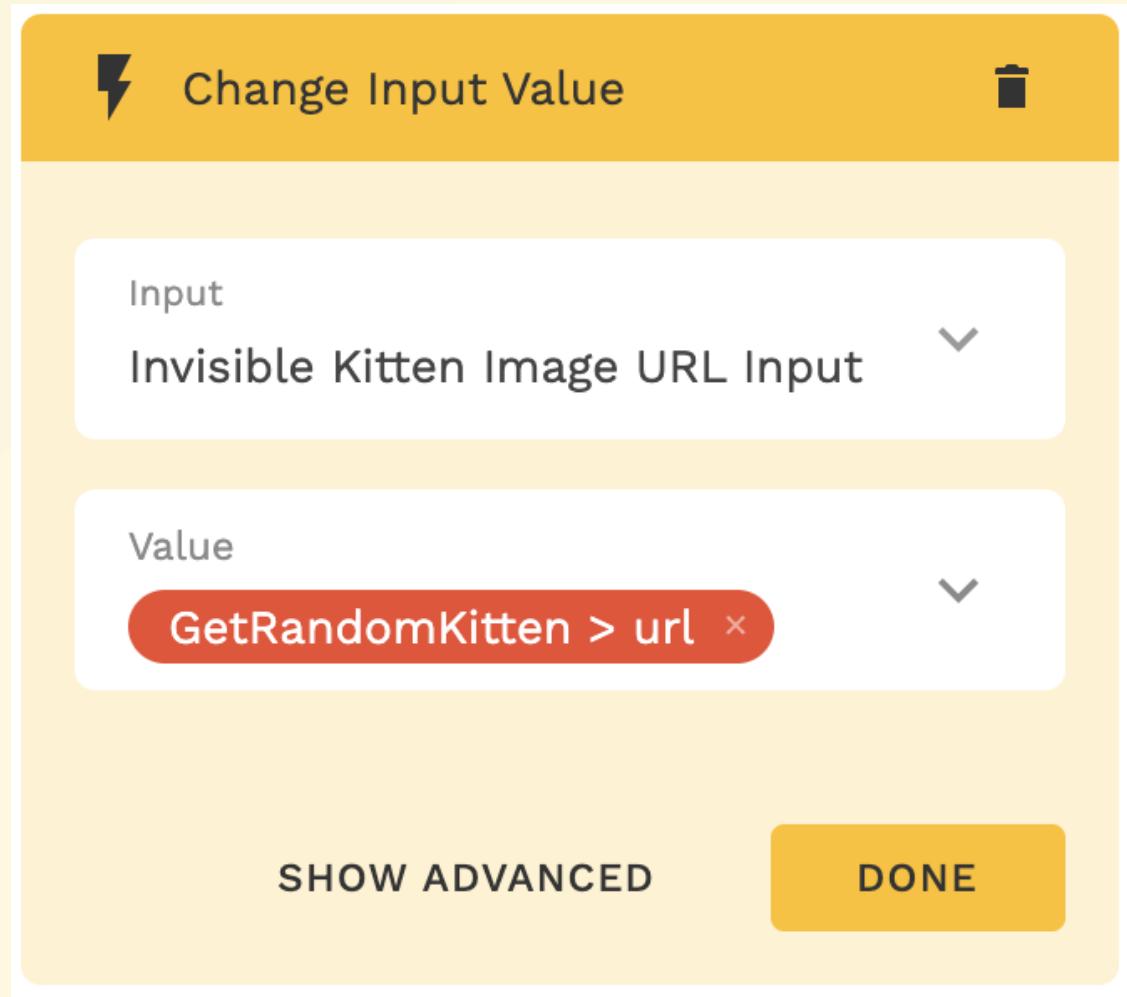
- Add a Text Input component to the screen
- Change the Name to "Invisible Kitten Image URL Input".



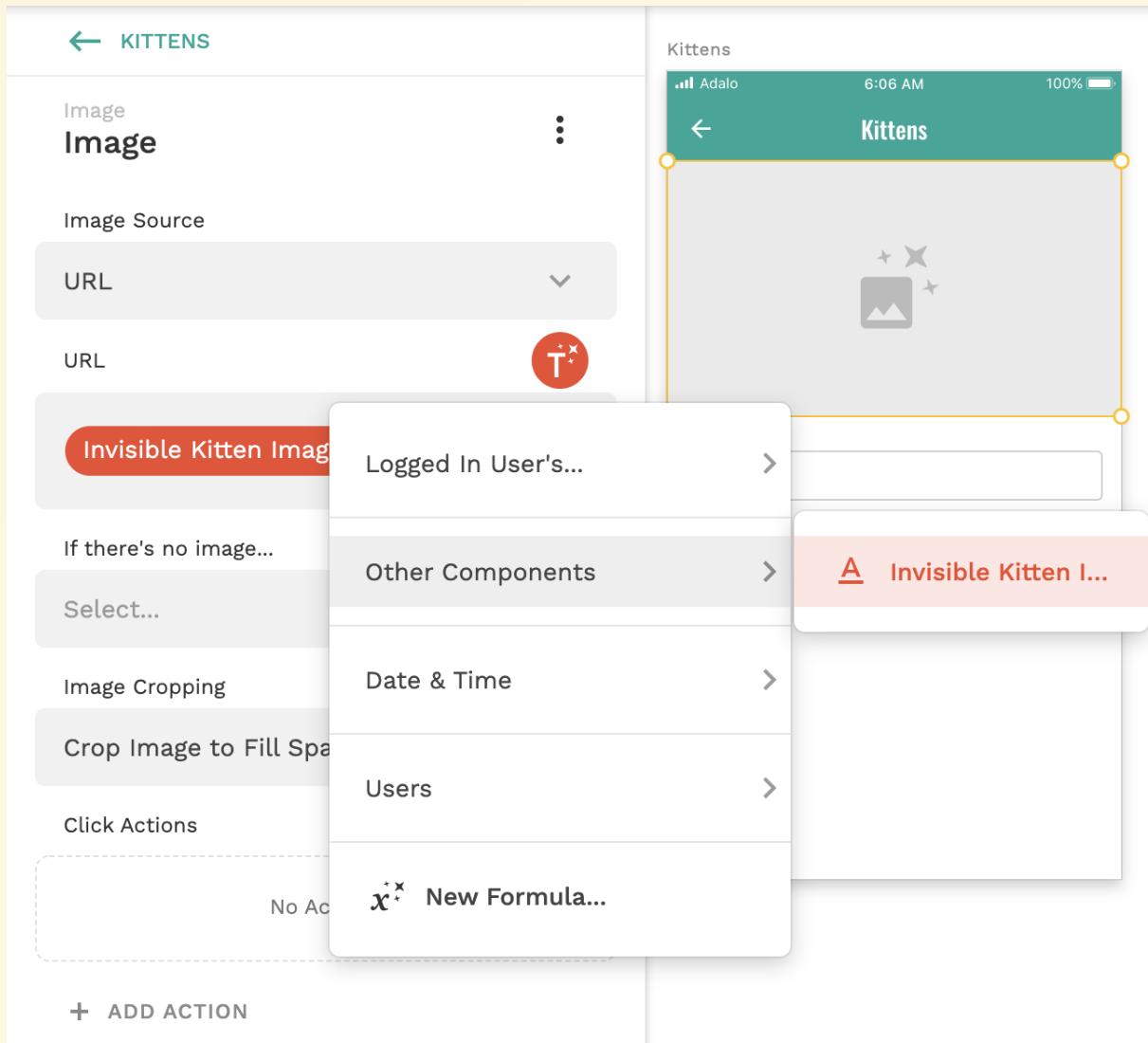
- Click on the "Change Kitten Image Button"
- Select "Change Input Value" from ADD ANOTHER ACTION



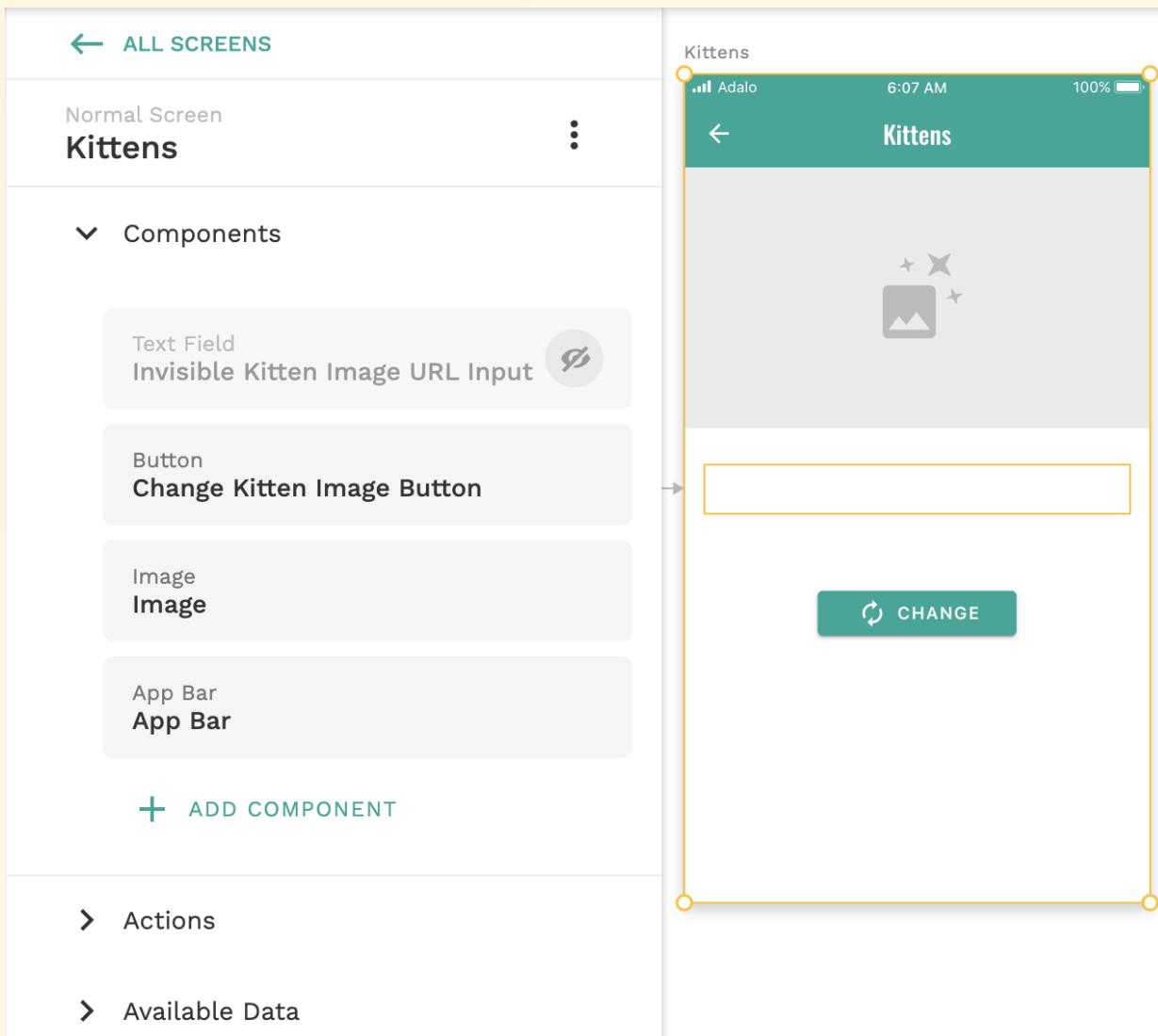
- Set "Invisible Kitten Image URL Input" as Input
- Set Value to "GetRandomKitten > url".
- Click "DONE"



- Click on the Image component
- Set the URL to "Invisible Kitten Image URL Input".

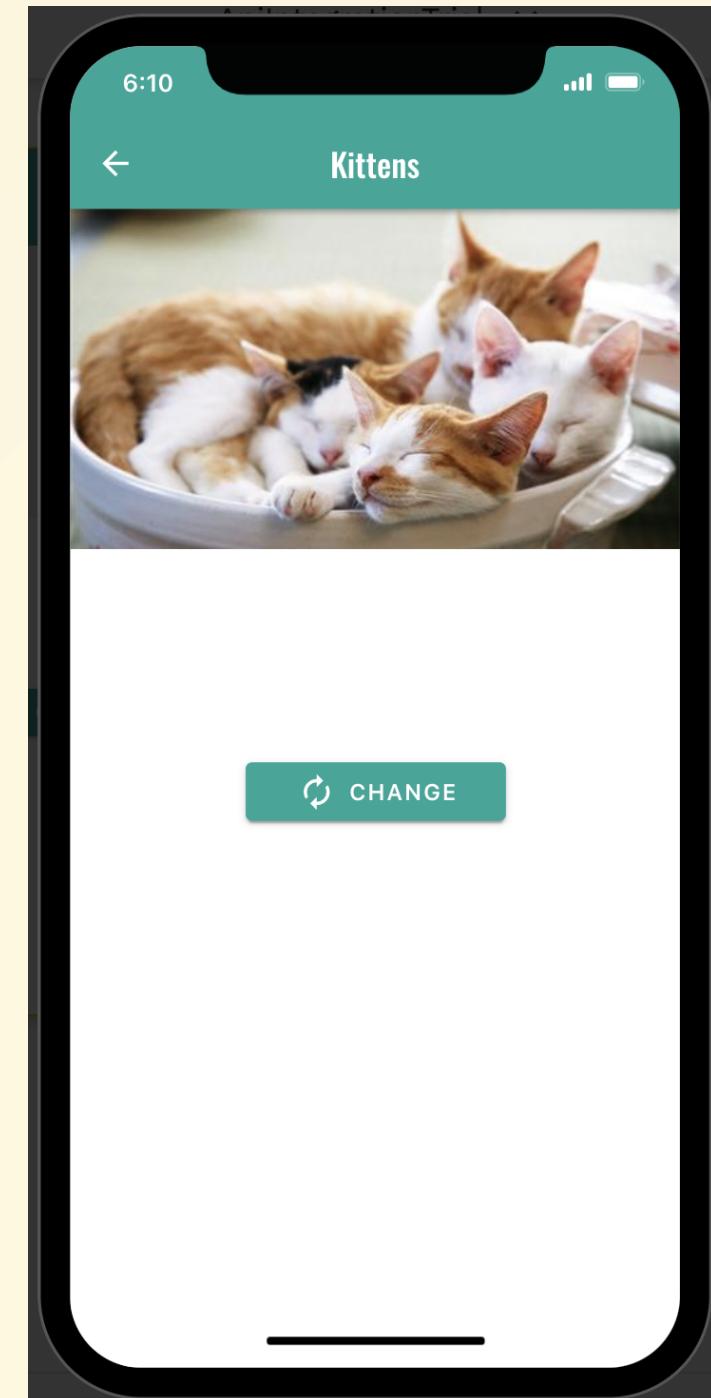


- Click on the screen name "Kittens".
- Click on the eye icon to the right of "Invisible Kitten Image URL Input" to hide it.



Check it with the Preview function.

Click on the CHANGE button and the kitten image is now displayed.



Supplemental

- Data acquired from the API in Custom Action can be used in subsequent Actions. If you want to use the data in your component, use one of the following methods.
 - Set the data to the Text Input value on the same screen using the Change Input Value of the subsequent Action and load it.
 - Alternatively, you can save the data to the database in a subsequent Action and load it from another component. In this case, it can be accessed even after the screen transition.
 - Example: <https://help.adalo.com/integrations/custom-actions>

Notes

Currently, Custom Action has some limitations.

- Custom Actions will not work with the Submit button of the Form component.
- If a Custom Action is used as an Action for the entire screen, data obtained as an API response cannot be used in the Imperial Action.
- Cloning an app does not copy the Custom Action; if you clone an app that contains a Custom Action, you will need to recreate it manually.

>Cloning an app does not copy the Custom Action; if you clone an app that contains a Custom Action, you will need to recreate it manually.

The 14-day Integration Pack Trial is expected to end in the middle of the Development Phase.

If you are likely to use a lot of Custom Actions, we recommend that you create a new Adalo account and start a new Integration Pack Trial before you start working on the Development Phase.

External Collection

This section describes how to handle data retrieved from the API as an Adalo collection.

If you want to retrieve multiple data at once and list them on the screen, use External Collection instead of Custom Action.

Let's use this API to retrieve and display a list of cat breeds.

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

🔗 Send a Test Request

The screenshot shows a user interface for sending a test request to the Cat API. At the top, there is an orange button labeled "Send". To its right, the method "GET" is selected. Below this, the URL "https://api.thecatapi.com/v1/breeds" is displayed. A navigation bar below the URL includes tabs for "Settings", "Headers [1]", "Query [3]", and "Code Generation". The "Headers" tab is currently active. Under the "Headers" tab, there is a button "Add Header" and a key-value pair "x-api-key : {\$\$.env.x-api-key}".

- Click "ADD COLLECTION" in External Collections of Database

The screenshot shows a user interface for managing database collections. On the left, there's a vertical sidebar with icons for adding a collection (a plus sign in a circle), a palette, a clipboard, a gear (settings), a checkmark, and a refresh symbol. The main area is divided into sections: "Database Collections" (containing a "Users" collection with 1 Record) and "External Collections". In the "External Collections" section, there's a red button labeled "+ ADD COLLECTION". Below this, there's a callout box titled "Connect to APIs" with the subtext "Make your app even more powerful by connecting to 3rd party APIs". At the bottom, there's a pink box for "Integration Pack Trial | 14 Days Left" with the text "Upgrade now to make sure all of your external collections stay up and running." and a red "UPGRADE NOW" button.

+ ADD COLLECTION

Database Collections

> **Users** 1 Record

+ ADD COLLECTION

External Collections

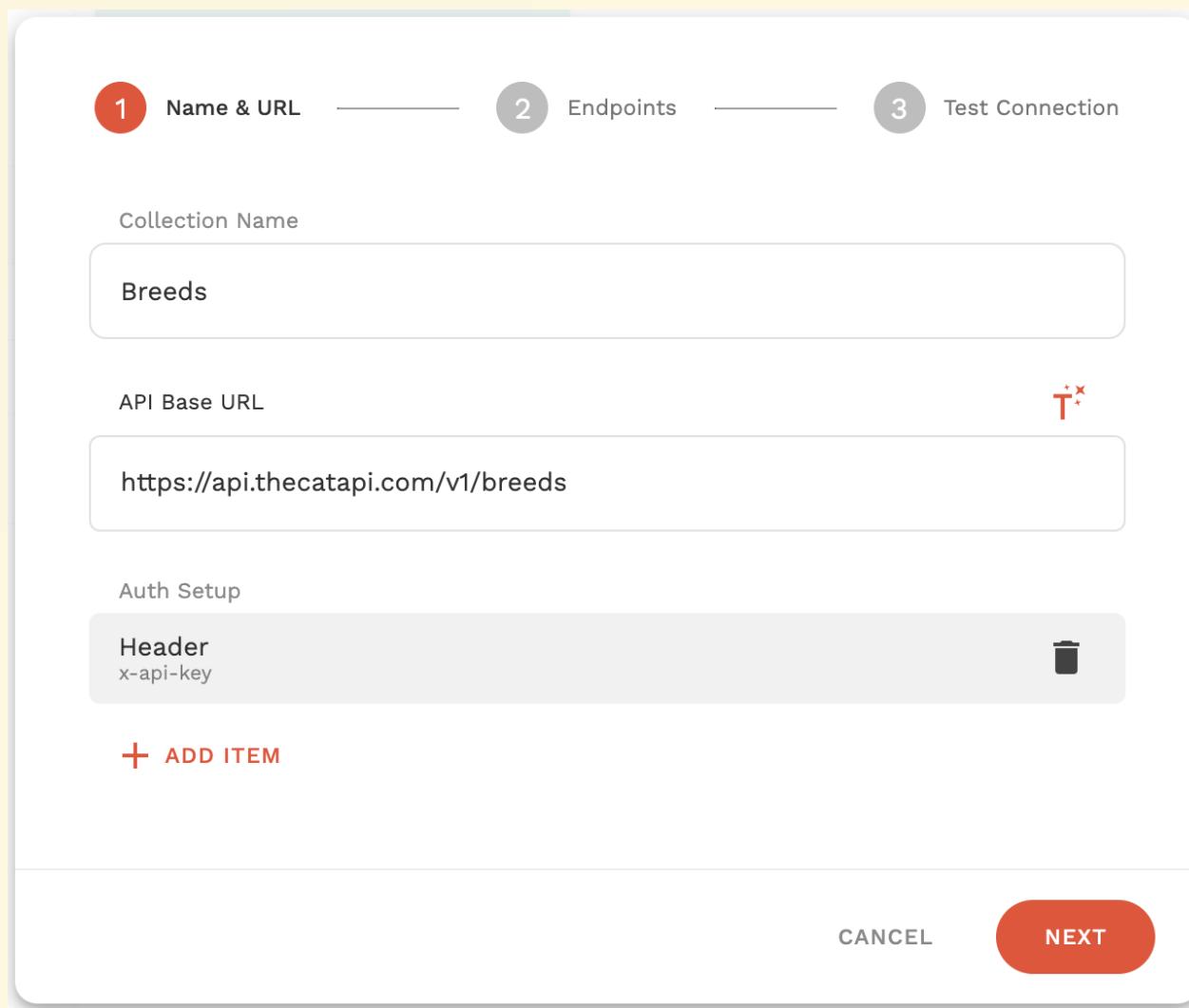
Connect to APIs
Make your app even more powerful
by connecting to 3rd party APIs

+ ADD COLLECTION

Integration Pack Trial | 14 Days Left
Upgrade now to make sure all of your external
collections stay up and running.

