

Programming Boot Camp

Bubble の基本 #2-2

東京工業大学 2022/11/26

Naotake KYOGOKU

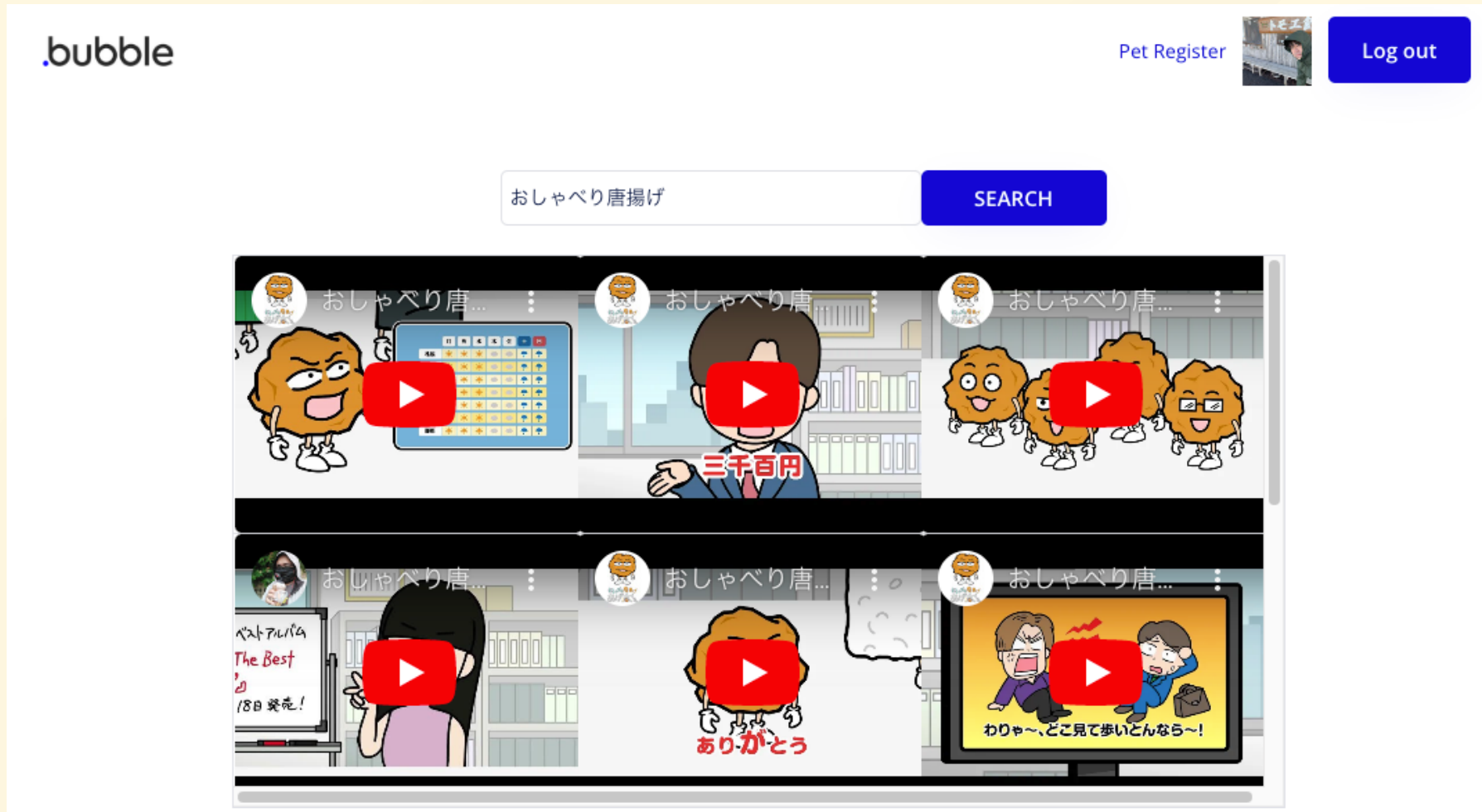
目次

- Bubble での外部連携
- Bubble での共同作業の Tips
- チーム開発演習
- まとめ

今回やることの確認

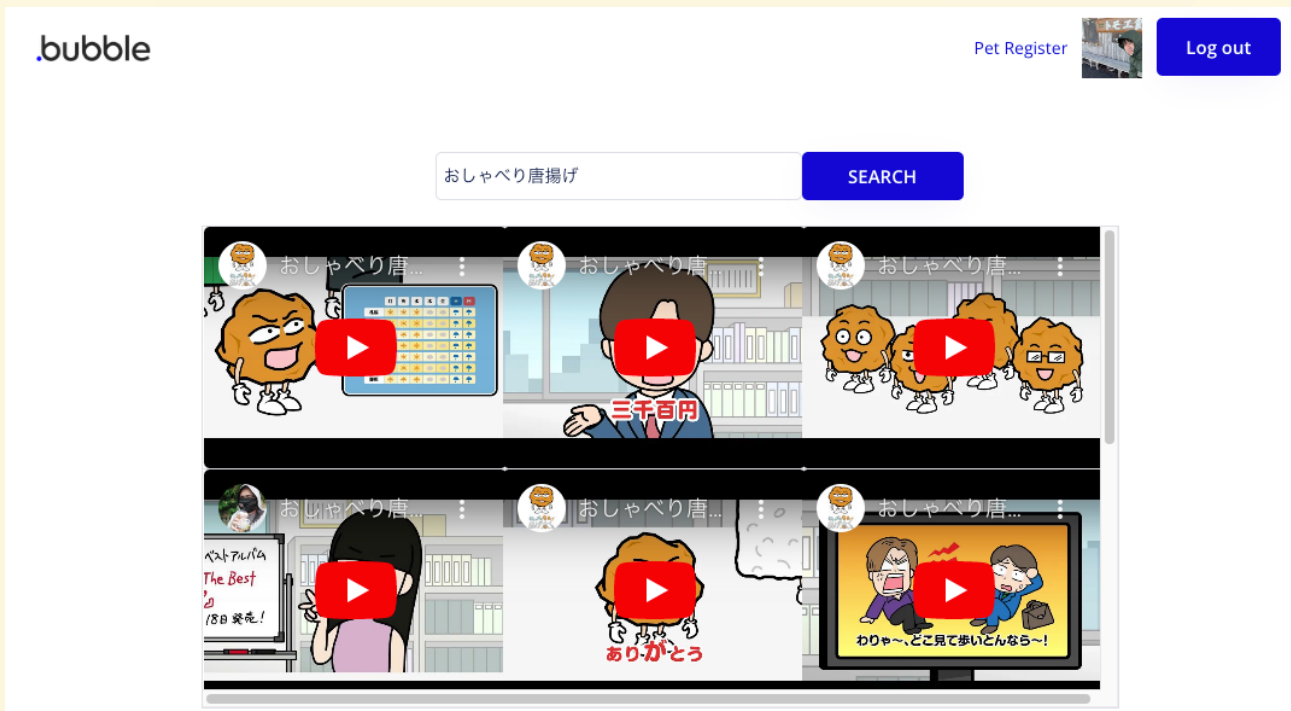
- 今回はまず、Bubble で作ったアプリケーションを外部のサービスと連携させる方法を学びます
- 次に、1つのアプリケーションを使った共同作業に関するコツや注意点を紹介します
- 最後に、Development Phaseのチームでアプリ開発の演習と発表を行います

Bubble での外部連携



機能としては下記を盛り込んだ画面となります

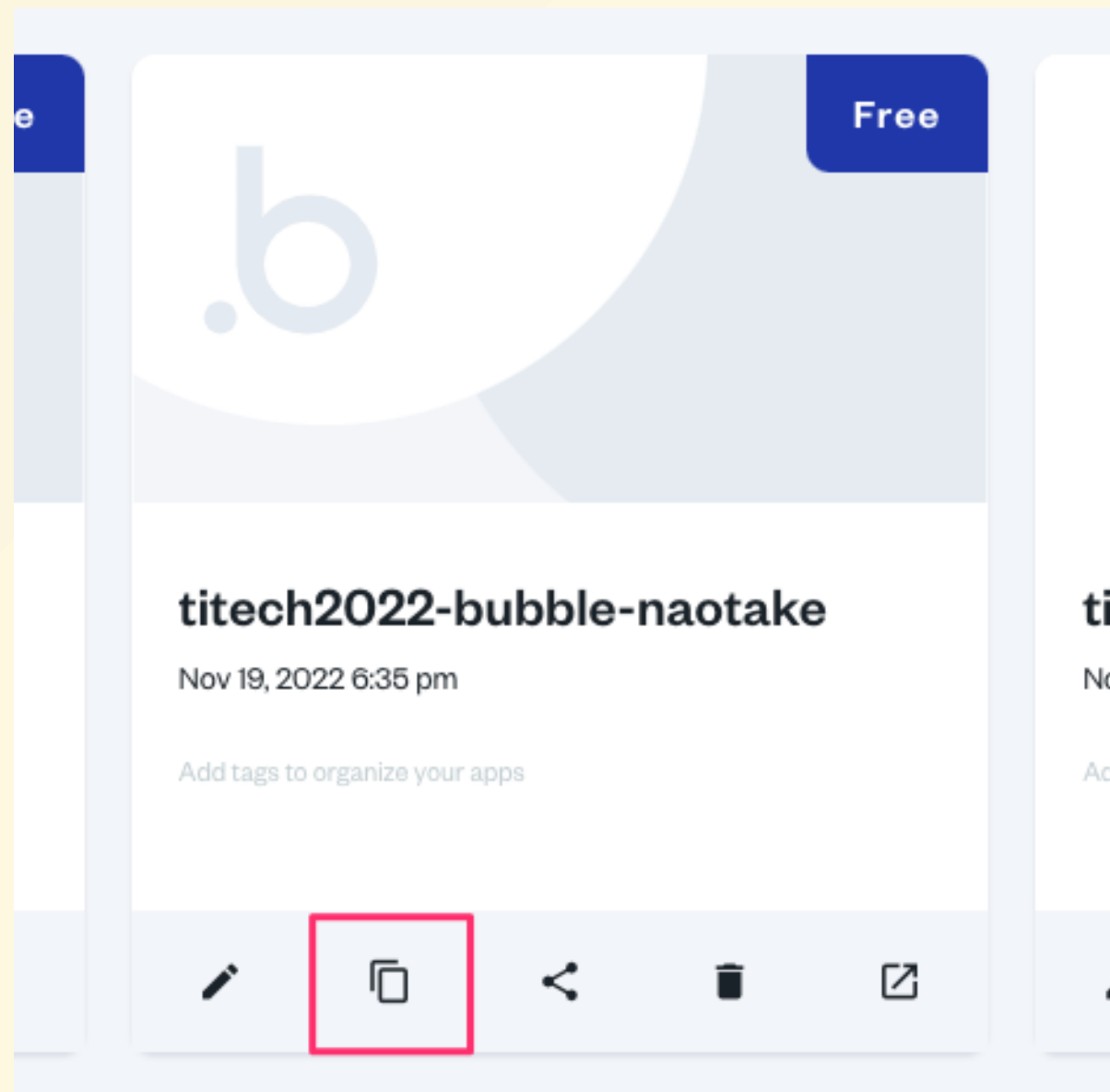
- YouTube の動画一覧を表示
- Google アカウントを使ったソーシャルログイン
- YouTube の動画一覧を動的に検索



まずは土台となるアプリケーションを新たに用意

- Bubble 1 回目の講義で使ったアプリケーションをコピーして使い回したいと思います
- <https://bubble.io/home?tab=apps> にアクセス

- 各自のアカウントで作成している Bubble アプリの一覧が表示されます
- その中で、Bubble 1 回目の講義で作成したアプリケーションの下にある COPY をクリック

