

Programming Boot Camp

Bubble Basic #2-2

Tokyo Institute of Technology 2022/11/26

Naotake KYOGOKU

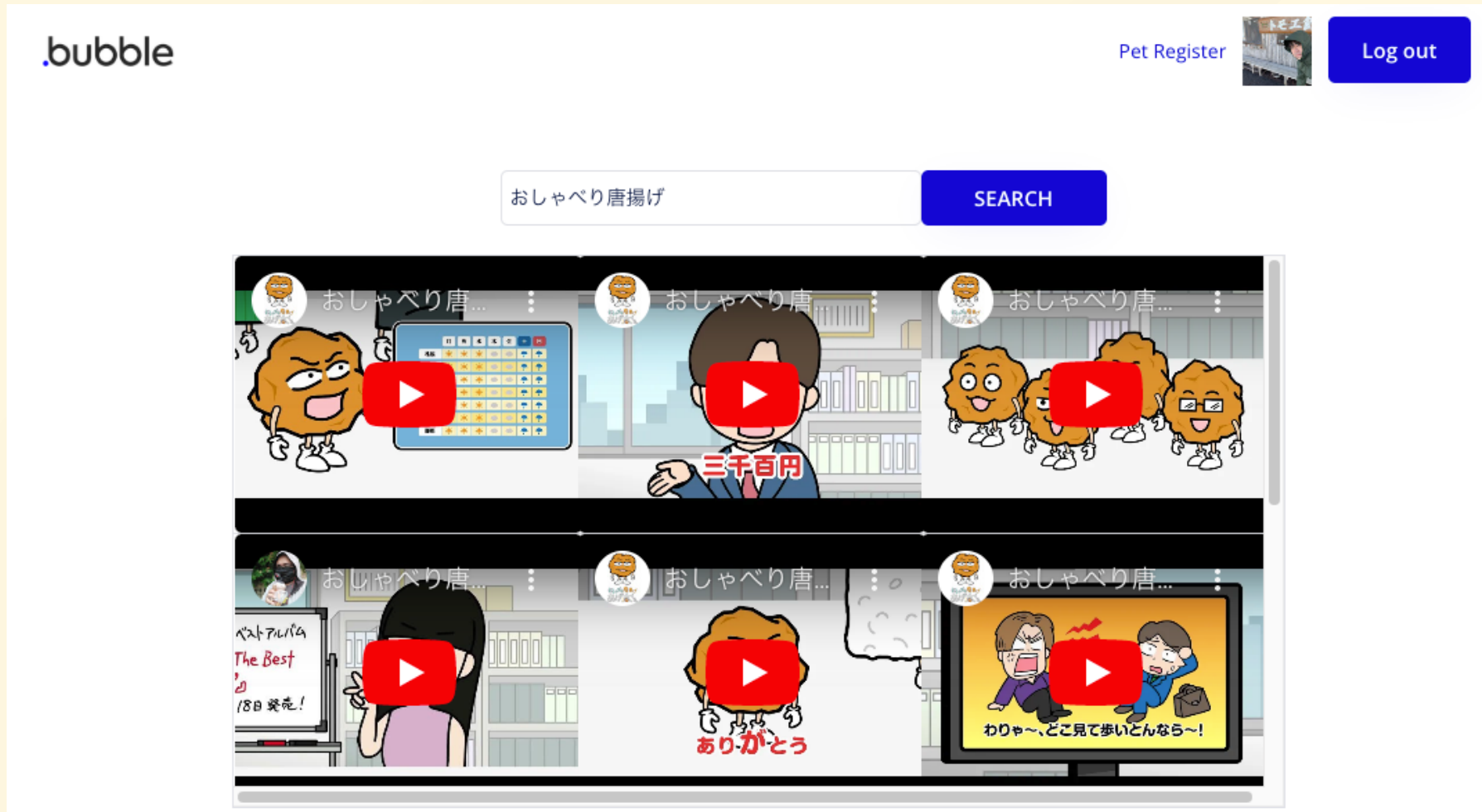
Table of Contents

- External Collaboration with Bubble
- Tips for Collaboration with Bubble
- Team Development Exercise
- Summary

Confirmation of what we will do this time

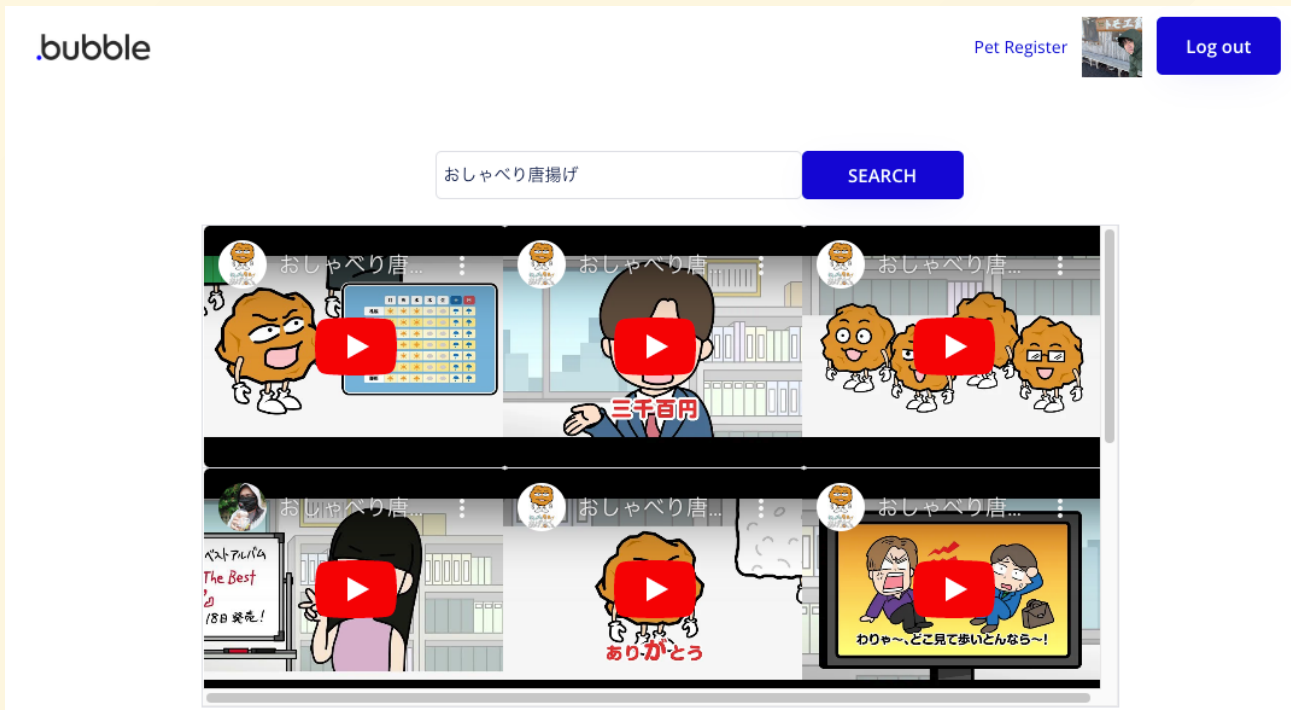
- First, you will learn how to integrate your Bubble application with external services.
- Next, we will show you some tips and tricks for working together on a single application.
- Finally, we will practice and present the application development in teams in the Development Phase.

External Collaboration with Bubble



The functionality of this screen includes the following

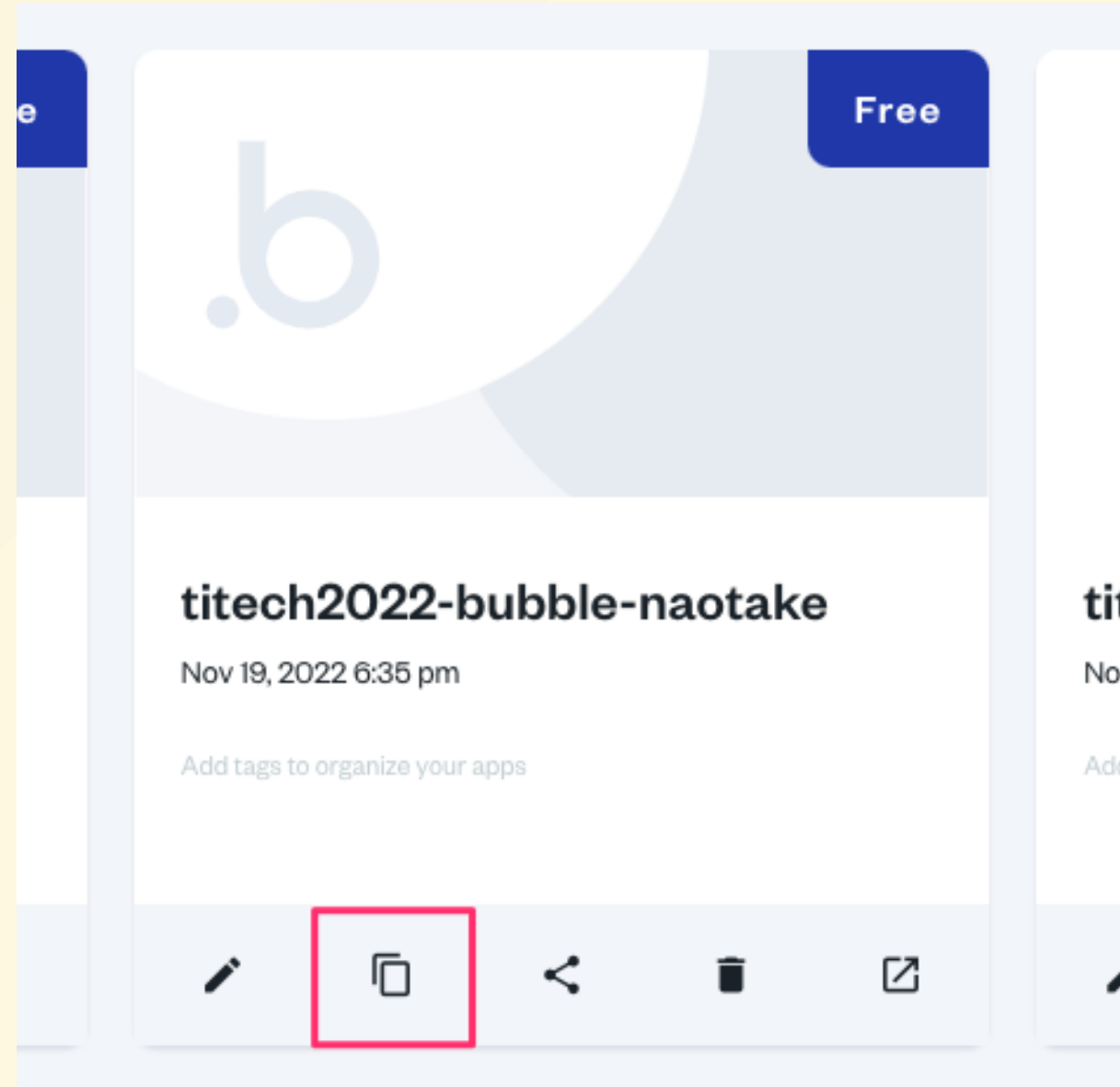
- YouTube video listings
- Social login using Google Account
- Dynamic search of YouTube video listings



First, we need to prepare a new application as a foundation.

- Bubble I would like to copy the application used in the first lecture and use it again.
- Go to <https://bubble.io/home?tab=apps>

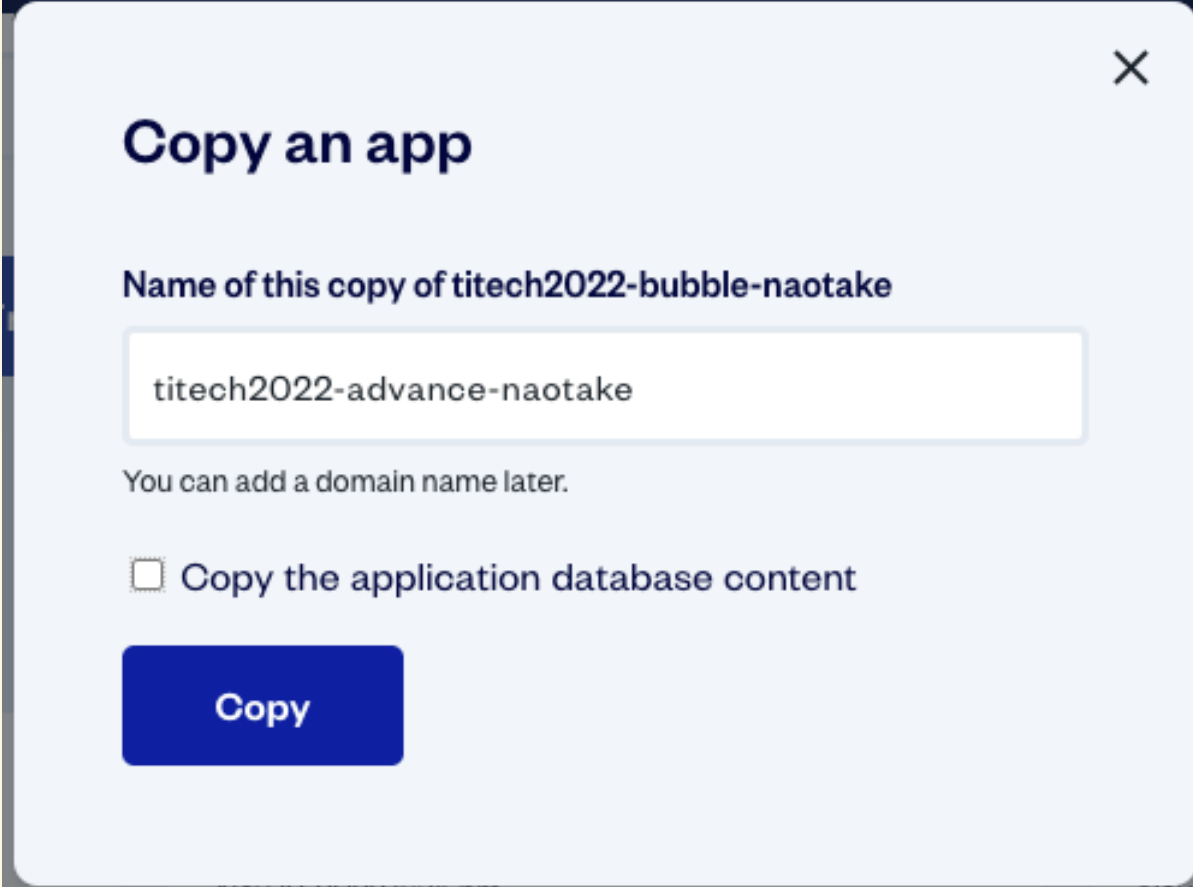
- A list of Bubble applications created under your account will be displayed.
- In the list, click COPY under the application you created in the first Bubble class.



- A dialog box for the name of the application after copying is displayed.
- Leave the

Copy the application database content

 checkbox OFF, since you will not be using the previous data.
- Click COPY



Copy an app

Name of this copy of titech2022-bubble-naotake

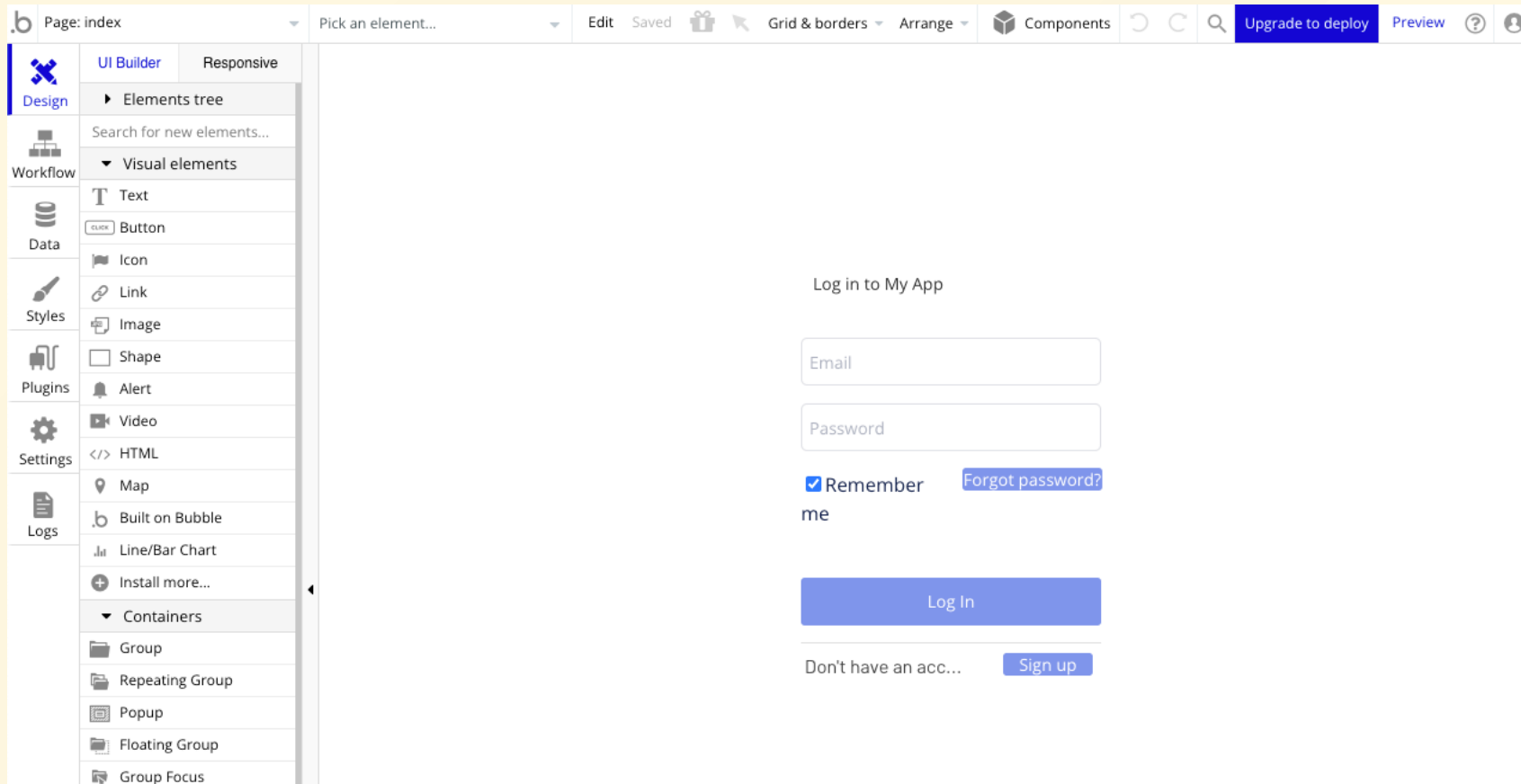
titech2022-advance-naotake

You can add a domain name later.