UPGRADE NOW

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



The screenshot shows a mobile application interface for managing API collections. At the top, there are three circular tabs labeled 1 Name & URL, 2 Endpoints, and 3 Test Connection. The first tab is highlighted with a red background.

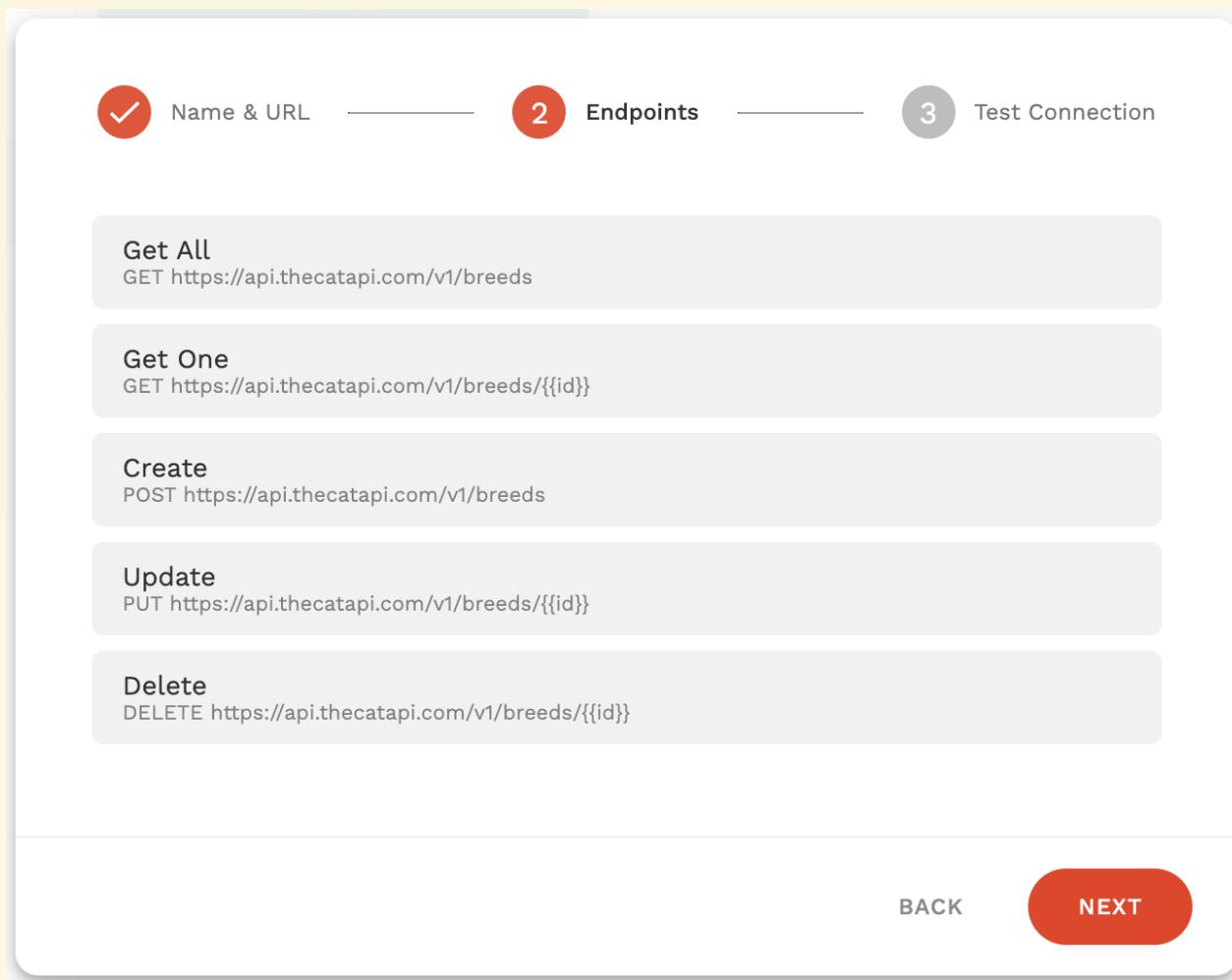
The main area contains the following fields:

- Collection Name:** Breeds
- API Base URL:** https://api.thecatapi.com/v1/breeds
- Auth Setup:** A section containing a single item: **Header** x-api-key. To the right of this item is a small trash can icon.

At the bottom right of the screen are two buttons: **CANCEL** and a large **NEXT** button.

In Adalo, you can set up five Endpoints (access methods) for each resource (in this example, breeds) that you want to access with the API.

Depending on the specifications of the API, you may need to modify it to fit your needs, but in this case, you can just click NEXT.



- Run the test and if it succeeds, click "CREATE COLLECTION".

The screenshot shows a user interface for testing a MongoDB API. 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, followed by the message "Here's the full response we got back:". A JSON object representing a cat breed is shown:

```
[
  {
    "weight": {
      "imperial": "7 - 10",
      "metric": "3 - 5"
    },
    "id": "abys",
    "name": "Abyssinian",
    "cfa_url": "http://cfa.org/Breeds/BreedsAB/Abyssinian.aspx",
    "vetstreet_url": "http://www.vetstreet.com/cats/abyssinian",
    "vcahospitals_url": "https://vcahospitals.com/know-your-pet/cat-breeds/",
    "temperament": "Active, Energetic, Independent, Intelligent, Gentle",
    "origin": "Egypt",
    "country_codes": "EG",
    "country_code": "EG",
    "description": "The Abyssinian is easy to care for, and a joy to have in your home. They are intelligent, active, and affectionate cats that bond well with their families. Their distinctive 'ticked' coat pattern is a result of a natural mutation that occurred in Egypt thousands of years ago. Abyssinians are known for their agility and grace, often leaping high into the air to catch small birds or insects. They also enjoy climbing trees and furniture, so it's important to provide them with safe places to explore. These cats are generally healthy, but like all breeds, they can be prone to certain health issues such as dental problems and eye infections. Regular veterinary check-ups and proper nutrition are key to maintaining their overall well-being.",
    "life_span": "14 - 15",
    "indoor": 0,
    "lap": 1,
    "alt_names": "",
    "adaptability": 5,
    "affection_level": 5,
    "child_friendly": 3,
    "dog_friendly": 4,
    "energy_level": 5,
    "grooming": 1,
    "health_issues": 2,
    "intelligence": 5,
    "shedding_level": 2,
    "social_needs": 5,
    "stranger_friendly": 5,
    "vocalisation": 1,
    "experimental": 0
  }
]
```

At the bottom right, there are two buttons: "BACK" and a large orange "CREATE COLLECTION" button.

The External Collection has been created.

All the data to be retrieved from the API is set as properties.

External Collections

▼  Breeds 

A id

A weight > imperial

A weight > metric

A name

A temperament

A origin

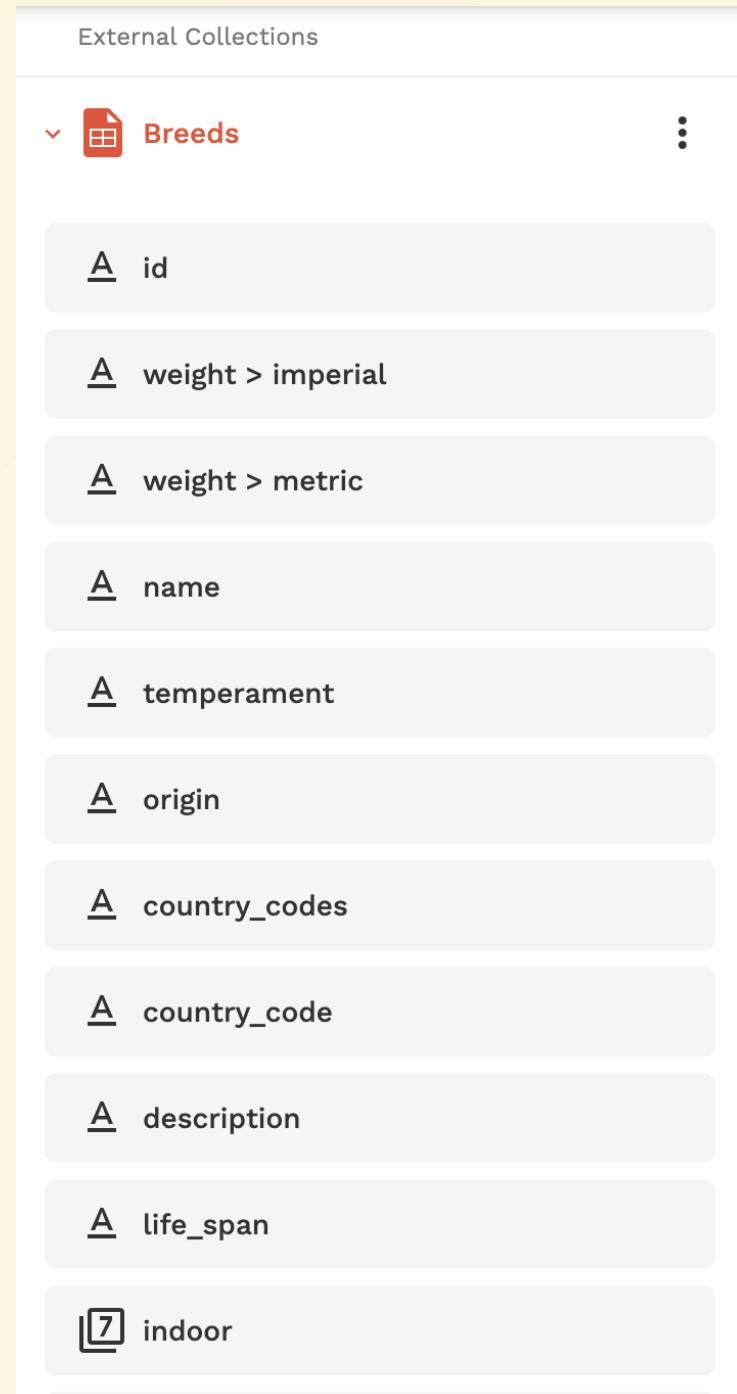
A country_codes

A country_code

A description

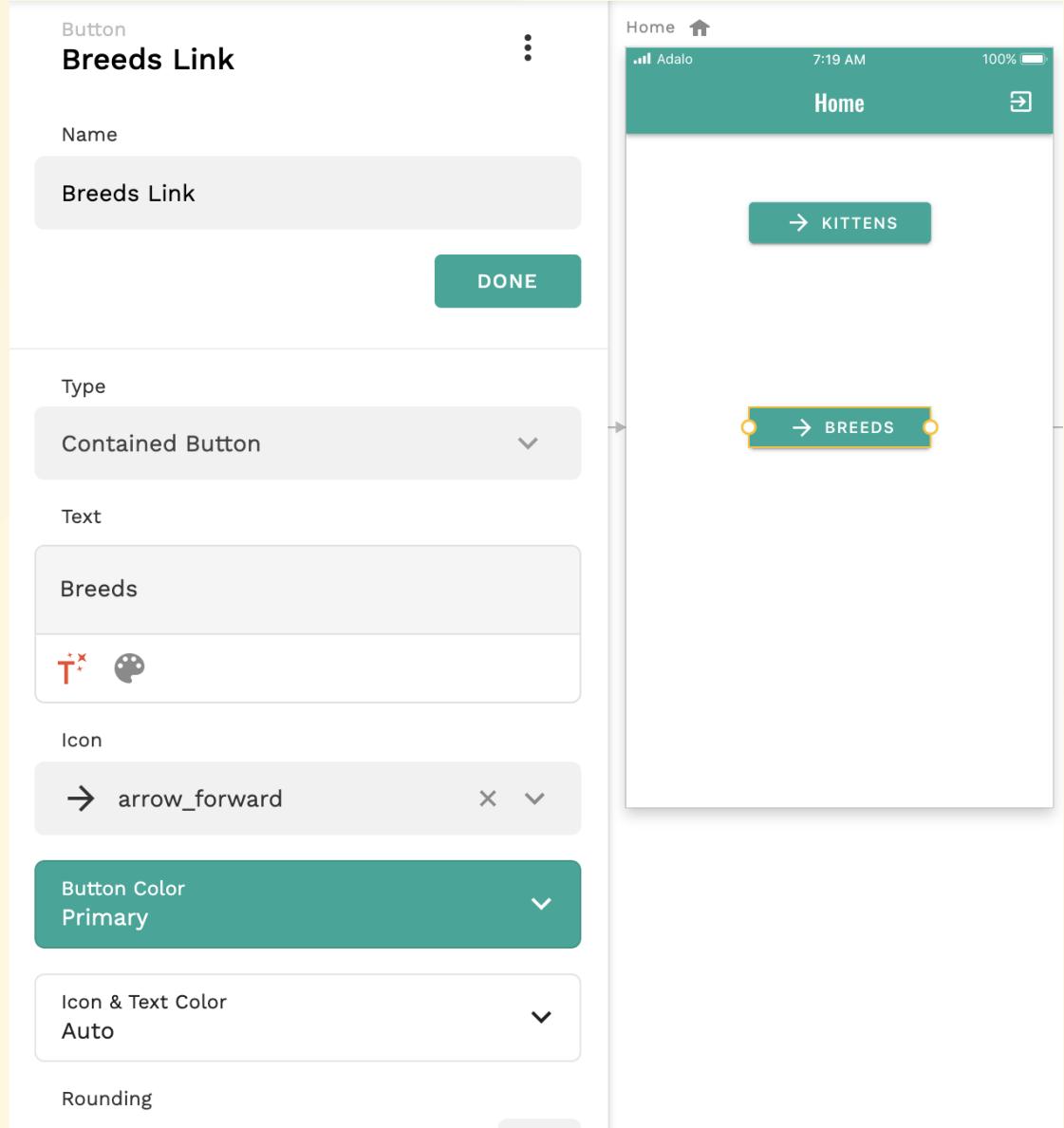
A life_span

 7 indoor

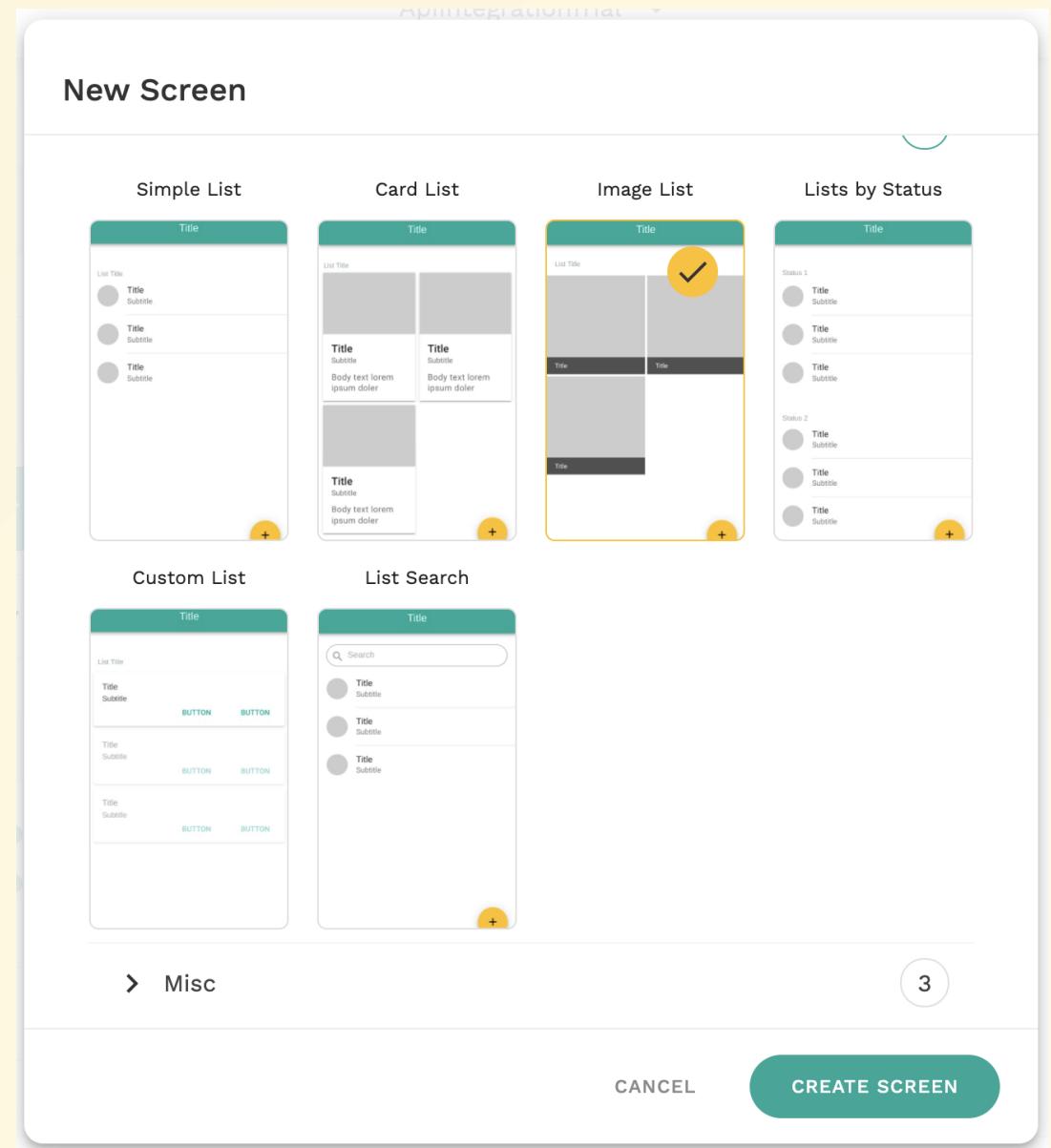


Let's list the retrieved data.

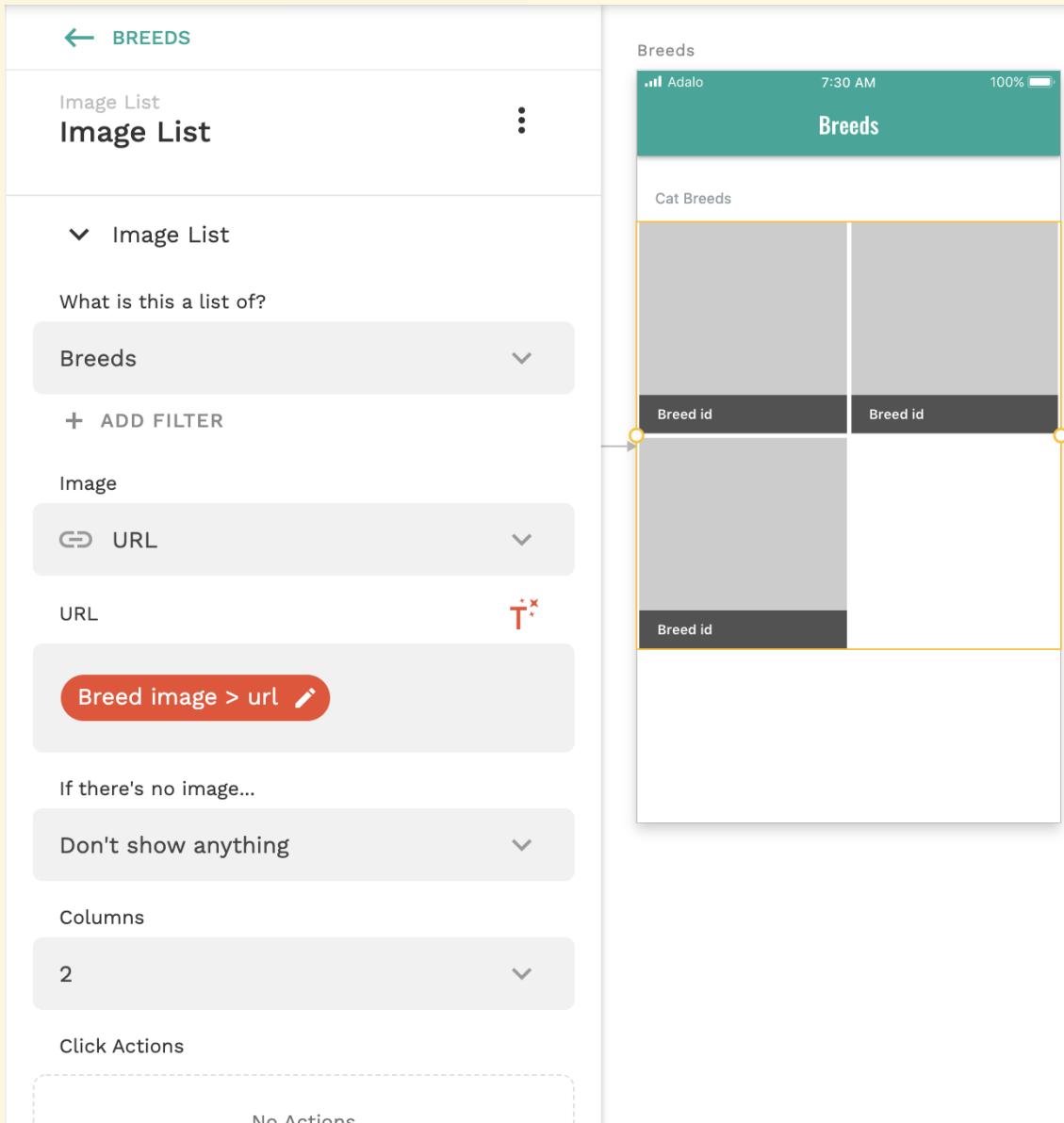
- Add a "Breeds Link" button to the Home screen.
- Add a LINK to the NEW SCREEN from ADD ACTION.