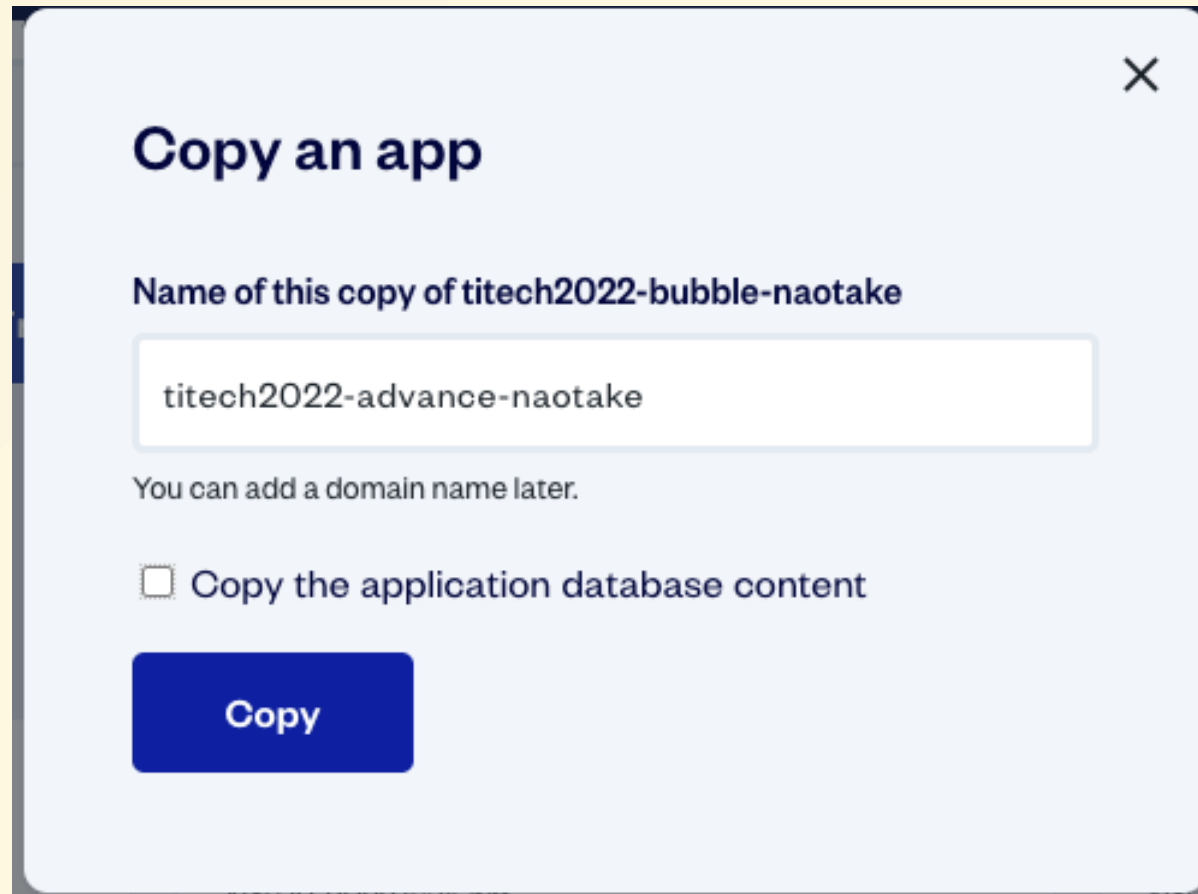


- コピー後のアプリケーション名の入力ダイアログが表示されるので、適当な名前を入力
- 以前のデータは使い回さないで

Copy the application
database content

のチェックは OFF のまま

- COPY をクリック



Copy an app

Name of this copy of titech2022-bubble-naotake

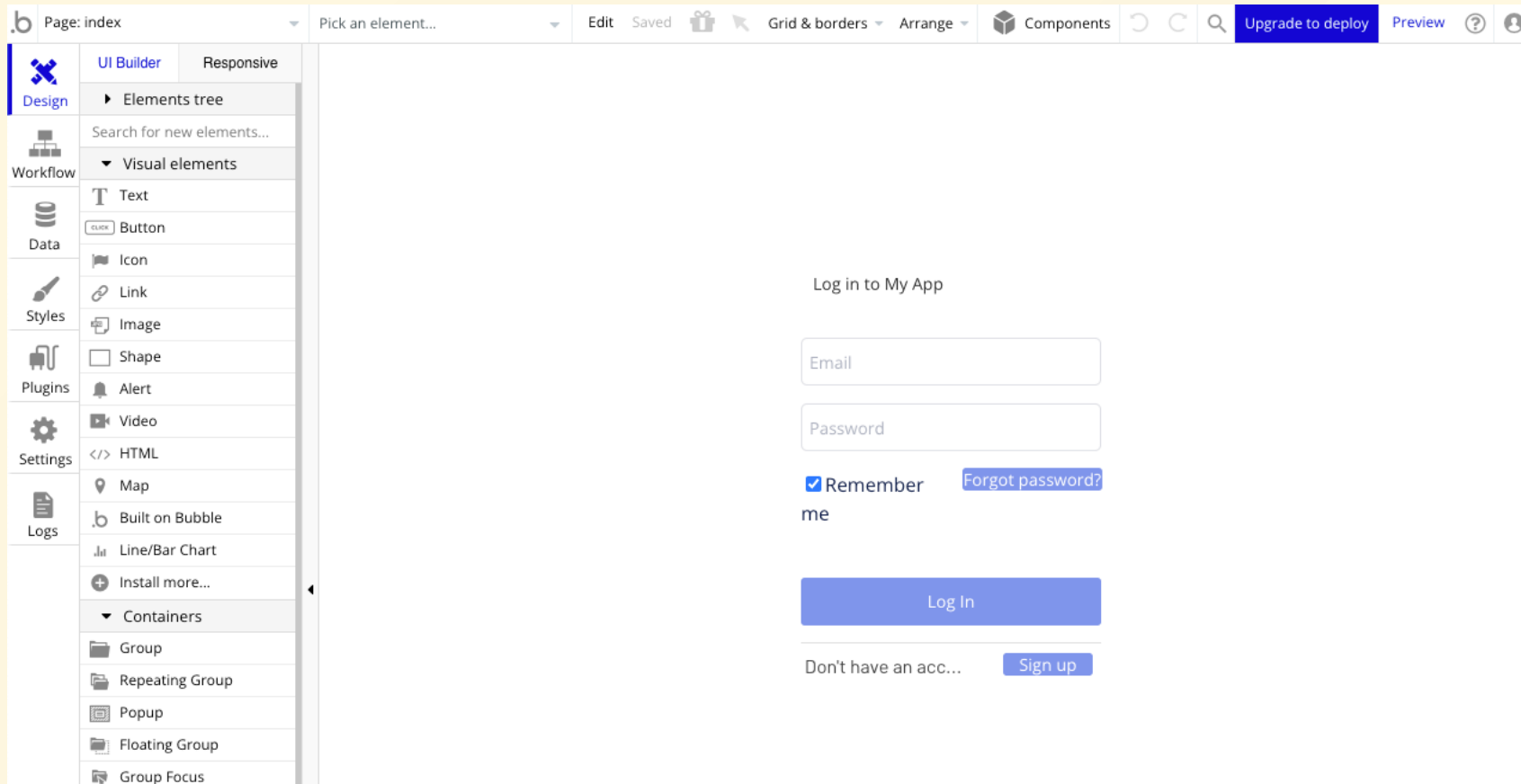
titech2022-advance-naotake

You can add a domain name later.

☐ Copy the application database content

Copy


- コピーが終わると、コピーしたアプリケーションの編集画面（index）が表示されれば OK です



続いて YouTube 連携の事前準備

- YouTube の動画一覧を表示するために必要な下準備を行います
- 今回の講義では Google が提供しているクラウドサービス "Google Cloud Platform" に用意されている各種 API を使っていきます
 - Amazon が提供している "Amazon Web Service" (AWS) の Google 版だと思ってください

- 下記 URL にアクセス
 - <https://console.developers.google.com/>
- ご自身の Google アカウントでログインしてください
 - もし、Google アカウントをお持ちでない方は、アカウントの作成をお願いします



The image shows a Google login page for Google Cloud Platform. 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 '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: '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 it's not your own PC, use Guest mode to log in anonymously.) 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

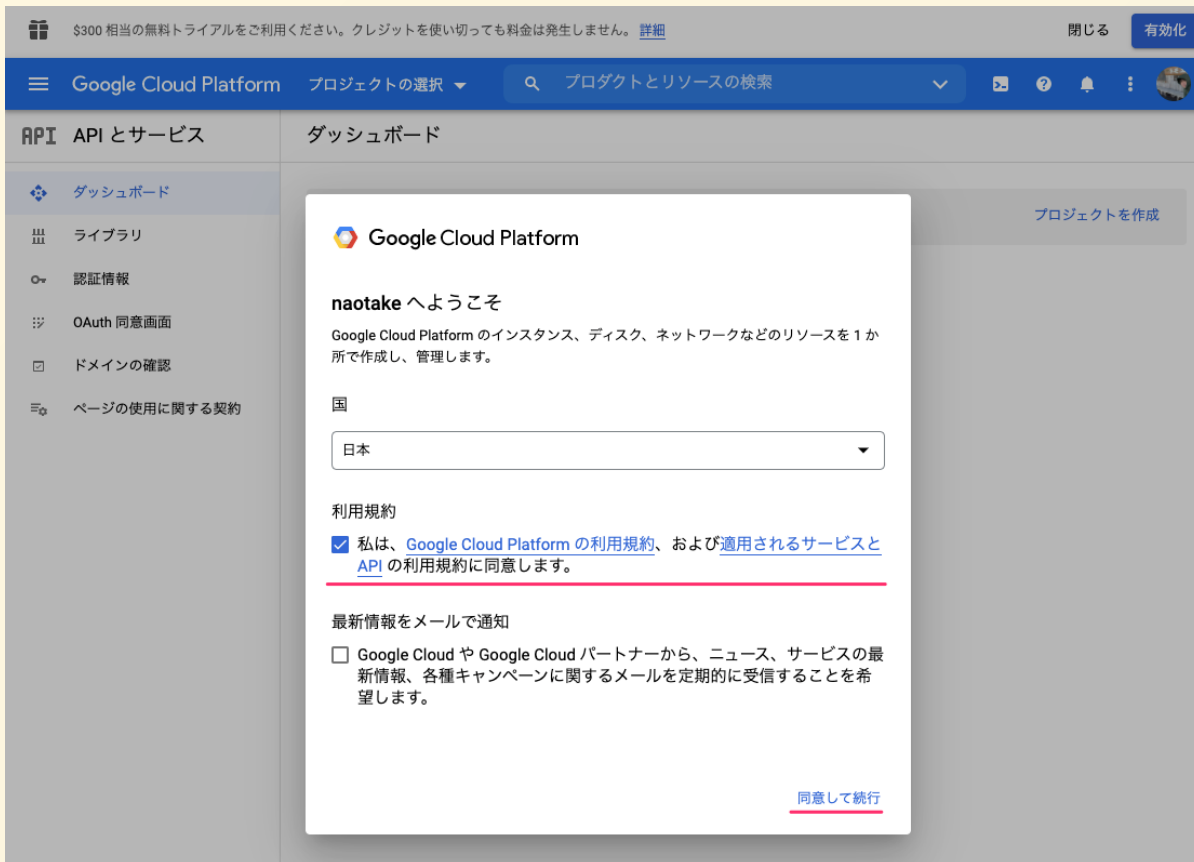
メールアドレスまたは電話番号

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](#)

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




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




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


新しいプロジェクト

 割り当て内の残りのプロジェクト数は 12 projects 件です。プロジェクトの増加をリクエストするか、プロジェクトを削除してください。[詳細](#)

[MANAGE QUOTAS](#)

プロジェクト名 *
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

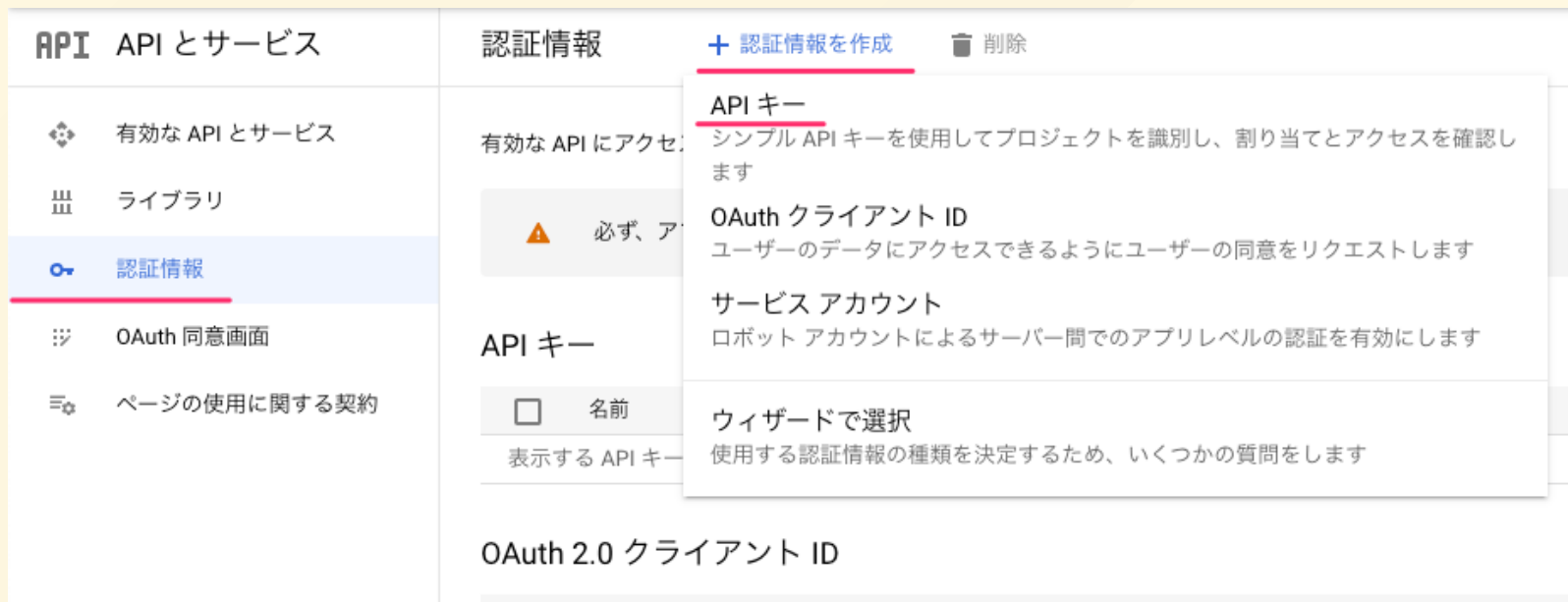
- 作成が完了すると、作成したプロジェクトのダッシュボードが表示されます
- 画面上部の Google Cloud の左隣に作成したプロジェクト名が表示されていれば OK



- 続いて YouTube の動画一覧を表示するために必要な認証キーを発行していきます
- API とサービスというメニューを開くため、画面上部の検索ボックスに "API" と入力
- サジェスト候補の中から "プロジェクトとページ" ブロックにある "API とサービス" を選択



- API とサービスページが開いたら、左メニューから「認証情報」を選択
- 右パネルが認証情報に変わるので、画面上部の「+ 認証情報を作成」をクリック
- ポップアップから「API キー」を選択





- すると API キー作成完了のポップアップが表示されるので、その値を控えておきます
- キーを制限しろと言われますが、講義の最後にこれらのキーを削除しますので一旦このまま行きます


API キーを作成しました

アプリケーション内で使用するには、このキーを `key=API_KEY` パラメータと一緒に渡します。

自分の API キー

AI 



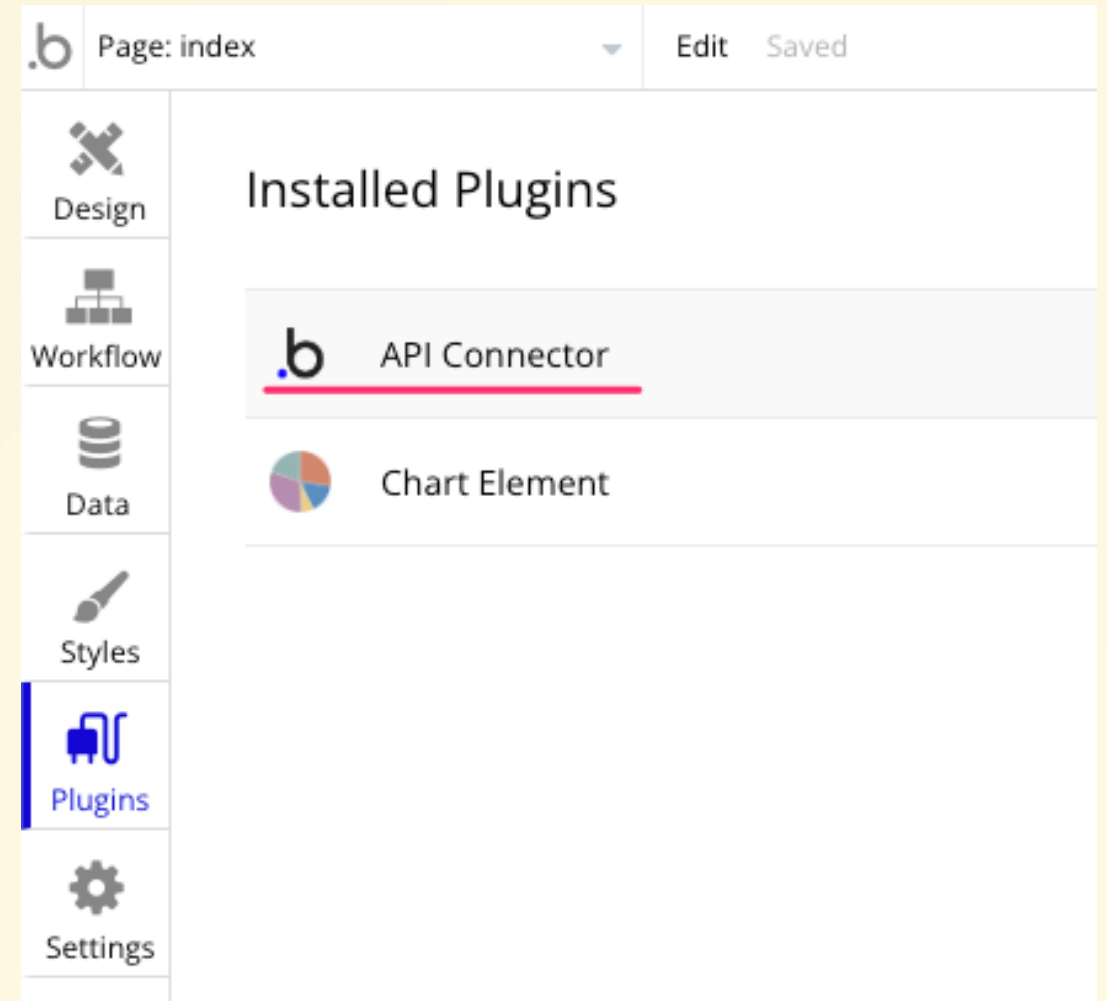
 このキーに制限はありません。不正使用を防ぐため、使用できる場所と対象の API を制限することをおすすめします。制限を追加するには、[API キーを編集](#)します。 [詳細](#)

