

Programming Boot Camp

Bubble の基本

東京工業大学 2021/11/13

Naotake KYOGOKU

目次

- Bubble とは
- Bubble に登録しよう
- Bubble でのアプリ開発の概要説明
- まずはペット登録から一覧表示まで作ってみよう
- ペット詳細画面を作ってみよう
- ペットの体重を管理しよう
- まとめ

Bubble とは

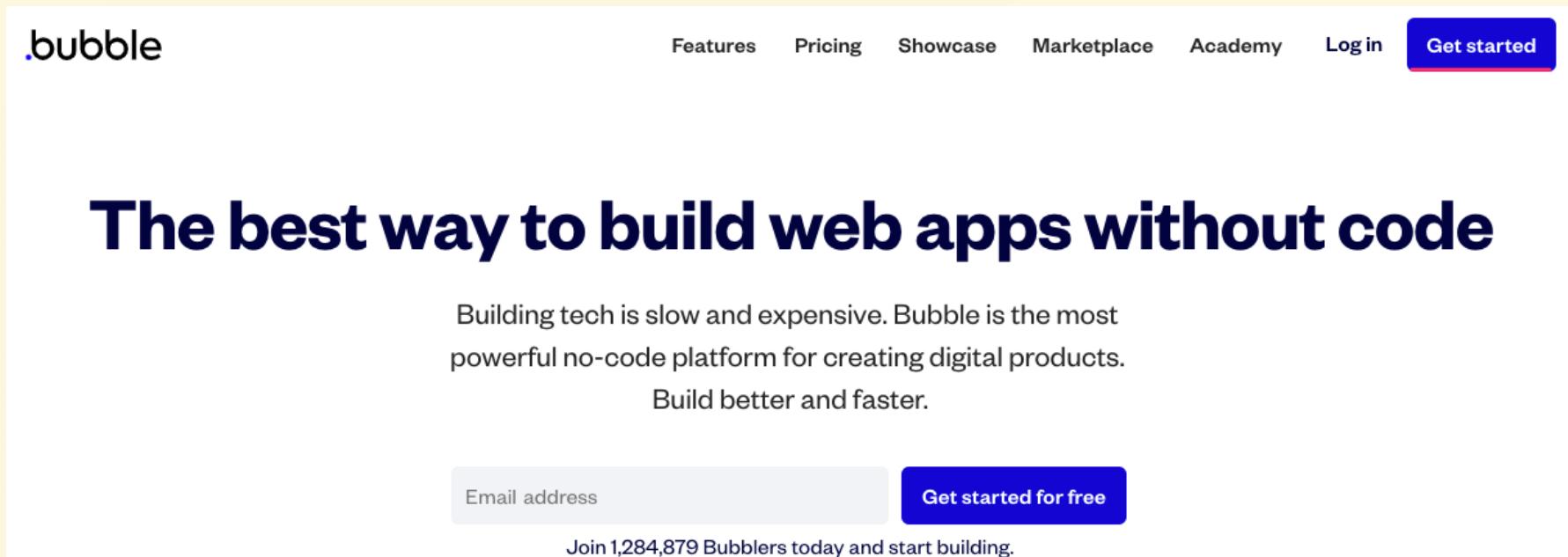
- Bubble はビジュアルプログラミングツールと呼ばれ、ノーコードツールではありますが、プログラミング的な発想が必要なツールとなります。
- 用意されているパーツから使用したいものを選び、それを画面にドラッグ&ドロップしていくなどの操作は Adalo と同じです。
- ただ、配置したパーツに対して動きをつけていく部分については、プログラミング的な発想が必要となります。
- また、Adalo と違い、Web アプリケーション前提での開発を行い、それをスマホ表示に対応させていく形となります。