☐ Copy the application database content

Copy

- When the copying is finished, it is OK if the edit screen (index) of the copied application is displayed



Preparation for YouTube integration

- We will do the necessary preparations for displaying a list of YouTube videos.
- In this lecture, we will use various APIs provided by Google's cloud service "Google Cloud Platform".
 - Think of it as Google's version of Amazon's "Amazon Web Service" (AWS).

- Access the following URL
 - <https://console.developers.google.com/>
- Sign in with your Google Account
 - If you do not have a Google Account, please create one.



The image shows a Google Cloud Platform login page. At the top is the Google logo, followed by the text 'ログイン' (Login) and 'Google Cloud Platform に移動する' (Move to Google Cloud Platform). Below this is a text input field with the placeholder text 'メールアドレスまたは電話番号' (Email address or phone number). Under the input field is a link 'メールアドレスを忘れた場合' (If you forgot your email address). Further down is a paragraph of text: 'ご自分のパソコンでない場合は、ゲストモードを使用して非公開でログインしてください。' (If you are not on your own PC, please log in anonymously using Guest mode.) followed by a link '詳細' (Details). At the bottom left is a link 'アカウントを作成' (Create account), and at the bottom right is a blue button labeled '次へ' (Next).

Google

ログイン

Google Cloud Platform に移動する

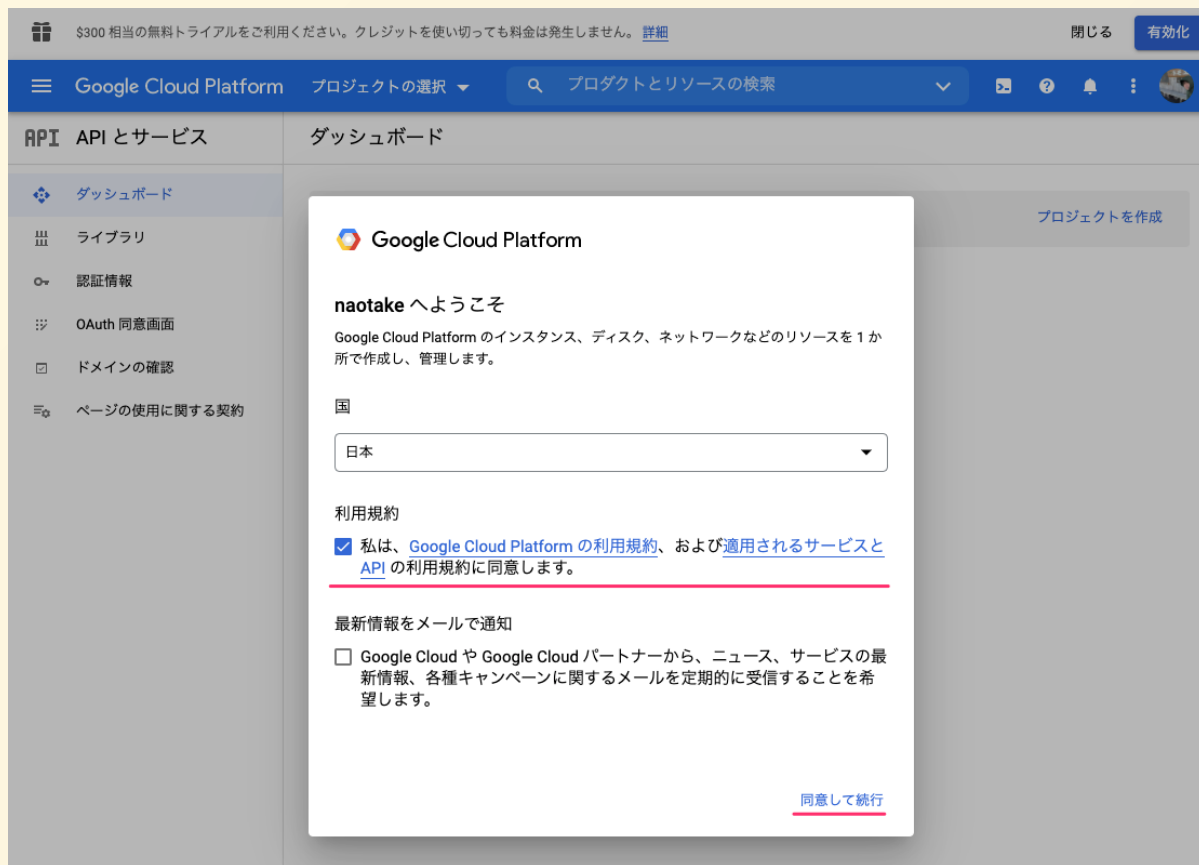
メールアドレスまたは電話番号

メールアドレスを忘れた場合

ご自分のパソコンでない場合は、ゲストモードを使用して非公開でログインしてください。 [詳細](#)

[アカウントを作成](#) [次へ](#)

- After logging in, when you access the Google Cloud Platform (GCP) screen for the first time, you will see a dialog like this, select your country, check the Terms of Service, and "Agree and Continue".




- Once you agree, you will see the API and Services dashboard, and click the Create Project link to create a project (box) for this lecture.



- When the Create New Project screen appears, specify an appropriate name for the project 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 *

bubble advance

?

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 *

 組織なし

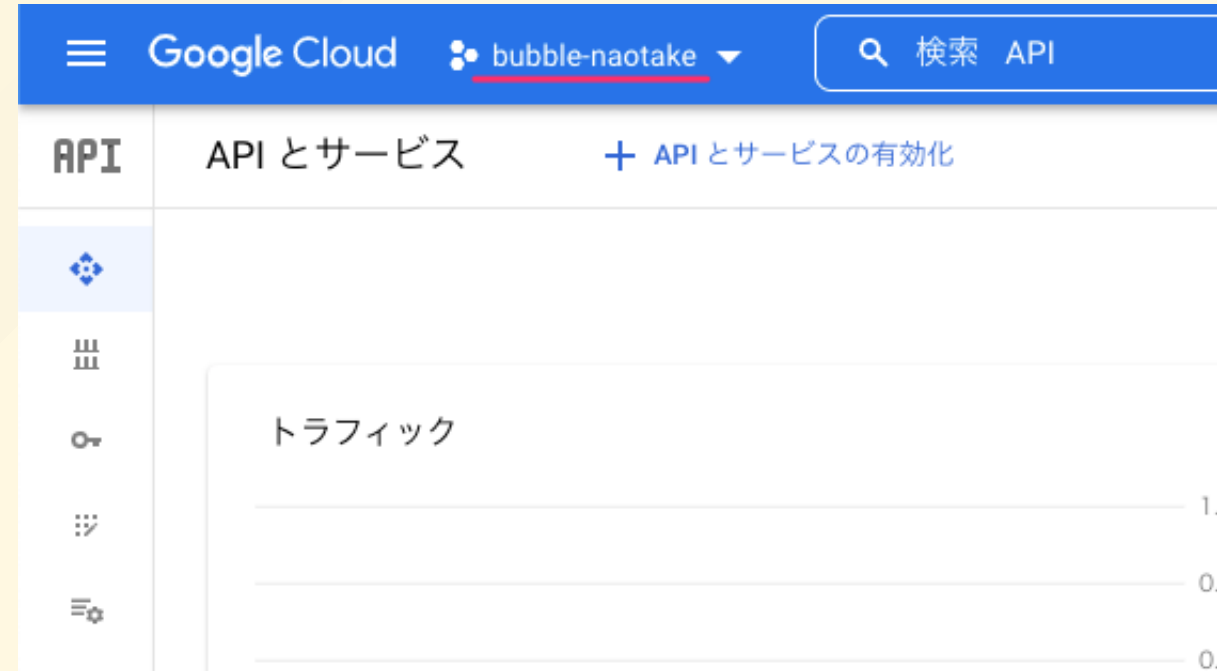
[参照](#)

親組織またはフォルダ

作成

キャンセル

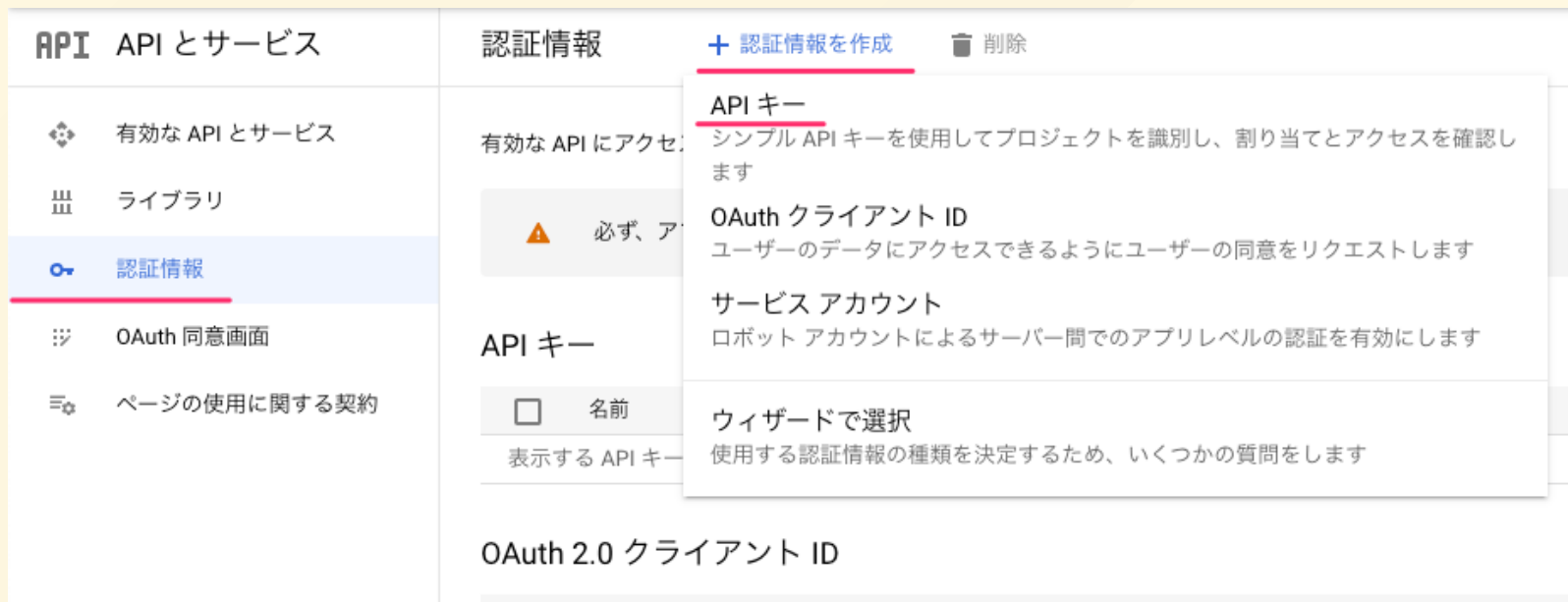
- When the creation is complete, the dashboard of the created project will be displayed.
- If you see the name of the project you created at the top of the screen to the left of Google Cloud, you are good to go.



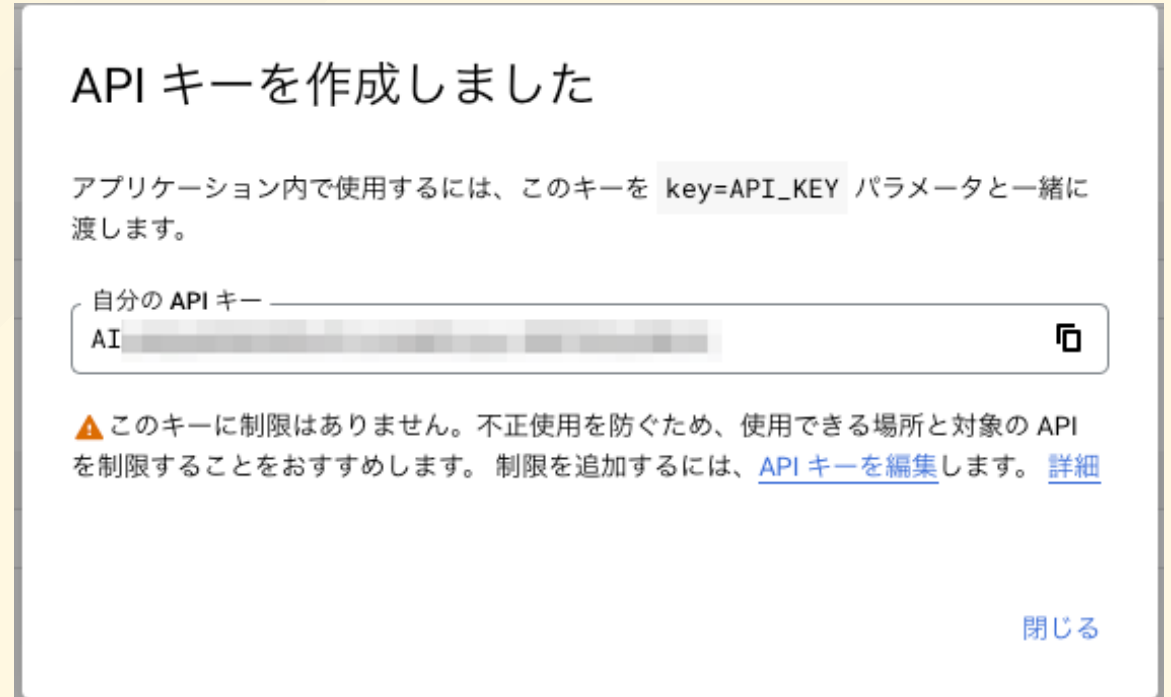
- Next, we will issue the authentication key needed to display the YouTube video list.
- Type "API" in the search box at the top of the screen to open the API and Services menu
- Select "APIs and Services" in the "Projects and Pages" block from the list of suggestions



- When the API and Services page opens, select "Credentials" from the left menu.
- The right panel will change to "Credentials", click "+ Create Credentials" at the top of the screen.
- Select "API Key" from the pop-up window.



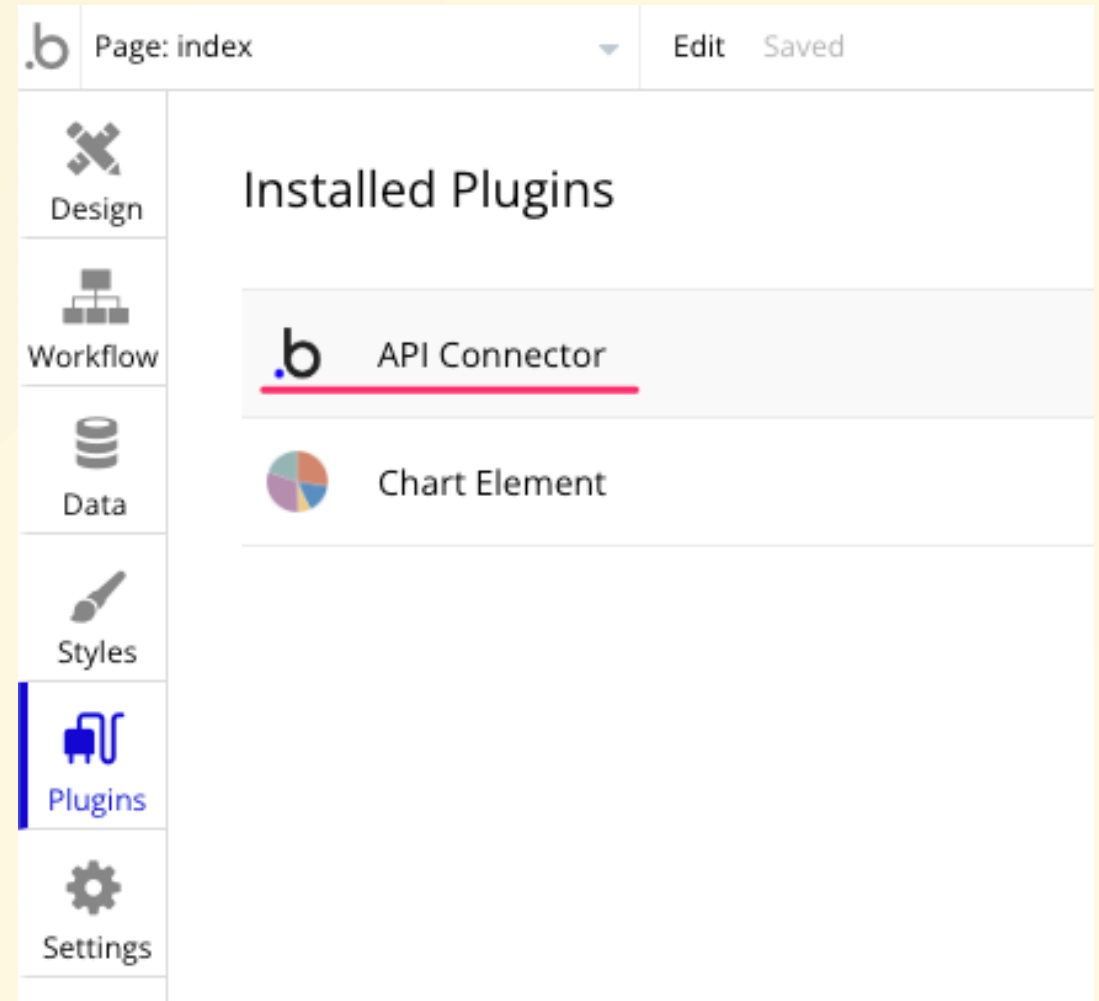
- You will then get a pop-up message saying that your API key creation is complete, so take note of the value.
- You will be asked to limit the keys, but we will delete these keys at the end of the lecture, so we will leave it as is.