閉じる

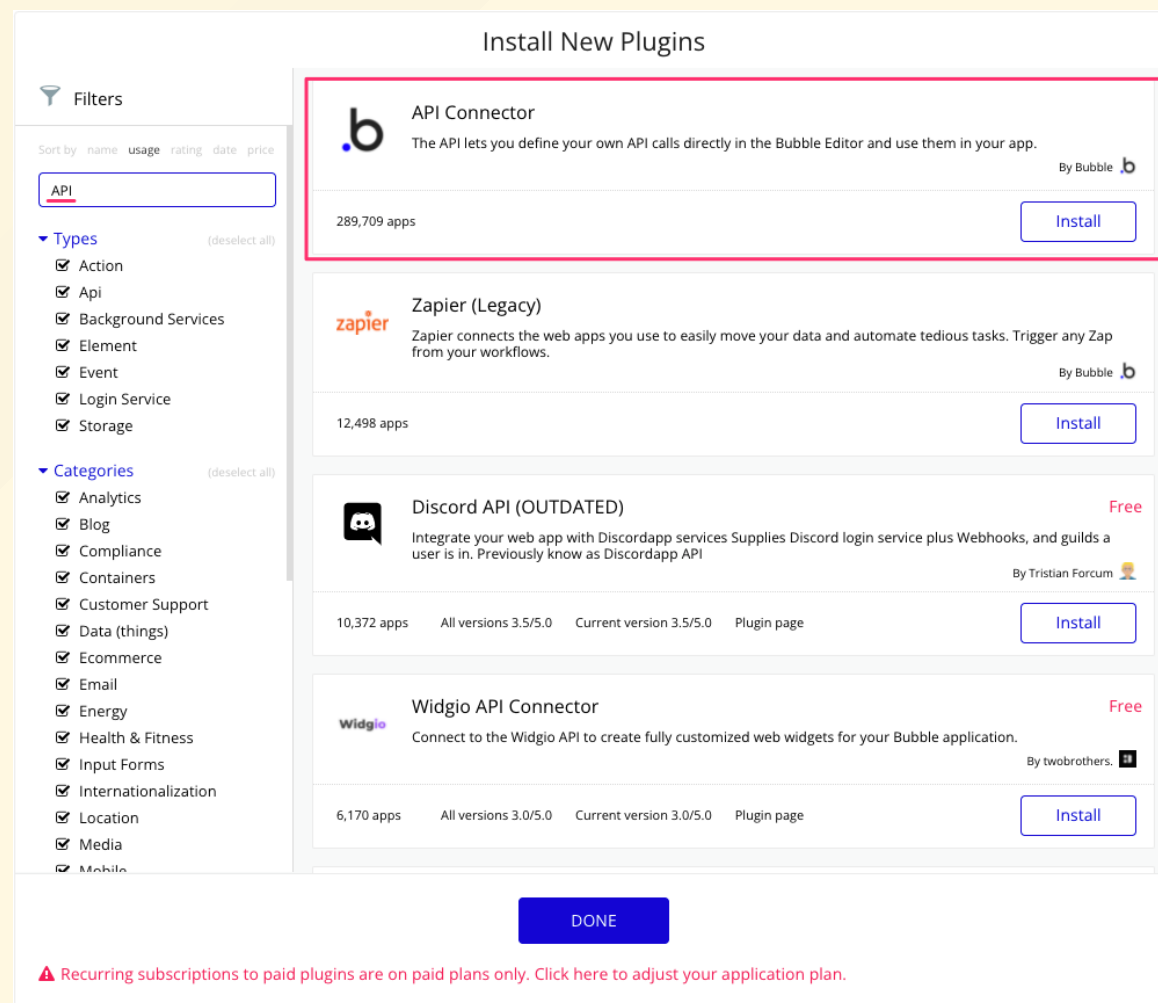
- これで Bubble から YouTube の一覧を取得するためのキーが発行できたので Bubble 側の設定を行っていきます
- まずは Bubble から YouTube の一覧を API を介して取得するための準備を行います

Bubble の API 連携の準備

- 左メニューから Plugins を選択し、Installed Plugins の中に **API Connector** が既に存在していることを確認

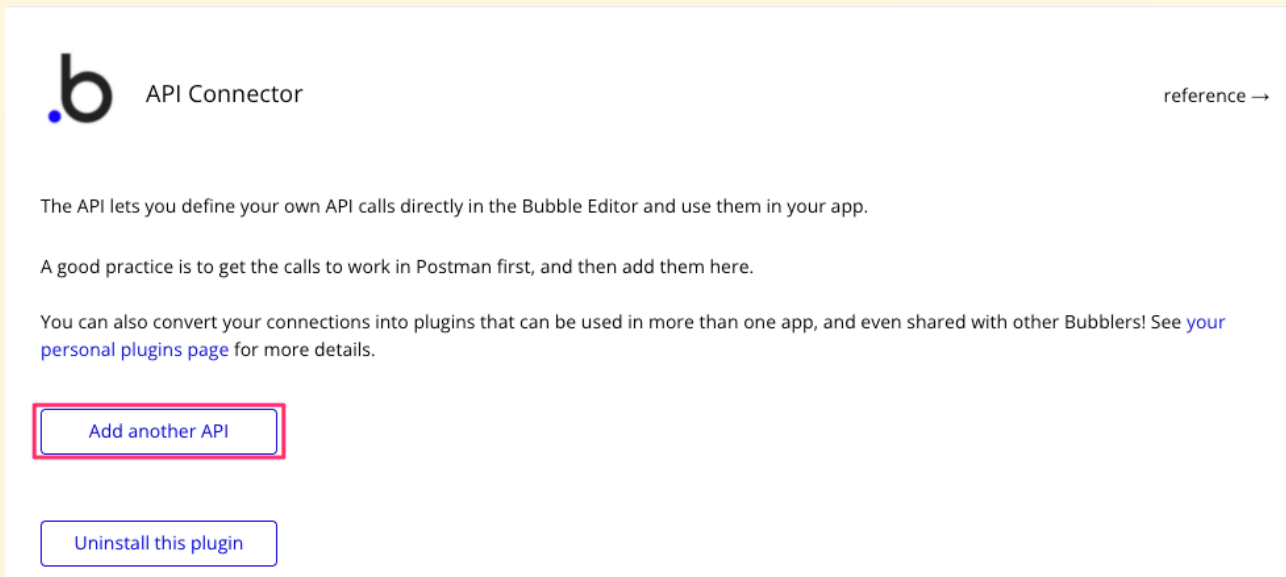


- もし含まれていない人は
Add plugins ボタンをクリック
- "API" で検索し、
API Connector というプラグインをインストールします



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

- それでは YouTube API を使うための設定を行っていきます
- Plugins タブで Installed Plugins の中から "API Connector" を選択
- 右パネルにこのプラグインの概要が記載されているので、その下にある "Add another API" をクリックし、新しい API 連携の設定を開始します



- "Add another API" をクリックすると、このような API 設定部品が表示されると思います
- 各項目の簡単な説明と設定をしていきます

The screenshot displays a web-based API configuration interface. At the top right, there is a "collapse" link. The main configuration area includes:

- API Name:** A text input field containing "New API".
- Authentication:** A dropdown menu currently set to "None or self-handled", accompanied by a trash icon.
- Shared headers for all calls:** A section with an "Add a shared header" button.
- Shared parameters for all calls:** A section with an "Add a shared parameter" button.
- API Call List:** A table with one visible entry:

Name
API Call

There is an "expand" link at the top right of this section.
- Import another call from cURL:** A section with an "Add another call" button.

At the bottom right, there is a link labeled "expand all calls".

1. API Name: この API 連携のグループ名

- 今回は YouTube

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

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

- 今回は "None or self-handled"

The screenshot shows a configuration panel for a group of APIs. It includes fields for API Name (YouTube) and Authentication (None or self-handled). Below these are sections for shared headers and parameters, each with an 'Add' button. At the bottom, there is a section for 'API Call' with a name field (API Call) and a button to 'Add another call'.

① API Name ② Authentication  collapse

③ Shared headers for all calls

④ Shared parameters for all calls

expand

Import another call from cURL

expand all calls

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

- 今回は特に指定しません

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

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

- 今回は特に指定しません

The screenshot shows a configuration panel for an API group. It includes fields for API Name (YouTube) and Authentication (None or self-handled). Below these are sections for shared headers and shared parameters, each with an 'Add' button. At the bottom, there is a section for 'API Call' with a name field (API Call) and a button to 'Add another call'.

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

- 続いて、このグループに対して具体的な API の内容を設定していきます
- 先ほどの画面で "API Call" となっているブロックの右側に expand というリンクがあるのでそれをクリック

The screenshot shows a web interface for configuring API calls. At the top, there's a header bar with 'API Name' set to 'YouTube', 'Authentication' set to 'None or self-handled', and a 'collapse' link. Below this, there are sections for 'Shared headers for all calls' and 'Shared parameters for all calls', each with an 'Add' button. The main section is titled 'API Call' and contains a 'Name' field with the value 'API Call'. To the right of this field is a red 'expand' link. At the bottom, there's a section for 'Import another call from cURL' with an 'Add another call' button and an 'expand all calls' link.

API Name: YouTube 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

- すると、具体的な API の設定項目が表示されますので、簡単に説明します

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: 具体的な API の名前

- 今回は "search" と入力します

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 marked with a red circle and the number 3, containing an "Add header" button.
- Parameters:** A section marked with a red circle and the number 4, containing two parameter rows:
 - Key: "key", Value: "AlzaSyAA3b34Zbc9rzxnw.", Private: ☒, Allow blank: ☐, Optional: ☐.
 - 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 triangle icon followed by the text "You need to initialize this call before it will work."
- Footer:** A link "Manually enter API response" next to the "Initialize call" button.

2. Path: 具体的な API の URL

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

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)
 - Parameter 1: `key` (Key), `AlzaSyAA3b34Zbc9rxnw.` (Value), `Private` (checked), `Allow blank` (unchecked), `Optional` (unchecked)
 - Parameter 2: `q` (Key), `おしゃべり唐揚げ` (Value), `Private` (checked), `Allow blank` (unchecked), `Optional` (unchecked)
 - `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` and `Manually enter API response`

3. Headers: この API 固有のヘッダー情報

- 今回は特に指定しません

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

- 1** Name: search
- 2** GET https://www.googleapis.com/youtube/v3/search (use [] for params)
- 3** Headers section with an "Add header" button.
- 4** Parameters section with two entries:
 - Key: key, Value: AlzaSyAA3b34Zbc9rxnw, Private: ☒, Allow blank: ☐, Optional: ☐
 - Key: q, Value: おしゃべり唐揚げ, Private: ☒, Allow blank: ☐, Optional: ☐
- Buttons: "Add parameter", "Initialize call", and "Manually enter API response".
- Options: "Include errors in response and allow workflow actions to continue" and "Capture response headers", both with unchecked checkboxes.
- Warning: "⚠ You need to initialize this call before it will work."
- Top right: "collapse" link and a trash icon.

4. Parameters: この API 固有のパラメータ情報

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

- すべて設定するとこんな感じです

collapse

①

Name

search

Use as

Data

Data type

JSON

②

GET

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

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

- おそらく Status code 403 のエラーメッセージが返ってきたはずです
 - 内容は書いてある通りですが、要約するとこんな感じ
です
- “ YouTube Data API v3 はあなたのプロジェクトで無効になっています。
この URL にアクセスして有効にしてから再試行してください。

”

API Connector

There was an issue setting up your call.

Raw response for the API
Status code 403

```
{
  "error": {
    "code": 403,
    "message": "YouTube Data API v3 has not been used in project 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",
        "link": "https://console.developers.google.com/apis/api/youtube.googleapis.com/overview?project=298727815636"
```

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

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



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 [🔗](#)

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

The screenshot shows the Google Cloud Console interface for the YouTube Data API v3. The left sidebar contains navigation links for 'API とサービス', '有効な API とサービス', 'ライブラリ', '認証情報', 'OAuth 同意画面', 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 with columns for 'サービス名', 'タイプ', and 'ステータス'. The status is '有効'. A red line is drawn under the 'API を無効にする' link in the top navigation bar. Below the table, there are tabs for '指標', '割り当て', and '認証情報', and a section for 'グラフ' with a dropdown menu and a '30 日' filter.

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

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

Returned values - search

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

kind <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

SAVECancel

- これは今回設定した
"<https://www.googleapis.com/youtube/v3/search>" の
API を実行した結果（レス
ポンス）の情報を表してい
ます

Returned values - search

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

kind <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>UCcRigzi_JBAZh_UySjv8xuw</small>	text
id videoId <small>SO94aNshTSU</small>	text

SAVECancel

詳しくは割愛しますが、ポイントとしては下記 3 つです。

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

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

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

Returned values - search

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

kind <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>WahficVNgYipX-I3rjRIPUrHi10</small>	Ignore field ▼
id kind <small>youtube#channel</small>	Ignore field ▼
id channelId <small>UCcRigzi_jBAZh_UySjv8xuww</small>	Ignore field ▼
id videoId <small>SO94aNshTSU</small>	text ▼