Adaloで作られたアプリの例

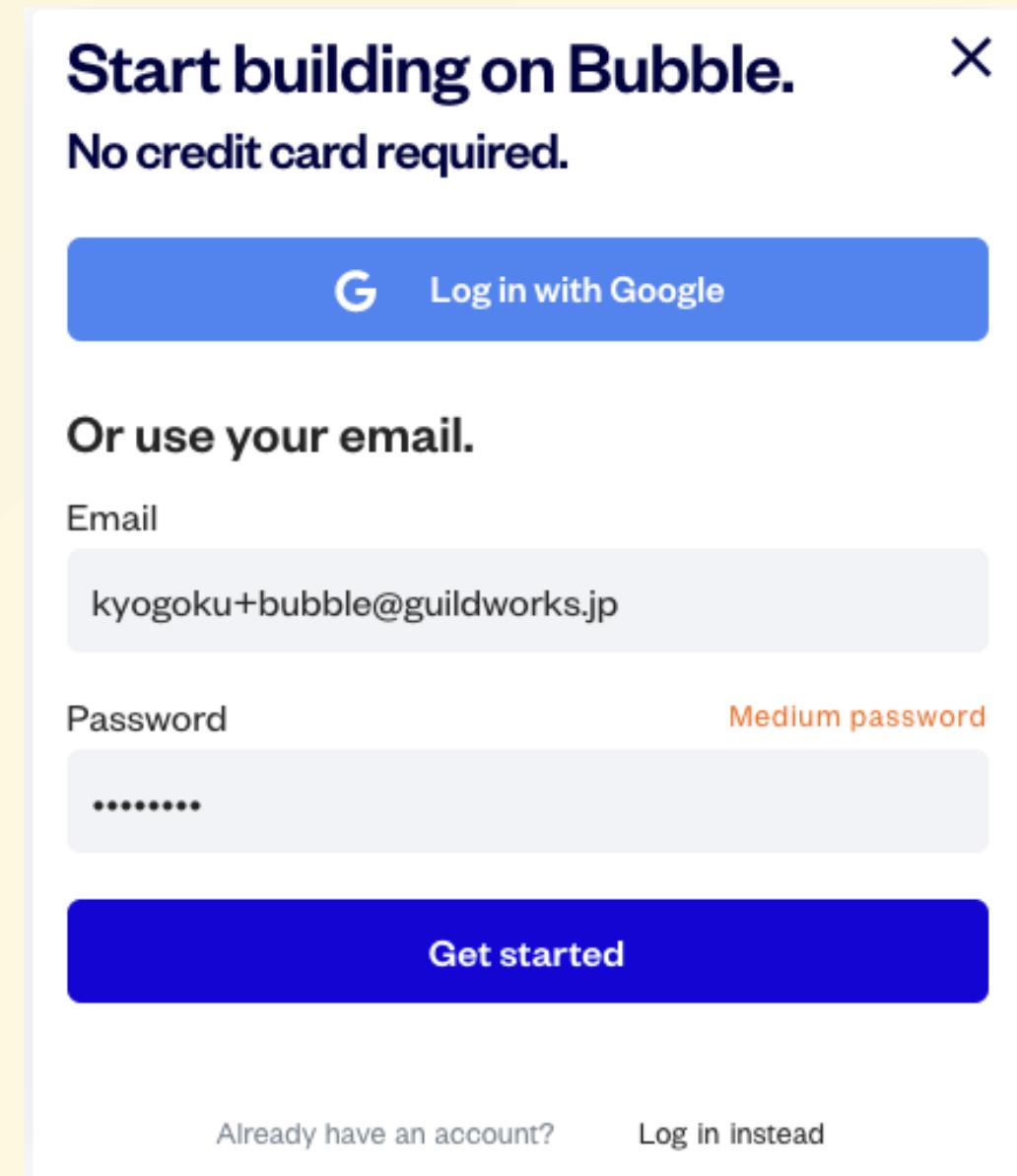
- LIBRIS（リブリス）：<https://libris.ne.jp/>
 - 2021年3月リリース
 - 行きたい本屋が見つかる検索サイト
 - 個人経営のお店やブックカフェやライブラリなど、全国の本屋を検索できる
 - 開発者は、本好きの非エンジニアの方だそうです
- その他
 - <https://nocode-freaks.com/?p=200>
 - ゲームもありました（笑）