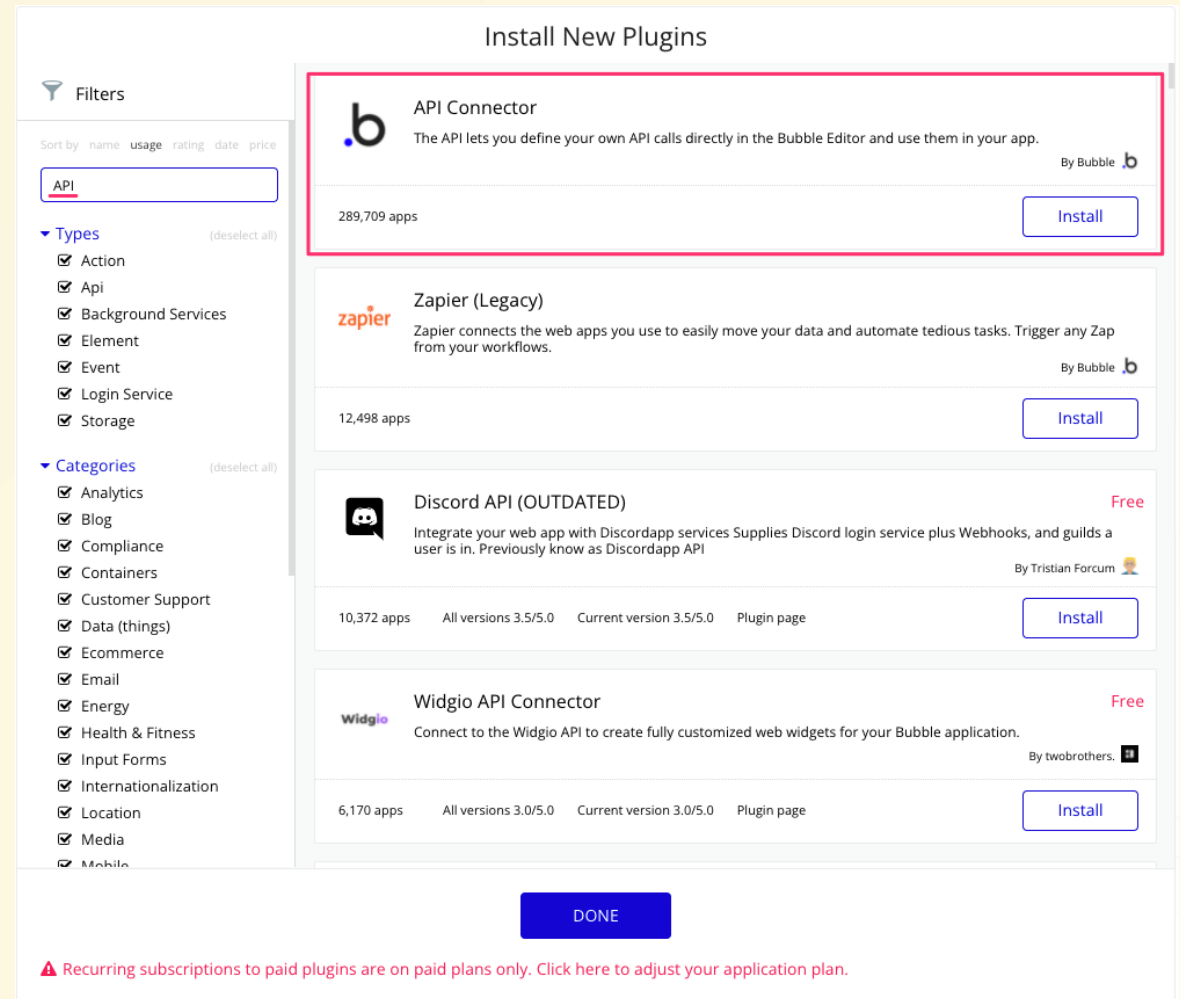
- Now that we have issued the key to retrieve the YouTube listings from Bubble, let's configure the Bubble side.
- First, let's prepare to get the YouTube listings from Bubble via API.

Preparing Bubble for API integration

- Select Plugins from the left menu and check that the **API Connector** is already present in the Installed Plugins.

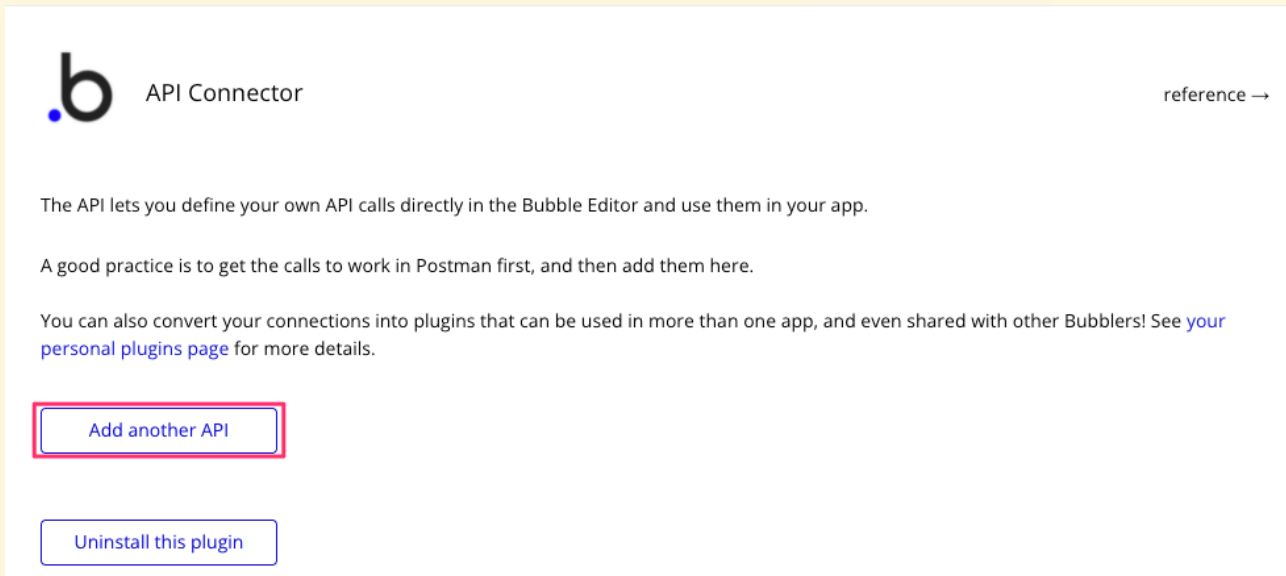


- If it is not included, click the Add plugins button.
- Search for "API" and install the plugin **API Connector**.



- This plugin is designed to connect Bubble apps to other APIs that are publicly available, and to information on the other side of those APIs.
- Some Bubble plugins are specialized for specific APIs and are simpler to configure than the API Connector.
- Let's use the YouTube API via the API Connector plugin in its most basic form to display a list of videos.

- Now let's configure the YouTube API to use it
- In the Plugins tab, select "API Connector" from the Installed Plugins list.
- In the right panel, you will see a summary of this plugin, and below that, click "Add another API" to start setting up your new API connection



- "Add another API", you should see API configuration components like this
- Here is a brief description of each item and its settings

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

1. API Name: Group name for this API linkage

- In this case, it is YouTube.

The screenshot shows an API configuration interface with the following elements:

- Annotation 1:** Points to the "API Name" field, which contains the text "YouTube".
- Annotation 2:** Points to the "Authentication" dropdown menu, which is currently set to "None or self-handled".
- Annotation 3:** Points to the "Shared headers for all calls" section, which includes a button labeled "Add a shared header".
- Annotation 4:** Points to the "Shared parameters for all calls" section, which includes a button labeled "Add a shared parameter".

Below these sections, there is a box for "API Call" configuration:

- The "Name" field contains the text "API Call".
- There is an "expand" button in the top right corner of this box.

At the bottom of the interface:

- There is a link "Import another call from cURL".
- Below it is a button labeled "Add another call".
- In the bottom right corner, there is a link labeled "expand all calls".

2. Authentication: Common authentication method for APIs under this group.

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

The screenshot displays a configuration interface for an API group. At the top, the 'API Name' is set to 'YouTube' (marked with a red circled 1), and the 'Authentication' is set to 'None or self-handled' (marked with a red circled 2). A 'collapse' link is in the top right. Below this, the 'Shared headers for all calls' section (marked with a red circled 3) contains an 'Add a shared header' button. The 'Shared parameters for all calls' section (marked with a red circled 4) contains an 'Add a shared parameter' button. A section for individual API calls is shown below, with a name 'API Call' (marked with a red circled 4) and an 'expand' link. At the bottom, there is a link to 'Import another call from cURL' and a button labeled 'Add another call'. An 'expand all calls' link is located in the bottom right corner.

API Name: YouTube

Authentication: None or self-handled

Shared headers for all calls

Add a shared header

Shared parameters for all calls

Add a shared parameter

Name: API Call

Import another call from cURL

Add another call

expand all calls

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

- Not specified at this time

The screenshot displays a configuration panel for an API group named 'YouTube'. The panel includes sections for 'Shared headers for all calls' and 'Shared parameters for all calls', each with an 'Add' button. Below these is a section for 'API Call' with a name field. At the bottom, there is a button to 'Add another call' from a cURL. The interface also features 'collapse' and 'expand' controls for the main panel and 'expand all calls' at the bottom right.

① API Name: YouTube

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

collapse

expand

expand all calls

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

- Not specified at this time

①

API Name

YouTube

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

- Next, let's set up the API for this group.
- Click on the "expand" link to the right of the "API Call" block on the previous screen.

collapse

API Name

YouTube

Authentication

None or self-handled

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

- Then you will see the specific API configuration items, which are briefly explained below

collapse

Name

API Call

Use as

Data

Data type

JSON

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

1. Name: A specific API name.

- In this case, enter "search".

The screenshot shows an API configuration interface with the following elements:

- Name:** A text input field containing "search", marked with a red circle and the number 1.
- Use as:** A dropdown menu set to "Data".
- Data type:** A dropdown menu set to "JSON".
- Method:** A dropdown menu set to "GET", marked with a red circle and the number 2.
- URL:** A text input field containing "https://www.googleapis.com/youtube/v3/search".
- Headers:** A section labeled "Headers" with an "Add header" button, marked with a red circle and the number 3.
- Parameters:** A section labeled "Parameters" with two parameter rows, marked with a red circle and the number 4:
 - Row 1: Key "key", Value "AlzaSyAA3b34Zbc9rxnw.", Private ☒, Allow blank ☐, Optional ☐.
 - Row 2: Key "q", Value "おしゃべり唐揚げ", Private ☒, Allow blank ☐, Optional ☐.
- Buttons:** "Add parameter" and "Initialize call".
- Options:** "Include errors in response and allow workflow actions to continue" and "Capture response headers", both with unchecked checkboxes.
- Warning:** A red warning icon and text: "You need to initialize this call before it will work."
- Footer:** A link "Manually enter API response".

2. Path: URL of the specific API

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

The screenshot shows an API configuration interface with the following elements:

- Name:** `search` (annotated with ①)
- Use as:** `Data` (dropdown)
- Data type:** `JSON` (dropdown)
- Method:** `GET` (dropdown, annotated with ②)
- URL:** `https://www.googleapis.com/youtube/v3/search` (annotated with ②)
- Headers:** (annotated with ③)
 - `Add header` button
- Parameters:** (annotated with ④)
 - Key: `key`, Value: `AlzaSyAA3b34Zbc9rxnw`, Private: ☒, Allow blank: ☐, Optional: ☐
 - Key: `q`, Value: `おしゃべり唐揚げ`, Private: ☒, Allow blank: ☐, Optional: ☐
 - `Add parameter` button
- Include errors in response and allow workflow actions to continue:** ☐
- Capture response headers:** ☐
- Warning:** `⚠ You need to initialize this call before it will work.`
- Buttons:** `Initialize call` (blue outline), `Manually enter API response`
- UI Elements:** `collapse` button (top right), trash icons for headers and parameters.

3. Headers: Header information specific to this API

- Not specified at this time

The screenshot shows an API configuration interface with the following elements:

- 1** Name: search
- 2** Use as: Data
- Data type: JSON
- Method: GET
- URL: https://www.googleapis.com/youtube/v3/search (use [] for params)
- 3** Headers: Add header
- 4** Parameters:
 - Key: key, Value: AlzaSyAA3b34Zbc9rxnw, Private: ☒, Allow blank: ☐, Optional: ☐
 - Key: q, Value: おしゃべり唐揚げ, Private: ☒, Allow blank: ☐, Optional: ☐
- Add parameter
- Include errors in response and allow workflow actions to continue: ☐
- Capture response headers: ☐
- Warning:** You need to initialize this call before it will work.
- Initialize call
- Manually enter API response

4. Parameters: Parameter information specific to this API

- In this case, the following two parameters are specified: 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.
 2. Key: `q`, Value: Keyword for searching behavior on YouTube (feel free to specify)

- When everything is set up, it looks like this

collapse

①

Name

search

Use as

Data

Data type

JSON

GET

②

https://www.googleapis.com/youtube/v3/search

(use [] for params)

③

Headers

Add header

④

Parameters

Key

key

Value

AlzaSyAA3b34Zbc9rxnw.

Private

☒

Allow blank

☐

Optional

☐

Key

q

Value

おしゃべり唐揚げ

Private

☒

Allow blank

☐

Optional

☐

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

- When you are done, check to make sure your settings are correct
- Click the "Initialize call" button at the bottom