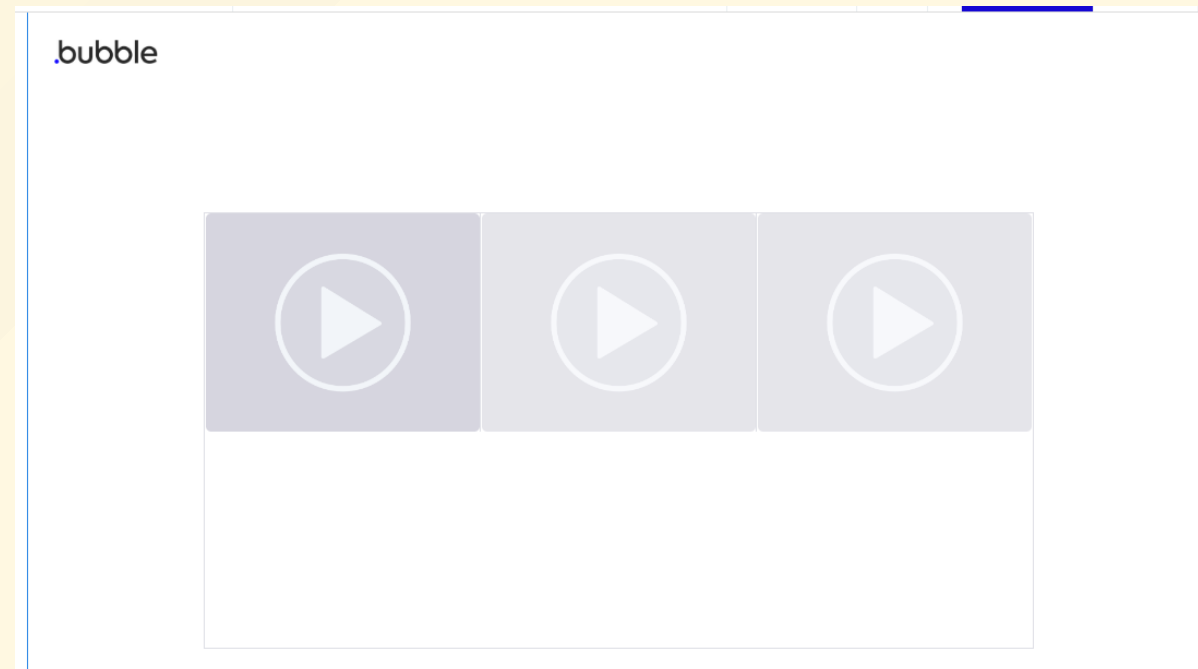
SAVE

Cancel

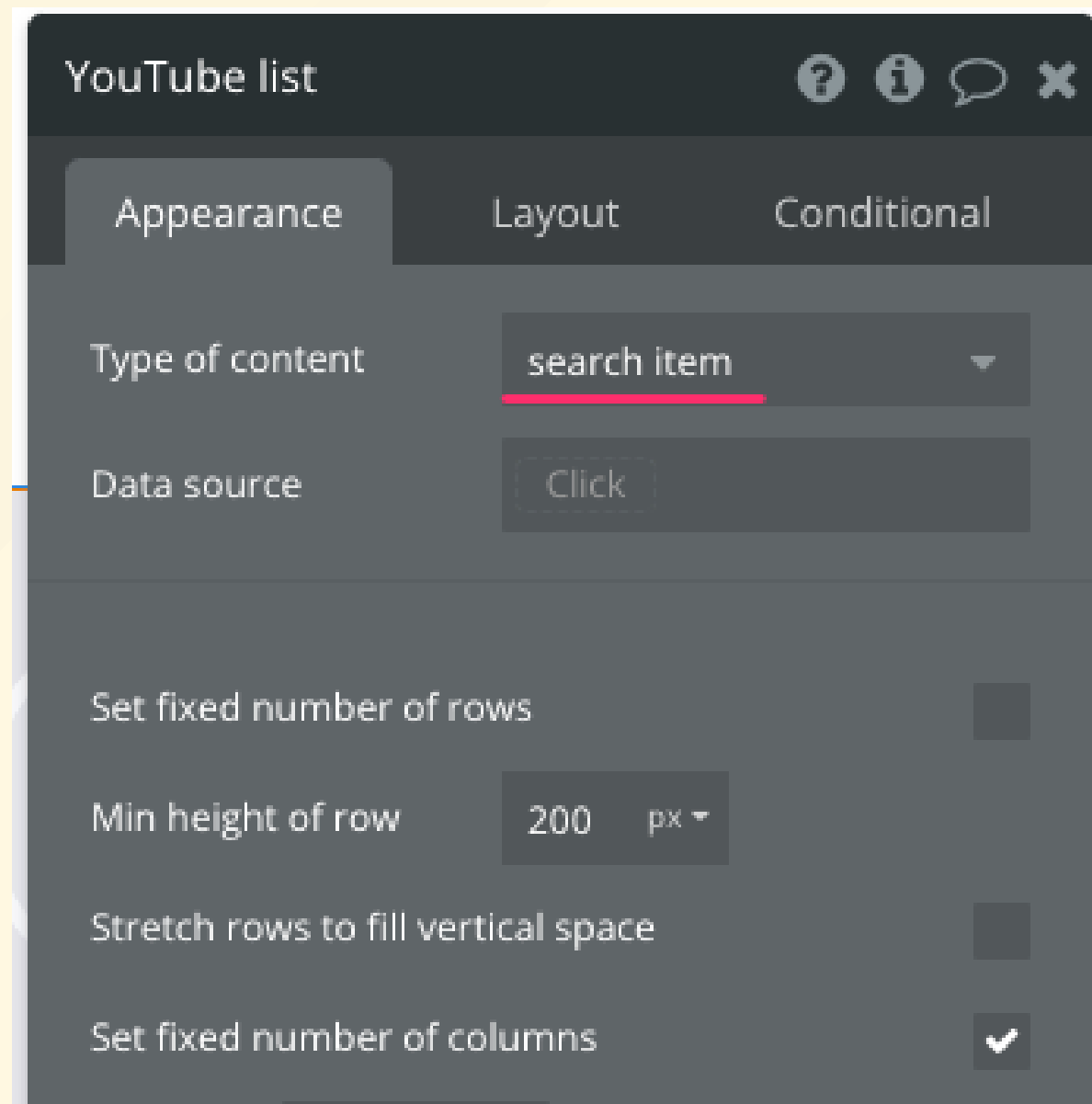
画面表示の準備

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

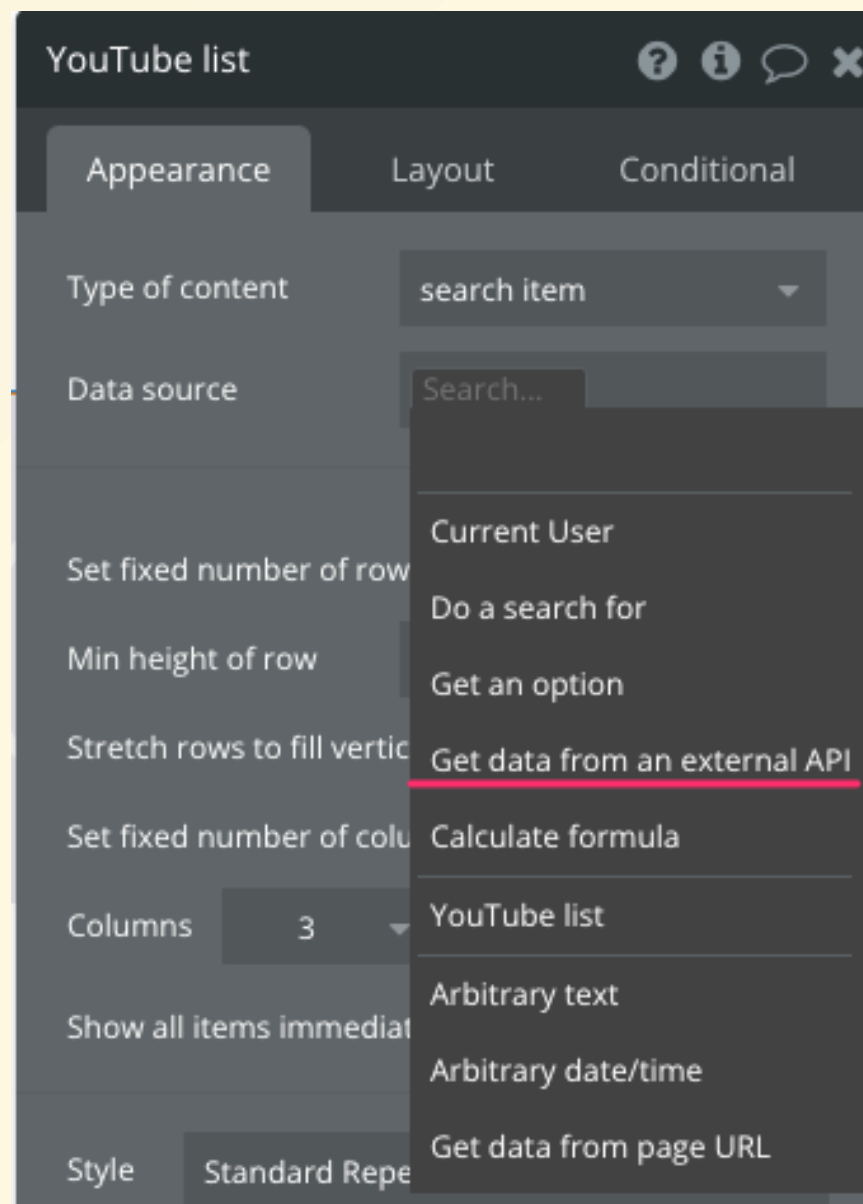
- pet_list ページを clone する形で "video_list" 画面を作成
- Repeating Group の各セルの中身を全て削除
- 代わりに Visual elements の Video をセル内いっぱいになるよう設置
- こんな感じですね



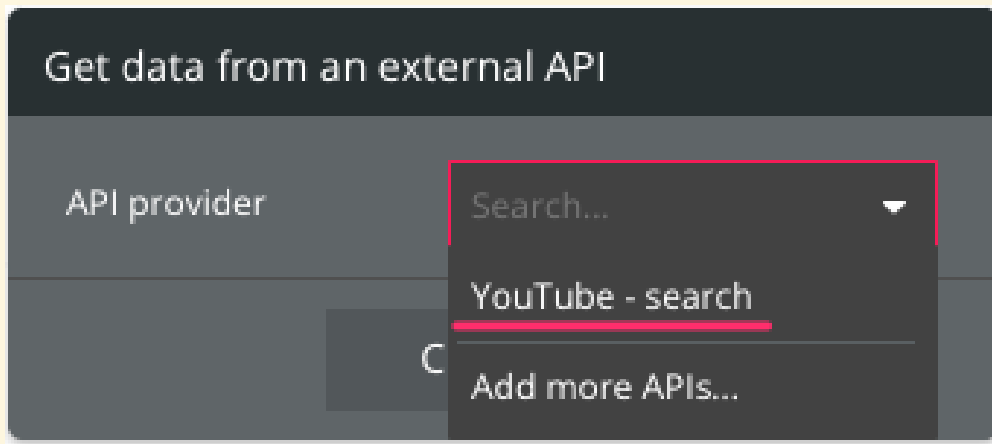
- レイアウトが出来たので実際に YouTube の一覧表示を設定していきます
- まず Repeating Group の Type of content は、先ほど API 実行時に確認した `search item` 型となります
- これで繰り返すデータの種類の YouTube の動画 1 つ 1 つとなります



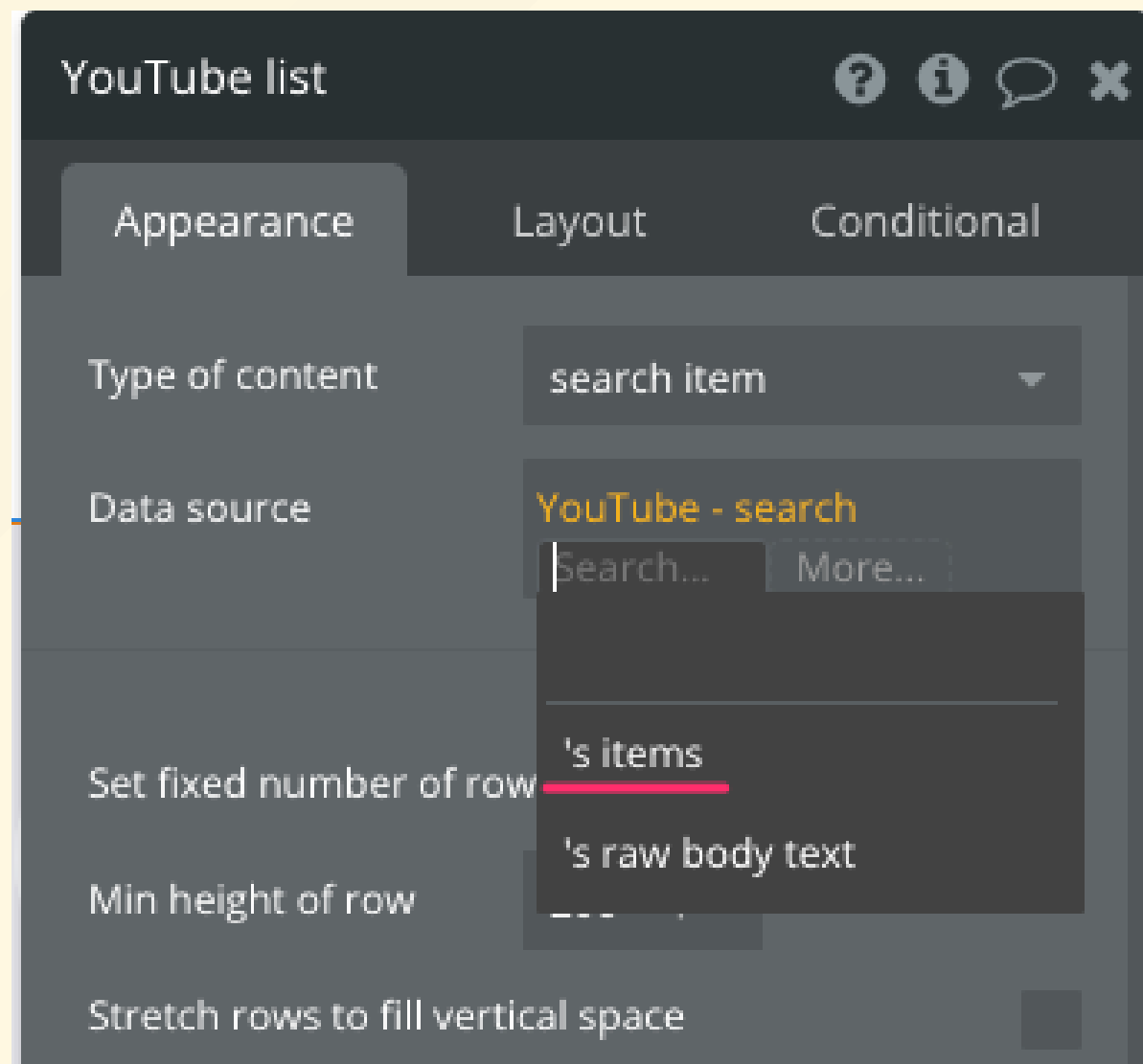
- 次に Data source ですが、今回は API から取得した YouTube の動画一覧を対象とします
- Click から候補を見てみると
Get data from an external API
という項目があると思うのでそちらを選択