Bubble に登録しよう

- Bubble の TOP 画面からメールアドレスを入力してアカウント登録
 - <https://bubble.io/>
 - Get started クリック



- アカウントは無料で登録できます
- ご自身のメールアドレスとパスワードを入力して "Get started" クリック



- 利用規約への同意して "I agree..." クリック

Agreeing to terms

Before you can use Bubble, you need to accept Bubble's Terms of Service. In particular, you must agree to Bubble's

- [Terms](#) - our Terms of Service are a legal agreement between you and Bubble and govern your use of our services.
- [Privacy Policy](#) - your privacy is important to us. We will only process your personal data as described in our Privacy Policy.
- [Cookies](#) - you consent to the placing of cookies on your computer or device to make this website better in accordance with our Privacy Policy.

Please read these documents carefully and click "I agree to Bubble's terms, privacy policy, and use of cookies" to use Bubble. If you have not read these documents, or if you do not agree to be legally bound by them, click "I do not agree" and do not use the platform.

I agree to Bubble's terms, privacy policy, and use of cookies

I do not agree

- 利用に向けていくつかアンケートに回答
- 最後に "Submit(thanks!)" クリック

Welcome to Bubble!

We have a couple of questions to help you get started.

Who do you want to build apps for?

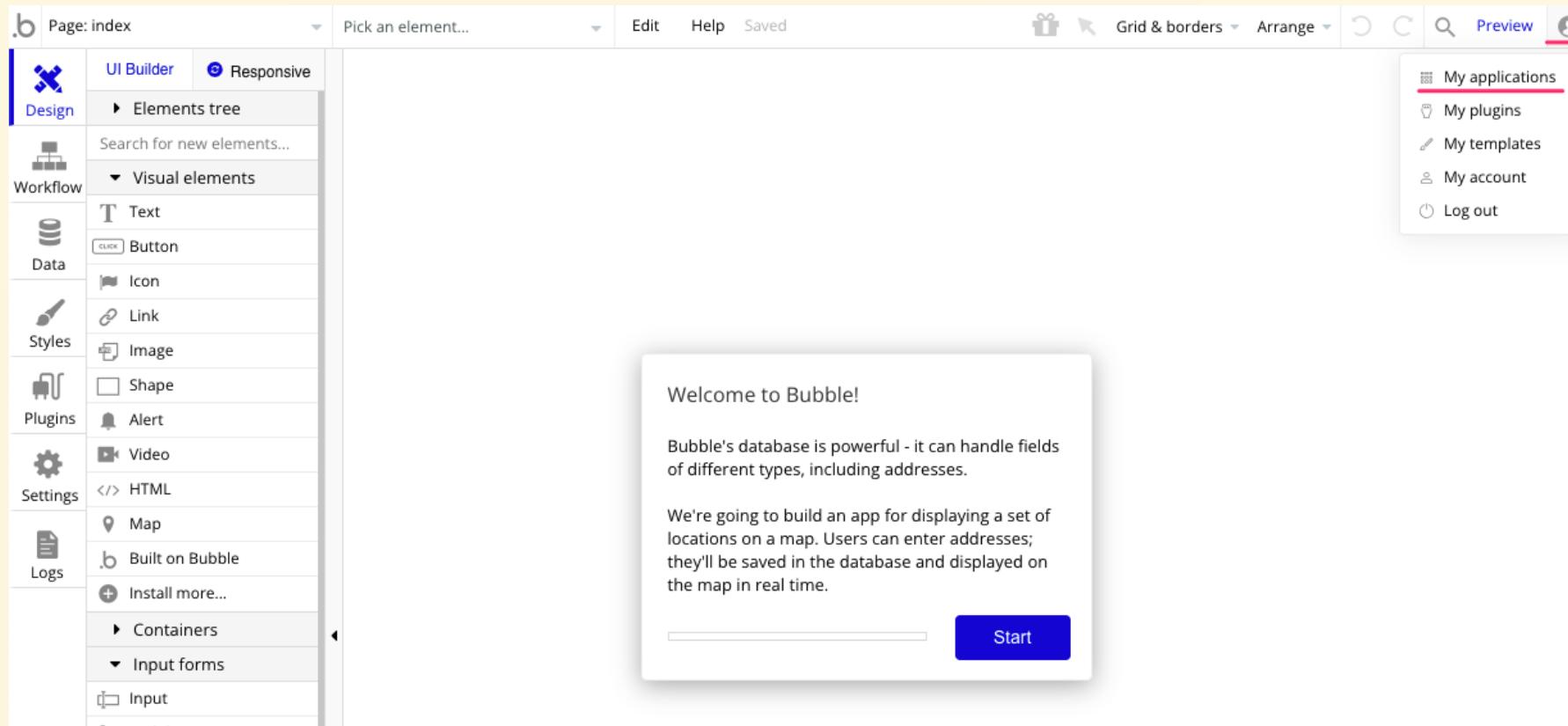
- Startup
- Small business
- Large, enterprise-size company
- Clients, as part of an agency
- Clients, as a freelancer
- School / university
- Non-profit or government agency
- Myself