- Enter "Breeds" in Name.
- Select Image List in Template
- Click CREATE SCREEN



- Change the List Title Text to "Cat Breeds".
- Set the Image List as the Breeds Collection list.
- Set Image URL to "Breed Image > url".
- "If there's no image..." Set "Don't show anything" to "If there's no image...".
- Remove the + button at the bottom right.



- Change Text to "Breed name".

The image shows a mobile application's configuration screen on the left and a preview of the app's interface on the right.

Configuration Screen (Left):

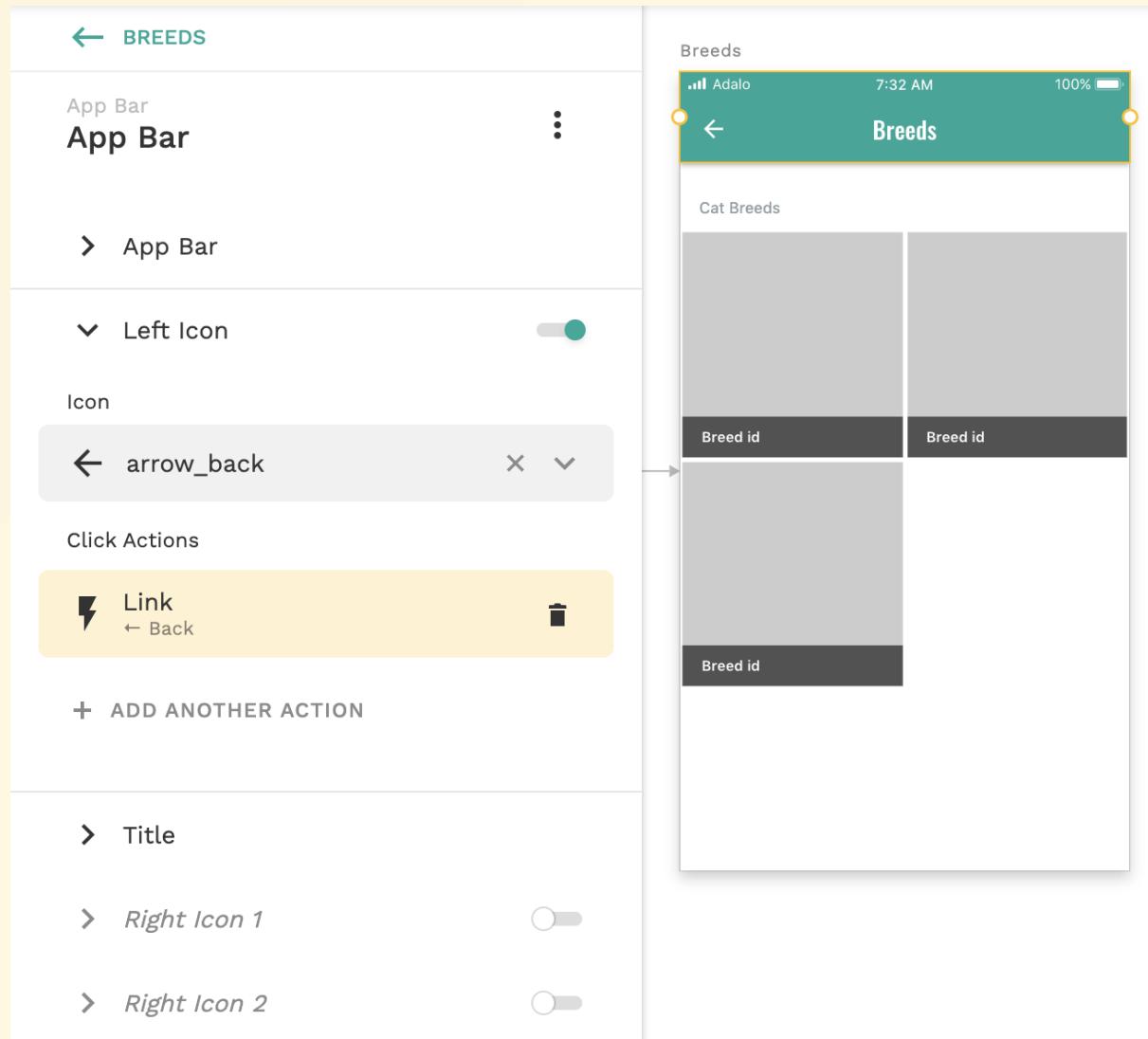
- List Header:** A toggle switch is turned off.
- Text:** A toggle switch is turned on.
- Title:** A red input field contains the text "Breed name" with a pencil icon.
- Subtitle:** An input field contains the placeholder "Enter Text..." with a red T icon and a black paint palette icon.
- Subtitle Position:** A dropdown menu is set to "Select...".
- Text Position:** A dropdown menu is set to "Select...".
- Background Effect:** A dropdown menu is set to "Select...".
- Image Style:** A section header with a right-pointing arrow.

Preview Screen (Right):

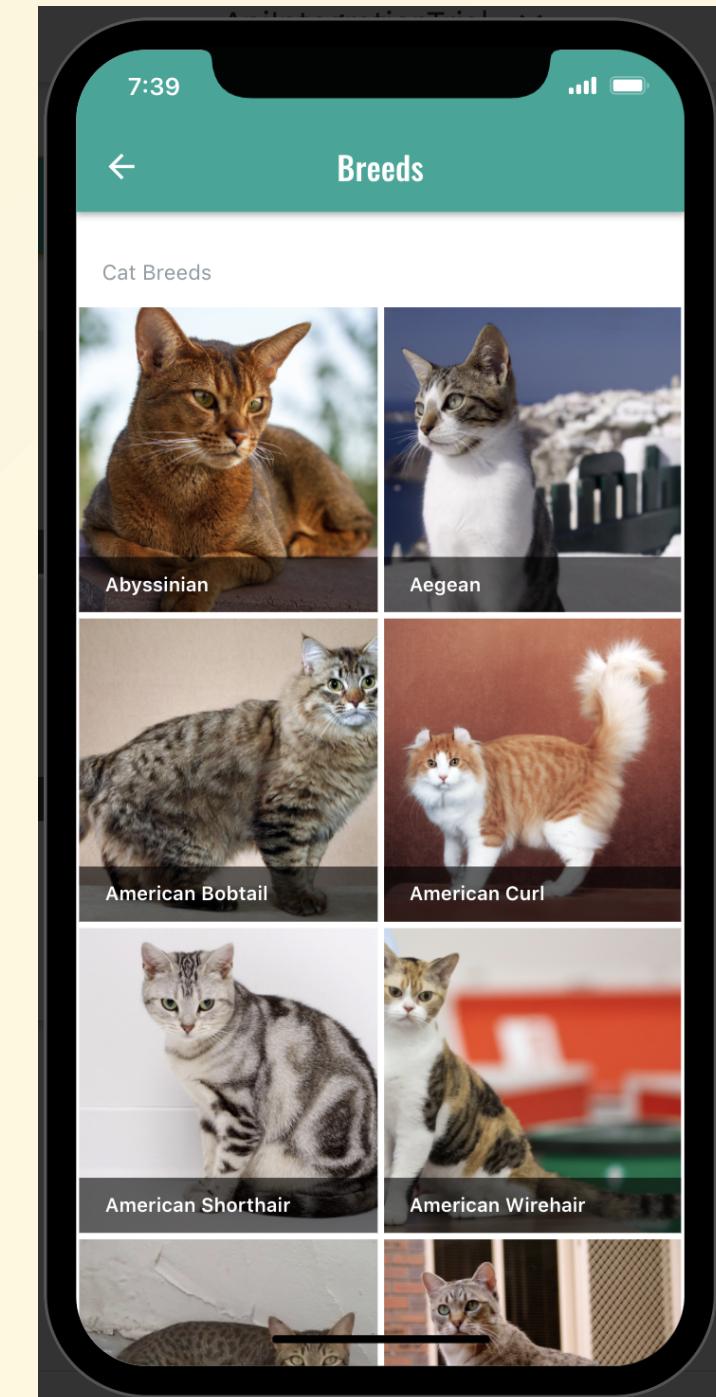
- Header:** Shows "Adalo" with signal strength, "7:37 AM", and "100% battery".
- Section:** "Breeds" with a teal header bar.
- Content:** "Cat Breeds" section with a grid of three cards. The first card has a yellow border and is highlighted. Each card contains the text "Breed name".

Show the back icon since it is hidden.

- Click on the App Bar
- Change the Left Icon toggle to ON.



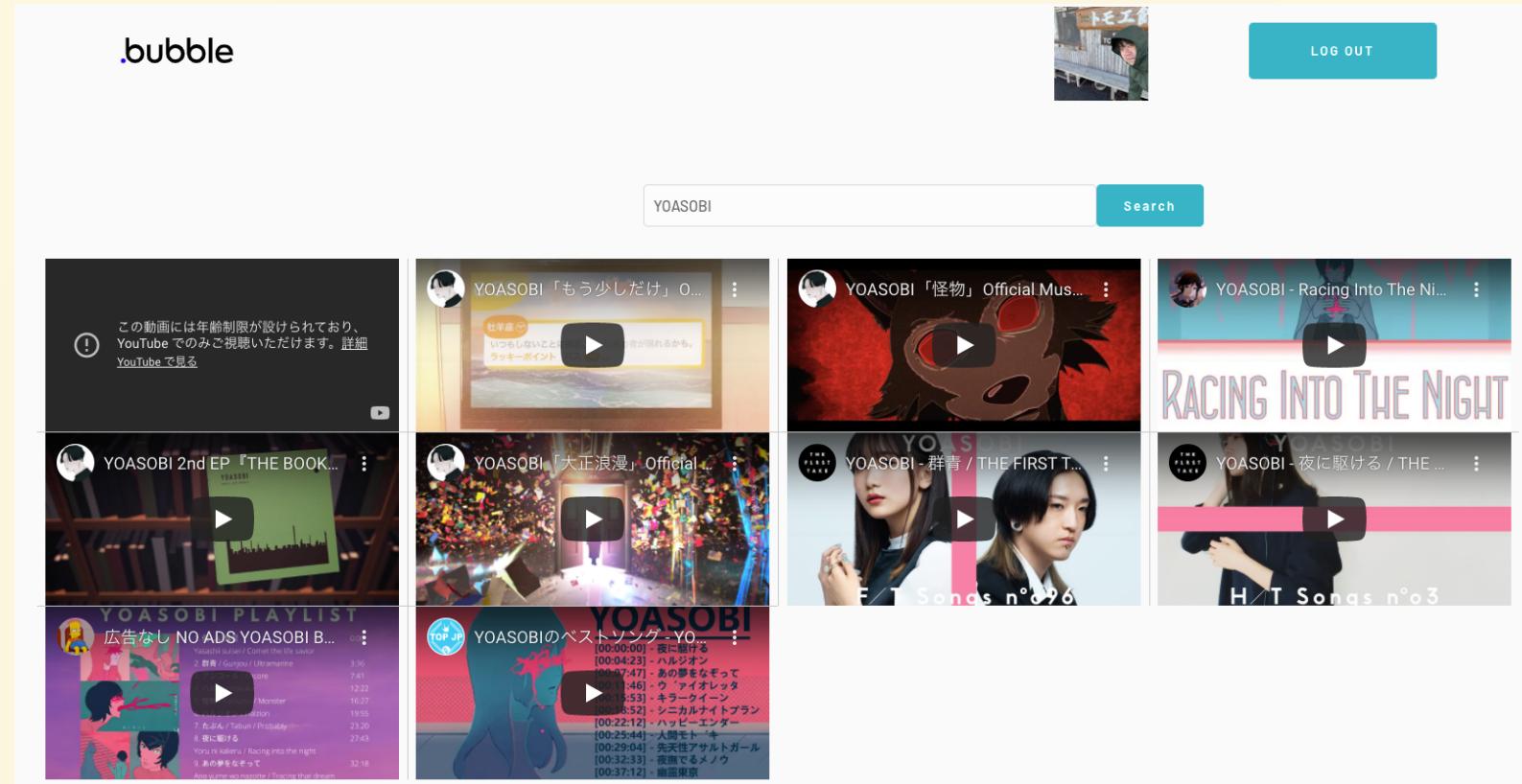
Checking with the Preview function, the list of cat breeds is now displayed.



That's it for the introduction of external integration in Adalo.

External Integration with Bubble

Next, let's try the external integration with Bubble.
The screen we will create looks like this.



The screen will have the following functions

- Display of YouTube video list
- Social login using your Google account
- Dynamic search of YouTube video list.

First, we need a new application to build on.

- Create a new application called Advanced-bubble-{your name}.

Create a new app X

Name of this new app
 Start from a template (optional)
Want to start from a template instead?
Explore the [templates on the marketplace](#).

You can add a domain name later.

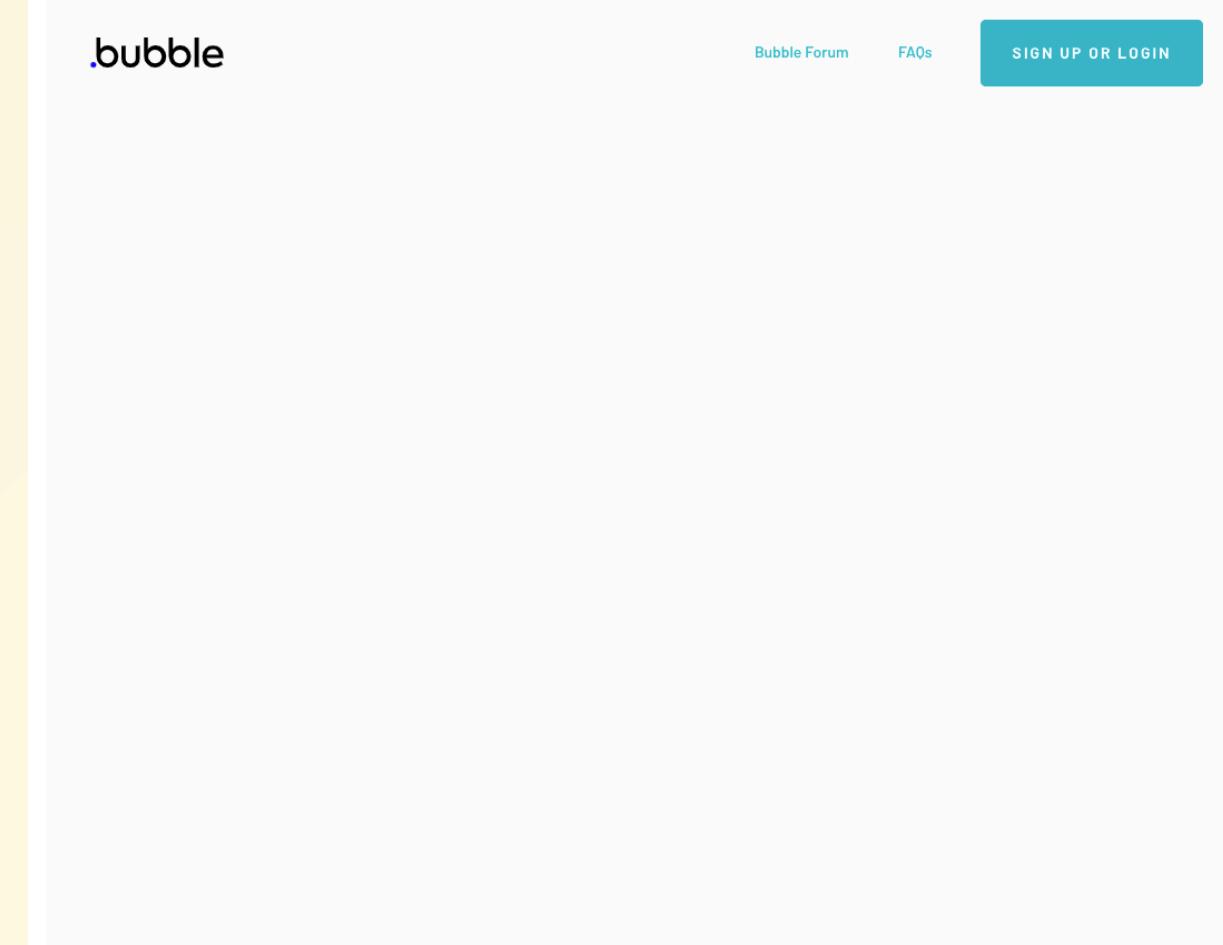
What kind of app are you building?
 Details of what you're building
 Try to be specific so that we can help!

Is it customer-facing or internal?
 What's your goal with this application?

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

Create a new app

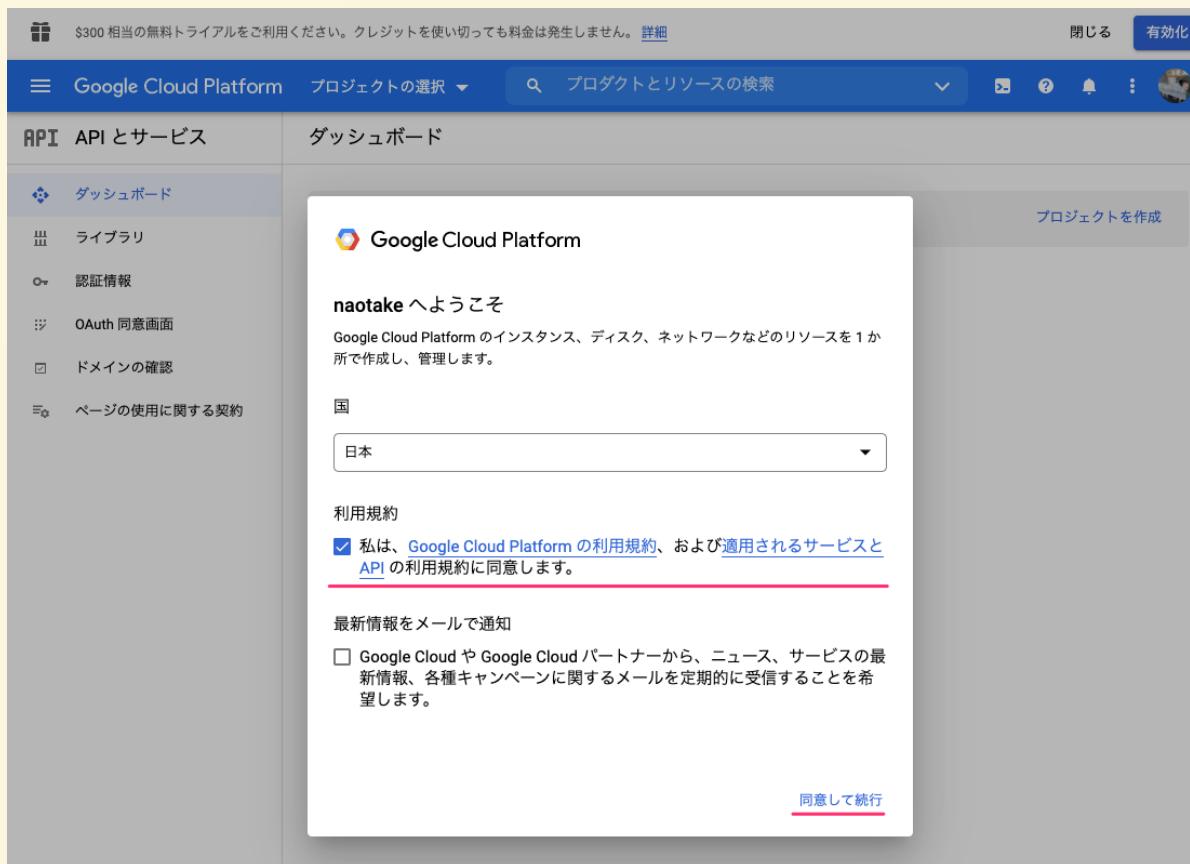
- As in the third article, delete all the elements except the header.
 - It is OK to delete everything and then reposition the header component.