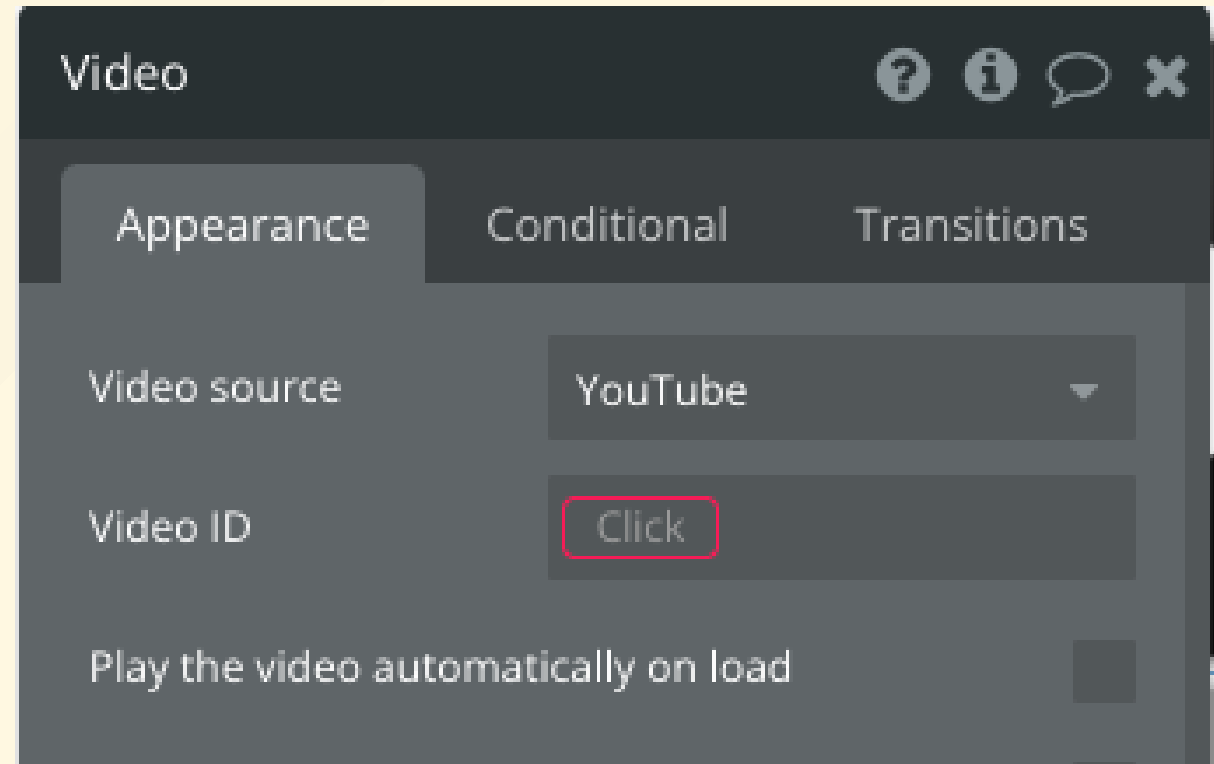
- 隣に Get data from an external API のポップアップが表示されるので、API provider のプルダウンを選んでください
- すると、プルダウンの中に先ほど設定した **YouTube - search** の候補があると思いますのでそれを選択してください
 - ハイフンの前が API のグループ名で、後ろが具体的な API の名前です



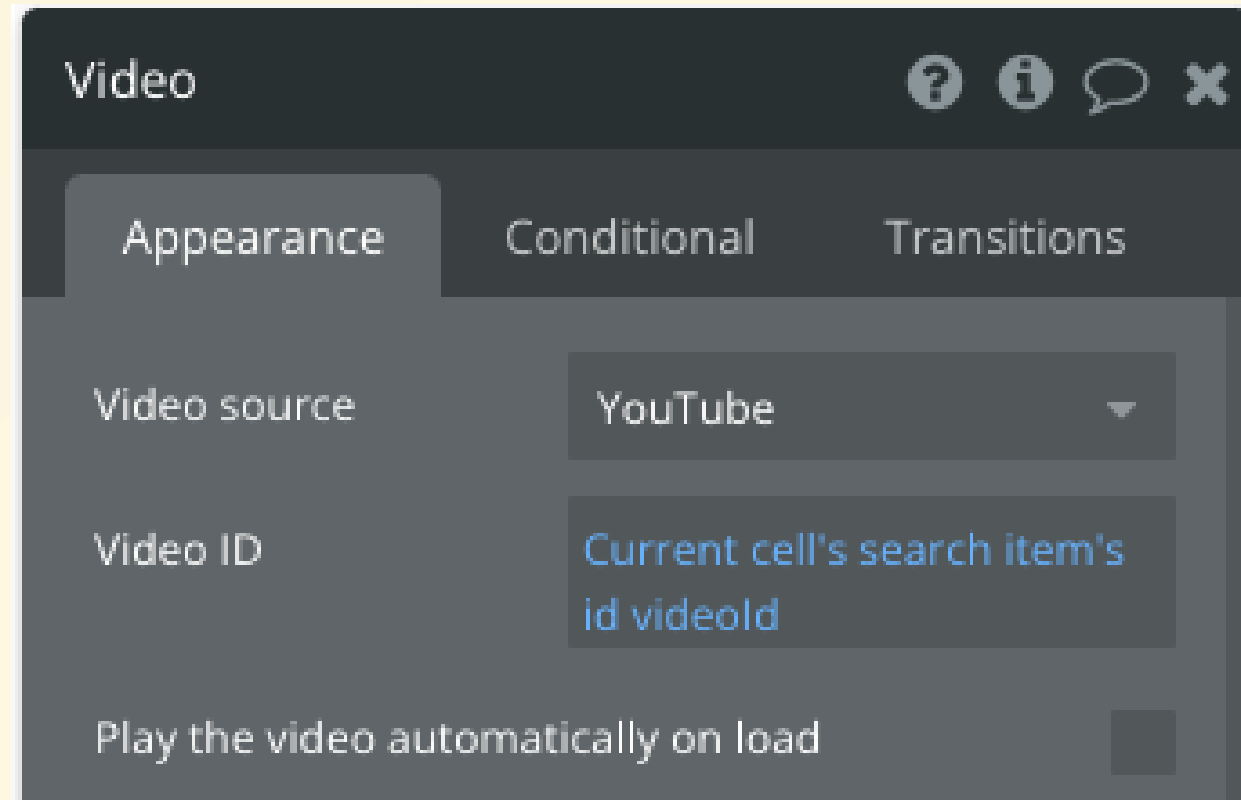
- API Provider が設定できたら、CLOSE で閉じます
- そして元の Repeating Group の Data source の選択として、さらに `'s items` を選択します
- これで、API の結果に含まれる動画一覧 (items) を繰り返し表示する、という指定ができました



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



- こんな感じですね



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

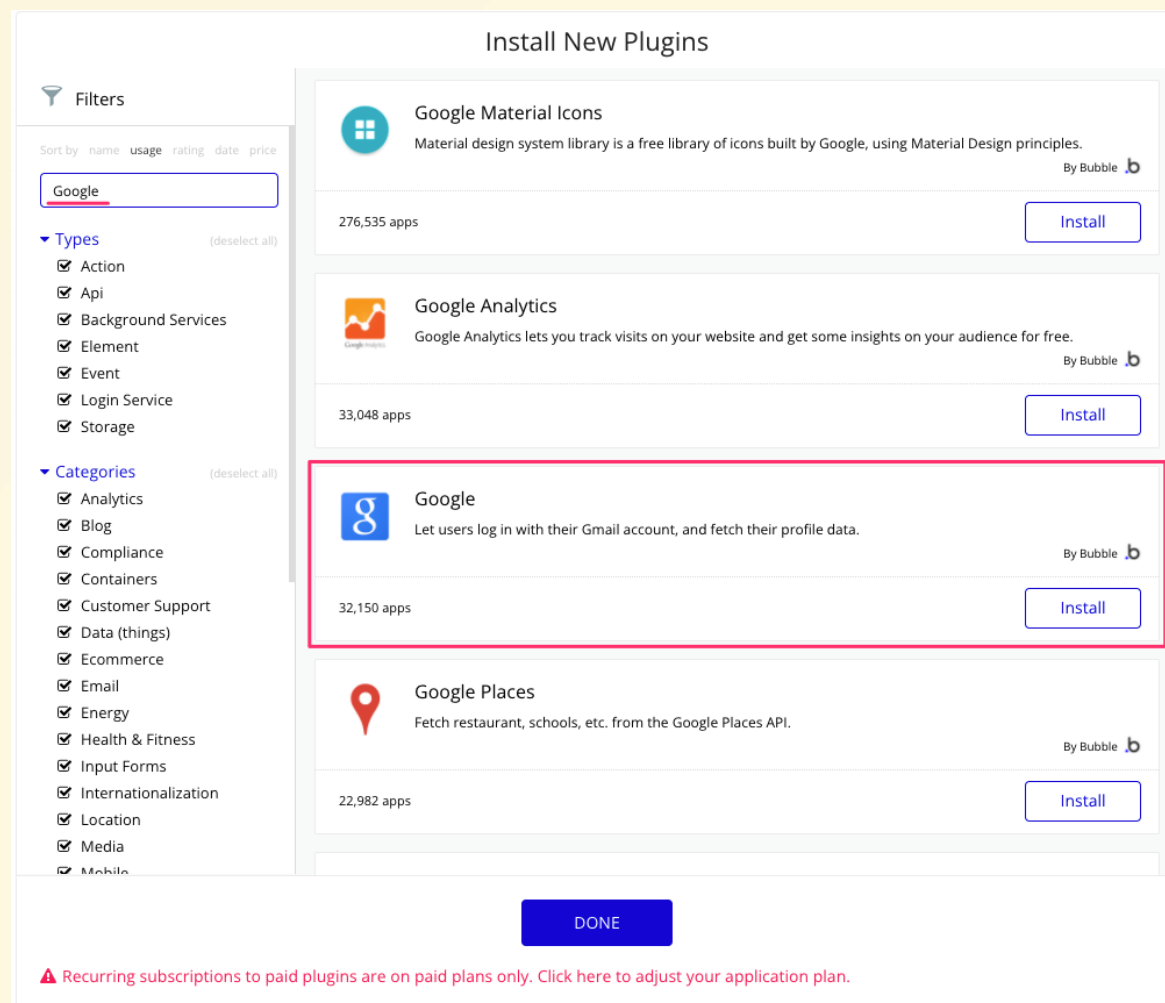
.bubble

Log in



Google ログインを試してみよう

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



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

Google へのログイン後にアプリに戻ってくるための認証準備

- Google Cloud Platform 画面にアクセス

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

- 先程と同じ API とサービスの画面を表示

- プロジェクトを選択し、左メニューから "OAuth 同意画面" を選択



The screenshot shows the Google Cloud console interface. The top navigation bar includes the Google Cloud logo, the project name 'bubble-naotake', a search bar, and various utility icons. The left sidebar menu is expanded, showing options like 'API とサービス', 'ライブラリ', '認証情報', 'OAuth 同意画面' (which is highlighted), 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 on how to configure the consent screen, including a 'User Type' section with radio buttons for '内部' (Internal) and '外部' (External). A '作成' (Create) button is visible at the bottom of the configuration area.

Google Cloud bubble-naotake

検索 プロダクト、リソース、ドキュメ...

API API とサービス

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ページの使用に関する契約

OAuth 同意画面

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

User Type

☐ 内部 ?

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

☐ 外部 ?

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

作成

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

- User Type に "外部" を選択して作成をクリック

API	API とサービス	OAuth 同意画面
🔗	有効な API とサービス	
📖	ライブラリ	
🔑	認証情報	
🔗	OAuth 同意画面	
📄	ページの使用に関する契約	

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

User Type

☐ 内部 ?

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

☒ 外部 ?

Google アカウントを持つすべてのテストユーザーが使用できます。アプリはテストモードで起動し、アプリを使用できるのは、テストユーザーのリストに追加されたユーザーに限られます。アプリを本番環境に移す準備ができれば、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](#)

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](#)。

- 次ページで必要事項を入力していきます

- アプリ情報

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

アプリ登録の編集

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

アプリ情報

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

アプリ名 *

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。

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

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 *

[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 *

[kyougoku2bubble@gmail.com](#) ✕

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

保存して次へ

キャンセル

- スコープは特に設定変更せず "保存して次へ"

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

アプリ登録の編集

✓ OAuth 同意画面

 —

✓ スコープ

 —

3 テストユーザー

 —

4 概要

テストユーザー

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

+ ADD USERS

≡ フィルタ

 プロパティ名または値を入力

?

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

保存して次へ

 キャンセル

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

The screenshot shows the 'API & Services' page in Google Cloud Platform. The left sidebar has a menu with 'API & Services', 'Effective APIs and Services', 'Library', 'Authentication' (selected), 'OAuth consent screen', and 'Agreements for page use'. The main content area is titled 'Authentication' and has a '+ Create authentication info' button and a 'Delete' button. Below this, there are sections for 'Effective APIs' and 'API Key'. The 'API Key' section has a table with columns 'Name' and 'API key'. The 'OAuth 2.0 Client IDs' section has a table with columns 'Name' and 'Created'. A dropdown menu is open from the '+ Create authentication info' button, showing options: 'API Key', 'OAuth Client ID' (highlighted), 'Service account', and 'Wizard selection'. The 'OAuth Client ID' option is underlined in red. Below the dropdown, there is a message 'No OAuth client IDs are displayed'.

API とサービス	
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	

認証情報	
+ 認証情報を作成 削除	
有効な API にアクセス	
API キー	
<input type="checkbox"/>	名前
<input type="checkbox"/>	API キー
OAuth 2.0 クラ	
<input type="checkbox"/>	名前
作成日 ↓	
表示する OAuth クライアントがありません	