- You probably got back an error message with Status code 403
 - The content of the message is as written, but in summary it looks like this
- “ YouTube Data API v3 is disabled in your project. Please visit this URL to 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 298727815636 before or it is disabled. Enable it by visiting https://console.developers.google.com/apis/api/youtube.googleapis.com/overview?project=298727815636 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 298727815636 before or it is disabled. Enable it by visiting https://console.developers.google.com/apis/api/youtube.googleapis.com/overview?project=298727815636 then retry. If you enabled this API recently, wait a few minutes for the action to propagate to our systems and retry.",
        "domain": "usageLimits",
        "reason": "accessNotConfigured",
        "extendedHelp": "https://console.developers.google.com"
      }
    ],
    "status": "PERMISSION_DENIED",
    "details": [
      {
        "@type": "type.googleapis.com/google.rpc.Help",
        "links": [
```

- It seems that the YouTube Data API we used this time is not enabled on each person's Google account.
- This is because Google has turned off the function that not everyone will use from the beginning, and only those who need it can turn it on by themselves.
- So, since we want to use the YouTube Data API, let's access the URL in the message to enable it.
<https://console.developers.google.com/apis/api/youtube.googleapis.com/overview?project={各人のプロジェクトID}>

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



YouTube Data API v3

[Google](#)

The YouTube Data API v3 is an API that provides access to YouTube data, such as videos, playlists,...

有効にする

この API を試す [🔗](#)

- After a few moments, you should be taken to a screen like this
- If the red line at the top of the screen reads "Disable API," the activation was successful.

The screenshot shows the Google Cloud Platform console interface for the YouTube Data API v3. The top navigation bar includes a back arrow, the title "API / サービスの詳細", and a red link "API を無効にする". The left sidebar lists navigation options: "有効な API とサービス", "ライブラリ", "認証情報", "OAuth 同意画面", and "ページの使用に関する契約". The main content area displays the API details for "YouTube Data API v3", including its description, owner (Google), and a table of service name, type, and status. The status is "有効" (Enabled). Below the table, there are tabs for "指標" (Metrics), "割り当て" (Quota), and "認証情報" (Credentials). The "指標" tab is selected, showing a graph selection dropdown and a "30 日" (30 days) time range selector.

サービス名	タイプ	ステータス
youtube.googleapis.com	公開 API	有効

- 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 saying "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 <small>youtube#searchListResponse</small>	text
etag <small>bMKu0aIXS6rekNwT5RVHZLOYR-o</small>	text
nextPageToken <small>CAUQAA</small>	text
regionCode <small>US</small>	text
pageInfo totalResults <small>1000000</small>	number
pageInfo resultsPerPage <small>5</small>	number
items (list) <small>(see fields below)</small>	search item
kind <small>youtube#searchResult</small>	text
etag <small>WahficVNgYipX-I3rjRiPUrHi10</small>	text
id kind <small>youtube#channel</small>	text
id channelId <small>UCcRigzl_jBAZh_UySjv8xuw</small>	text
id videoid <small>LZSaFPQI4_g</small>	text

SAVE

Cancel

- This shows the result (response) of executing the "<https://www.googleapis.com/youtube/v3/search>" API that you have set up this time

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 <small>youtube#searchListResponse</small>	text
etag <small>mjrGGMgtIMWG6aVWolrNtjmgTg8</small>	text
nextPageToken <small>CAUQAA</small>	text
regionCode <small>US</small>	text
pageInfo totalResults <small>946629</small>	number
pageInfo resultsPerPage <small>5</small>	number
items (list) <small>(see fields below)</small>	search item
kind <small>youtube#searchResult</small>	text
etag <small>WahlicVNgYipX-I3rjRIPUrHi10</small>	text
id kind <small>youtube#channel</small>	text
id channelId <small>UCcRigziJBaZh_UySjv8xuw</small>	text
id videoId <small>SO94aNshTSU</small>	text

SAVE

Cancel

I won't go into details, but the three main points are as follows.

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

- As you may know if you have seen YouTube before, this Videoid is the information that uniquely identifies each video, and this Id is also required to display the video on the screen.
- Other items that do not refer to any particular field are confusing, so select `Ignore field` from the pull-down menu.
- By doing so, when you handle it as Dynamic data, it will not be displayed in the choices, and you will not have to worry when you select a setting item.

- 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 <small>youtube#searchListResponse</small>	Ignore field ▼
etag <small>mjrGGMgtlMWG6aVWolrNtjmgTG8</small>	Ignore field ▼
nextPageToken <small>CAUQAA</small>	Ignore field ▼
regionCode <small>US</small>	Ignore field ▼
pageInfo totalResults <small>946629</small>	Ignore field ▼
pageInfo resultsPerPage <small>5</small>	Ignore field ▼
items (list) <small>(see fields below)</small>	search item ▼
kind <small>youtube#searchResult</small>	Ignore field ▼
etag <small>WahfivNgYipX-I3rjRIPUrHi10</small>	Ignore field ▼
id kind <small>youtube#channel</small>	Ignore field ▼
id channelId <small>UCcRigzi_jBAZh_UySjv8xuw</small>	Ignore field ▼
id videoid <small>SO94aNshTSU</small>	text ▼

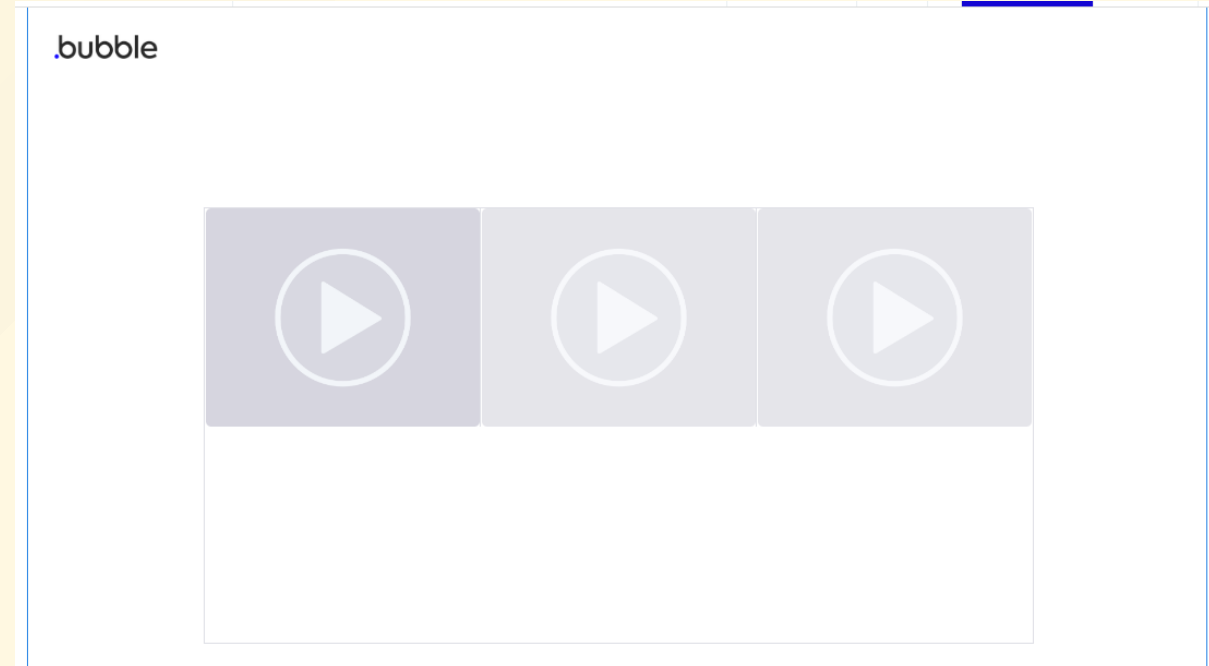
SAVE

Cancel

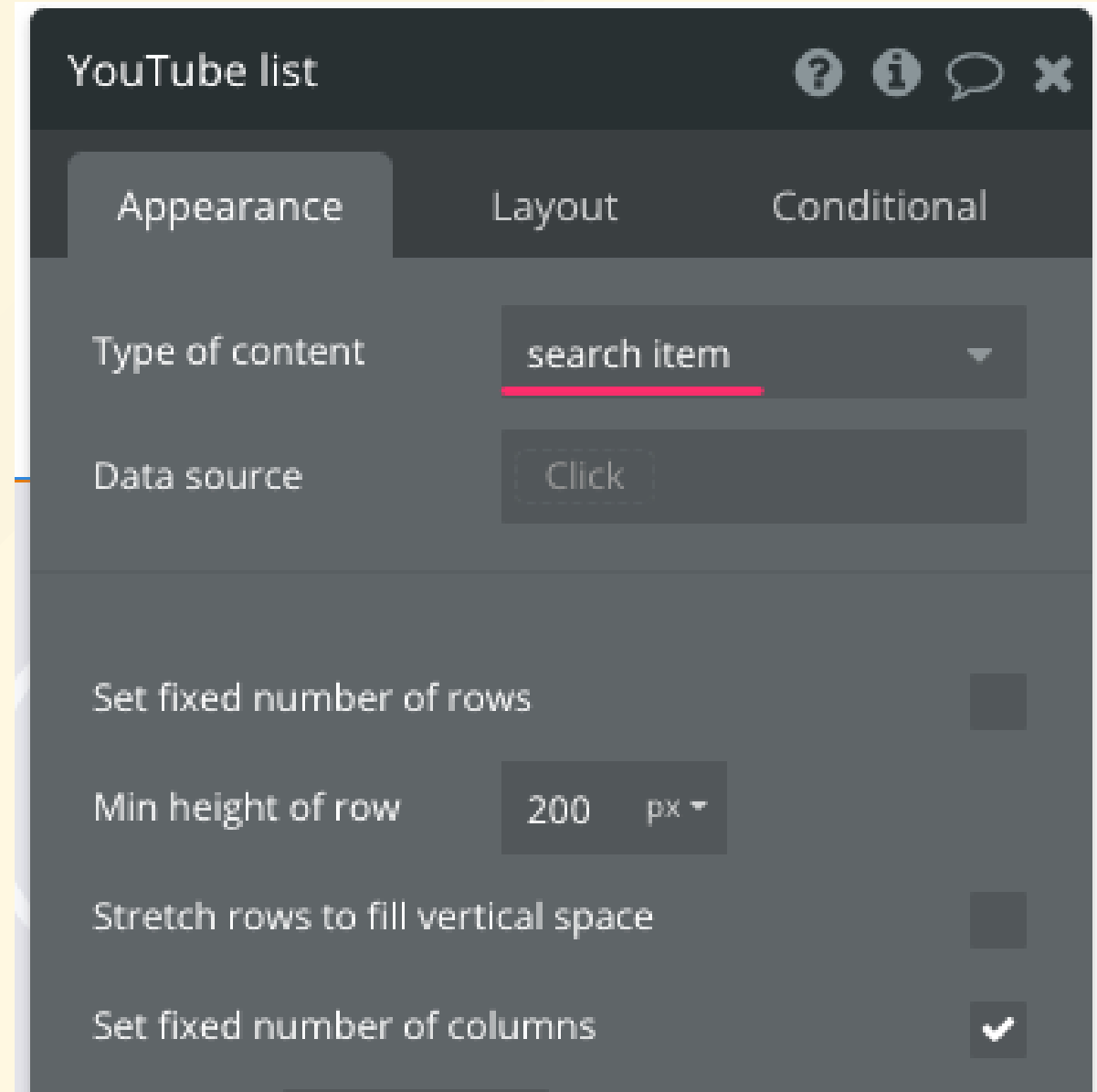
Preparation of Screen Display

- Now that Bubble's API integration is ready, it's time to set up the screens!
- We believe that you have already mastered screen setup in the previous and current lectures, so we will just give you an overview and let you try to build your screen layout.

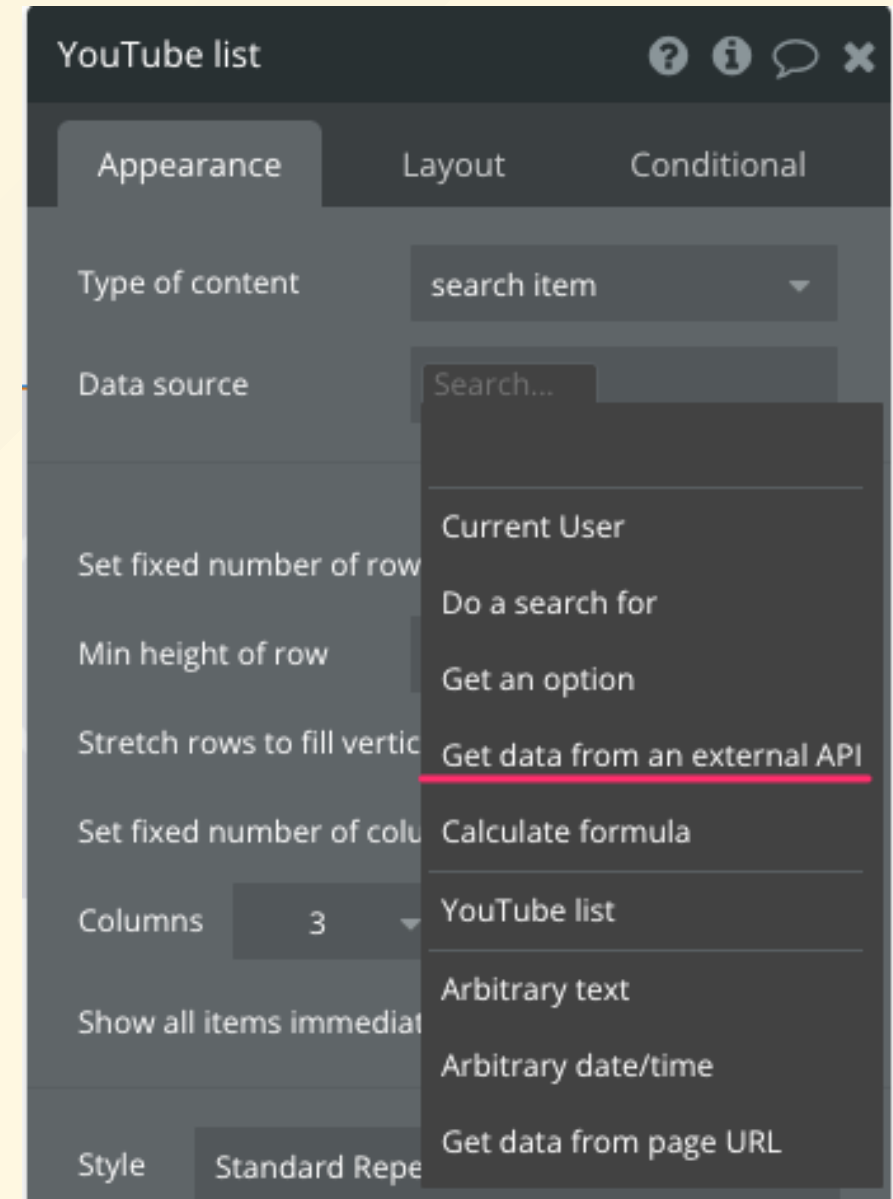
- Create the "video_list" screen by cloning the pet_list page.
- Delete all the contents of each cell in the Repeating Group.
- Instead, place Video in the Visual elements to fill the cell.
- It looks like this.



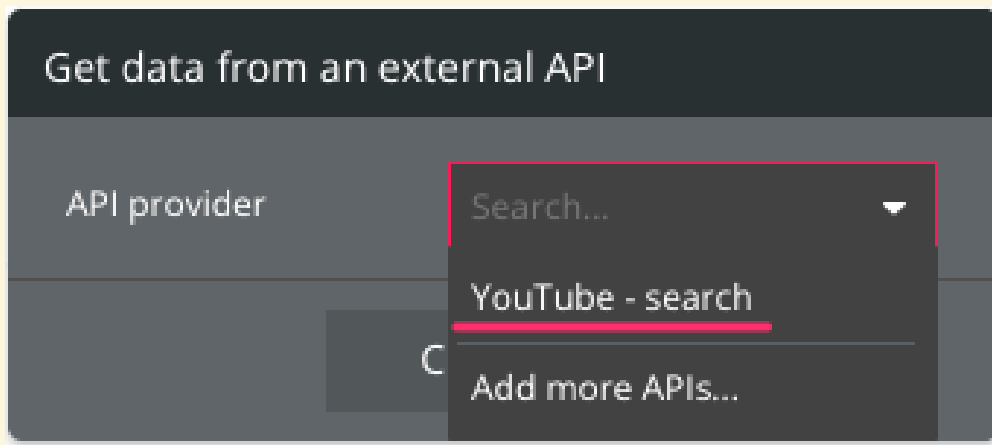
- Now that we have a layout, let's actually set up the YouTube listing.
- First, the Type of content in the Repeating Group will be the `search item` type that we just checked when executing the API.
- The type of data to be repeated will now be each YouTube video.



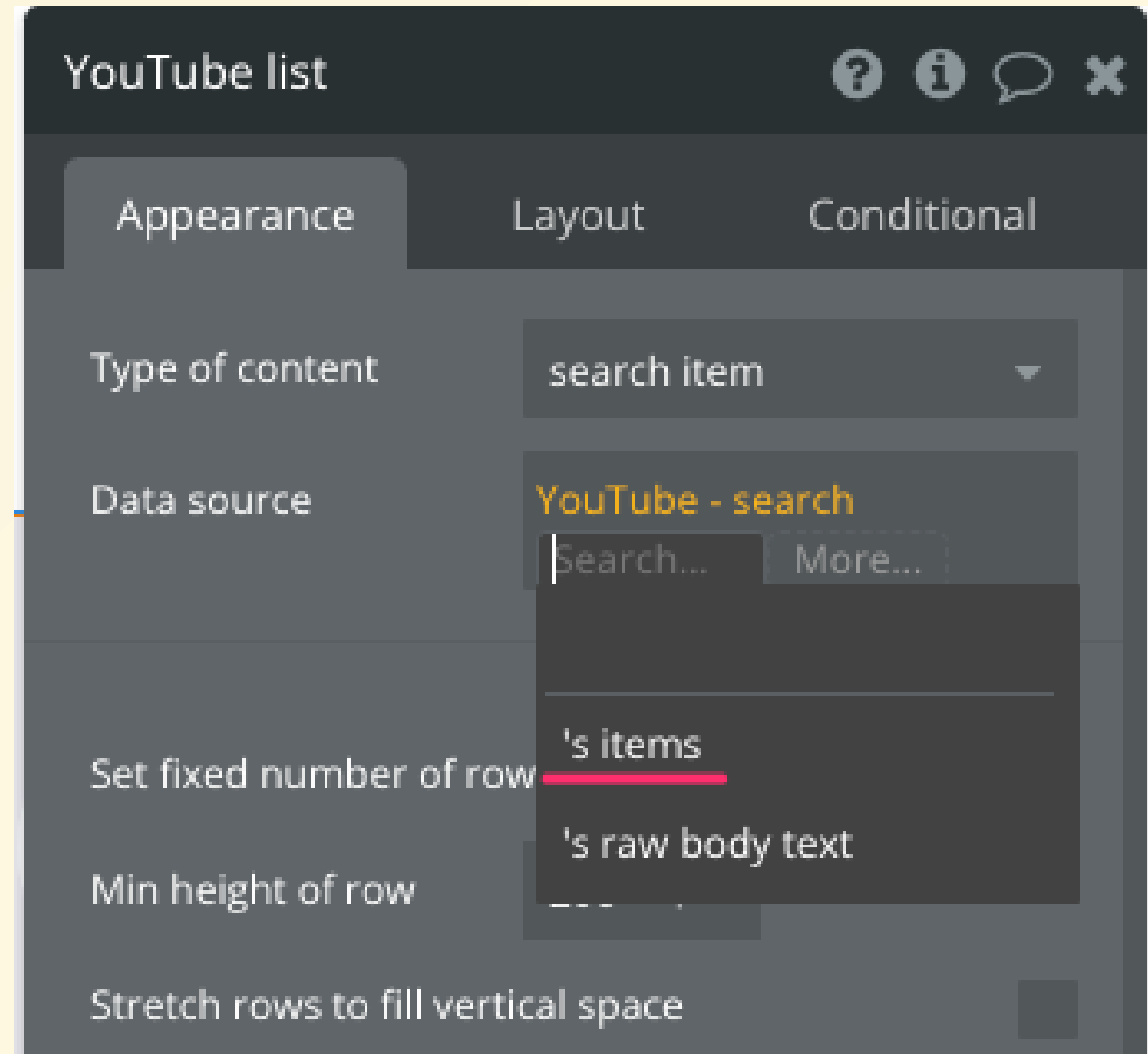