Next, let's prepare the Google account side.

- Do the necessary preparations to display the YouTube video list
- Sign in to your Google account and access the following URL
<https://console.developers.google.com/>

- When you access the Google Cloud Platform (GCP) screen for the first time, you will see a dialog box like this. Select your country, check the Terms of Service, and click "Agree and Continue".



- After agreeing, you will see the dashboard of APIs and Services, and click on the Create Project link to create a project (box) for this lecture

The screenshot shows the Google Cloud Platform API & Services dashboard. At the top, there is a banner with a gift icon and the text: '\$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。' with a '詳細' link. To the right of the banner are '閉じる' and '有効化' buttons. Below the banner is the navigation bar with the 'Google Cloud Platform' logo, a 'プロジェクトの選択' dropdown, a search bar, and other navigation icons. The main content area has a sidebar on the left with icons for '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', and 'ページの使用に関する契約'. The main panel displays a message: '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.' with a 'プロジェクトを作成' button highlighted with a red underline.

- When the Create New Project screen appears, specify an appropriate name for the project name and create it.

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](#)



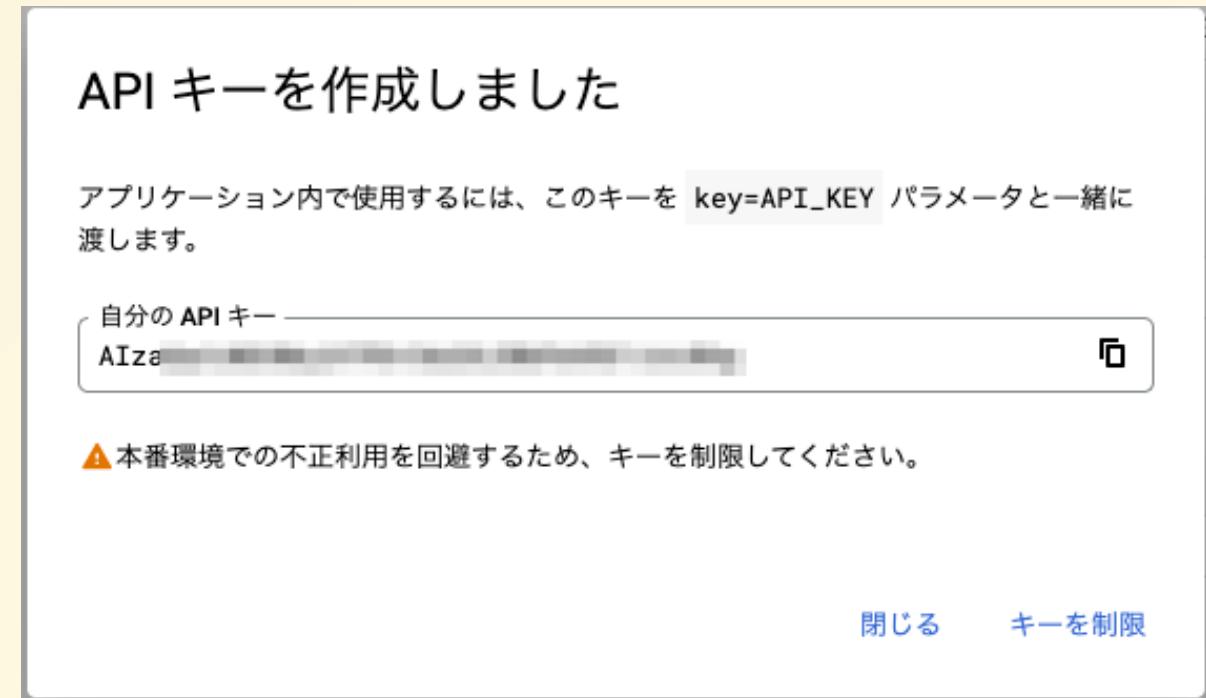
- Once the creation is complete, the dashboard for the created project will appear
 - If the name of the project you created is displayed at the top of the screen, to the left of Google Cloud Platform, you are good to go.

The screenshot shows the Google Cloud Platform API & Services dashboard. The navigation bar at the top includes the Google Cloud logo, the project name 'bubble advance', a search bar, and a dropdown menu. The main content area has a sidebar on the left with links: ダッシュボード, ライブラリ, 認証情報 (which is 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 panel is titled '認証情報' and contains a sub-section 'API キー'. It displays a table with columns '名前', '作成日', '制限', and 'キー'. There are two entries: one with a redacted key and another with a yellow key. Below this is a section for 'OAuth 2.0 クライアント ID' with a similar table. The final section is 'サービス アカウント' with a table showing a single entry for 'メール'.

- Next, we will issue the authentication key that is required to display the YouTube video list.
- Select "Authentication Information" from the left menu.
- The right panel will change to Authentication Information, and click "+ Create Authentication Information" at the top of the screen.
- Select "API Key" from the pop-up window.

The screenshot shows the Google Cloud Platform interface. The left sidebar has a red underline under '認証情報'. The main area shows '認証情報' with a sub-menu for 'API キー'. A modal dialog is open over the page, also titled '認証情報を作成'. It contains three sections: 'API キー' (with a warning about OAuth Client ID), 'OAuth クライアント ID' (describing user consent), and 'サービス アカウント' (describing bot accounts). At the bottom of the dialog is a '同意画面を設定' button. The status bar at the bottom says 'OAuth 2.0 クライアント ID'.

- You will then see a popup saying that the API key creation is complete, so write down the value.
- You will be asked to limit the number of keys, but we will delete these keys at the end of the lecture, so we'll leave it at that.



- Now that we have the key to get the YouTube list from Bubble, let's configure the Bubble side
- First, we need to prepare Bubble to retrieve the YouTube list via the API

Preparing Bubble for API integration

- Select Plugins from the left menu and click the Add plugins button
- Search for **API** and install the **API Connector** plugin

The screenshot shows the 'Install New Plugins' page in the Bubble app editor. On the left, there's a sidebar with filters for 'API', 'Types' (Action, API, Background Services, Element, Event, Login Service, Storage), and 'Categories' (Analytics, Blog, Compliance, Containers, Customer Support, Data (things), Ecommerce, Email, Energy, Health & Fitness, Input Forms, Internationalization, Location, Media, Mobile). The main area lists four plugins: 'API Connector' by Bubble (selected with a red border), 'Zapier (Legacy)' by Zapier, 'Discord API (OUTDATED)' by Tristan Forum, and 'Widgio API Connector' by twobrothers. Each plugin card includes its name, developer, description, number of apps, version information, and an 'Install' button.

Install New Plugins

API

Sort by: name usage rating date price

Filters

API Connector

By Bubble

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

289,709 apps

Install

Zapier (Legacy)

zapier

Zapier connects the web apps you use to easily move your data and automate tedious tasks. Trigger any Zap from your workflows.

By Bubble

12,498 apps

Install

Discord API (OUTDATED)

Free

Integrate your web app with Discordapp services Supplies Discord login service plus Webhooks, and guilds a user is in. Previously known as Discordapp API

By Tristan Forum

10,372 apps All versions 3.5/5.0 Current version 3.5/5.0 Plugin page

Install

Widgio API Connector

Free

Connect to the Widgio API to create fully customized web widgets for your Bubble application.

By twobrothers.

6,170 apps All versions 3.0/5.0 Current version 3.0/5.0 Plugin page

Install

DONE

⚠ Recurring subscriptions to paid plugins are on paid plans only. Click here to adjust your application plan.

- This plugin is used to connect the Bubble app to the other side of the API using the public API.
- Some of the Bubble plugins are specific to a particular API, and some of them are simpler to configure than this API Connector.
- In this example, we will use the YouTube API to display a list of videos via the API Connector plugin, which is the most basic form.

- Now let's configure the settings to use the YouTube API.
- First, click on "Add another API" to start setting up a new API integration

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 this, there is descriptive text about the API and a note about good practices. A red box highlights the 'Add another API' button at the bottom left. Another red box highlights the 'Uninstall this plugin' button at the bottom right.

.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

- After clicking "Add another API", you will see the API configuration part like this

The screenshot shows the configuration interface for a new API. 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' and 'Shared parameters for all calls', each with an 'Add a shared header' and 'Add a shared parameter' button respectively. A large expandable section for 'API Call' is shown, containing a 'Name' field (set to 'API Call') and an 'expand' button. At the bottom, there is a link to 'Import another call from cURL' and a blue 'Add another call' button, with an 'expand all calls' button to its right.

API Name New API Authentication None or self-handled collapse

Shared headers for all calls

Add a shared header

Shared parameters for all calls

Add a shared parameter

Name API Call expand

Import another call from cURL

Add another call expand all calls

- Here is a brief description of each item
- Please set 1 and 2 together.

1. API Name: The group name for this API integration.

- In this case, YouTube

The screenshot shows the configuration interface for a new API integration:

- ① API Name:** A text input field containing "New API".
- ② Authentication:** A dropdown menu set to "None or self-handled".
- ③ Shared headers for all calls:** A section with a "Add a shared header" button.
- ④ Shared parameters for all calls:** A section with a "Add a shared parameter" button.
- Name:** A text input field for naming the API call, currently empty.
- Import another call from cURL:** A button labeled "Add another call".
- expand** and **collapse** buttons are located in the top right corner of the main configuration area.
- expand all calls** is located in the bottom right corner of the expanded API call section.

2.Authentication: The common authentication method used by APIs under this group.

- In this case, "None or self-handled".

The screenshot shows a user interface for managing API configurations. At the top, there are fields for 'API Name' (set to 'New API') and 'Authentication' (set to 'None or self-handled'). Below these are sections for 'Shared headers for all calls' and 'Shared parameters for all calls', each with an 'Add a shared header' and 'Add a shared parameter' button respectively. A large expandable section labeled 'Name' and 'API Call' is shown below. At the bottom, there is a link to import from cURL and a blue button to add another call.

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

collapse

expand all calls

3. Shared headers for all calls: Common header information for all APIs under this group.

- Not specified at this time.

The screenshot shows the configuration interface for a new API. At the top, there are fields for 'API Name' (set to 'New API'), 'Authentication' (set to 'None or self-handled'), and a trash bin icon. Below these are two sections: 'Shared headers for all calls' and 'Shared parameters for all calls'. Each section has an 'Add a shared header' or 'Add a shared parameter' button. The 'Shared headers for all calls' section is currently expanded, showing a table with a single row labeled 'API Call'. The 'Shared parameters for all calls' section is collapsed. At the bottom, there is a link to 'Import another call from cURL' and a blue 'Add another call' button. A yellow 'expand' button is located next to the collapsed 'Shared parameters for all calls' section, and a grey 'expand all calls' button is located at the bottom right.

① API Name New API

② Authentication None or self-handled

collapse

③ Shared headers for all calls

Add a shared header

④ Shared parameters for all calls

Add a shared parameter

Name API Call

expand

Import another call from cURL

Add another call

expand all calls

4. Shared parameters for all calls: Common parameter information for all APIs under this group.

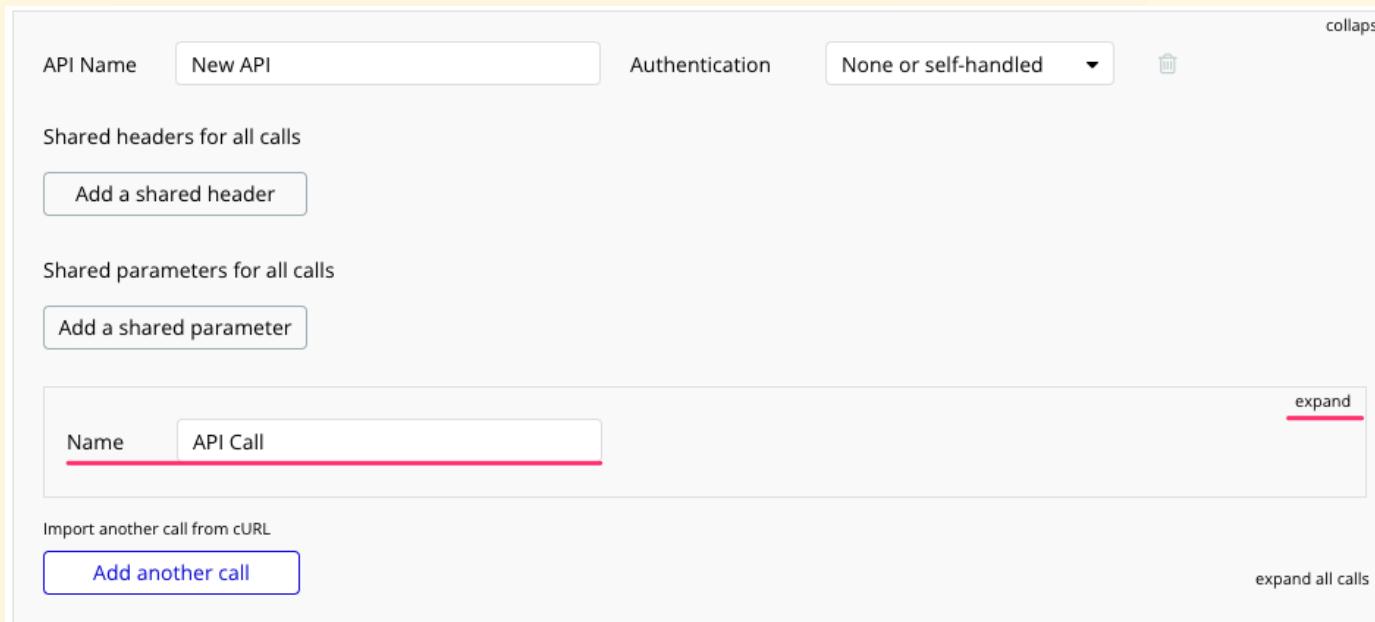
- Not specified this time.

The screenshot shows the configuration page for a new API. At the top, there are fields for 'API Name' (set to 'New API'), 'Authentication' (set to 'None or self-handled'), and a trash bin icon. A 'collapse' button is in the top right corner. The main area is divided into sections:

- ① API Name:** 'New API'
- ② Authentication:** 'None or self-handled'
- ③ Shared headers for all calls:** Contains a button 'Add a shared header'.
- ④ Shared parameters for all calls:** Contains a button 'Add a shared parameter'.

Below these sections is a collapsed panel labeled 'Name' with 'API Call' inside. It has an 'expand' button in the top right corner. At the bottom left, there is a link 'Import another call from cURL' and a button 'Add another call'. At the bottom right, there is an 'expand all calls' button.

- Next, let's set up the specific API contents for this group.
- To the right of the block labeled "API Call" in the previous screen, there is a link called "expand".
- Then the specific API settings will be displayed, and I will briefly explain them.



1. Name: The specific name of the API.

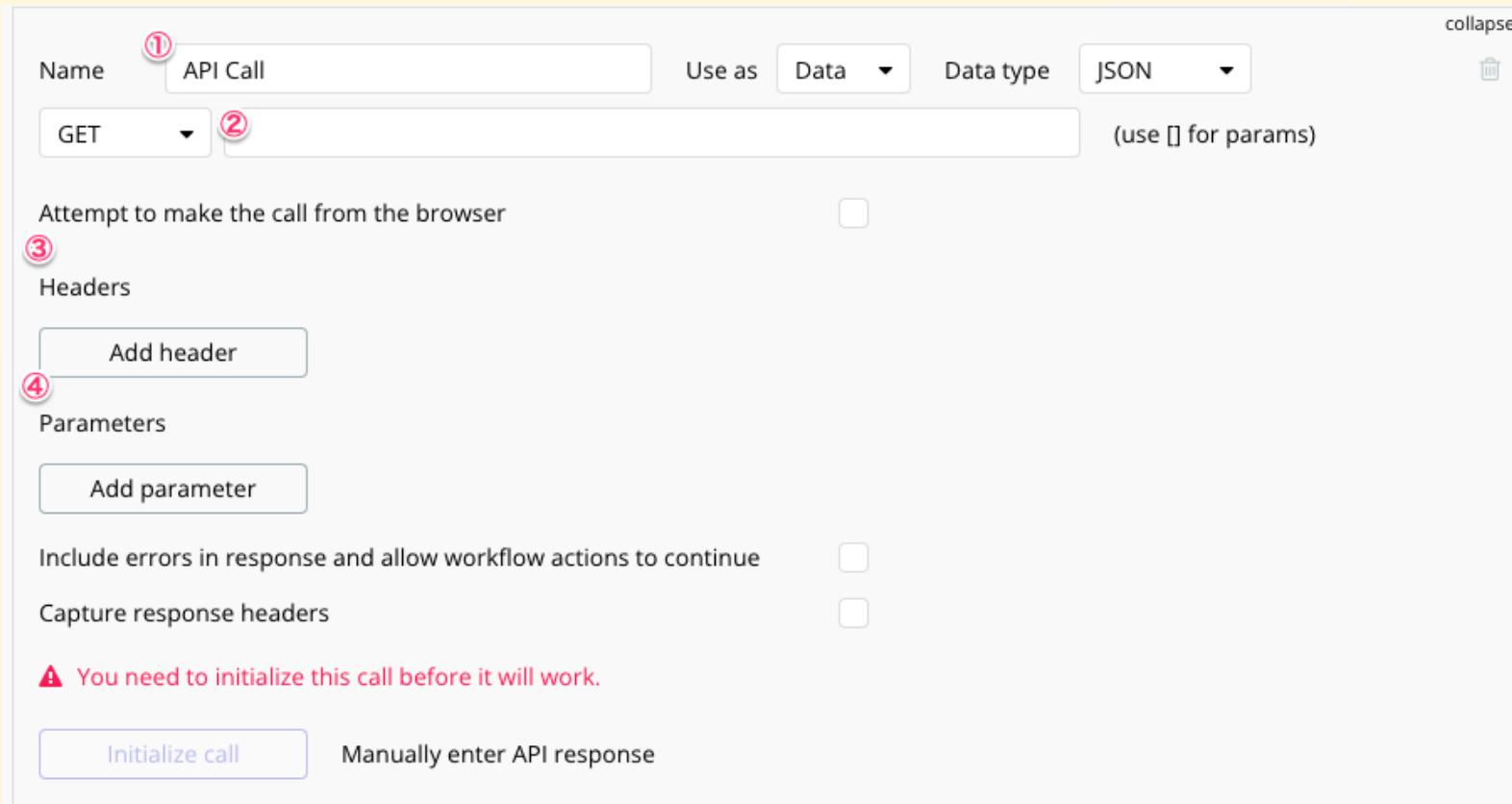
- In this case, enter "search".

The screenshot shows a configuration interface for an API call. At the top, there is a 'Name' field containing 'API Call' with a circled '1' above it. To its right are dropdown menus for 'Use as' (set to 'Data') and 'Data type' (set to 'JSON'). A 'collapse' button and a trash bin icon are also present. Below the name field is a 'Method' dropdown set to 'GET' with a circled '2' above it, and a text input field '(use [] for params)'. The main body of the form contains sections for 'Headers' (with an 'Add header' button, circled '3'), 'Parameters' (with an 'Add parameter' button, circled '4'), and several checkboxes: '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 two buttons: '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
Add header
Parameters
Add parameter
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: The specific URL of the API.

- In this case, enter <https://www.googleapis.com/youtube/v3/search>.



3.Headers: Header information specific to this API.

- Not specified this time.