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

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

← OAuth クライアント ID の作成

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

アプリケーションの種類 *
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

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

```
https://{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。

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

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

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

発行した認証情報を Bubble 側に設定

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

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://favorites-kyogoku.bubbleapps.io/api/1.1/oauth_redirect) ☒

App Secret - dev.

App ID/API Key - dev.

Plugin content

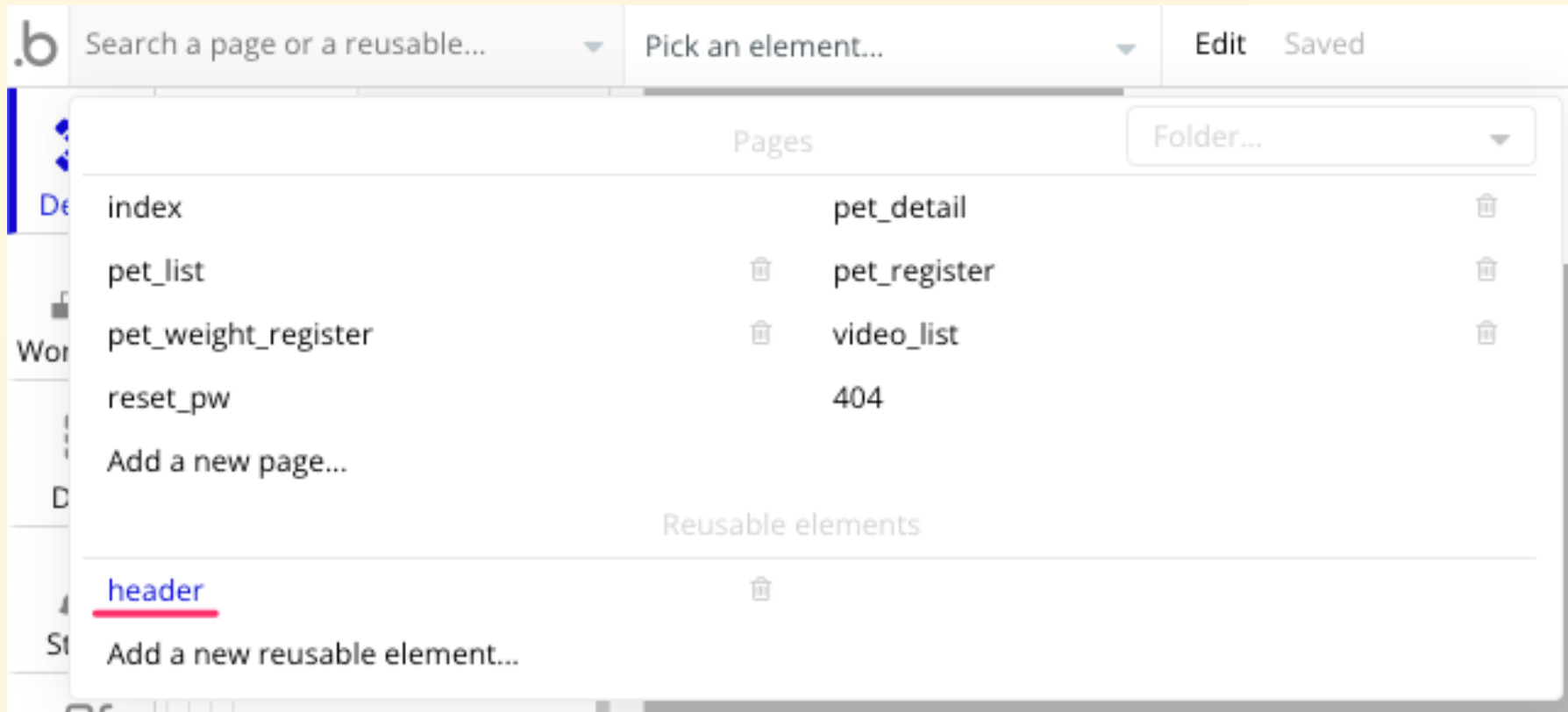
This plugin is an authentication provider.

[Uninstall this plugin](#)

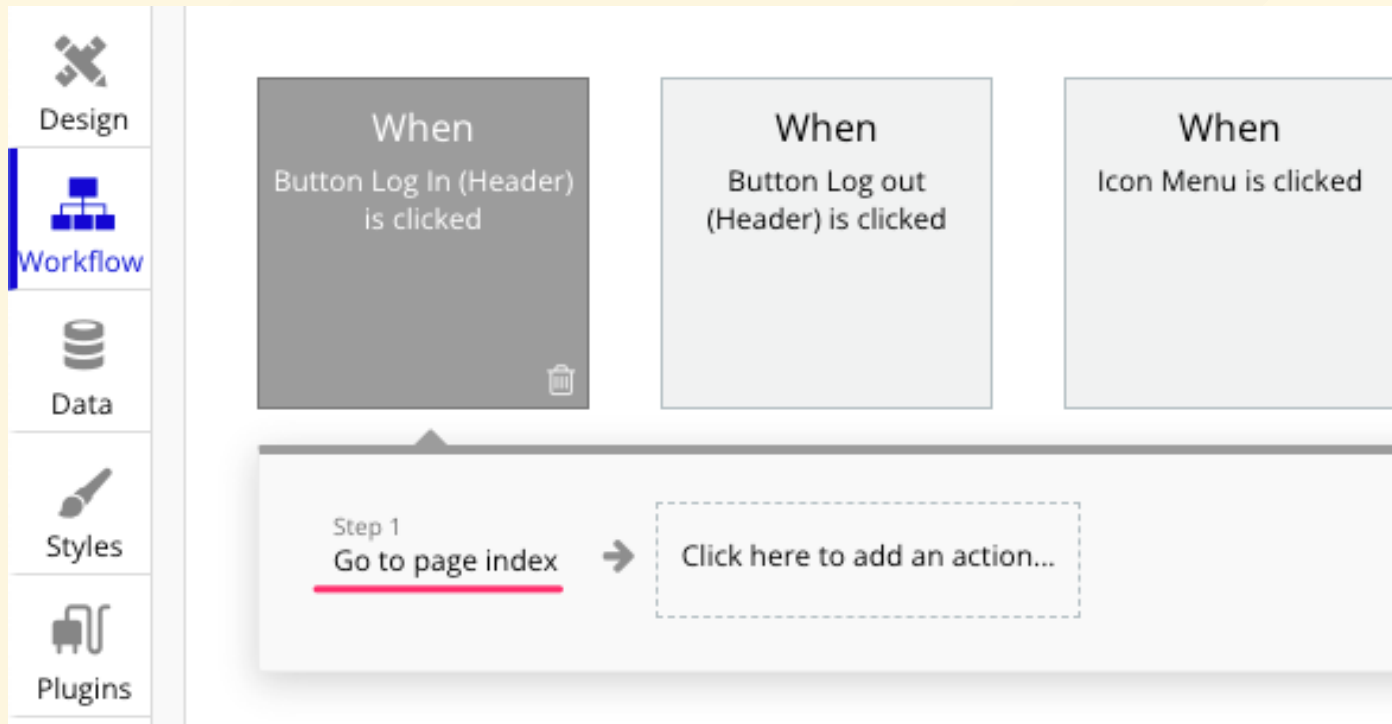
- これで事前設定は完了
- 続いてログイン機能を作り込んでいきます

ログイン機構を Google に置き換える

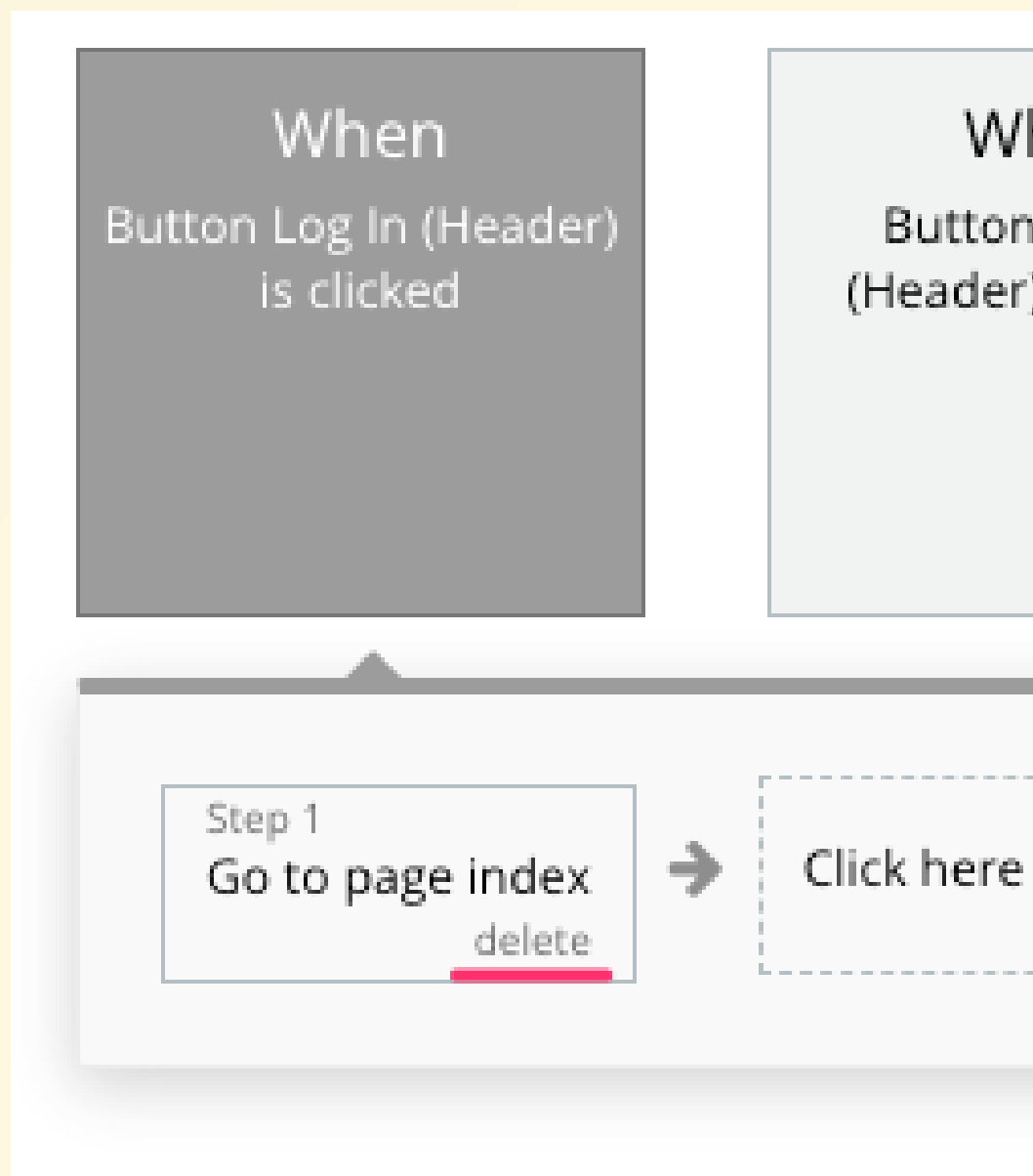
- 前回の講義で作成した Reusable elements の header を開きます



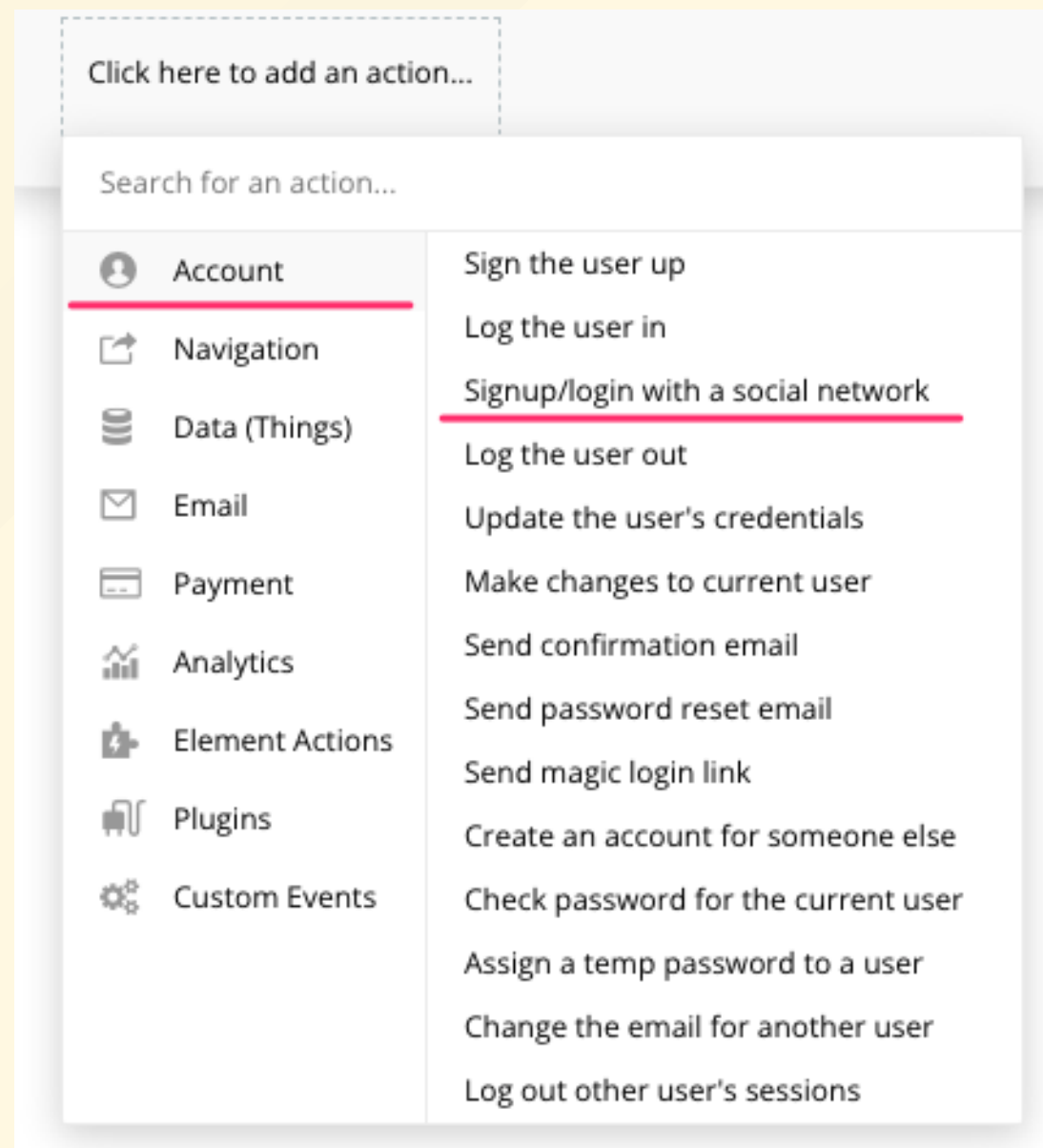
- Workflow タブを開き、ログインボタンに対するワークフローを開きます
- 今はログイン画面（index ページ）へ遷移するアクションだけが設定されているので、これを見直していきます



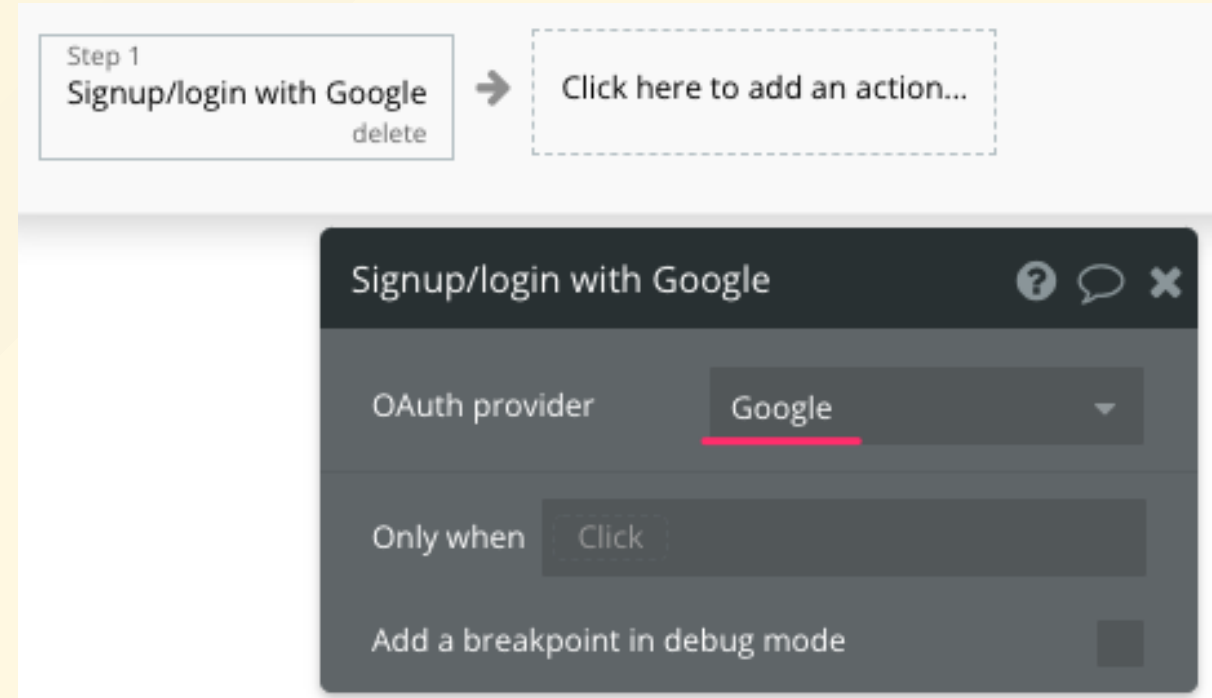
- 今回ログインは Google が用意しているログイン画面を使うことになるため、Bubble 側のログイン画面（index ページ）へ遷移するアクションは削除します



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



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



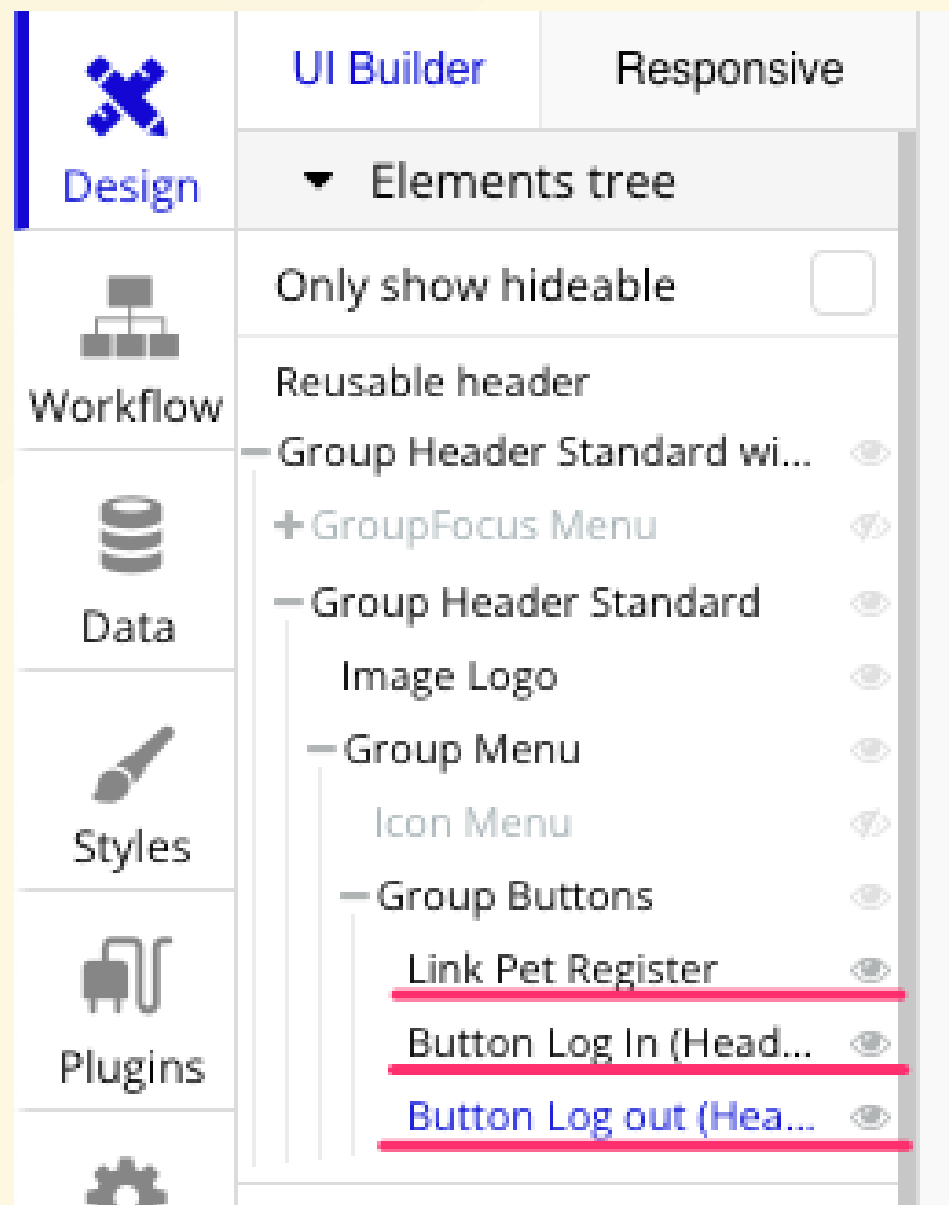
- 折角なのでログイン成功したらヘッダー上部にアカウント名と画像を表示してみましょう

Pet Register

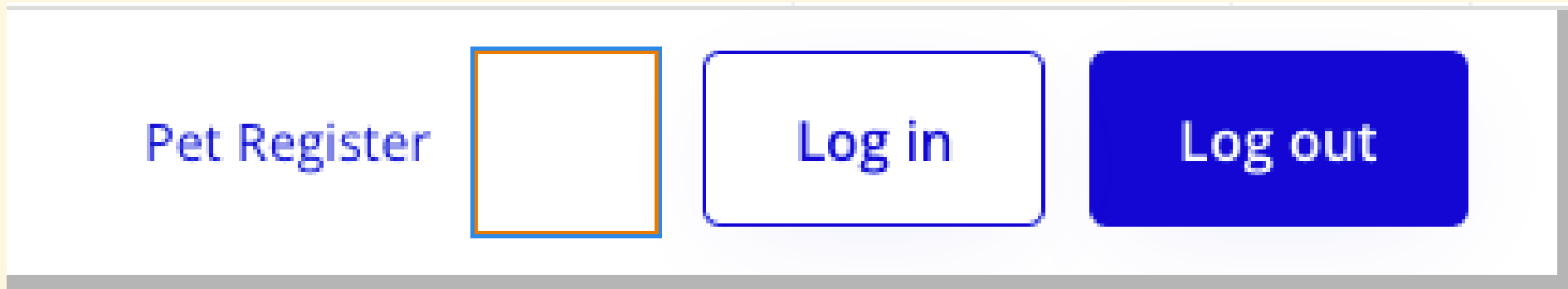


Log out

- Design タブに切り替え、現在非表示になっている下記3つのコンポーネントをElements tree から表示しておきます
 - こうしておかないと、画像追加後のイメージが分かりづらいため

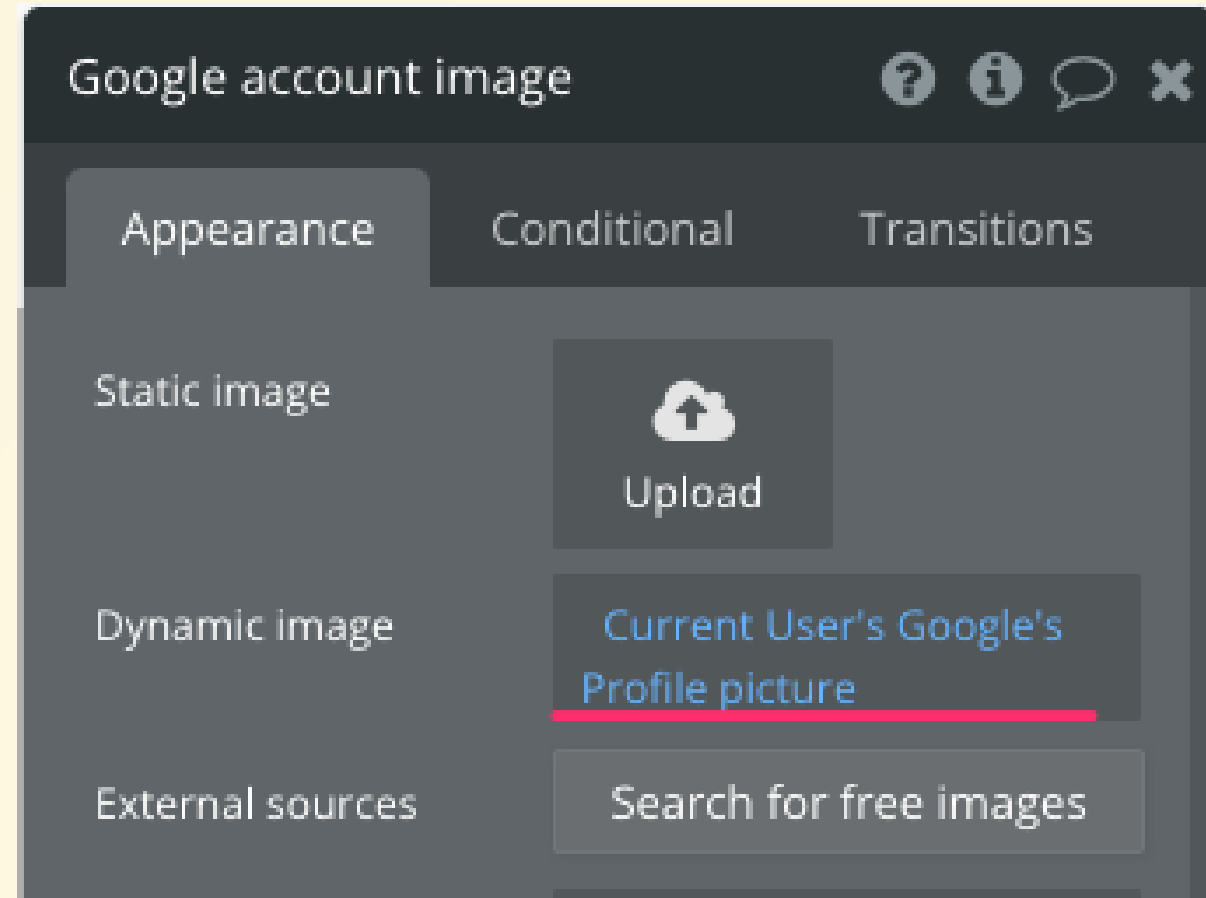


- Visual elements から Image を選択し、"Pet Register" リンクと "Log in" ボタンの間に画像を配置します
 - 縦横 50 x 50 の正方形にしておきましょう



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

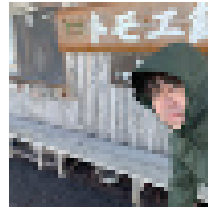
- こんな感じですね



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