- Next is Data source. This time, we will target the list of YouTube videos obtained from the API.
- If you look at the candidates from Click, you will find an item called `Get data from an external API`, so select it.



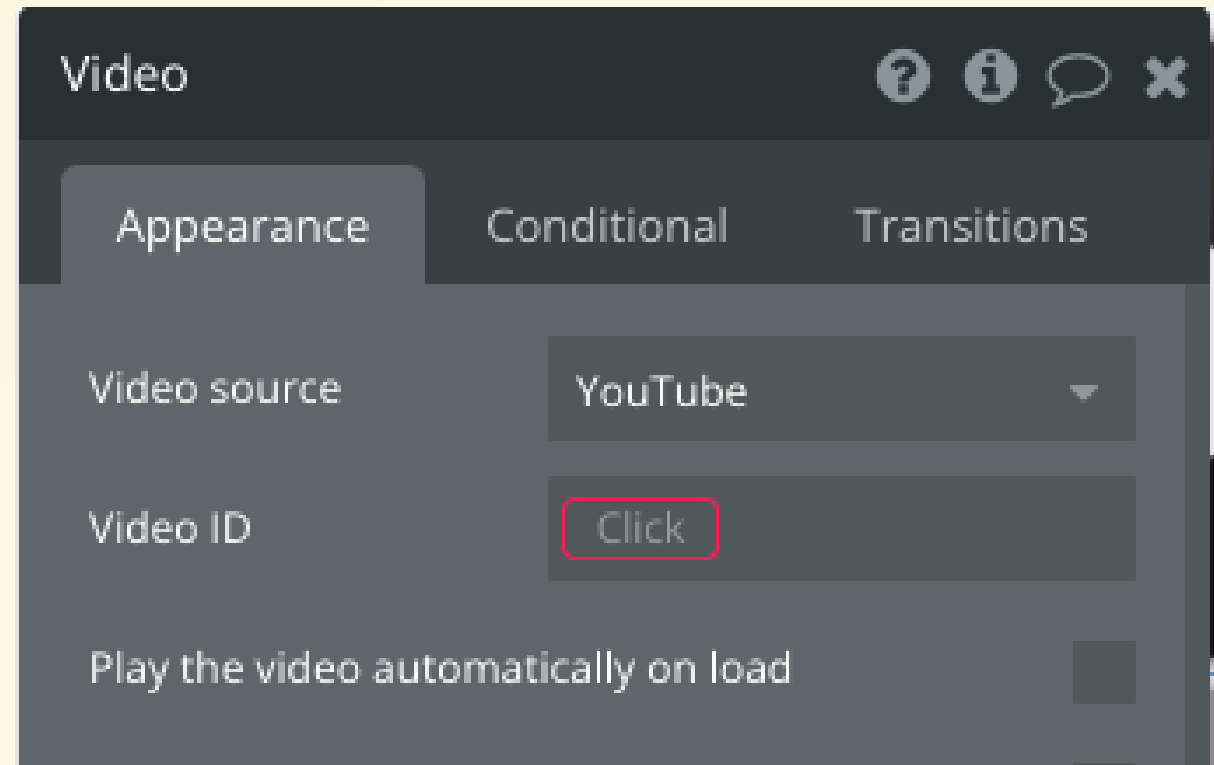
- Next to the Get data from an external API popup, select the API provider pull-down.
- Then select the **YouTube - search** that you have just set in the pull-down menu.
 - (The hyphen before is the API group name, and the hyphen after is the name of the specific API.)



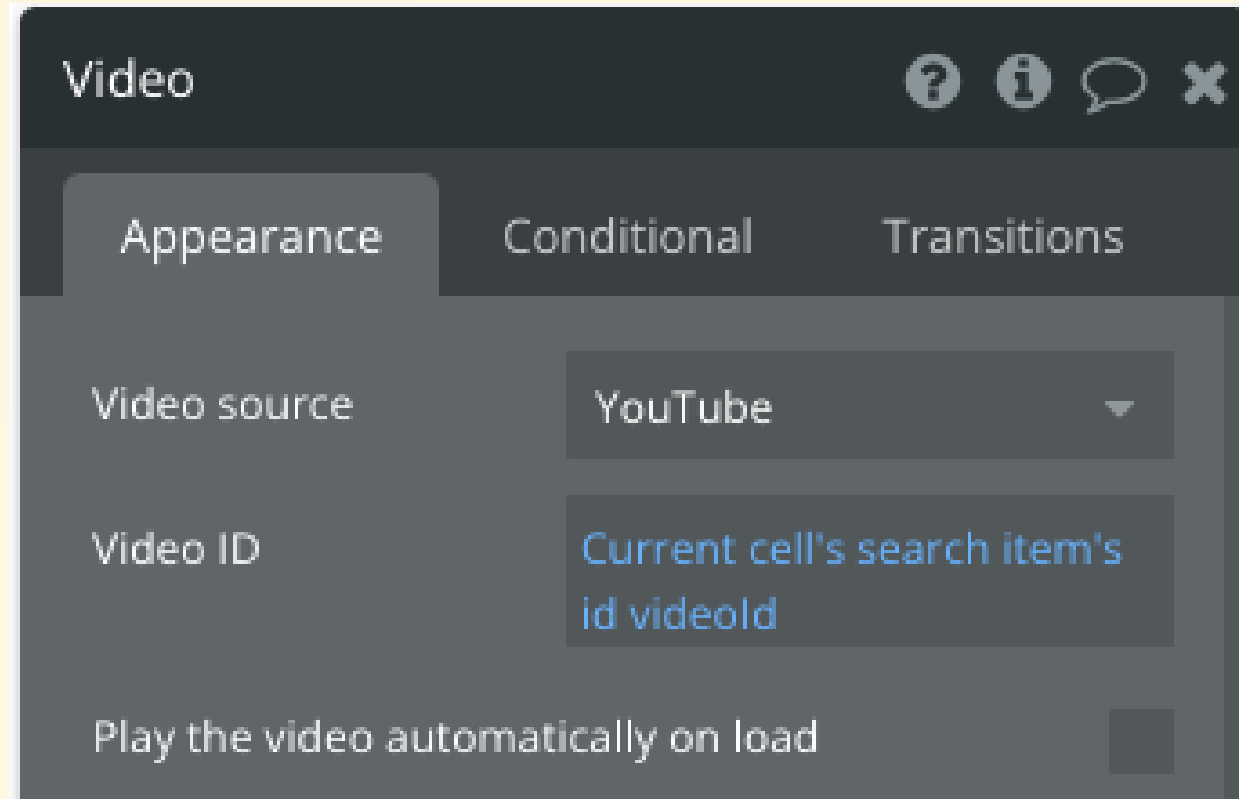
- Once the API Provider is set, close it with CLOSE
- Then, in the original Repeating Group, select `'items'` as the Data source.
- Now you have specified that the list of videos (items) included in the API results will be repeated.



- Next, let's set up each cell, but first, let's set up everyone!
- Hint 💡
 - Video source is YouTube
 - Then you will see an item called Video ID, which should refer to the **Video Id** you saw when you set up the API earlier.



- Something like this.



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

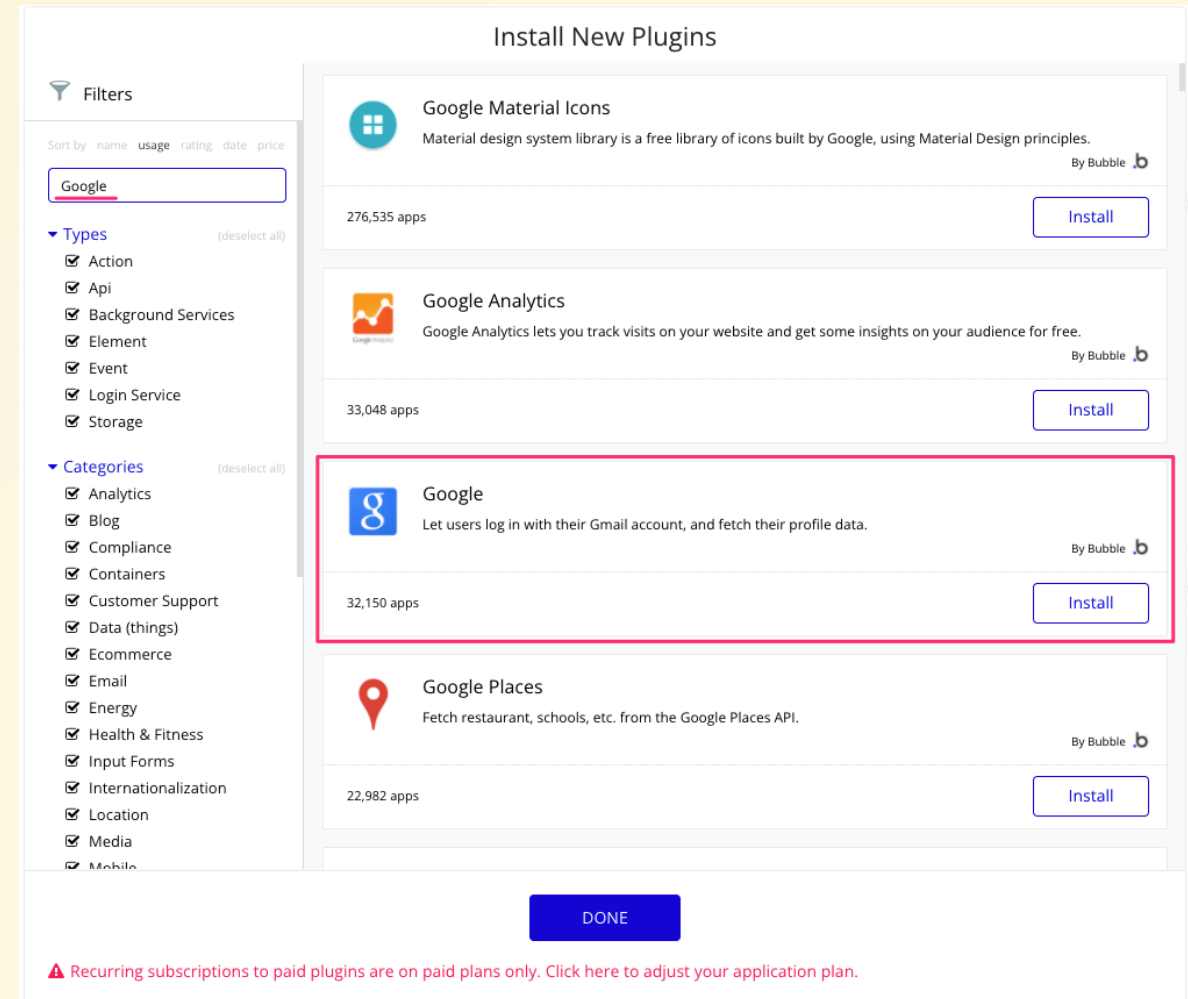
.bubble

Log in



Let's try Google Login!

- Next, let's try signing in to the Bubble app with your Google account!
- First, install the Google plugin



- Select the Google plugin you have installed
- Check the `Use a generic redirect URL` box in the right panel.
 - Note down this URL as well, but since you cannot select it on the screen, please replace the app name part of the URL below with your own app name and write it down. 😊
 - `https://{Your App 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.

Prepare to authenticate back to your app after logging in to Google

- Access the Google Cloud Platform screen

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

- Display the same API and Services screen as before

- Select the project and choose "OAuth Consent Screen" from the left menu

The screenshot shows the Google Cloud console interface. At the top, there's a blue header with the Google Cloud logo, a dropdown menu for 'bubble-naotake', and a search bar. Below the header, the left sidebar contains a list of menu items: 'API とサービス', 'ライブラリ', '認証情報', 'OAuth 同意画面' (which is highlighted with a blue bar), 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 the following text: 'アプリをどのように構成および登録するか（ターゲット ユーザーを含む）を選択します。プロジェクトに関連付けることができるアプリは1つだけです。' Below this, there's a section titled 'User Type' with two radio button options: '内部' (Internal) and '外部' (External). The '内部' option is selected. Below the '内部' option, there's a paragraph: '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](#)' The '外部' option has a paragraph: 'Google アカウントを持つすべてのテストユーザーが使用できます。アプリはテストモードで起動し、アプリを使用できるのは、テストユーザーのリストに追加されたユーザーに限られます。アプリを本番環境に移す準備ができたなら、アプリの確認が必要となる場合があります。 [ユーザーの種類の詳細](#)' At the bottom of the main content area, there's a blue button labeled '作成' (Create). At the very bottom, there's a link: '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 click Create

API	API とサービス	OAuth 同意画面
有効な API とサービス		アプリをどのように構成および登録するか（ターゲット ユーザーを含む）を選択します。プロジェクトに関連付けることができるアプリは 1 つだけです。
ライブラリ		
認証情報		
OAuth 同意画面		<p>User Type</p> <p><input type="radio"/> 内部 ?</p> <p>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</p> <p><input checked="" type="radio"/> 外部 ?</p> <p>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れば、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</p> <p>作成</p> <p>Google の OAuth に関する ご意見やご要望をお聞かせください。</p>
ページの使用に関する契約		

- Fill in the required information on the next page

- App Information
 - App name: The name of your app as created in bubble
 - User Support Email: Your Google Account email address

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 *
titech2022-advance-naotake
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 Domains
 - bubbleapps.io
- Developer Contact Information
 - Your Google Account email address
- "Save and Continue"

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。

承認済みドメイン 1*

[+ 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](#)

- Scope is "Save and Next" without changing any settings.

- Enter your Google email address as the test user.
- After setting up, save and continue.

アプリ登録の編集

✓ OAuth 同意画面

 —

✓ スコープ

 —

3 テストユーザー

 —

4 概要

テストユーザー

公開ステータスが「テスト中」に設定されている間は、テストユーザーのみがアプリにアクセスできます。アプリの確認前の許可済みユーザー数の上限は 100 で、この上限はアプリの全期間でカウントされます。 [詳細](#)

+ ADD USERS

≡ フィルタ

 プロパティ名または値を入力

?

ユーザー情報	
<u>kyougoku2bubble@gmail.com</u>	

保存して次へ

キャンセル

- Now that registration is complete, we will issue the credentials.
- Select "Credentials" from the left menu and click "+ Create Credentials".
- Select OAuth Client ID from the submenu

The screenshot shows the Google Cloud Platform 'API & Services' interface. On the left, the 'API とサービス' (API & Services) menu is visible, with '認証情報' (Credentials) selected. The main area shows the '認証情報' (Credentials) section, which includes a '+ 認証情報を作成' (Create Credentials) button and a '削除' (Delete) button. Below these, there are three options for creating credentials: 'API キー' (API Key), 'OAuth クライアント ID' (OAuth Client ID), and 'サービス アカウント' (Service Account). The 'OAuth クライアント ID' option is highlighted with a red underline. A tooltip is displayed over the 'OAuth クライアント ID' option, providing details about each option: 'API キー' (Simple API key for project identification), 'OAuth クライアント ID' (User consent for data access), 'サービス アカウント' (Robot account for server-to-server authentication), and 'ウィザードで選択' (Wizard selection for decision-making). At the bottom, it states '表示する OAuth クライアントがありません' (No OAuth clients to display).

API	API とサービス	認証情報
有効な API とサービス	有効な API にアクセス	
ライブラリ		
認証情報		
OAuth 同意画面		
ページの使用に関する契約		

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

OAuth クライアント ID
ユーザーのデータにアクセスできるようにユーザーの同意をリクエストします

サービス アカウント
ロボット アカウントによるサーバー間でのアプリレベルの認証を有効にします

ウィザードで選択
使用する認証情報の種類を決定するため、いくつかの質問をします

作成日 ↓

表示する OAuth クライアントがありません

- The Create OAuth Client ID screen will appear, enter the following information and "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 *
ウェブ アプリケーション

名前 *
Bubble OAuth

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

i 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 1 *
https://titech2022-advance-naotake.bubbleapps.io/api/1.1/oauth_redirect

+ URI を追加

注: 設定が有効になるまで 5 分から数時間かかることがあります

作成 キャンセル

- Authorized redirect URI:
The URL that the Bubble
side has noted when
installing the Google plugin.