The screenshot shows a configuration panel for an API call. At the top, there are fields for 'Name' (labeled ①), 'Method' (labeled ②, showing 'GET'), 'Data type' (labeled ③, set to 'JSON'), and a note '(use [] for params)'. Below these are sections for 'Headers' (labeled ④) and 'Parameters'. Under 'Headers', there is an 'Add header' button. Under 'Parameters', there is an 'Add parameter' button. At the bottom, there are checkboxes for 'Attempt to make the call from the browser' and 'Include errors in response and allow workflow actions to continue'. A note at the bottom states '⚠ You need to initialize this call before it will work.' with buttons for 'Initialize call' and 'Manually enter API response'.

Name: API Call
Method: GET
Data type: JSON
(use [] for params)
Attempt to make the call from the browser
Headers
Add header
Parameters
Add parameter
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

4.Parameters: Parameter information specific to this API

- This time we will specify the following two parameters: Private / Allow blank / Optional as shown in the attachment. 1.
 1. key: **key**, Value: the API key you just issued in your Google account.
 2. key: **q**, Value: Keyword to search for the action on YouTube (feel free to specify).

- Once everything is set up, it looks like this

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

- After the settings are done, check if the settings are correct
- Click on the "Initialize call" button at the bottom

- You will probably get a Status code 403 error message.
- This is what it says, but in summary it looks like this

YouTube Data API v3 has been disabled in your project.
Please go to this URL, enable it, and try again.