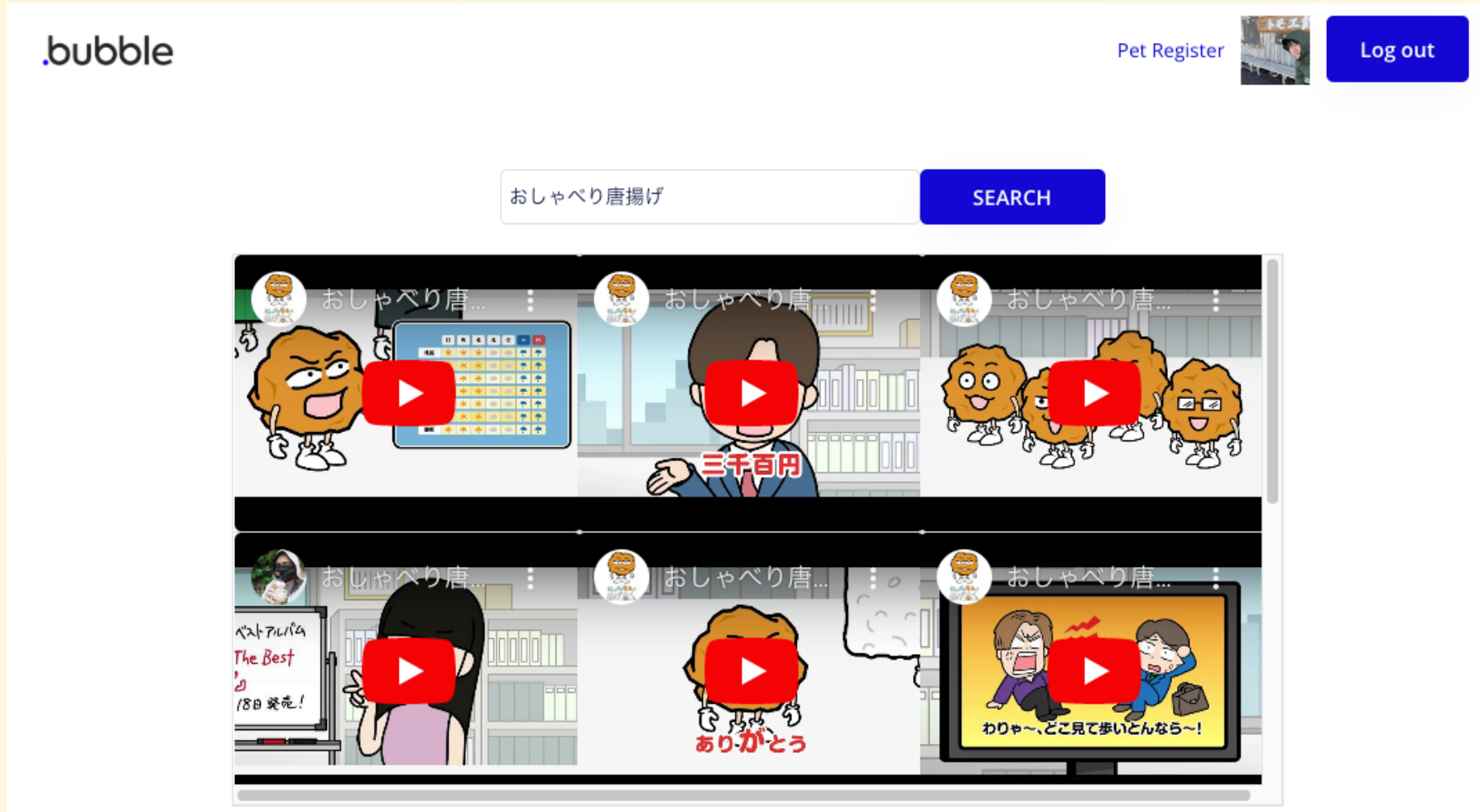
Log in

Pet Register



Log out

最後に YouTube API を使った検索機能を組み込んでみましょう



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


次のページにヒントを書いておきますのでまずはトライしてみましょう！

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



- 重要なポイントを解説します
- まずは User の型に対して **key** という field を追加します

Fields for type User

Type name 

<input type="text" value="key"/>	text	default <input type="text"/>
<hr/>		
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](#)

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

Data types

Privacy

App data

Option sets

File manager

Database views

New view

Primary 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]

- 次に YouTube API 側の設定
はこんな感じですね

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

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

The screenshot shows an API configuration window. 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' (empty value, Private unchecked), 'q' (empty value, Private unchecked), 'maxResults' (value '10', Private checked), and 'type' (value 'video', Private checked). Each row has checkboxes for 'Allow blank' and 'Optional'. 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 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 '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	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"/>

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.

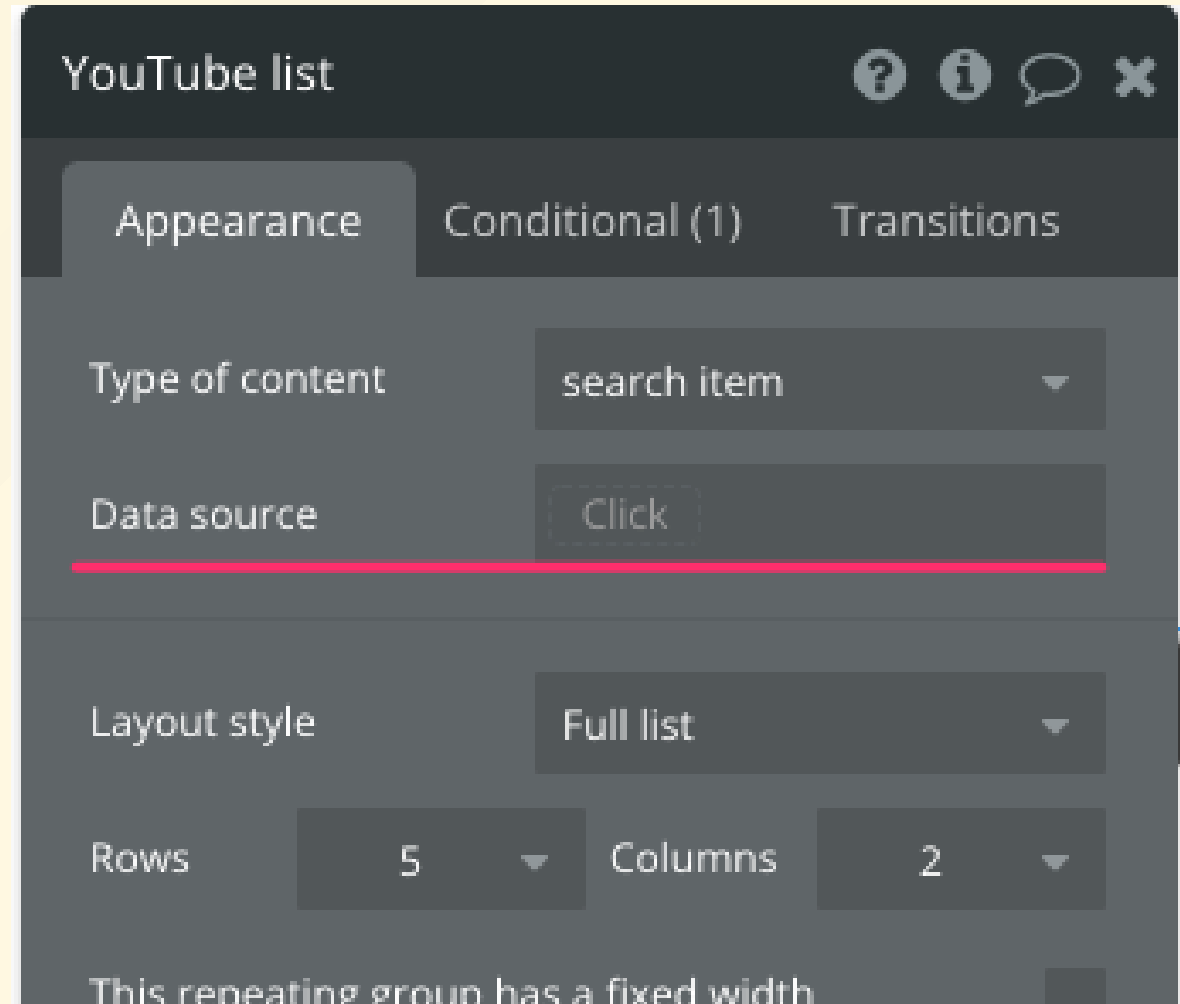
Reinitialize call Manually enter API response

- ついでに、検索のオプションを 2 つ追加しておきましょう
- key: **maxResults**、1 回の検索で取得するデータの件数
 - デフォルトは 5
- key: **type**、検索対象のデータ種類
 - **video** を指定することで、動画だけの検索が可能

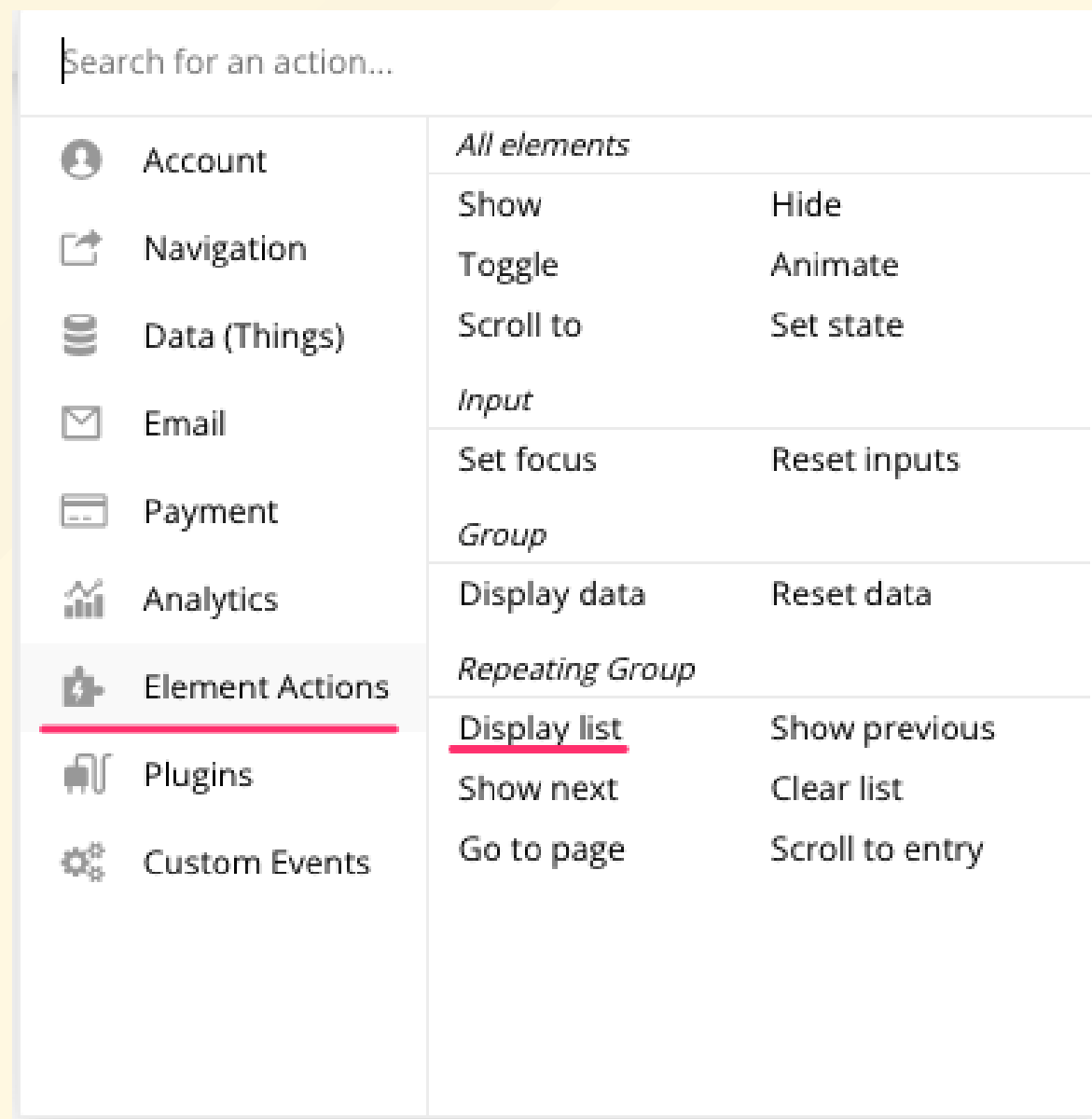
The screenshot shows an API configuration interface. At the top, the 'Name' field is 'search', 'Use as' is 'Data', and 'Data type' is 'JSON'. The 'GET' method is selected, and the URL is 'https://www.googleapis.com/youtube/v3/search'. Below this, the 'Parameters' section contains a table with four rows. The third row, with 