```
https://{Your App  
Name}.bubbleapps.io/api/1.1/ou
```

← OAuth クライアント ID の作成

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

アプリケーションの種類 *
ウェブ アプリケーション

名前 *
Bubble OAuth

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



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

承認済みの JavaScript 生成元 ?

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 1 *
https://titech2022-advance-naotake.bubbleapps.io/api/1.1/oauth_redirect

+ URI を追加

注: 設定が有効になるまで 5 分から数時間かかることがあります

作成

キャンセル

- Click "Create."
- Then you will see a popup saying "OAuth client created", where the client ID and client secret are displayed, so write them down.
 - If you forget to write them down, you can download the JSON file.
- After writing down the information, click OK to close.

OAuth クライアントを作成しました

クライアント ID とシークレットには、常に API とサービスの認証情報からアクセスできます

i OAuth アクセスは、[OAuth 同意画面](#)に表示されている[テストユーザー](#)に制限されます

クライアント ID

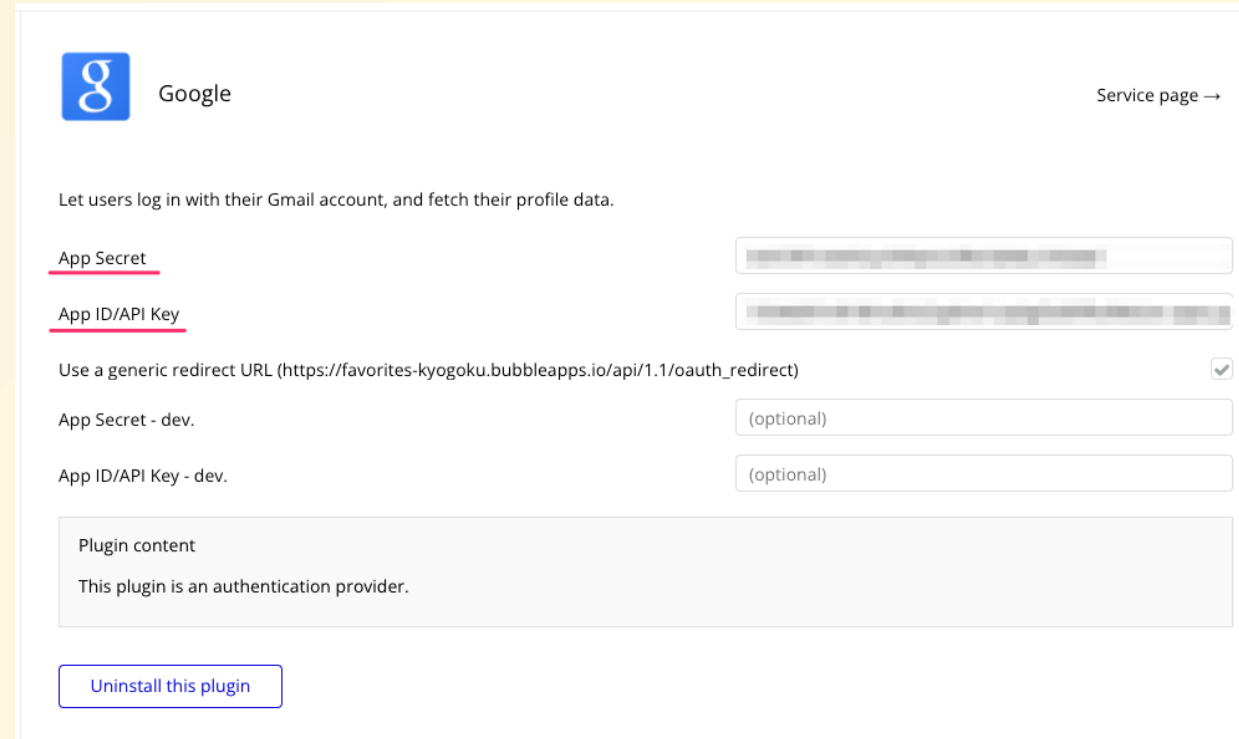
クライアント シークレット

↓ JSON をダウンロード

OK

Set the issued credentials on Bubble

- Return to the Google Plugin screen of Bubble, and enter the keys you have just obtained into the respective fields.
 - Client ID → AppID/API Key
 - Client Secret → App Secret

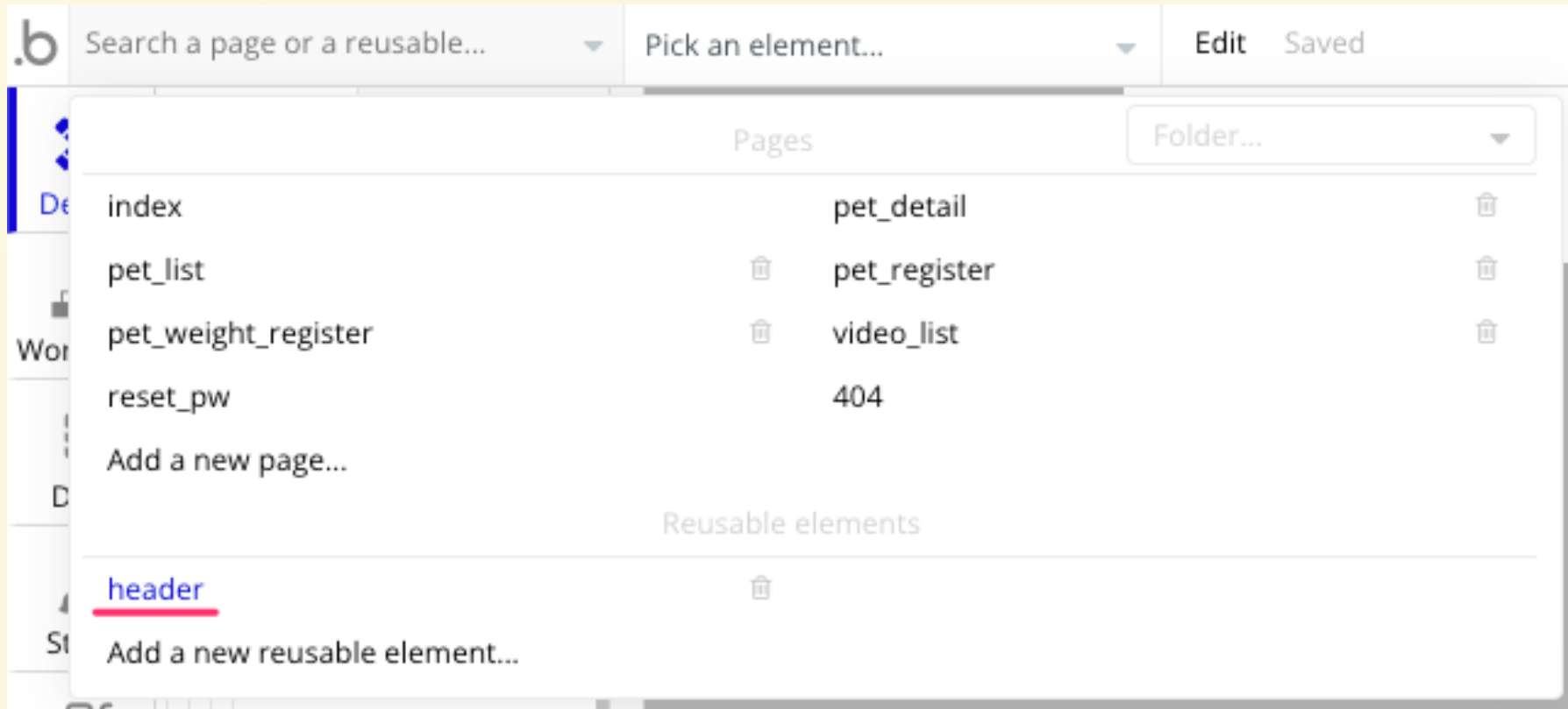


The screenshot shows the Google OAuth2 consent screen for a Bubble plugin. At the top left is the Google logo and the word "Google". At the top right is a link "Service page →". Below this is the instruction "Let users log in with their Gmail account, and fetch their profile data." There are two input fields: "App Secret" and "App ID/API Key", both with red underlines and containing masked text. Below these is a checkbox labeled "Use a generic redirect URL (https://favorites-kyogoku.bubbleapps.io/api/1.1/oauth_redirect)" which is checked. There are two more input fields: "App Secret - dev." and "App ID/API Key - dev.", both labeled "(optional)". At the bottom is a section titled "Plugin content" with the text "This plugin is an authentication provider." and a button labeled "Uninstall this plugin".

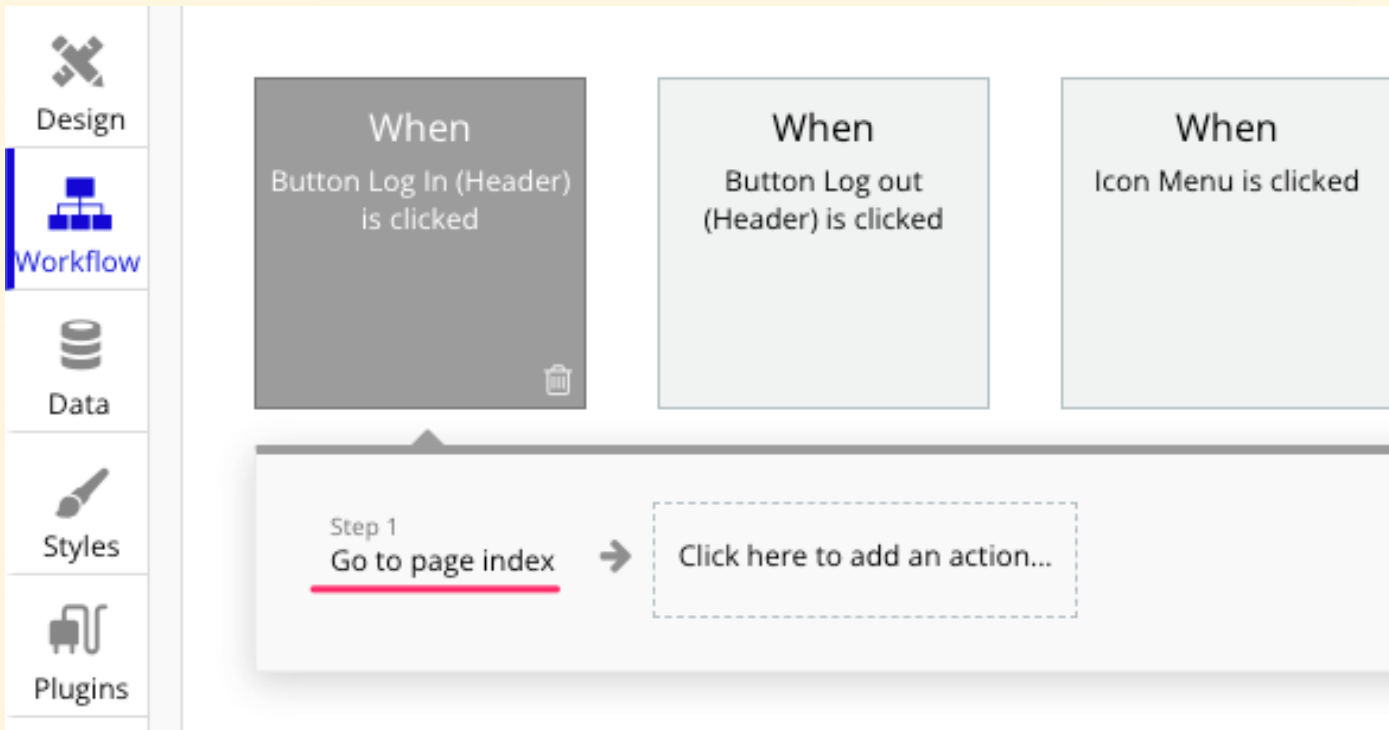
- This completes the preconfiguration.
- Next, we will build the login function.

Replace the login mechanism with Google

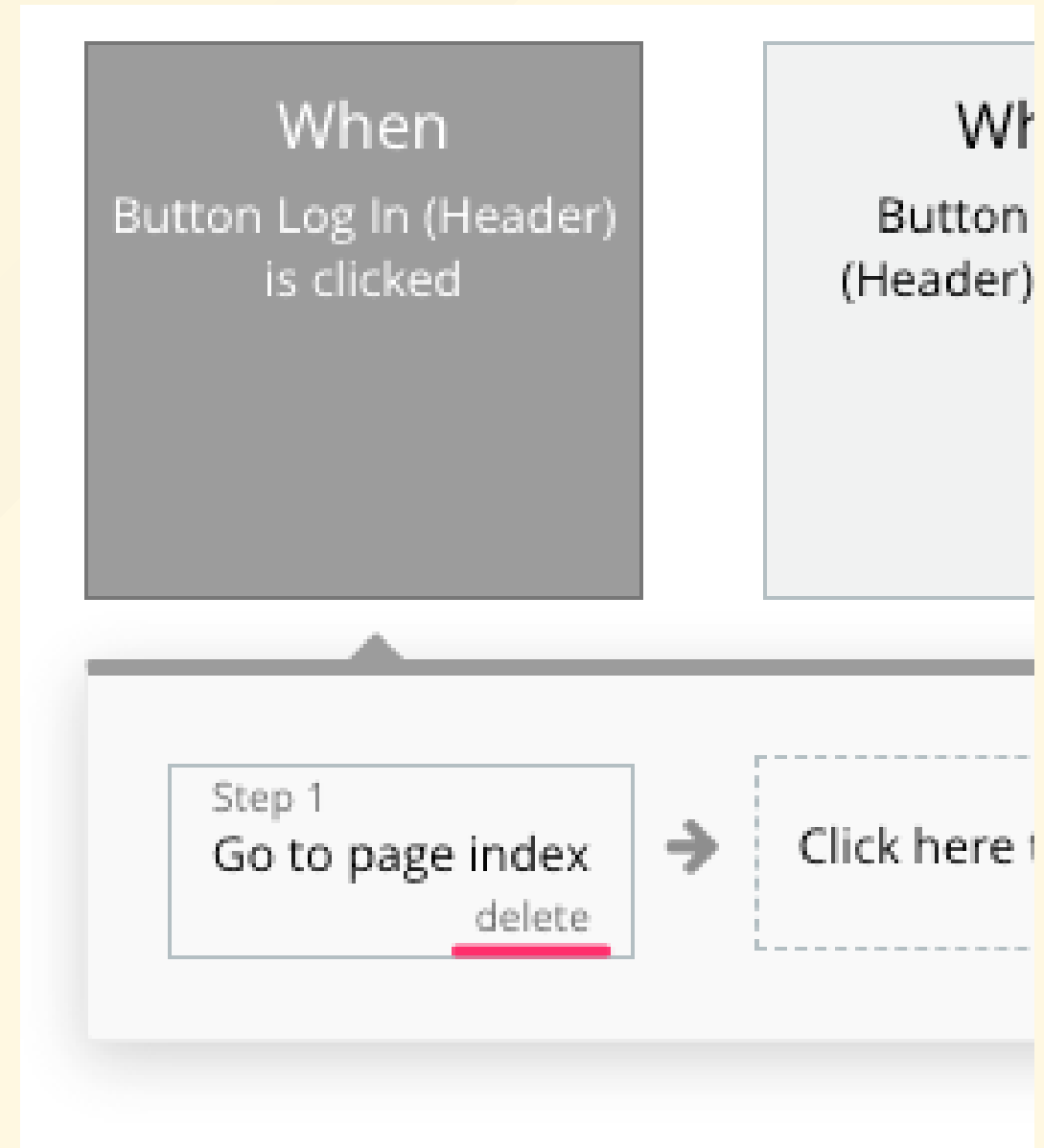
- Open the header of the Reusable elements created in the previous lecture



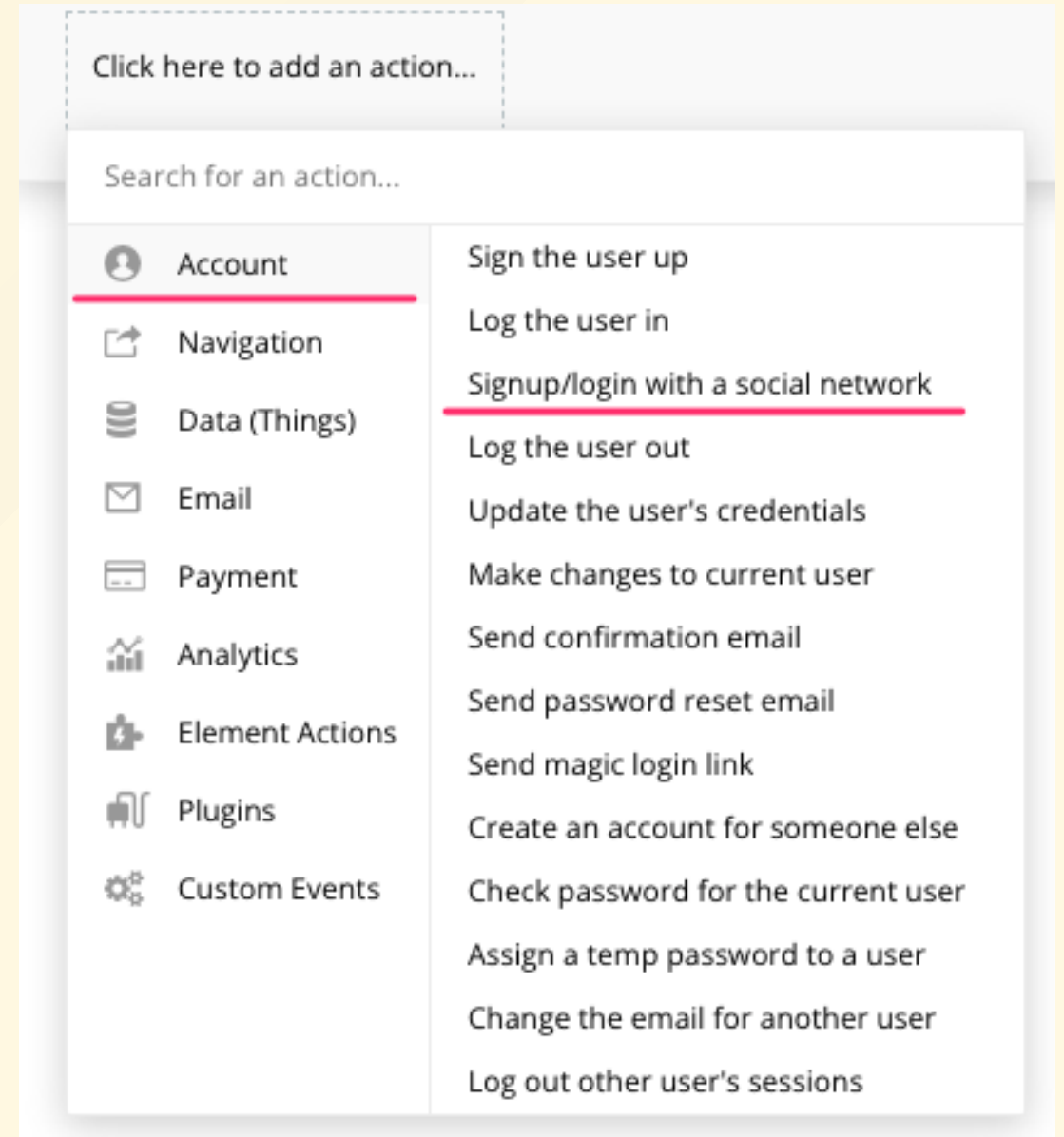
- Open the Workflow tab and open the workflow for the login button
- Now only the action to go to the login page (index page) is set, so we will review this.



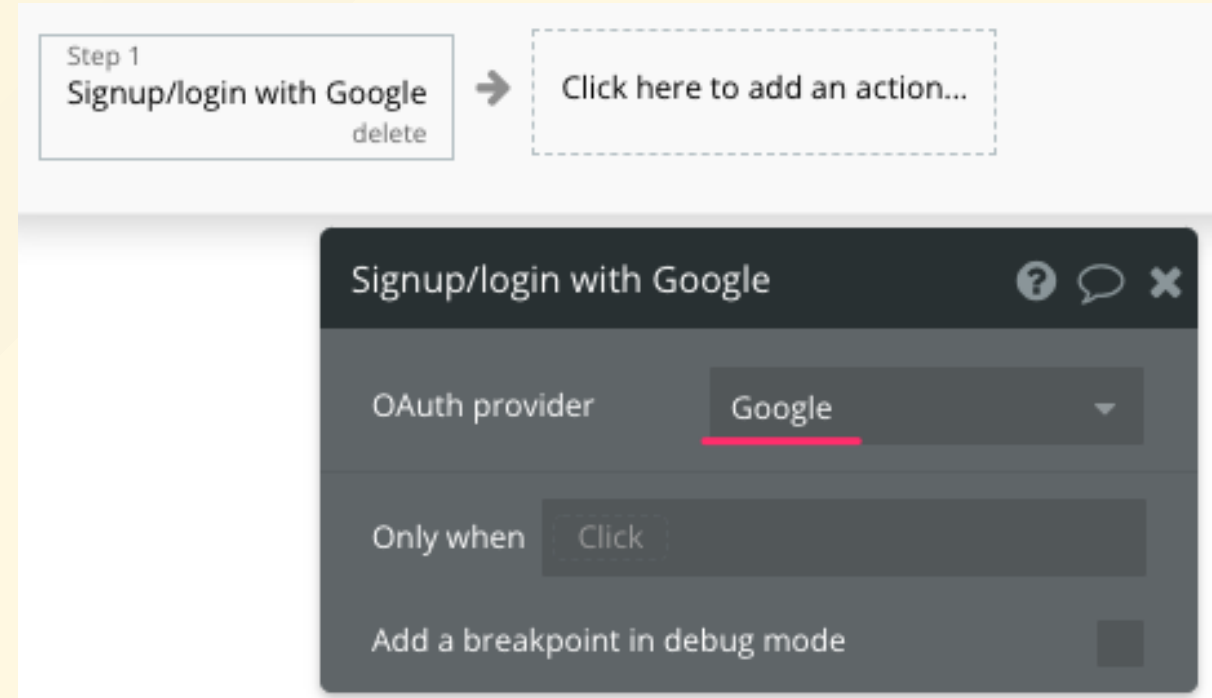
- Since we will be using the login screen provided by Google, we will delete the action to move to the login screen (index page) on the Bubble side.



- Next we will set up a workflow for Google Login.
- Click here to add an action... Click here to add an action...
- Select Signup/login with a social network from Account



- Signup/login with Google popup will appear
- Select Google as OAuth provider
- Since you have configured Google and Bubble plugins in advance, you can now use Google Authentication! 🙌



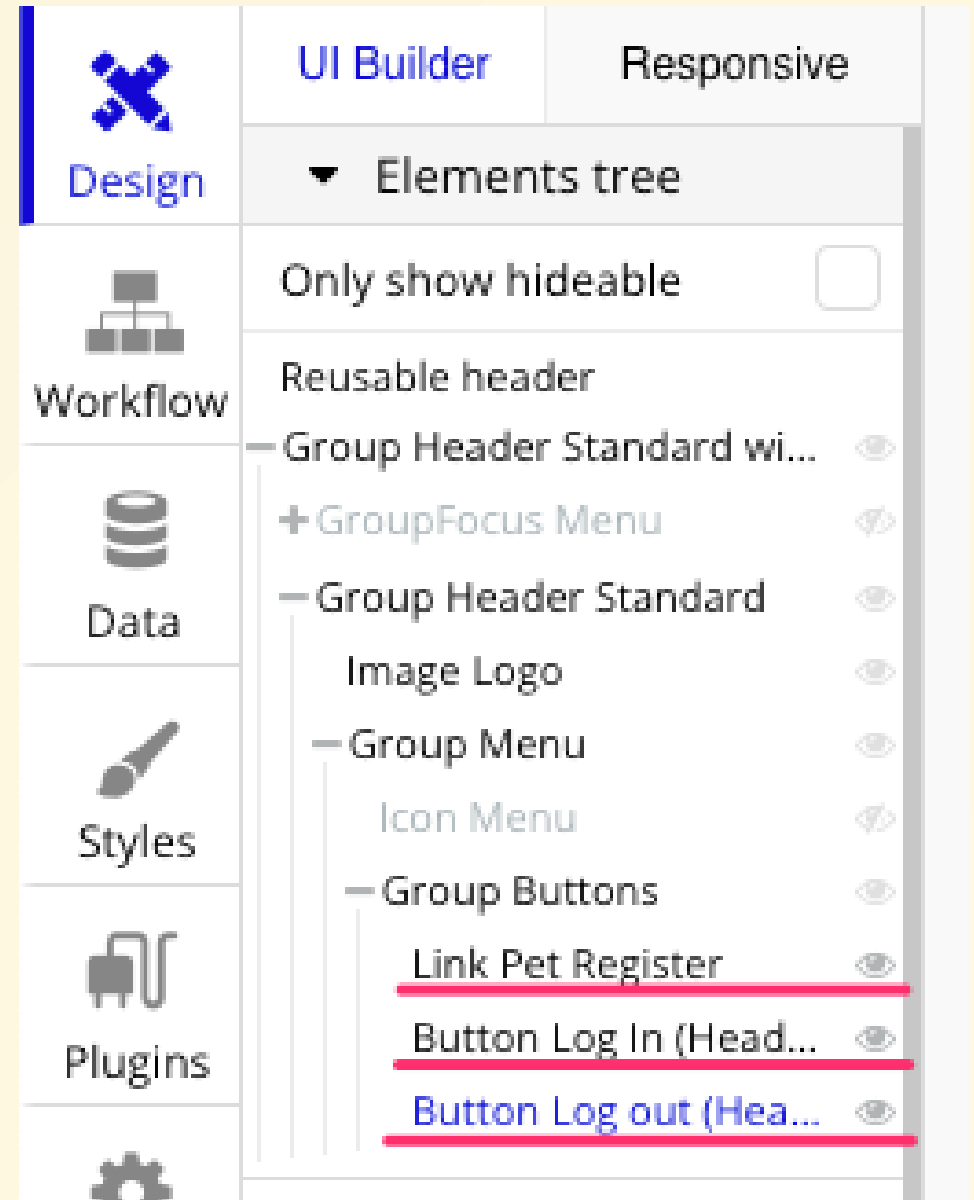
- 折角なのでログイン成功したらヘッダー上部にアカウント名と画像を表示してみましょう

Pet Register



Log out

- Switch to the Design tab and display the following three components from the Elements tree, which are currently hidden
 - (If you don't do this, it will be difficult to see the image after the image is added.)

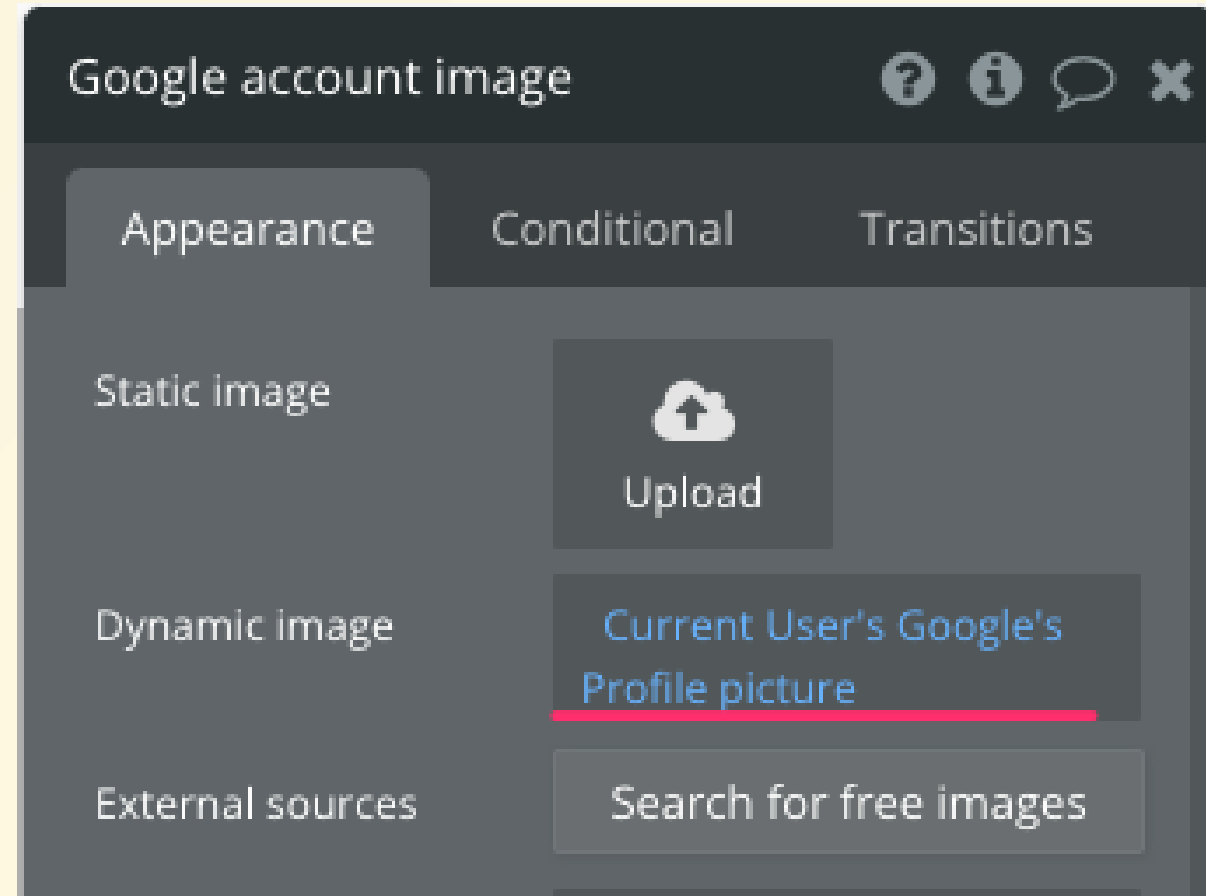


- Select Image from Visual elements and place the image between the "Pet Register" link and the "Log in" button.
 - Make sure it is a 50 x 50 square



- Then, set up the image references.
- If you are a Bubbler, you know what to do!
- Dynamic image for dynamic display
- The image you want to display is the profile image of the currently logged in user's Google Account.

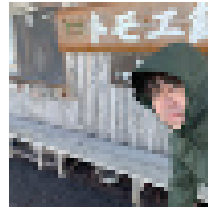
- Something like this.



- Now let's preview!
 - If you have the same email address in User of Data as the Google email address you are going to use to check the operation, please delete it.
- When you click the "Login" button, you will be redirected to the Google login screen, and when you login there, you will be redirected back to the Bubble side, right?
- By the way, when you log in, the email address information of the Google account you logged in is registered in the User data on the Bubble side.
 - Of course, your password is not saved, so don't worry! 😊

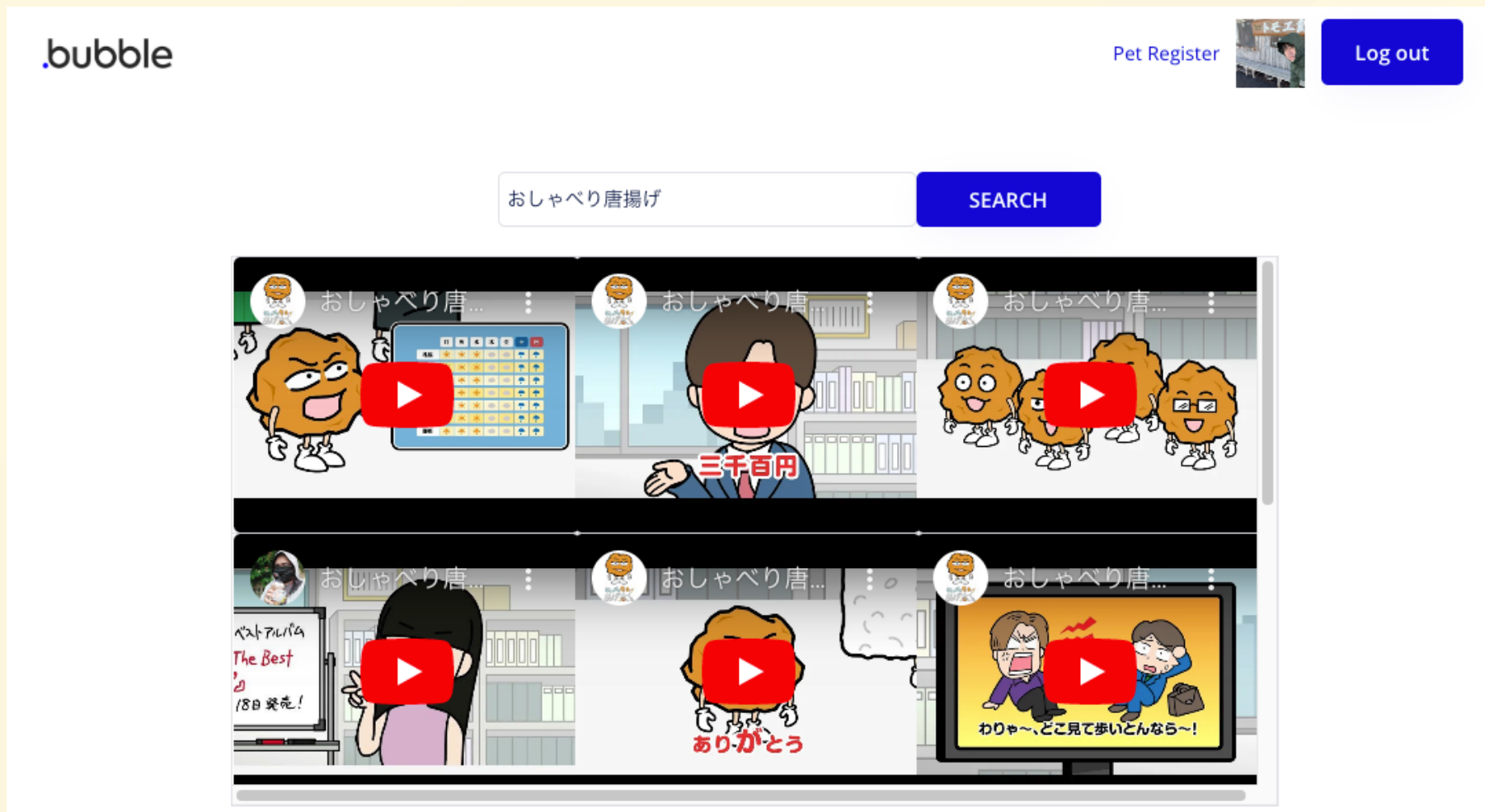
Log in

Pet Register



Log out

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



- Currently, the YouTube API authentication key and search keywords are fixed.
- So let's change them as follows
 - Authentication keys are 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 a hint on the next page so you can try it first!