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"
      }
    ]
  }
}
```

- It seems that the YouTube Data API that we used is not enabled on each person's Google account.
- This is because Google has turned off features that not everyone will use, and only those who need them can turn them on themselves.
- So, since we want to use the YouTube Data API, let's access the URL in the message and enable it

<https://console.developers.google.com/apis/api/youtube.googleapis.com/overview?project={project ID of each person}>

- You should now be kindly redirected to the "YouTube Data API v3" screen.
- Click on "Enable" to activate it as shown on the screen.

The screenshot shows the Google Cloud Platform interface with a blue header bar. On the left, there's a navigation menu icon and the text "Google Cloud Platform". Next to it is a user profile icon with the name "bubble advance" and a dropdown arrow. To the right of the header are several icons: a magnifying glass for search, a gear for settings, a question mark for help, a bell for notifications, and three vertical dots for more options. A circular profile picture of a person is also present.

The main content area has a white background. At the top left is a red button with a white play icon. To its right, the text "YouTube Data API v3" is displayed in bold black font, followed by the word "Google" below it. Below this section, a descriptive text reads: "The YouTube Data API v3 is an API that provides access to YouTube data, such as videos, playlists,...".

At the bottom of the main content area, there are two buttons: a blue rectangular button with the Japanese text "有効にする" (Enable) and a white rectangular button with the Japanese text "この API を試す" (Try this API) with a small arrow icon. Below these buttons, there is some very small, illegible text.

At the very bottom of the page, there is a horizontal navigation bar with three items: "概要" (Overview), "ドキュメント" (Documentation), and "サポート" (Support). The "概要" item is underlined, indicating it is the active page.

- After a while, you should see a screen like this
- If the red line at the top of the screen says "Disable API", you have successfully enabled it!



- Now that the YouTube Data API has been activated, go back to the Bubble screen and click the "Initialize call" button.
- If successful, you should see a popup that says "Returned values - search"

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

SAVE Cancel

- This shows the results of running the
`https://www.googleapis.com/youtube/v3/search` API that we set up.

I'll spare you the details, but the point is that

- The key point is that the `items(list)` part is a list of videos of search results.
- The type of the list is `search item`, which means that it is the data of multiple systems included in the search results.
- And the most important part of this setting is the `id videoId`.

- If you've ever watched YouTube, you'll know that this Videoid is the video that uniquely identifies each video, and this Id is also required to display the video on the screen.
- For other items that you don't want to refer to, select **Ignore field** from the pull-down menu because it is confusing.
- By doing this, when you handle them as Dynamic data, they will not be shown in the choices, and you will not have to worry about them when selecting settings.

- This is what it looks like when everything is set up
- When you are done, click "SAVE" to save your settings.

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.

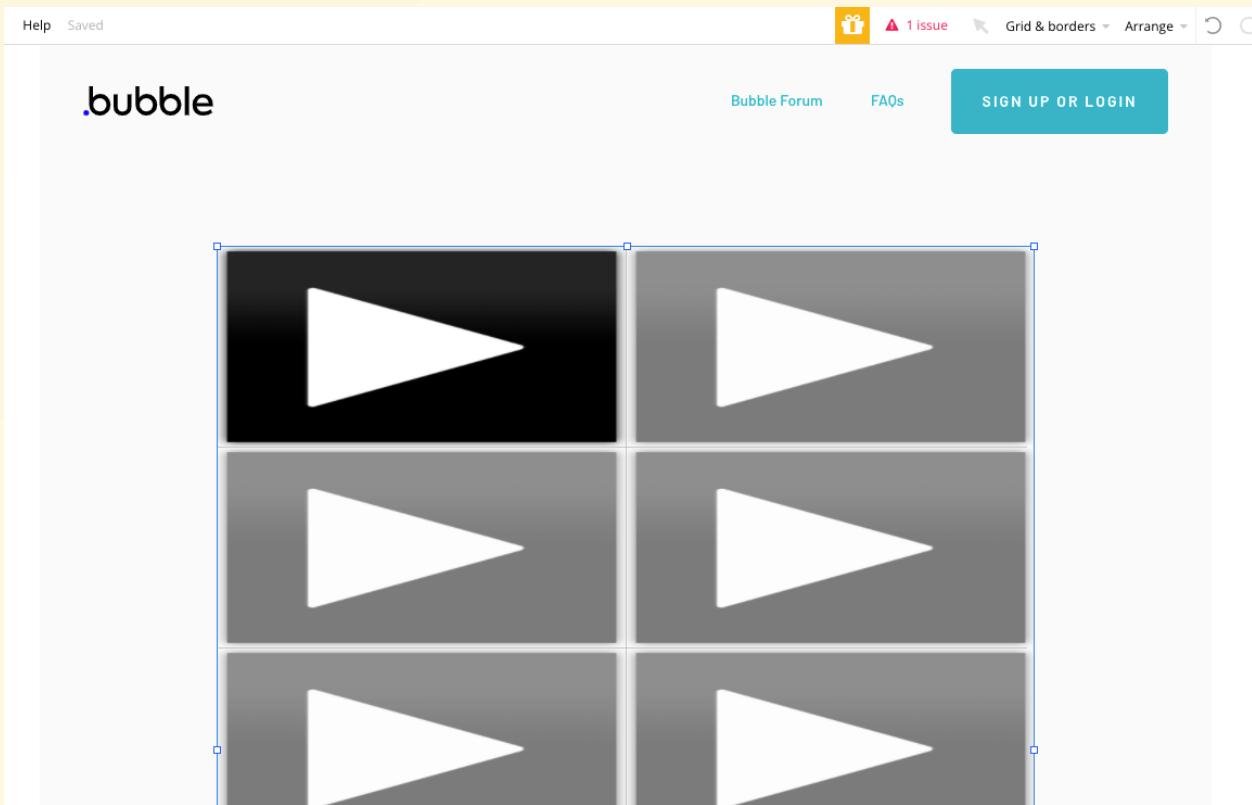
kind youtube#searchListResponse	Ignore field ▾
etag bMKu0aIXS6rekNwT5RVHZLOYR-o	Ignore field ▾
nextPageToken CAUQAA	Ignore field ▾
regionCode US	Ignore field ▾
pageInfo totalResults 1000000	Ignore field ▾
pageInfo resultsPerPage 5	Ignore field ▾
<u>items (list)</u> (see fields below)	search item ▾
kind 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

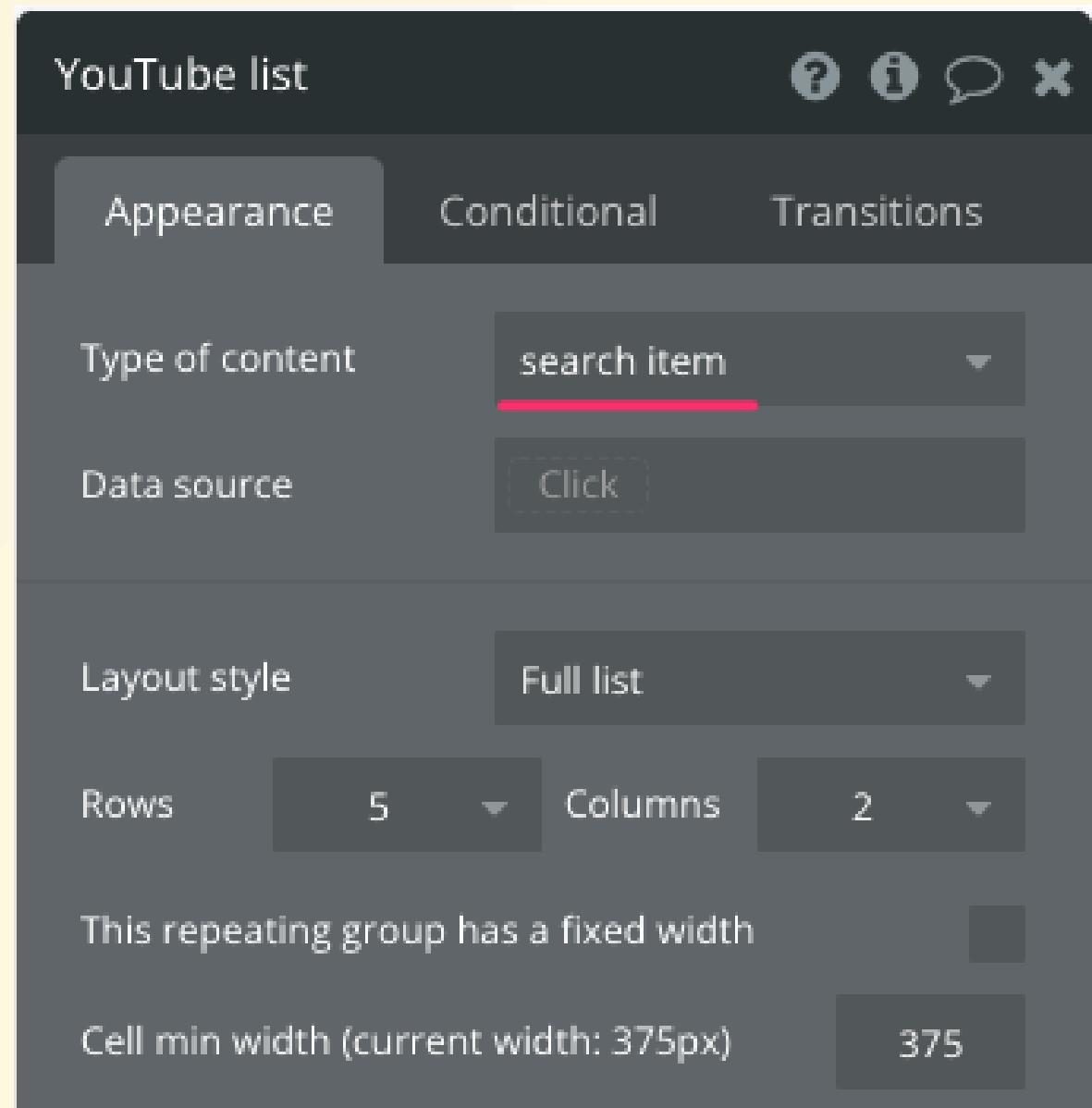
Preparing for Screen Display

- Now that Bubble's API is ready, it's time to set up the screen.
- Since you have mastered the screen setup in the 3rd and 4th lectures, we will just give you an overview and let you build the screen layout.

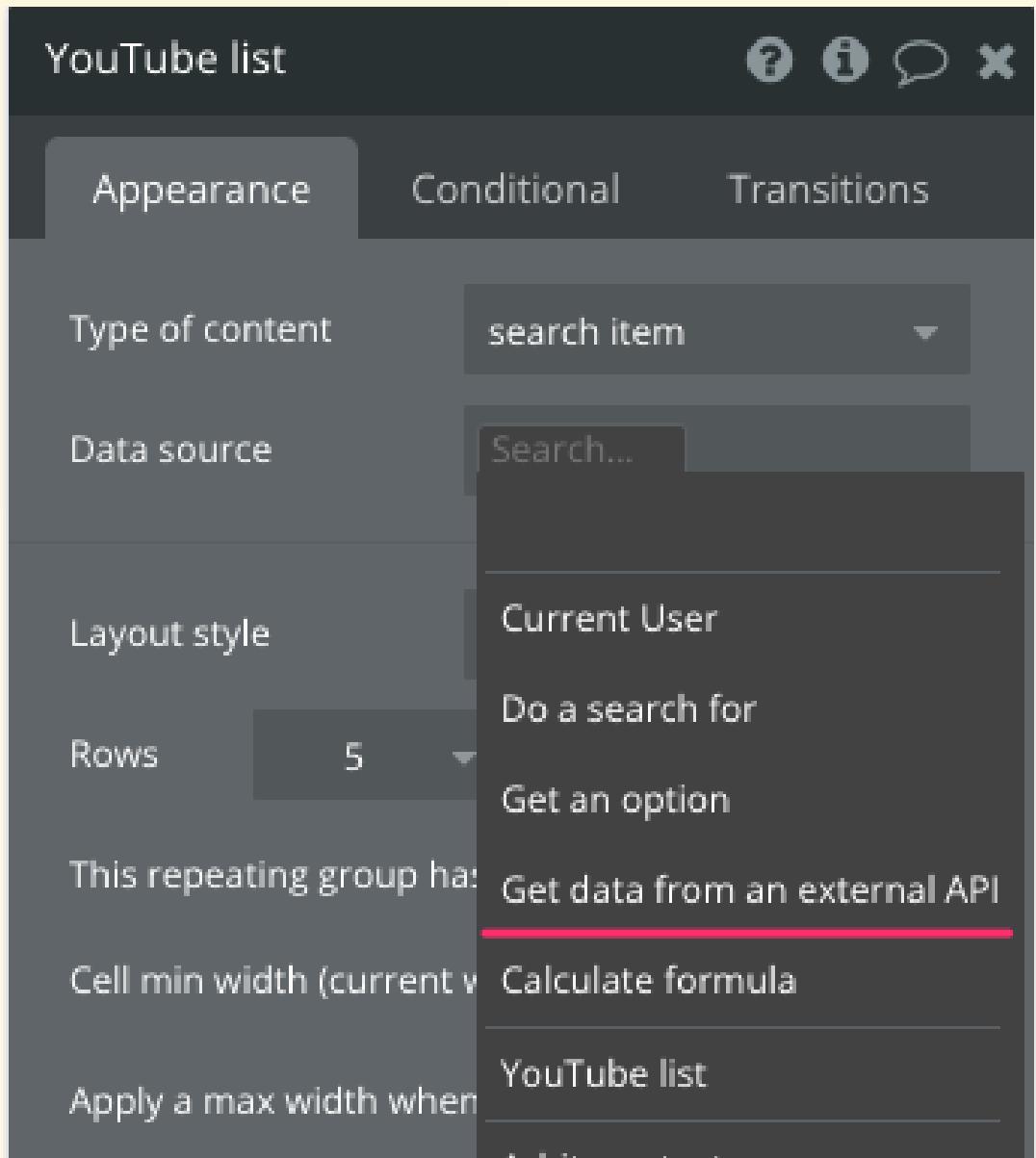
- On the index page, we will construct a list using Repeating Group.
- Place a Video in each cell of the Repeating Group
- It looks like this



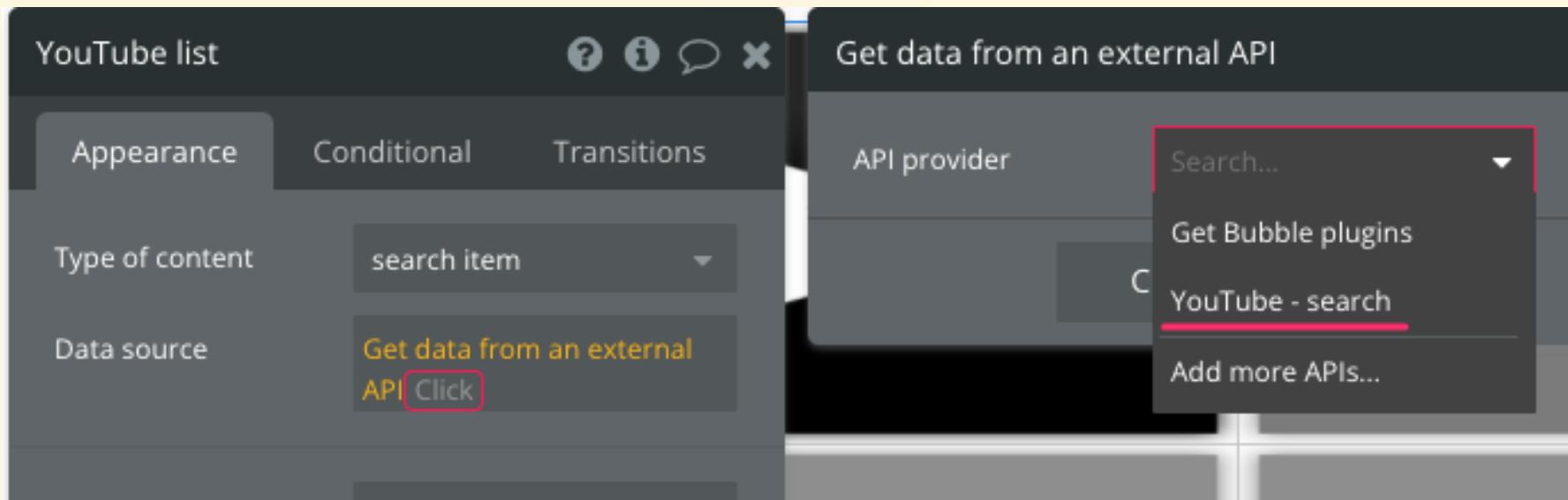
- Now that the layout is done, let's actually set up the YouTube list display.
- First, the Type of content for the Repeating Group will be the `search item` type that we checked when we ran the API earlier.



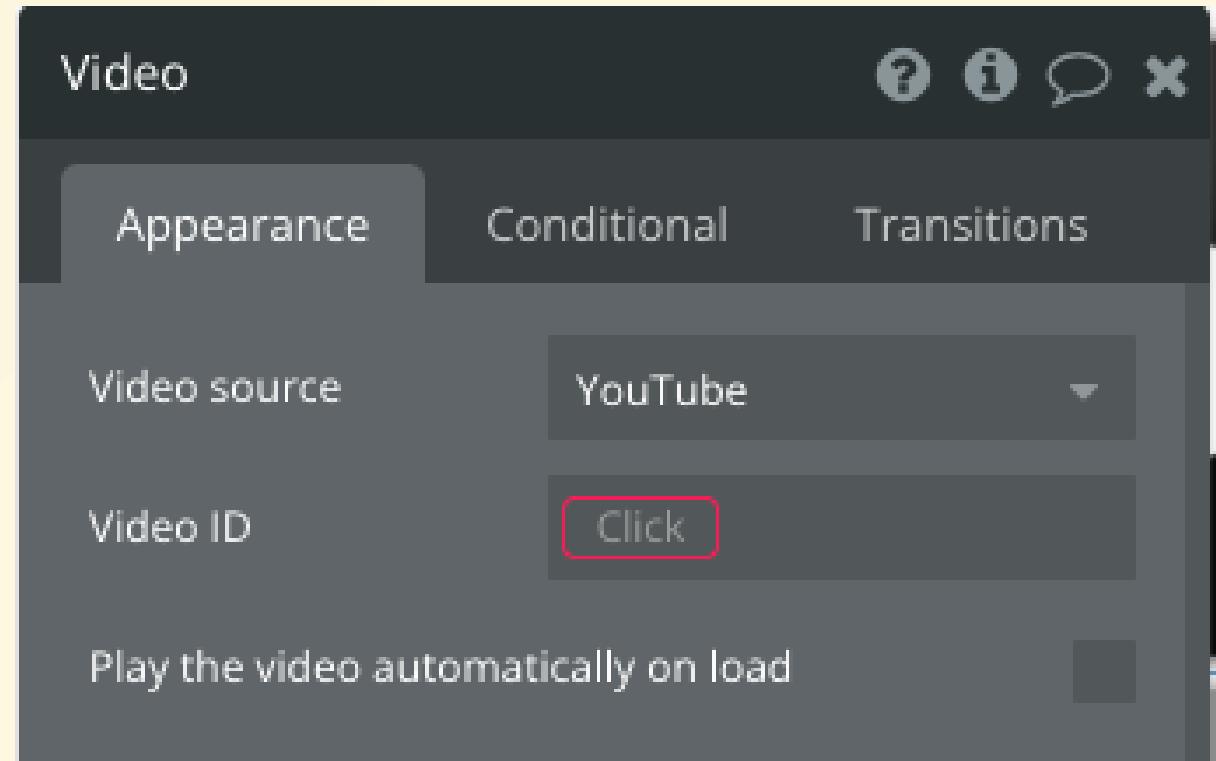
- Next, let's look at the data source, which in this case is a list of YouTube videos obtained from the API.
- If you look at the suggestions from Click, you will see an item called Get data from an external API



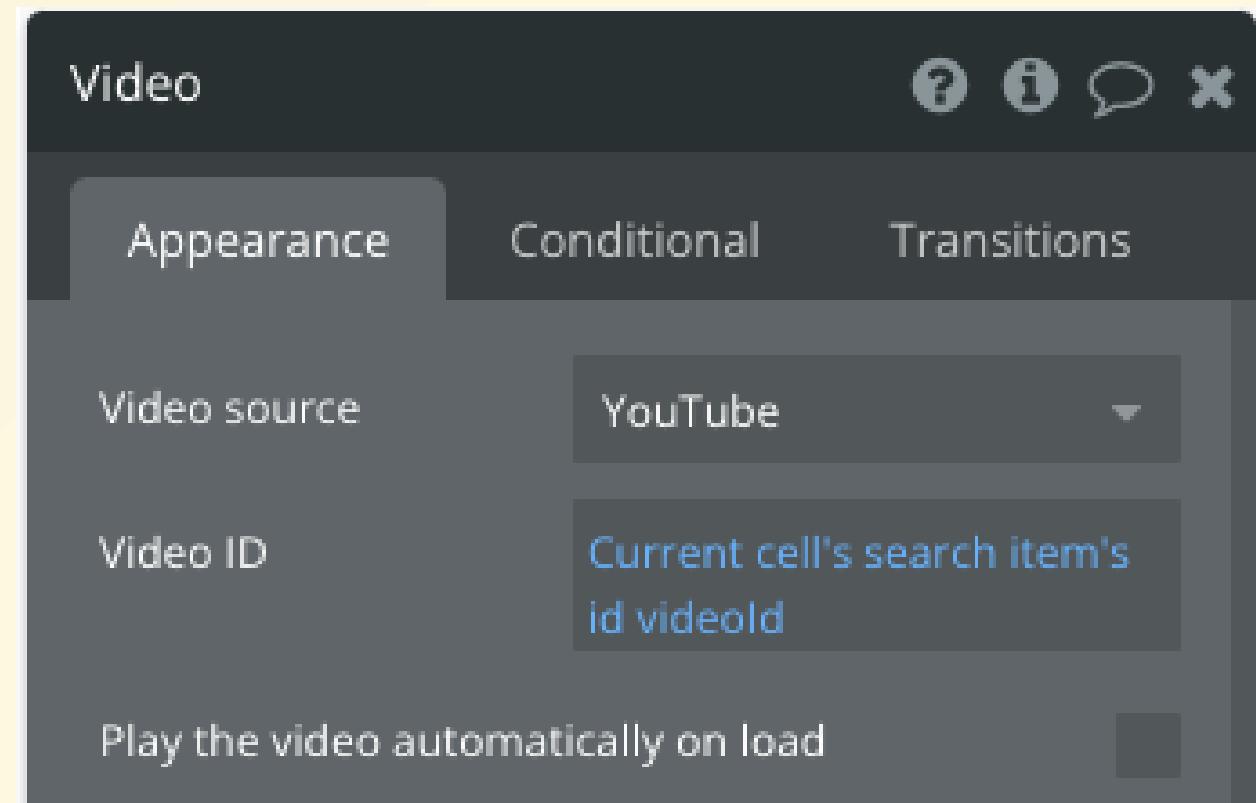
- When the Get data From an external API popup appears next to it, select the API provider pull-down
- In the pull-down, you will see the candidate YouTube - search that you set up earlier, select it
 - Before the hyphen is the group name of the API, and after it is the specific name of the API



- Next, let's configure the settings for each cell, but first, let's configure everyone
- Video source is YouTube
- Then you will see the Video ID field, so you can refer to the **Video Id** that you saw when you set up the API earlier.



- It looks like this.

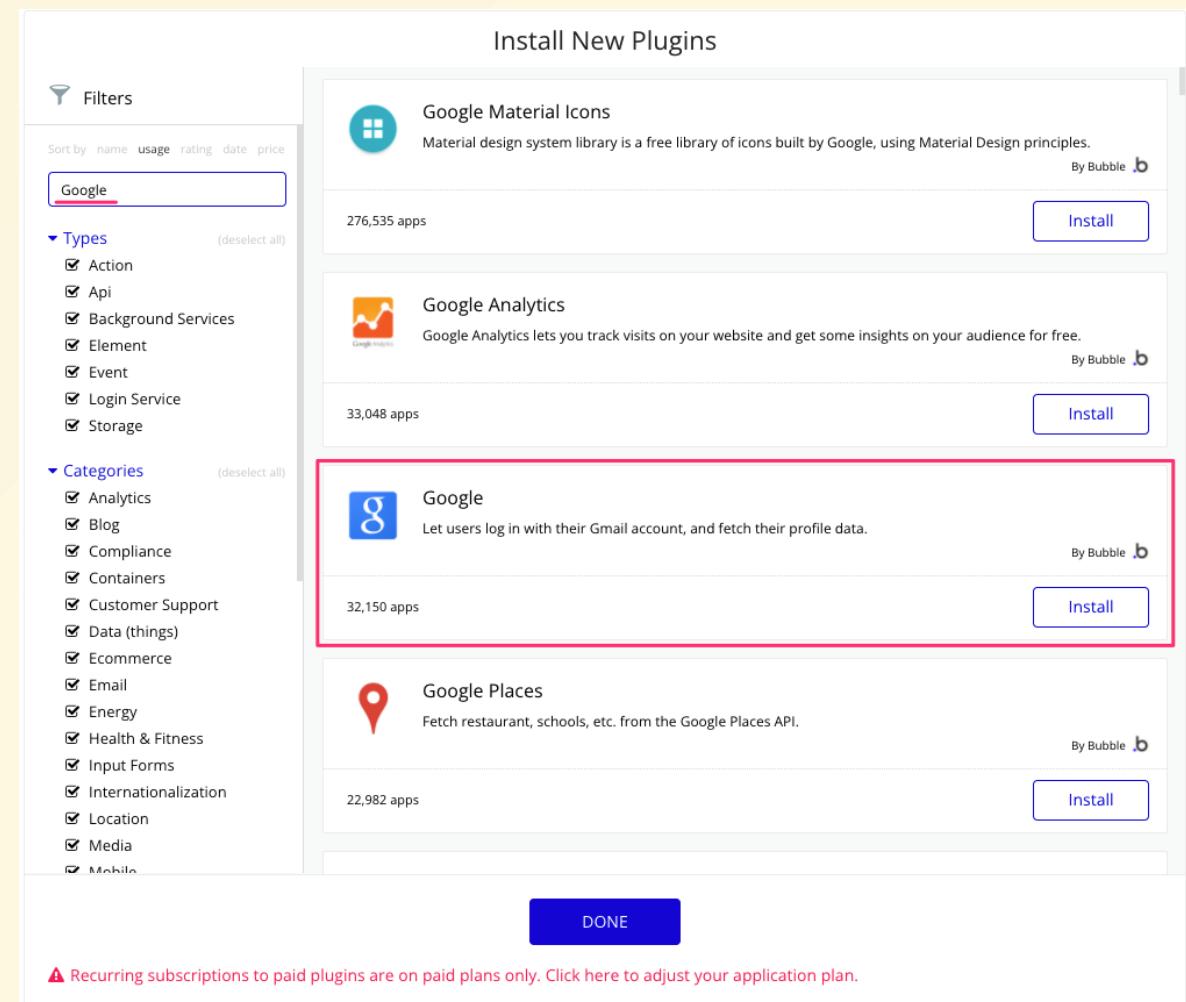


- Now you're all set! Let's preview it right away!
- Do you see the list of videos for the keywords you specified in the API beforehand?



Let's try Google Login

- Now let's try to log in to the Bubble app using our Google account!
- First, install the Google plugin



- Select the Google plugin you installed.
- Check the `Use a generic redirect URL` box in the right panel.
 - Note down this URL as well, but since you can't select it on the screen, just replace the app name in the following URL with your own app name and note it down
`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.

Preparing to authenticate back to the app after logging in to Google

- Access the Google Cloud Platform screen

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

- Select your project and choose OAuth Consent screen from the left menu

The screenshot shows the Google Cloud Platform interface for configuring an OAuth Consent screen. The top navigation bar includes the Google Cloud Platform logo, the project name "Bubble test", and a search bar. On the left, a sidebar lists several options under "API & Services": ダッシュボード, ライブラリ, 認証情報, and OAuth 同意画面, with "OAuth 同意画面" currently selected. The main content area is titled "OAuth 同意画面". It contains a section for "User Type" with two radio button options: "内部" (selected) and "外部". The "内部" option is described as allowing only users within the organization to use the app, while "外部" allows users with Google accounts. At the bottom right of the main content area is a blue "作成" (Create) button.

Google Cloud Platform Bubble test プロダクトとリソースの検索

API & Services

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](#)。

- Select "External" for User Type and "Create".
- After entering the required information, save and go to next

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" (selected with a red underline). Below each option is a detailed description. 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。](#)

- App Information