'maxResults' as the key and '10' as the value, and the fourth row, with 'type' as the key and 'video' as the value, are highlighted with a red line. The 'Private' checkbox is checked for both. Other options like 'Allow blank' and 'Optional' are unchecked. At the bottom, there are checkboxes for '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.' and there is a 'Reinitialize call' button.

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"/>

- 続いて Repeating Group の Data source を空にします
- 表示する対象データの指定は、新たに設ける検索機能のワークフローの中で設定するため、ここが空でも問題ありません

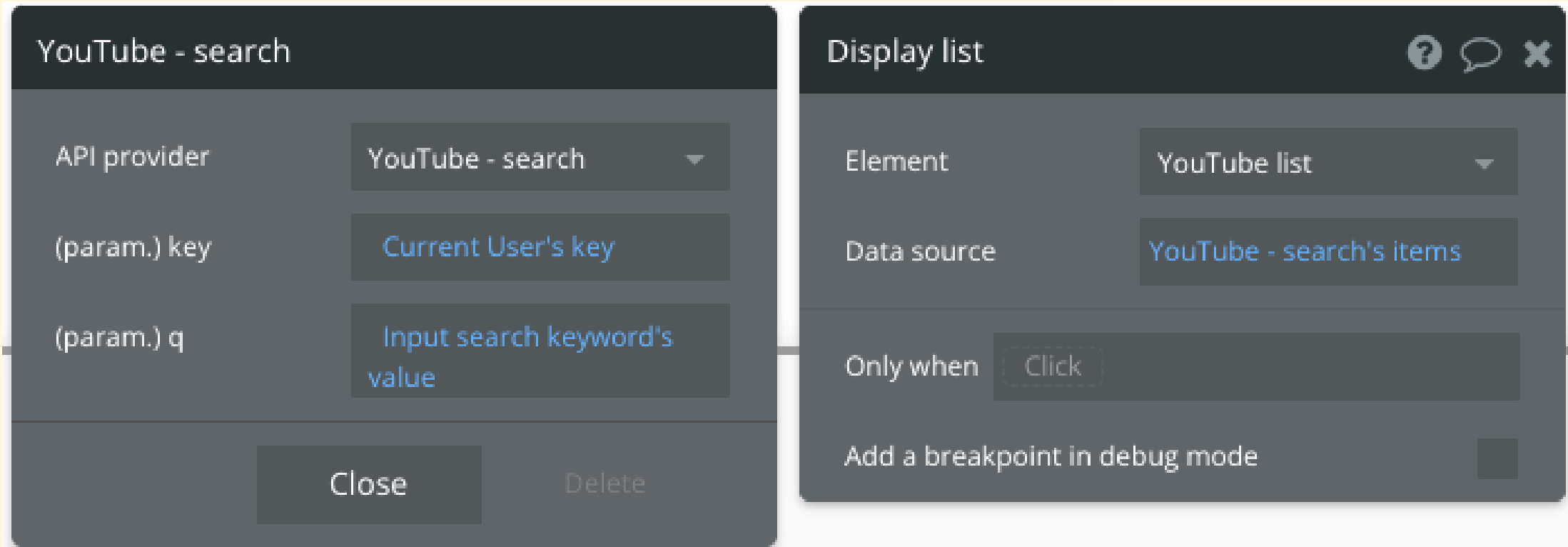


- 次にキーワード検索が行われた時のワークフローの設定となるため、新たに用意した検索ボタンに対してワークフローを設定します
- 今回は要素（Repeating Group）に対するアクションとなるため "Element Actions" の中にある "Display list" を選択します



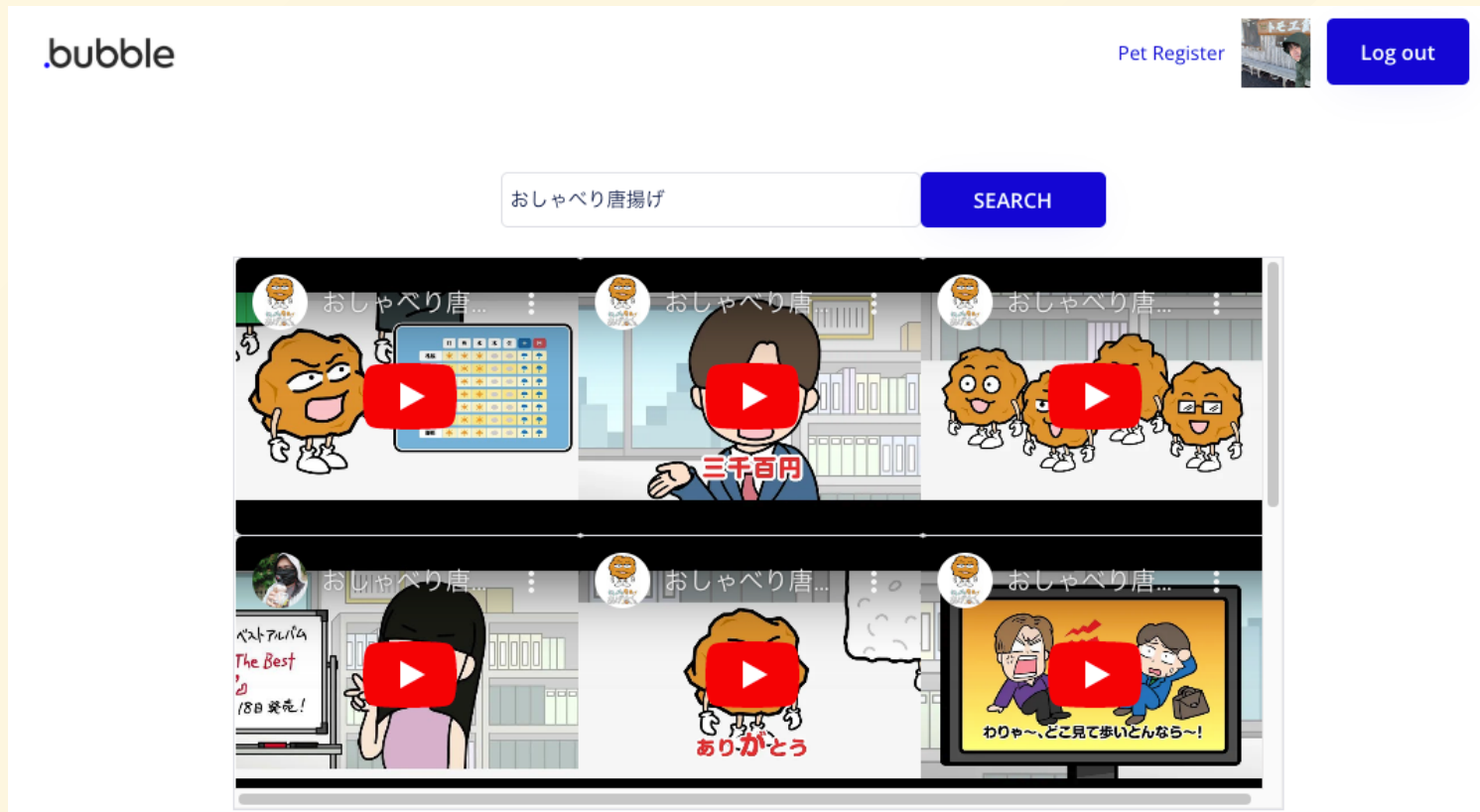
- Data source には Get data from an external API を選択し、API 経由のデータ表示とします
- そして API provicer に **YouTube - search** を選択すると、先ほどはなかった 2 つの param が表示されるはずです
- これは先ほど YouTube API 側の設定で動的に指定したいパラメータについて Private のチェックを外したためとなります
- ここで指定する値はいつもの Dynamic data ですね

- 設定するとこんな感じです



プレビューしてみましょう

- ログインしている状態で、キーワード欄に入力した動画一覧が表示されましたか？



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

Bubble での共同作業の Tips

チームで Bubble を使って共同作業をする時に役立つ情報をお伝えします。

Bubbleでのチームメンバーの招待方法

チームメンバーと一つのアプリを共同編集するには...

- チームで一つのアカウントを用意し、そのアカウントをみんなでログインして使いまわすことになります

- Bubble には正規の方法による共同編集機能はついているのですが...



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	Suggested 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

Bubbleでの同時編集の注意点

- 同じエレメントに対する同時編集は避けましょう(後から行われた編集で上書きされます)
 - そのためには、画面ごとに担当者を決めて同じ画面の同時編集を避けるのがおすすめです。
- それ以外の注意点は特にありません。
- 編集は他の人の画面にリアルタイムで反映されます。
 - 画面の編集だけでなく、データベースやWorkflowの編集もリアルタイムで反映されます。
- 画面ごとに作業分担すれば、難なく開発できそうです。

チーム開発演習

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

作業分担のやり方の例

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



演習結果の発表

チームごとに演習で作ったアプリについて発表してみましょう！

まとめ

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

最後に

- Google Cloud Platform で発行した API キーや OAuth の情報を削除しておきましょう

以上です！

お疲れさまでした！