- First, add a field called `key` for the User and set it in advance.
 - (Normally, it would be more natural to have the user enter the key himself, but for the sake of time, we will skip the API key setting screen this time.)
- If you want to dynamically set API parameters in the process, first remove the fixed values and uncheck `Private` so that you can specify the parameters when calling the API from a DataSource, etc.
- Place a new search element (text box and search button) on the video_list page, and when the search button is clicked, search via YouTube API and set the result to the Repeating Group.



- Here are the important points
- First, add a field called **key** for the User type

Fields for type User

Type name 

<input type="text" value="key"/>	text	default <input type="text"/>
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) obtained this time to the registered User in advance

Data typesPrivacyApp dataOption setsFile manager

Database views

New viewPrimary fields

Search view names or data types...

All Users

Application data - All Users - Development version

Search

1 entries (displaying 1)

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

- Next, the YouTube API settings look like this

collapse

Name Use as Data Data type JSON

GET (use [] for params)

Headers

Add header

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"/>	

Add parameter

Include errors in response and allow workflow actions to continue ☐

Capture response headers ☐

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

Reinitialize call [Manually enter API response](#)

- The key point is that we have removed the Value values for items that we want to specify dynamically, and unchecked the Private checkbox.
- This allows the values of these parameters to be specified when using the API.

The screenshot shows an API configuration interface. At the top, the Name is 'search', Use as is 'Data', and Data type is 'JSON'. The method is 'GET' and the URL is 'https://www.googleapis.com/youtube/v3/search'. Below this is a 'Headers' section with an 'Add header' button. The 'Parameters' section contains a table with four rows: 'key', 'q', 'maxResults', and 'type'. The 'key' and 'q' rows have empty value fields and the 'Private' checkbox is unchecked. The 'maxResults' row has a value of '10' and the 'Private' checkbox is checked. The 'type' row has a value of 'video' and the 'Private' checkbox is checked. Below the parameters is an 'Add parameter' button. At the bottom, there are two checkboxes: 'Include errors in response and allow workflow actions to continue' and 'Capture response headers', both of which are unchecked. A message states 'This call has been modified since you last initialized; consider reinitializing.' At the very bottom, there is a 'Reinitialize call' button and a link to 'Manually enter API response'.

Name: search Use as: Data Data type: JSON

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

Headers

Add header

Parameters

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

Add parameter

Include errors in response and allow workflow actions to continue ☐

Capture response headers ☐

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

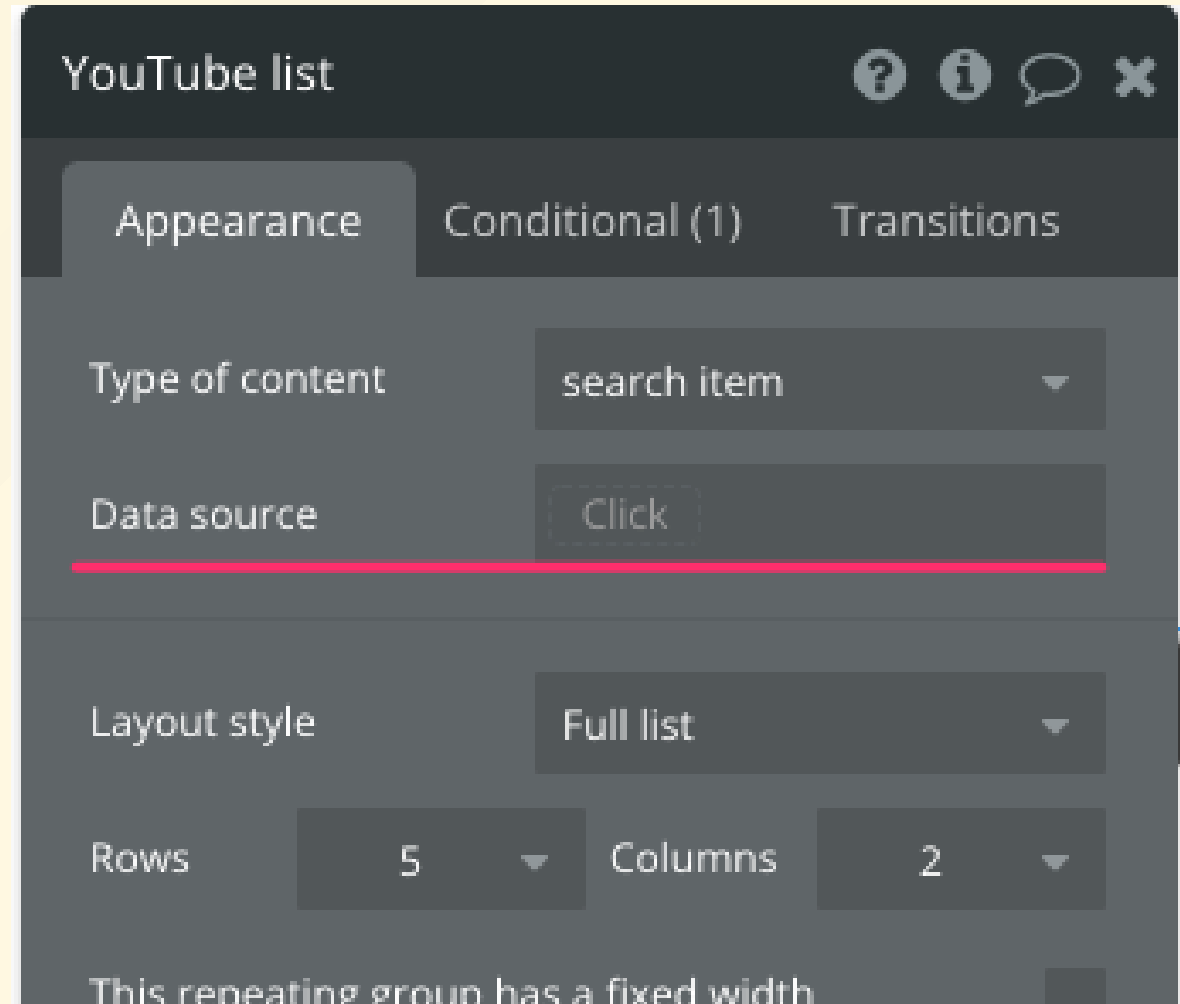
Reinitialize call Manually enter API response

- In addition, let's add two search options
- key: `maxResults`, the number of data to retrieve in one search.
 - Default is 5.
- key: `type`, type of data to search.
 - By specifying `video`, you can search only videos.

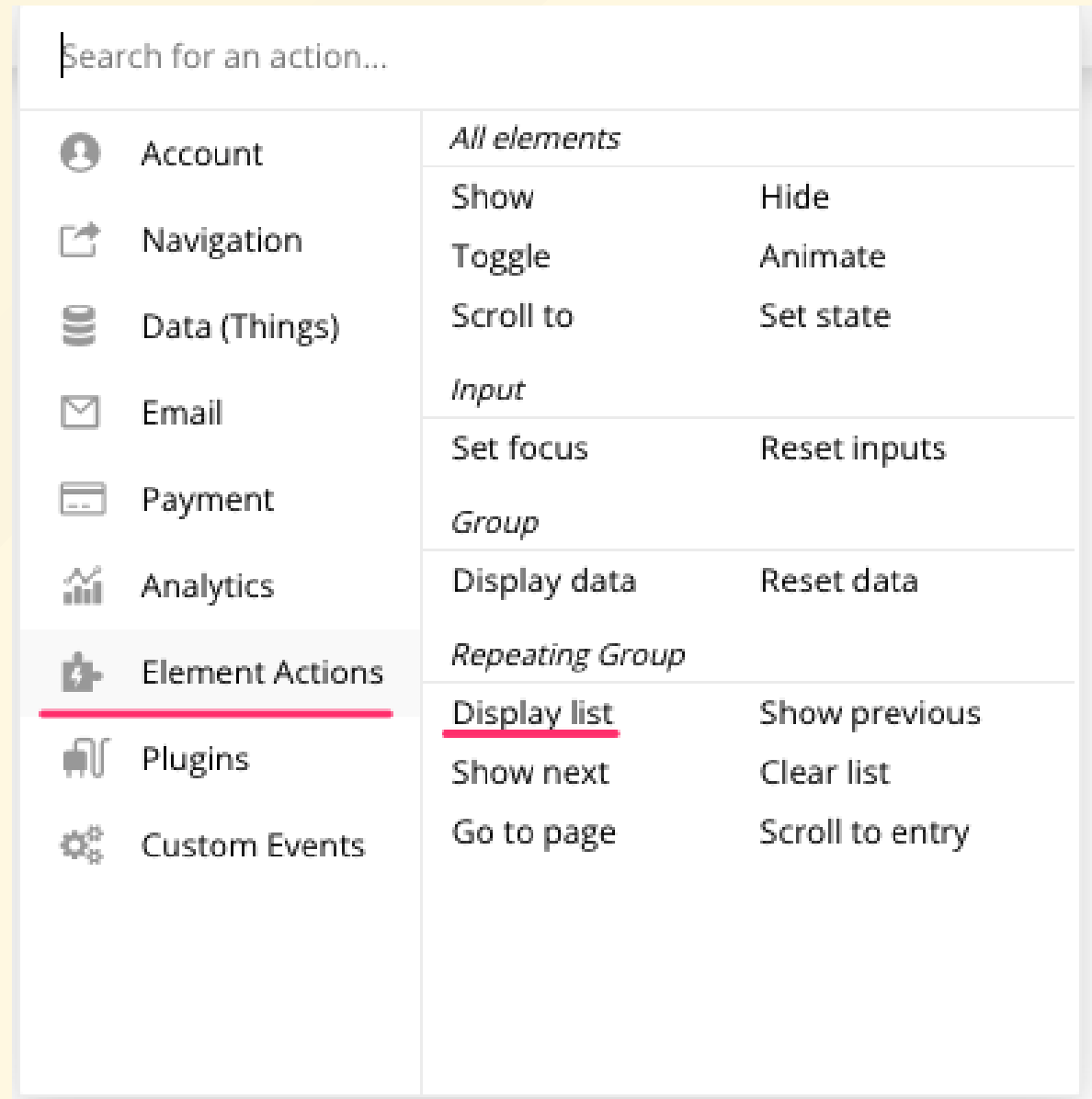
The screenshot shows an API configuration interface. At the top, the 'Name' field is 'search', 'Use as' is 'Data', and 'Data type' is 'JSON'. The 'Method' is 'GET' and the 'URL' is 'https://www.googleapis.com/youtube/v3/search'. Below this is a 'Headers' section with an 'Add header' button. The 'Parameters' section contains a table with four rows. The third and fourth rows are highlighted with a red line. The first two rows have empty value fields, while the third and fourth have '10' and 'video' respectively. Below the parameters is an 'Add parameter' button. At the bottom, there are two checkboxes: 'Include errors in response and allow workflow actions to continue' and 'Capture response headers', both of which are unchecked. A message states 'This call has been modified since you last initialized; consider reinitializing.' At the very bottom, there is a 'Reinitialize call' button and a link to 'Manually enter API response'.

Key	Value	Private	Allow blank	Optional
key		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
q		<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"/>

- Next, empty the Repeating Group's Data source.
- The target data to be displayed is set in the workflow of the new search function, so there is no problem even if this field is empty.