- App name: The name of the app you created with bubble
- User Support Email: Email address of your Google account.

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。

- Approved domain
 - bubbleapps.io
- Developer Contact Information
 - Email address of your Google account

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](#)

- Do not change the scope settings, save and go to the next step.

- Enter your own Google email address as the test user.
- After setting, save and go next.



- Now that the registration is complete, let's issue the authentication information.
- Select Authentication Information from the left menu, and click "+ Create Authentication Information".
- Select OAuth Client ID from the submenu.

The screenshot shows the Google Cloud Platform interface. The top navigation bar includes the Google Cloud logo, the project name "Bubble test", and a search bar. On the left, a sidebar menu is open, with "認証情報" (Authentication Information) selected, indicated by a red underline. The main content area is titled "認証情報" (Authentication Information) and shows a sub-section for "API キー" (API Key). A callout box highlights the "OAuth クライアント ID" (OAuth Client ID) option, which is described as requesting user consent for data access. Below this, there are sections for "サービス アカウント" (Service Account) and "OAuth 2.0 クラウド" (Cloud OAuth 2.0). At the bottom, there is a "wizardで選択" (Select via Wizard) section. The top right of the main content area has a "+ 認証情報を作成" (Create Authentication Information) button, which is also underlined in red.

- When the Create OAuth Client ID screen appears, enter the following information and click "Save."
- Application Type: Web Application
- Name: 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](#)

- Accepted redirect URI: The URL that you wrote down when you installed the Google plugin on Bubble
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 *

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 [?](#)

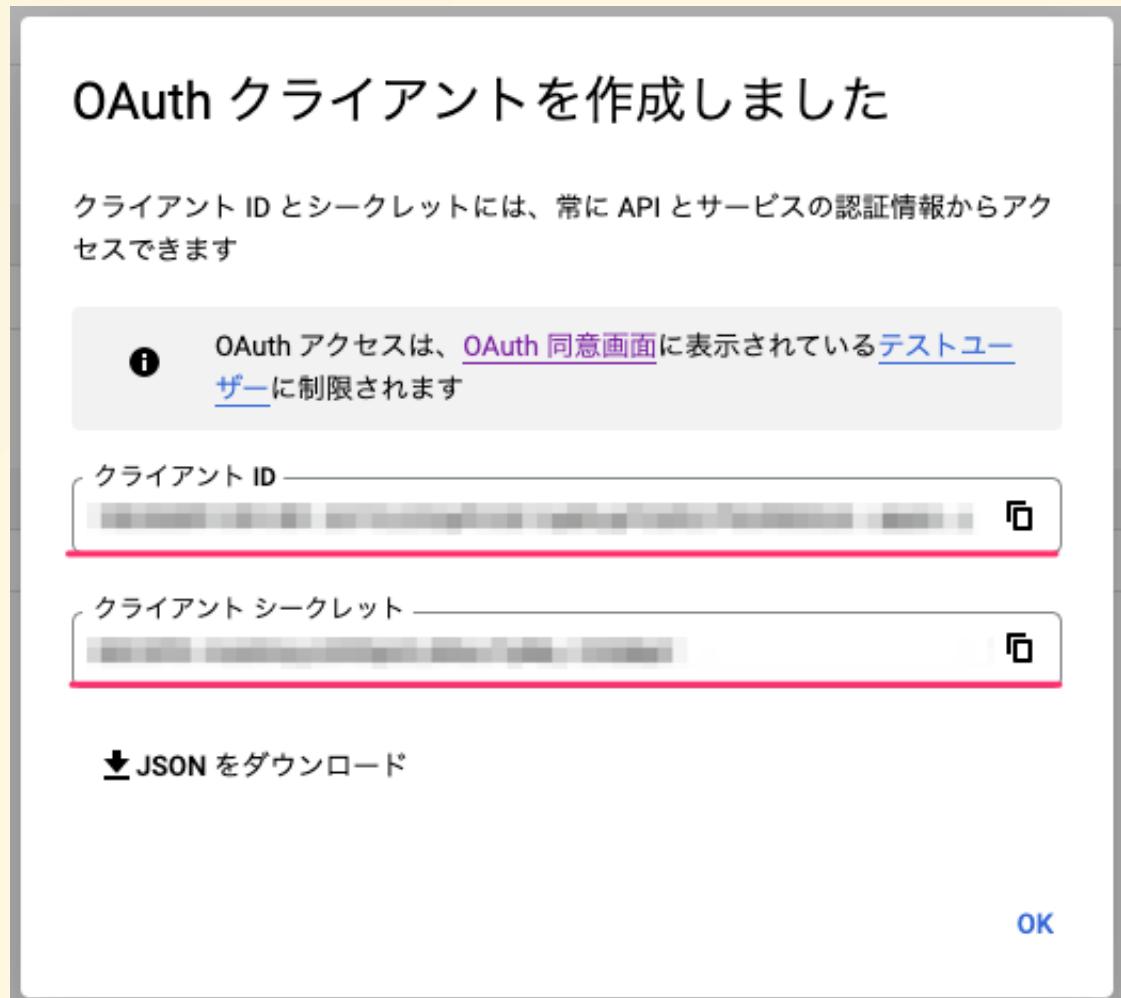
ウェブサーバーからのリクエストに使用します

URI *

[+ URI を追加](#)

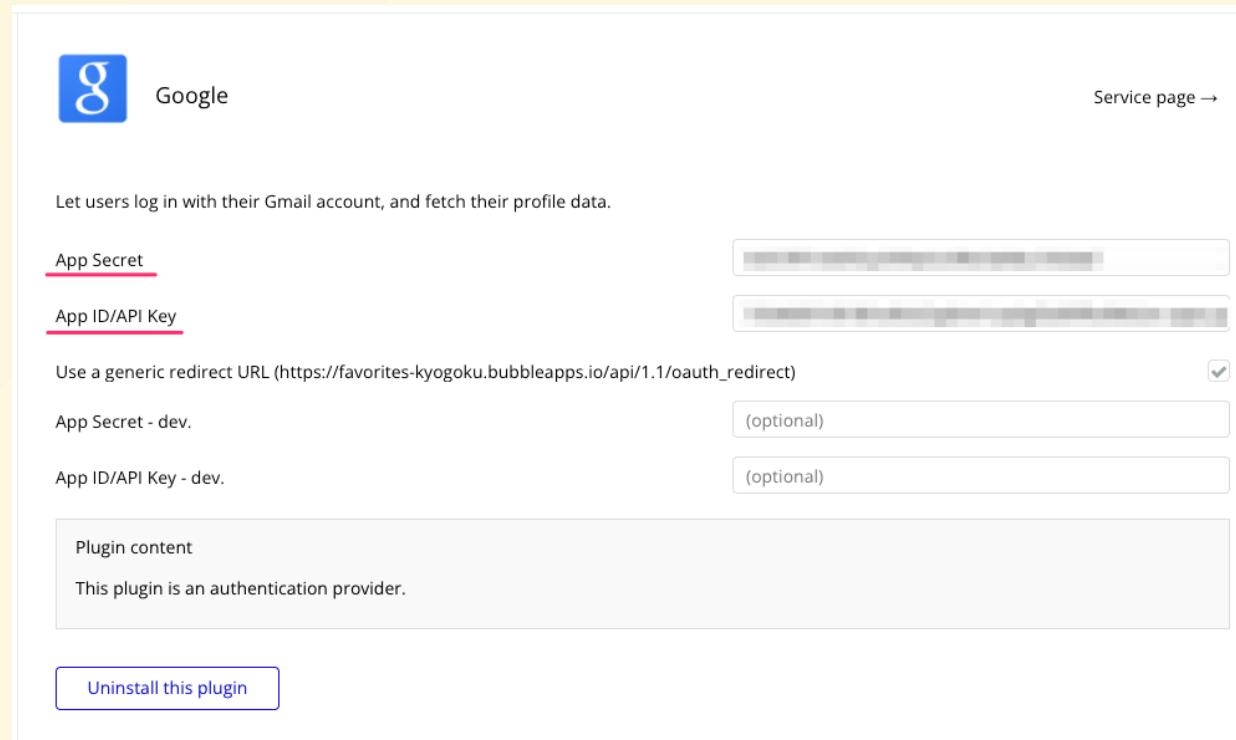
[作成](#) [キャンセル](#)

- You will see a popup that says "OAuth client created", and the client ID and client secret will be displayed there, so make a note of it.
 - You can also download the JSON file if you might forget it.
- Close with OK after writing it down



Set up Bubble with the issued credentials

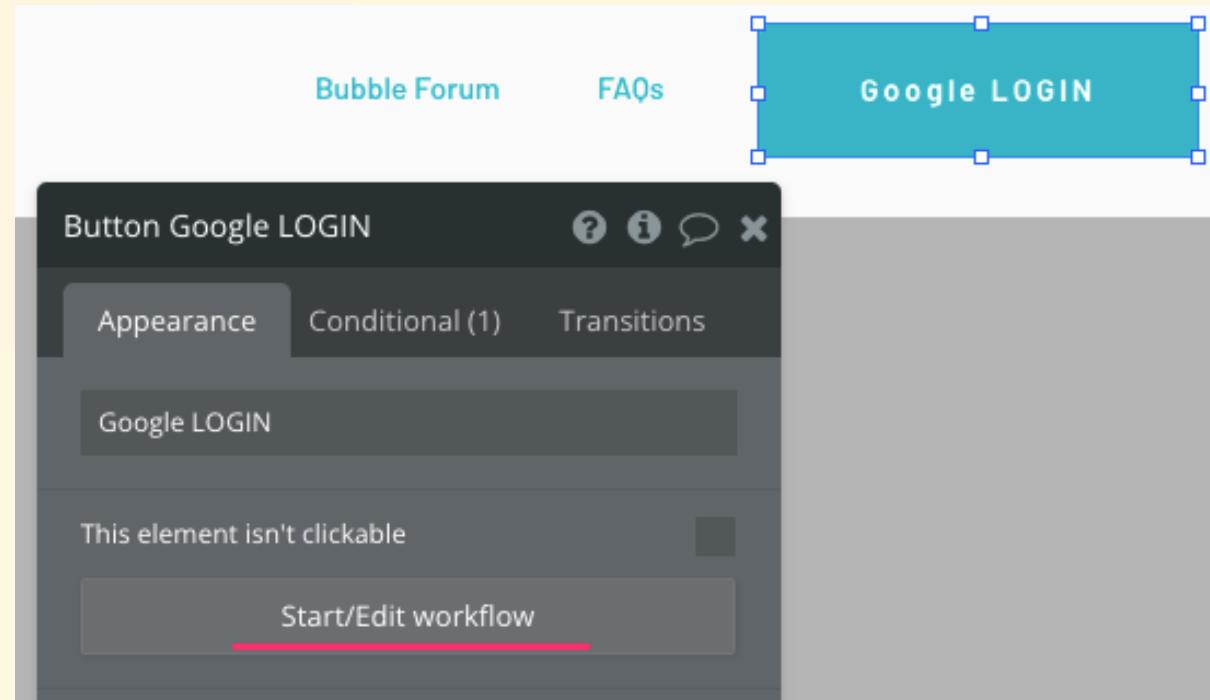
- Go back to the Google Plugin screen of Bubble, and enter the key you have just obtained into the respective field
 - Client ID → AppID/API Key
 - Client Secret → App Secret



- This completes the preconfiguration.
- Next, let's create the login function.

Replace the login mechanism with Google

- Open the header component and select the SIGN UP OR LOGIN button to go to the workflow screen.
 - And change the label of the button to "Google LOGIN".



- When you go to the Workflow screen, you will see that Button Google LOGIN has been configured to behave in two different ways

When
Button Google LOGIN
is clicked and Current
User is logged in

When
Button Google LOGIN
is clicked and Current
User is logged out

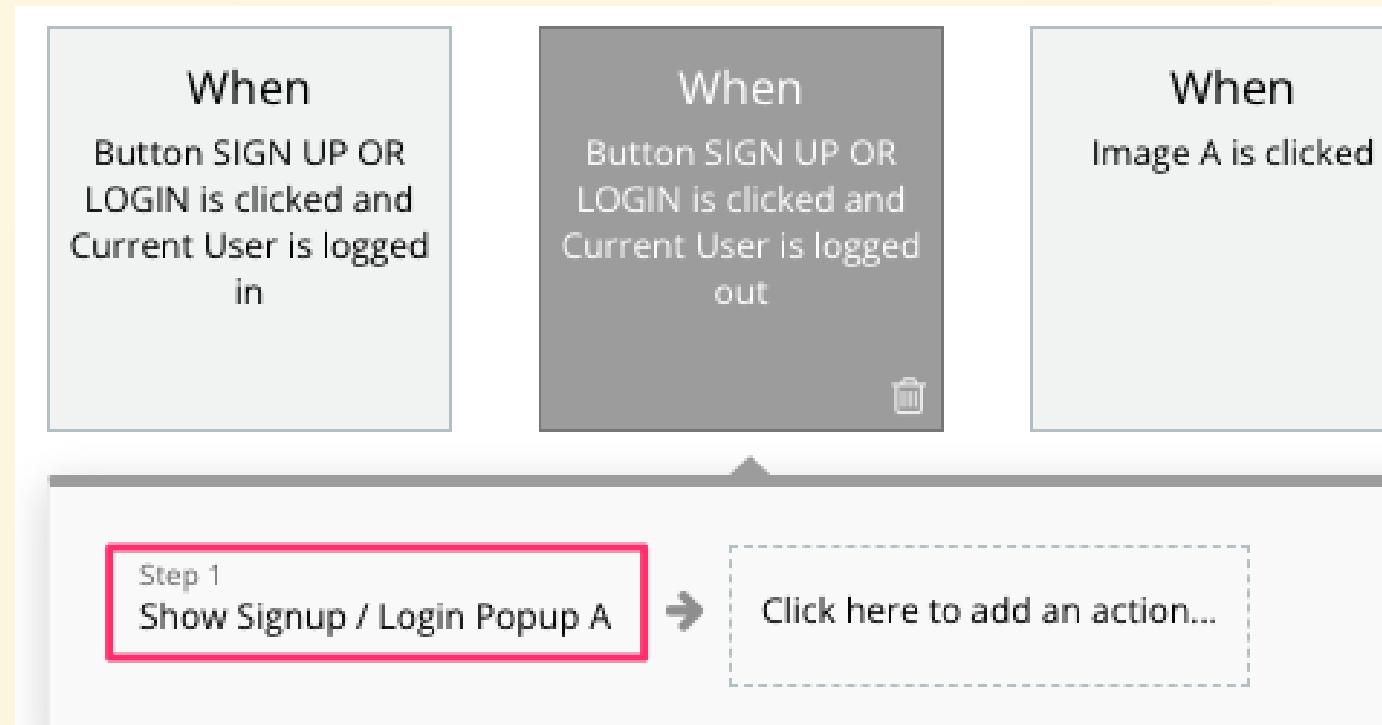
When
Image A is clicked

Click

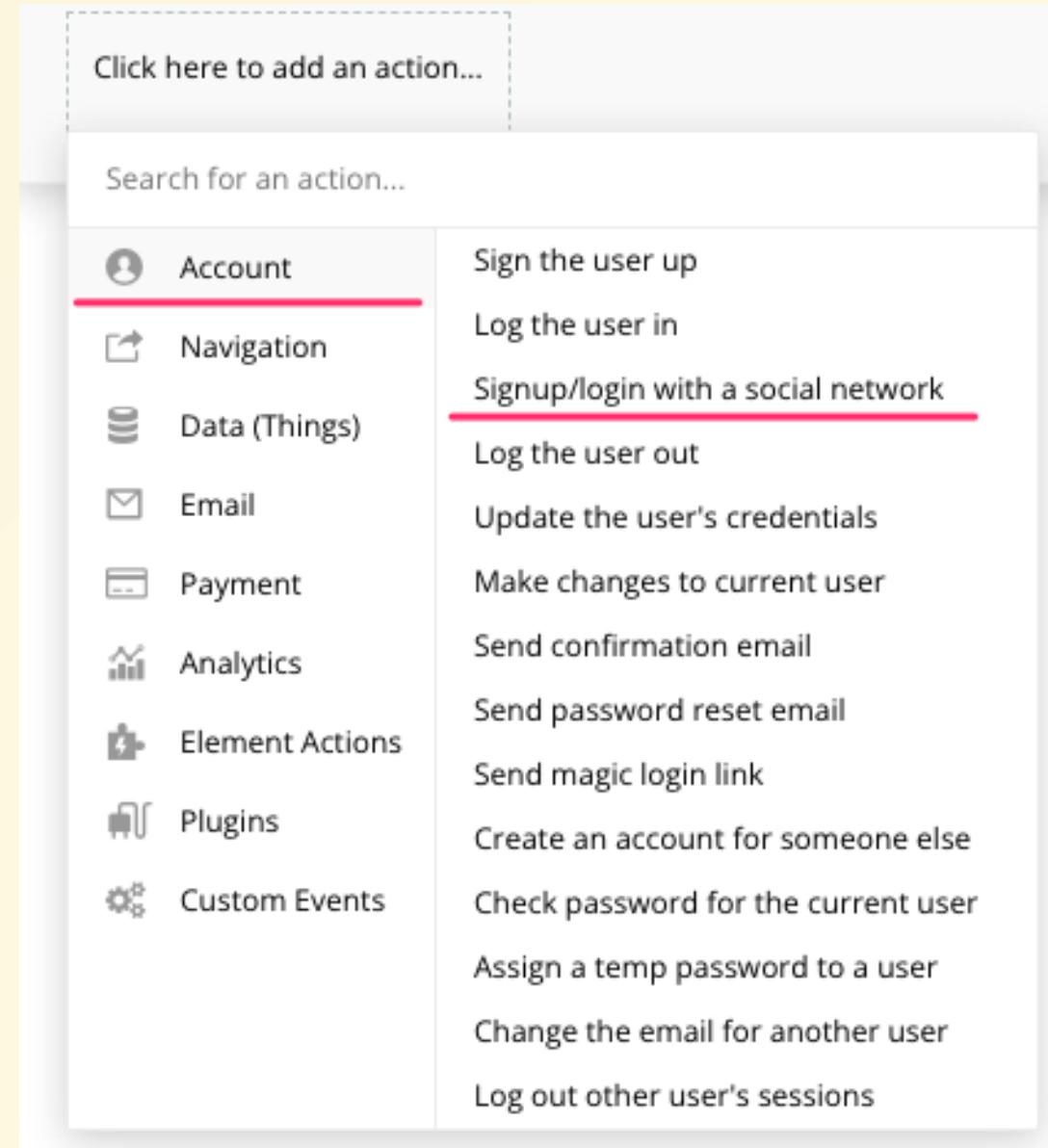
- When: Button Google LOGIN is clicked and Current User is logged in
- This is literally a workflow for when the current user is logged in and the "Google LOGIN" button is clicked.

- When: Button Google LOGIN is clicked and Current User is logged out
- This is also literally the workflow when the current user is logged out and the "Google LOGIN" button is clicked

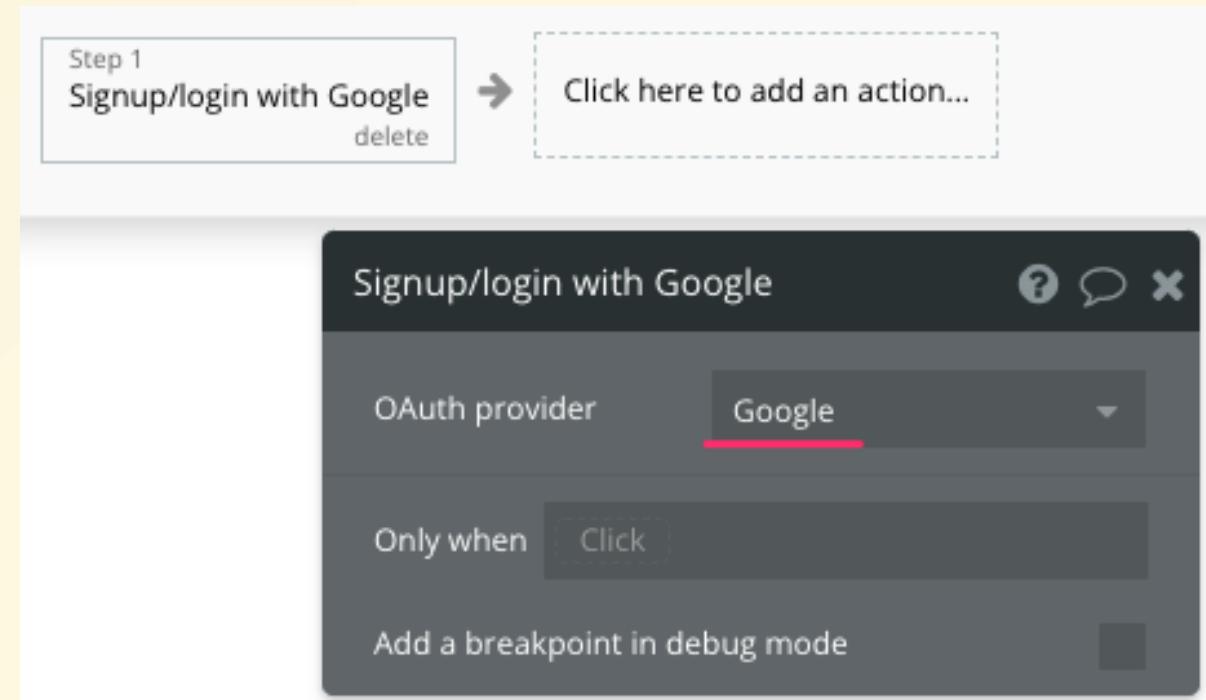
- First of all, we want to set up a workflow for login, so we'll set up a workflow for Current User is logged out.
- Since the "Show Sign up / Login Popup A" step is already defined, we will delete it.



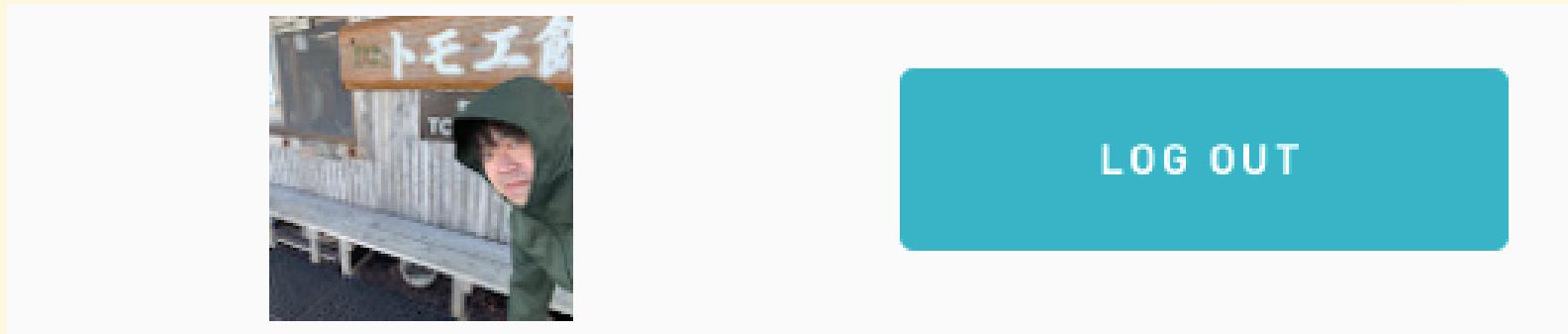
- Next, let's configure the Google Login workflow.
- Click here to add an action... Select
- From Account, select Signup/login with a social network



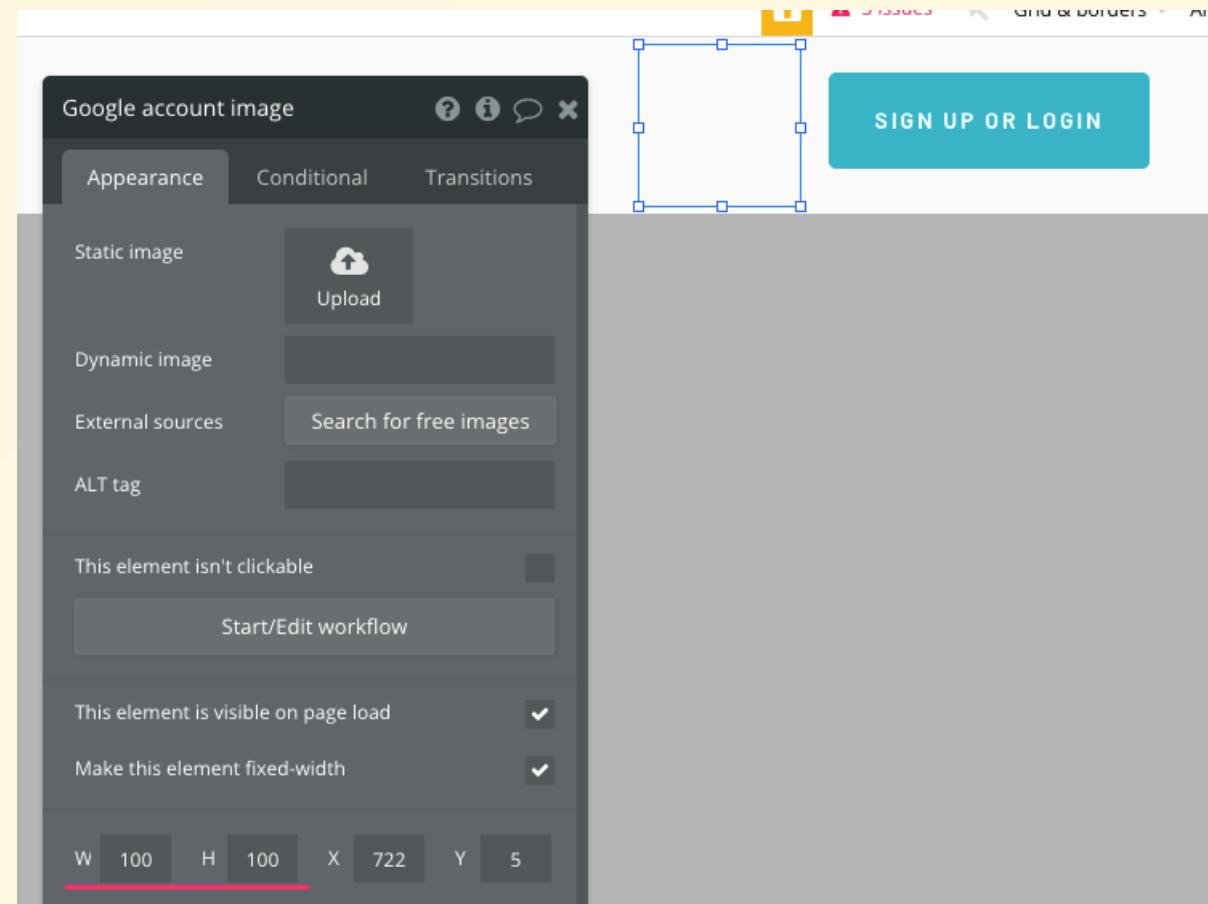
- The Signup/login with Google popup will appear
- Select Google as OAuth provider
- Now that we've configured the Google and Bubble plugins, we're ready for Google authentication 🙌!



- Now that we've successfully logged in, let's display the account name and image at the top of the header!

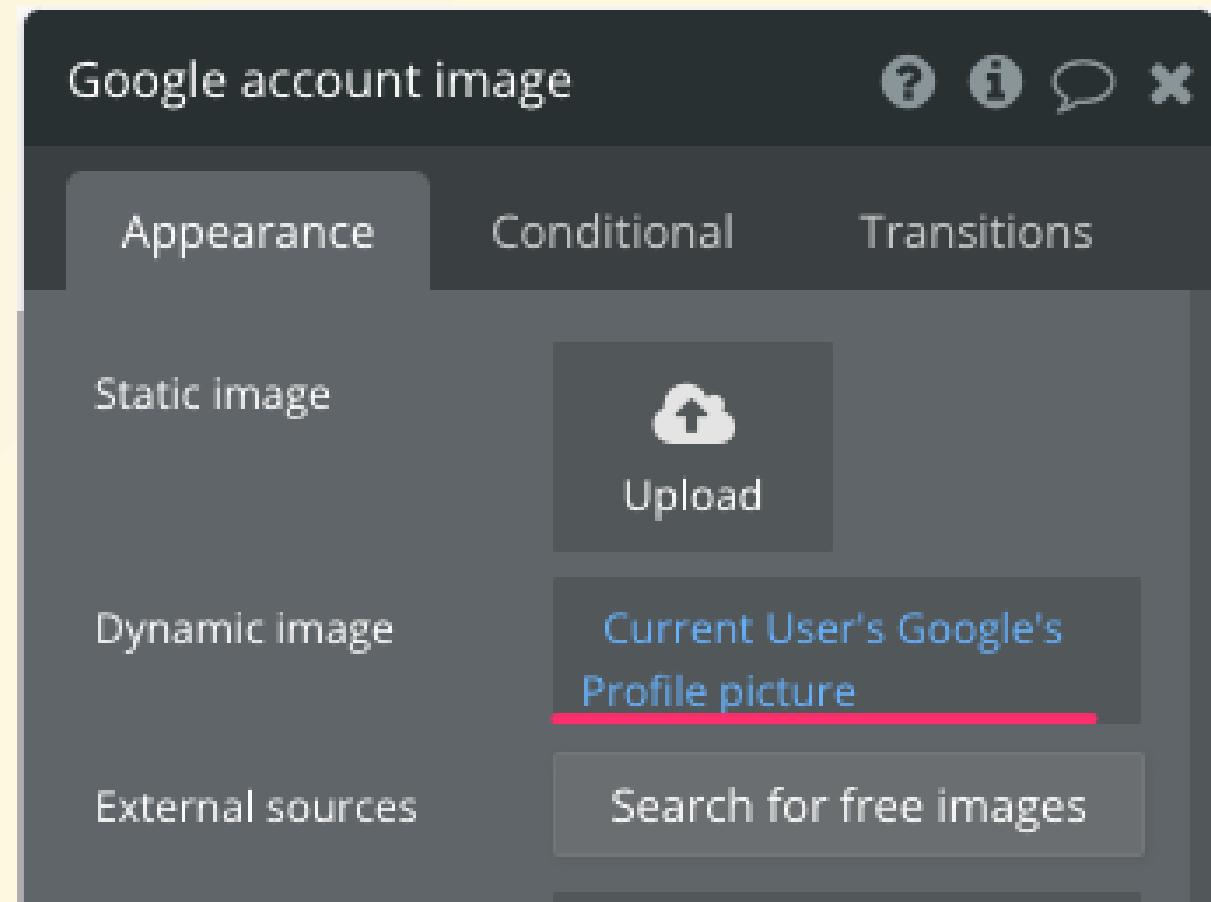


- Open the Header component and delete the element to the left of the LOGIN button
- Select Image from Visual elements and place it to the left of the LOGIN button.
 - Make it a square of 100 x 100 in height and width.



- Then set the referrer for the image
- If you're a Bubbler, you'll know how to set this up.
- Dynamic image for dynamic display settings
- We want to display the Google Account profile image of the currently logged in user

- It looks like this



- Now let's preview it!
 - If you have the same Google email address as the one you are going to use in the User section of the Data, please delete it.
- When you click on the "Login" button, you will be redirected to the Google login screen, and when you login, you will be redirected back to the Bubble side, right?
- When you log in, the email address of the Google account you signed in with will be registered in the User data of Bubble.

Google LOGIN



LOG OUT

Finally, let's incorporate a search function using the YouTube API

This application allows users to search for and view YOASOBI's content on YouTube.

The search results include:

- YOASOBI 「もう少しだけ」 Official Music Video
- YOASOBI 「怪物」 Official Music Video
- YOASOBI - Racing Into The Night
- YOASOBI 2nd EP『THE BOOK』
- YOASOBI 「大正浪漫」Official Music Video
- YOASOBI - 群青 / THE FIRST TIME
- YOASOBI - 夜に駆ける / THE FIRST TIME
- 広告なし NO ADS YOASOBI Best Songs
- YOASOBIのベストソング-YOASOBI

The bottom section shows a detailed view of the 'YOASOBIのベストソング-YOASOBI' playlist, listing tracks:

Rank	Title	Length
1	Yasashii susume / Comet the life savior	3:36
2	群青 / Gumbo / Ultramanne	7:41
3	ココロノココロ / Core	12:22
4	ハルジオン / Monstar	19:55
5	アメイジング・グレイズ / Amazing Grays	23:20
6	たぶん / Tatsumi / Probably	27:43
7	夜に駆ける / Tatsumi / Probably	32:18
8	あの夢をなぞって / Ann yume wo nazotte	
9	Ann yume wo nazotte / Training that dream	

- Currently, the YouTube API authentication key and search keywords are fixed.
- So, let's change them as follows
 - The authentication key is obtained from the logged in user information (add a new field)
 - Search keywords can be entered and searched from the screen.

I'll give you some hints on the next page, so let's try it first!

- First, add a field `key` to User and set it up beforehand.
 - Normally, it would be more natural to let the user input the key himself, but for the sake of time, we will skip the API key setting screen.
- If you want to set the parameters of the API dynamically in the process, first remove the fixed value and uncheck `Private` to specify the parameters when calling the API with a DataSource, etc.
- Place a new search element (text box and search button) on the index page, and when the search button is clicked, perform a search via the YouTube API and set the results to the Repeating Group.

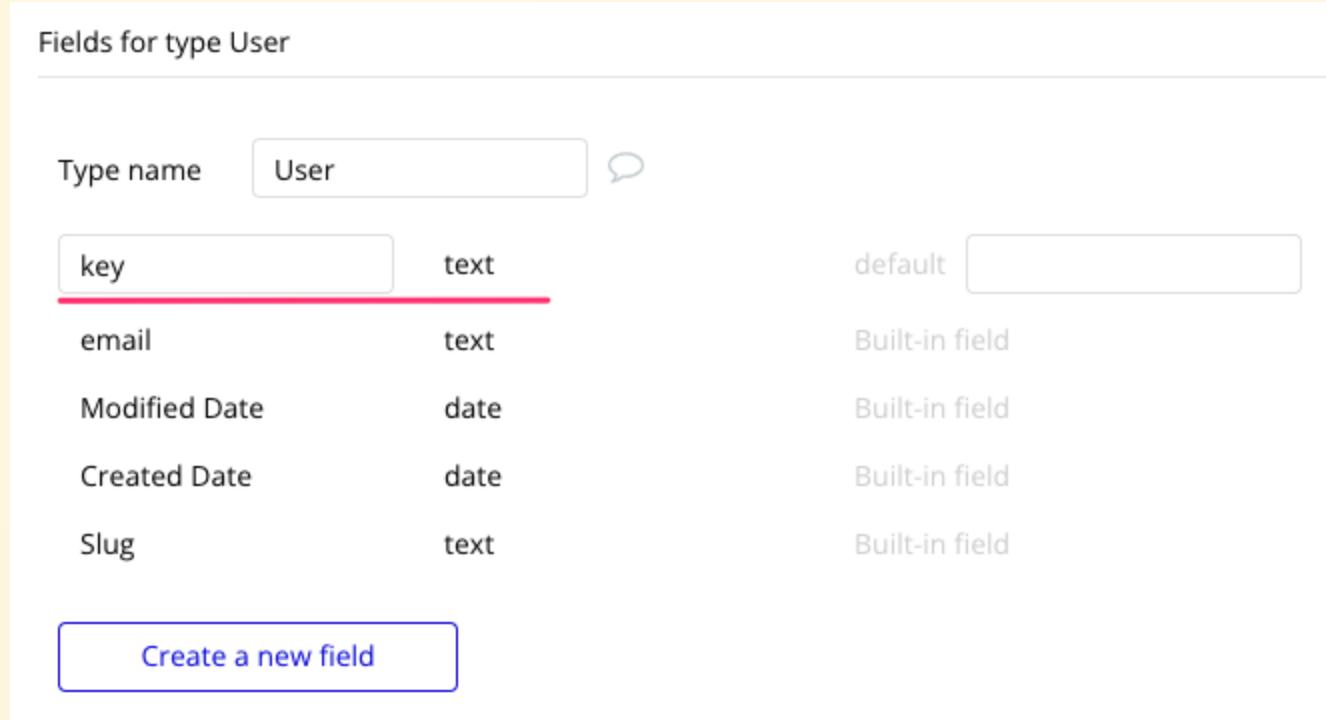


- First, we'll add a field called **key** for the User type

Fields for type User

Type name	User	
key	text	default
email	text	Built-in field
Modified Date	date	Built-in field
Created Date	date	Built-in field
Slug	text	Built-in field

[Create a new field](#)



- Then, set the authentication key (API key) to the registered User in advance.

The screenshot shows the 'App data' section of the Bubble platform. At the top, there are tabs for 'Data types', 'Privacy', 'App data' (which is selected), 'Option sets', and 'File manager'. Below the tabs, it says 'Database views' and 'Application data - All Users - Development version'. There are buttons for 'New view' and 'Primary fields'. A search bar contains the text 'Search' and indicates '1 entries (displaying 1)'. A search input field has the placeholder 'Search view names or data types...'. Below this is a table with one row. The table columns are: a checkbox, another checkbox, 'Email', and 'key'. The 'Email' column contains 'kyougoku2bubble@gmail.com' and the 'key' column contains 'AlzaSy...'. The entire 'key' value is highlighted with a red underline. At the bottom left, there is a 'All Users' button and a small icon. On the right side of the table, there is a 'Run as →' button followed by the email address 'kyougoku2bubble@gmail.com'.

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

- Next, the YouTube API side settings look like this

collapse

Name Use as Data type

GET (use [] for params)

Headers

Parameters

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

Include errors in response and allow workflow actions to continue

Capture response headers

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

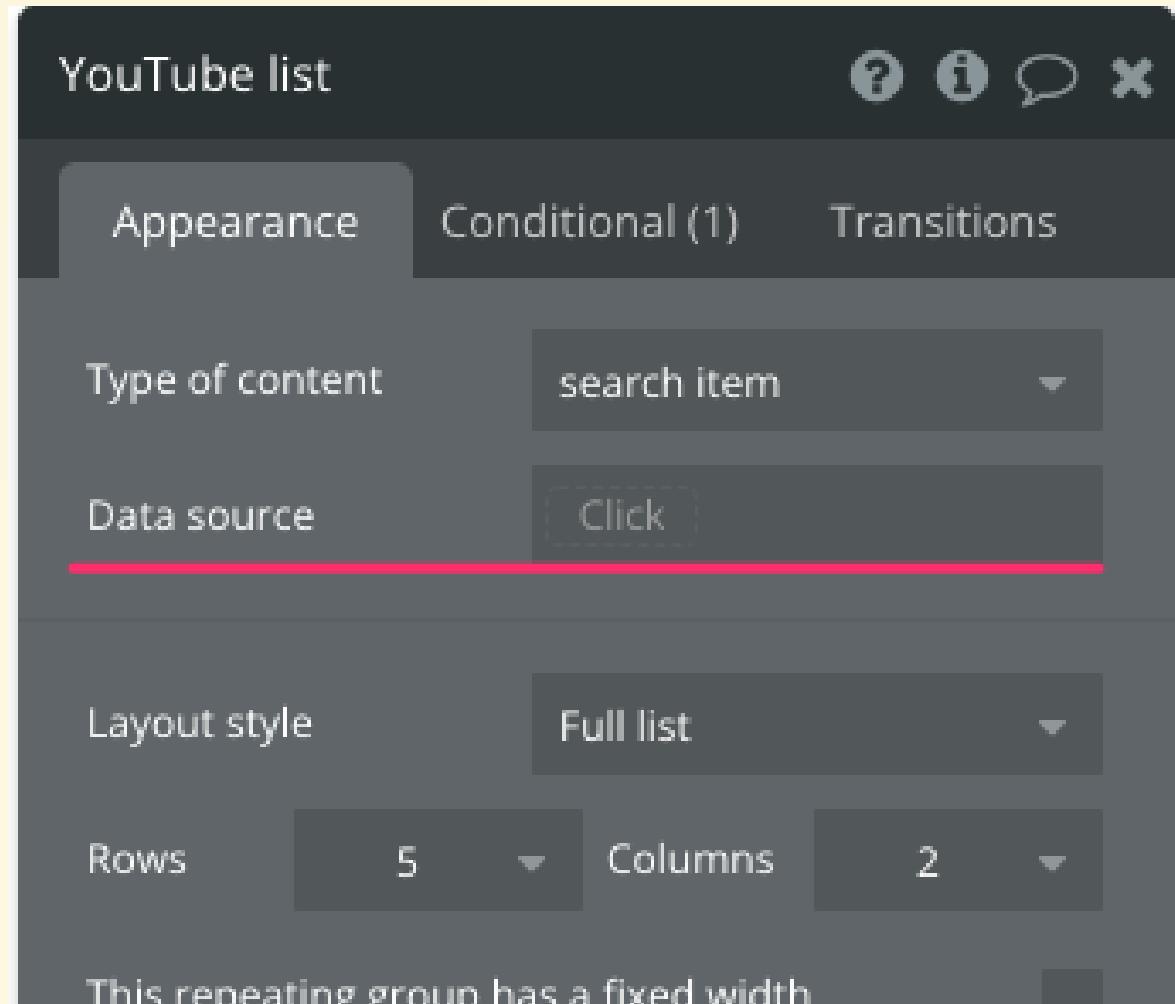
- As a point, I removed the Value value for the items I want to specify dynamically, and unchecked Private.
- This will allow you to specify the values of these parameters when using the API

The screenshot shows a configuration interface for a workflow action. The action is named "search" and is set to use JSON data type. It is a GET request to <https://www.googleapis.com/youtube/v3/search>. The parameters are defined as follows:

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"/>

Below the parameters, there are buttons for "Add header" and "Add parameter". There are also checkboxes for "Include errors in response and allow workflow actions to continue" and "Capture response headers". A message at the bottom states "This call has been modified since you last initialized; consider reinitializing." At the bottom right are buttons for "Reinitialize call" and "Manually enter API response".

- Next, empty the Data source of Repeating Group.
- This is to specify the data to be displayed here in the workflow of the new search function.

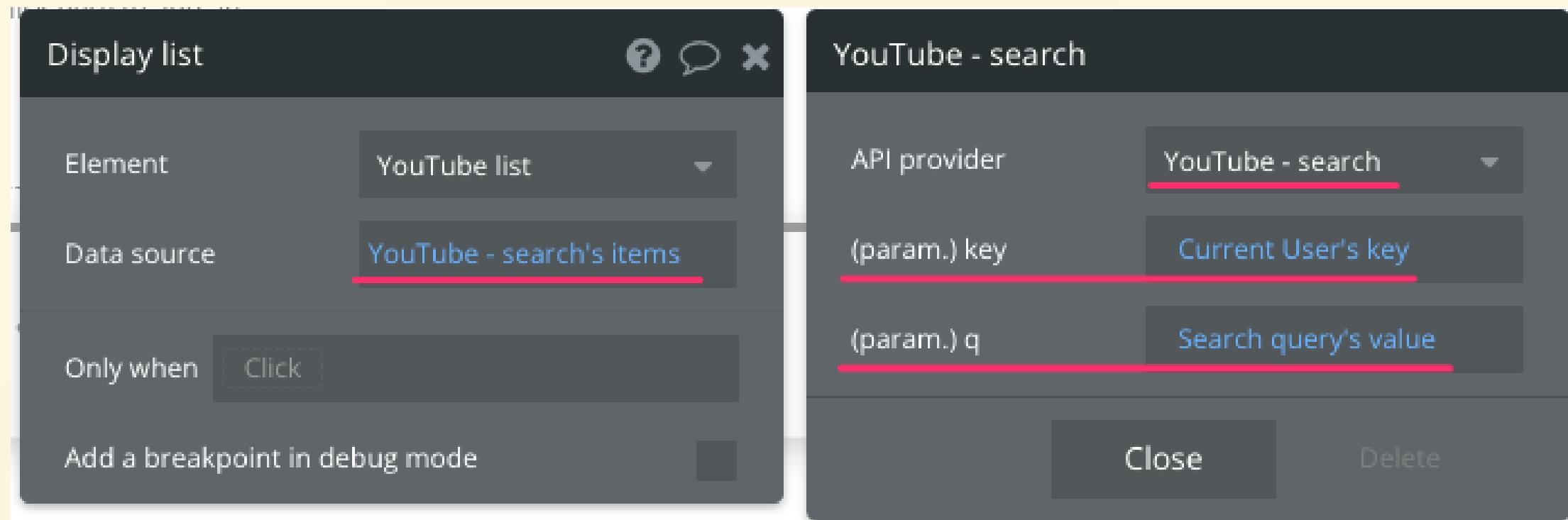


- The next step is to set up the workflow when a keyword search is performed.
- Since this is an action on an element, we will select "Display list" in "Element Actions".

Search for an action...		
 Account	All elements	
Show	Hide	
 Navigation	Toggle	Animate
Scroll to	Set state	
 Email	<i>Input</i>	
Set focus	Reset inputs	
 Payment	<i>Group</i>	
Display data	Reset data	
 Analytics	<i>Repeating Group</i>	
 Element Actions	<u>Display list</u>	Show previous
Plugins	Show next	Clear list
 Custom Events	Go to page	Scroll to entry

- For Data source, select Get data from an external API to display the data via the API.
- And if you select YouTube - search as the API provider, you should see two param's that were not there before.
- This is because we unchecked the "Private" checkbox for the parameters we want to specify dynamically in the YouTube API settings earlier.
- The value you specify here is the usual Dynamic data

- This is what it looks like when you set it



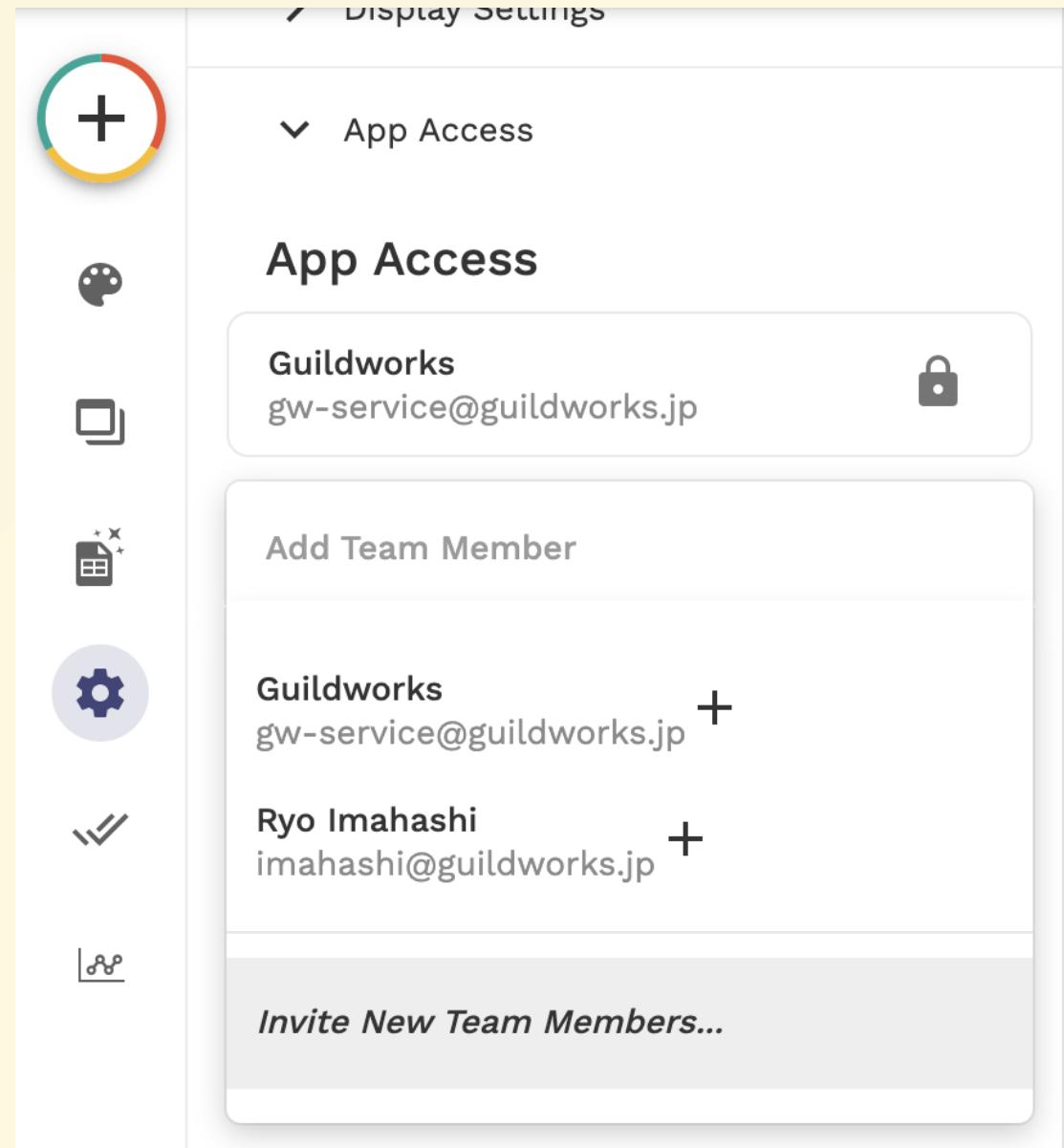
- How did you like it?
- I think the scope of what we can do with Bubble has expanded even more by using API integration!
- There may be teams that use external APIs at the camp, so let's use what we learned today and practice API integration!

Tips for Collaborating with No-Code Tools

Here are some tips to help you collaborate with your team using no-code tools.

(repost) How to invite team members in Adalo

- To co-edit one app with a team member, select Settings > AppAccess > Add Team Member > Invite New Team Member and enter the team member's email address.



How to invite team members in Bubble

- To co-edit a single app with a team member...

	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

¹ on an annual basis

Notes on simultaneous editing in Adalo

- Edits made by others will not be reflected on your screen in real time. You need to reload to reflect them.
- If you edit the same screen at the same time, the edits made earlier will be overwritten by the edits made later, and the screen will be in the state of the last person who made the edits.
 - The earlier edits will be undone, even if the edited components are different.
- For Actions, if you add them to the same screen at the same time, only the later one will be saved.
- Database, too, will be overwritten with the state of the last person who edited it if it is edited at the same time.

- If you edit on different screens at the same time, both will be saved and you will be fine. When editing, share with your team members which screen you are editing. It is also recommended to decide who is in charge of each screen.
- Before switching the screen to be edited, reload the screen to reflect the latest state (this will prevent you from reverting to the old state when someone else has edited the screen).
- To avoid inadvertently touching other people's screens, it is also recommended that you develop the screens apart from each other on Canvas.
- For the Database, decide on one person to be in charge and let that person update the database.

Points to note about simultaneous editing in Bubble

- Avoid simultaneous editing of the same element (it will be overwritten by later edits).
 - For this reason, it is recommended to assign a person in charge of each screen to avoid simultaneous editing of the same screen.

There are no other precautions to be taken.

- Edits will be reflected on other people's screens in real time.
 - Not only screen edits, but also database and Workflow edits are reflected in real time.
- If you divide the work by screen, you won't accidentally touch other screens (Canvas is divided), so you can develop comfortably.

Team Development Exercise

- With your DevelopmentPhase team, try to develop one new app.
 - It should be a different app from the one you plan to develop in DevelopmentPhase.
- You are free to use either Adalo or Bubble.
 - You can choose the one you plan to use in DevelopmentPhase, or both.
- Let's make sure everyone has a hand in this.

Example of how to divide the work

- Identify the screens and functions and design the database together at the beginning.
- Decide who will be in charge of each screen (registration screen, list screen, detail screen, update screen, etc.)
- There is also a way for multiple people to work on a single PC. In this case, the person who operates the PC and the person who gives instructions should be rotated periodically.



Presentation of the exercise results

Each team should present the application that they made in the exercise.

Summary

- This is the end of the Learning Phase lecture.
 - Have you mastered Adalo and Bubble?
 - Did you get an idea of how to do team development?
- In the next Development Phase, we will brush up and develop the ideas presented by each team.
 - In preparation for this, let's reconfirm our ideas and select tools (Adalo, Bubble, or other?) to be used for development. Bubble?

That's all!

Thanks for your time!