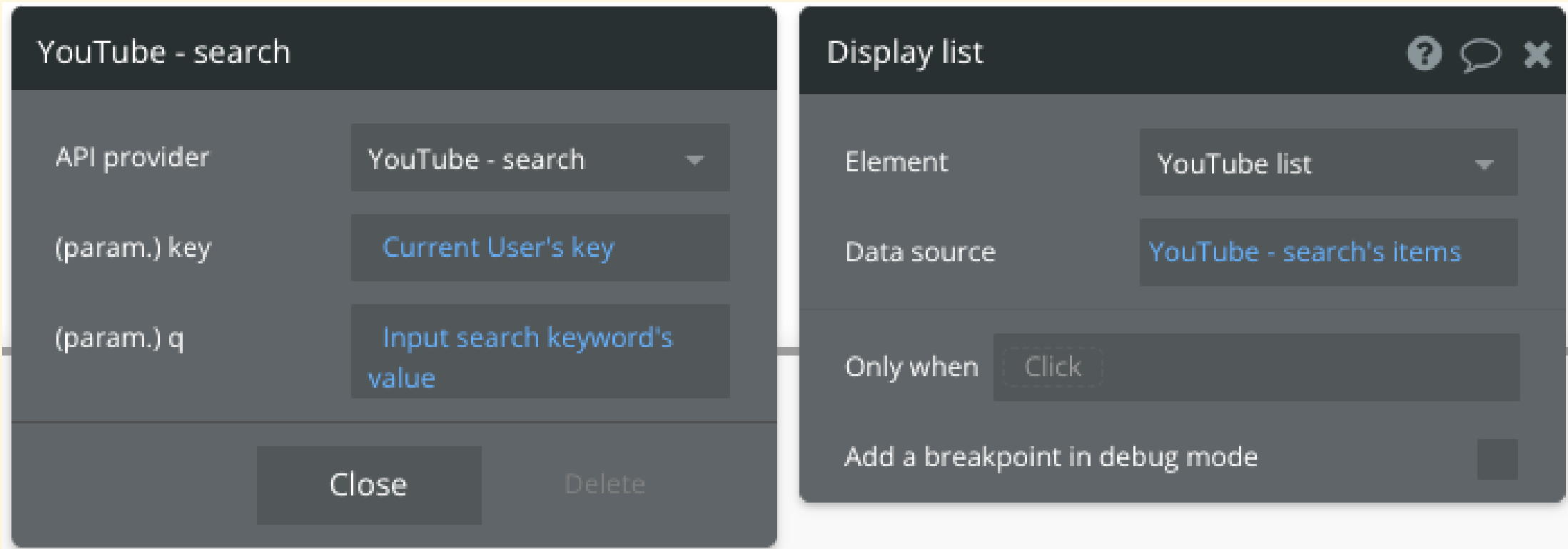


- Next, set up a workflow for the newly prepared search button for the keyword search.
- In this case, the action is for an element (Repeating Group), so select "Display list" in the "Element Actions" section.



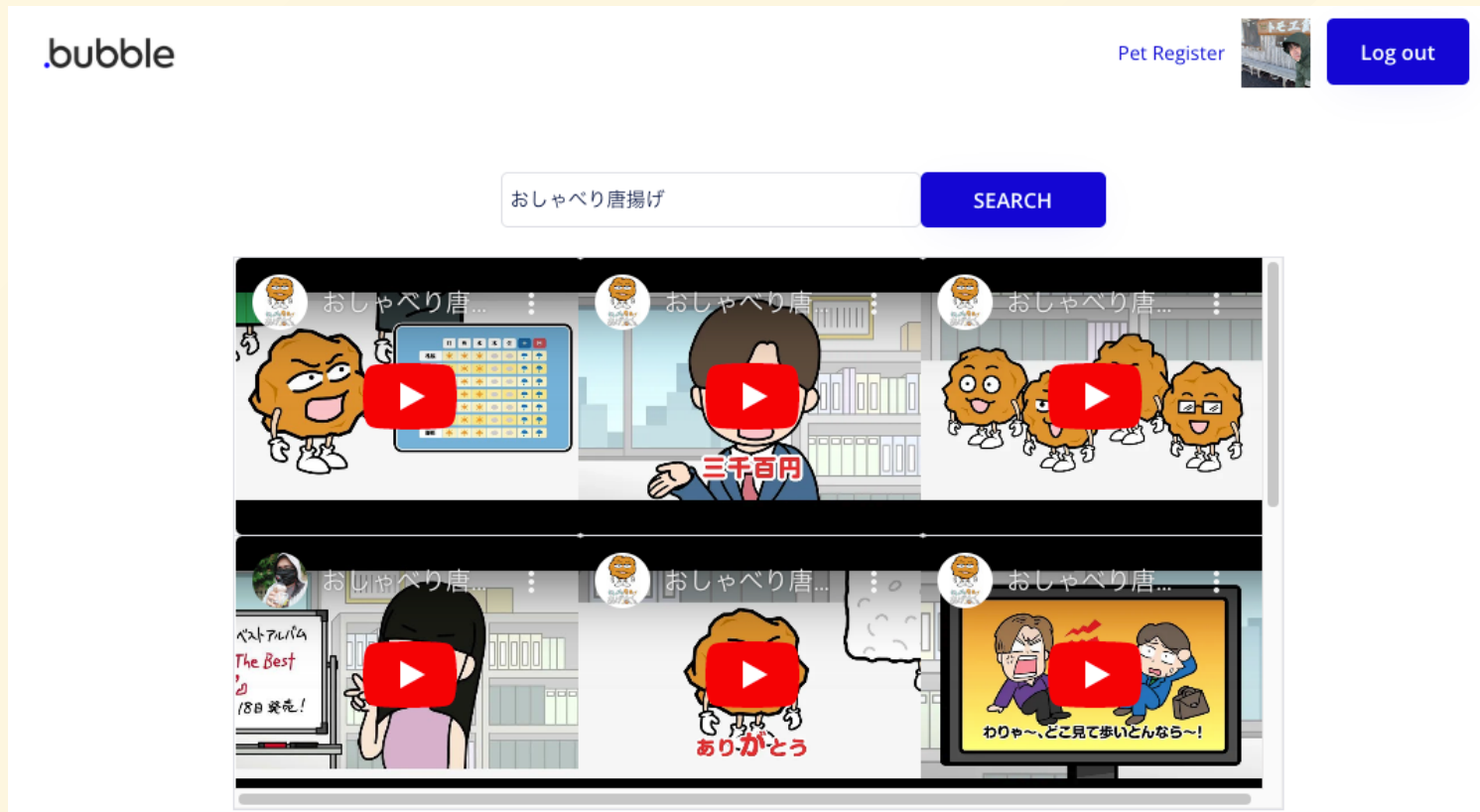
- For Data source, select Get data from an external API to display data via API.
- Then select `YouTube - search` for the API provider, and you should see two params that were not there before.
- This is because you have just unchecked the Private checkbox for the parameters you want to specify dynamically in the YouTube API settings.
- The value to be specified here is the usual Dynamic data.

- Here's what it looks like when set up



Let's preview it!

- Did you see the list of videos you entered in the keyword field while logged in?



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

Tips for Collaborating with Bubble

Here are some tips to help you and your team collaborate with Bubble

How to invite team members in Bubble

To co-edit one app with your team members...

- You will need to prepare one account for your team, and then everyone will log in to that account and use it together.

- Bubble has the ability to co-edit in the regular way, but... 💰💰💰

Upgrade to access this feature
Adding collaborators to this app is only available on professional plan and above

Bill monthly Bill annually & save

Free	Personal	Professional	Production
\$0/mo	\$29/mo	\$129/mo	\$529/mo
Current plan	Choose plan	Selected plan	Choose plan
<ul style="list-style-type: none">✓ Bubble branding✓ Password protection: Editor only✓ Community support✓ Basic server capacity✓ 1 application editor✓ 30 minutes of logs✓ 0.5 GBs file storage	<p>Everything on Free, plus</p> <ul style="list-style-type: none">✓ White labeling✓ Access to live version✓ Custom domain✓ Run monthly scheduled workflows✓ API✓ Password protection: Editor & live mode✓ Import / Export CSV✓ Email support✓ Data backup & restore last 7 days✓ 24 hours of logs✓ 10 GBs file storage	<p>Everything on Personal, plus</p> <ul style="list-style-type: none">✓ Run daily scheduled workflows✓ Discourse SSO✓ Algolia search✓ 3 reserved units of capacity✓ Ability to add up to 7 units of capacity✓ <u>2 application editors</u>✓ 2 development versions✓ Data backup & restore last 30 days✓ 14 days of logs✓ 20 GBs file storage	<p>Everything on Professional, plus</p> <ul style="list-style-type: none">✓ Main apps + Sub apps✓ Priority email support✓ 10 reserved units of capacity✓ Ability to add up to 30 units of capacity✓ <u>15 application editors</u>✓ 20 development versions✓ Data backup & restore last 365 days✓ 50 GBs file storage

Cancel Bill \$129 monthly

Notes on simultaneous edits in Bubble

- Avoid simultaneous editing on the same element (it will be overwritten by edits made later).
 - For this reason, it is recommended to assign a person in charge of each screen and avoid simultaneous editing of the same screen.
- Other than that, there are no special notes.
- Edits are reflected in real time on other people's screens.
 - Not only screen edits, but also database and Workflow edits are reflected in real time.
- If you divide the work by screen, you can develop without difficulty.

Team Development Exercise

- Try developing one new app with your DevelopmentPhase team!
 - Make it different from the one you are planning to develop in DevelopmentPhase!
- You can choose to use either Adalo or Bubble!
 - You can choose the one you plan to use in DevelopmentPhase, or you can choose both!
- Make sure everyone has a hand in it!

Example of how to divide up the work

- First, everyone should work together to identify screens and functions and design the database.
- Assign a person to be in charge of each screen (e.g., in charge of the registration screen, the list screen, the detail screen, the update screen, etc.)
- There is also a method in which multiple people work on a single PC. In this case, the person in charge of operating the PC and the person giving instructions should switch periodically.



Presentation of Exercise Results

Let's have each team present about the apps they made in the exercise!

Summary

- That's all for this Learning Phase lecture!
 - Have you mastered Adalo and Bubble?
 - Did you get an idea of how team development works?
- From next time, we will start the Development Phase, where we will brush up and develop the ideas presented by each team.
 - In preparation for this phase, let's reconfirm our ideas and select tools (Adalo, Bubble, or others?) to be used in the development phase. (Adalo? Bubble? Other?) to be used in the development.

Translated with www.DeepL.com/Translator (free version)

Finally.

- Remove API keys and OAuth information issued by Google Cloud Platform!

That's all!

Thank you for your hard work!