Skip

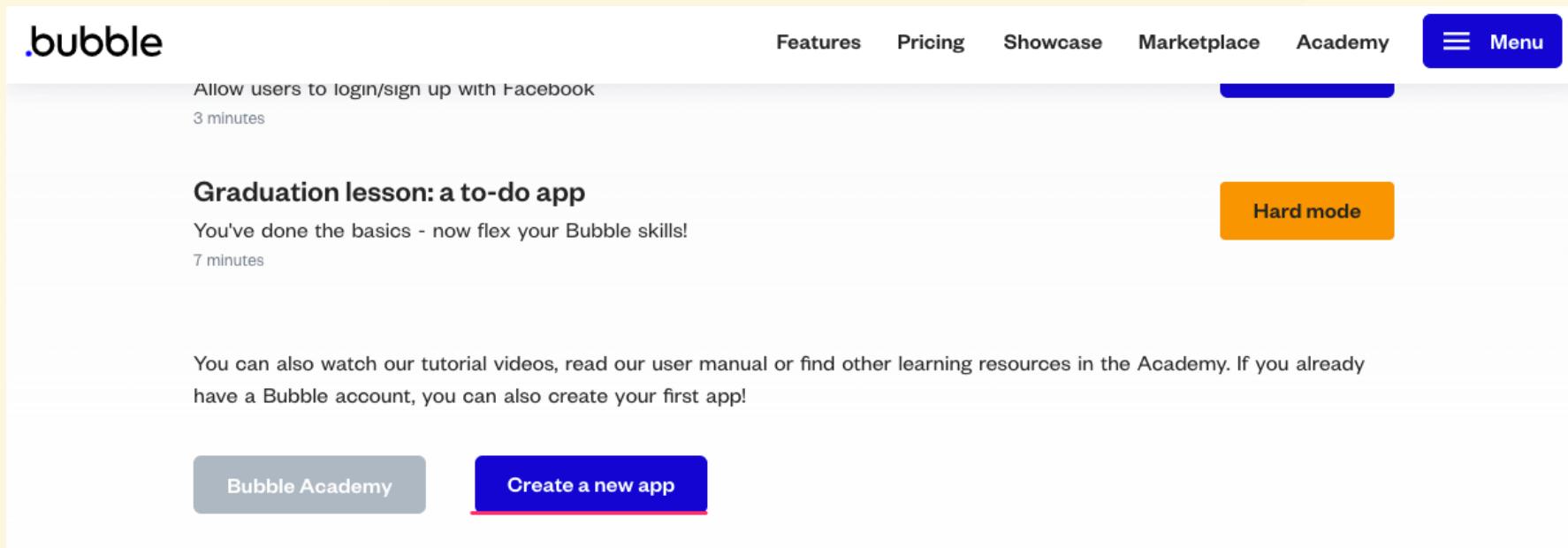


Submit (thanks!)

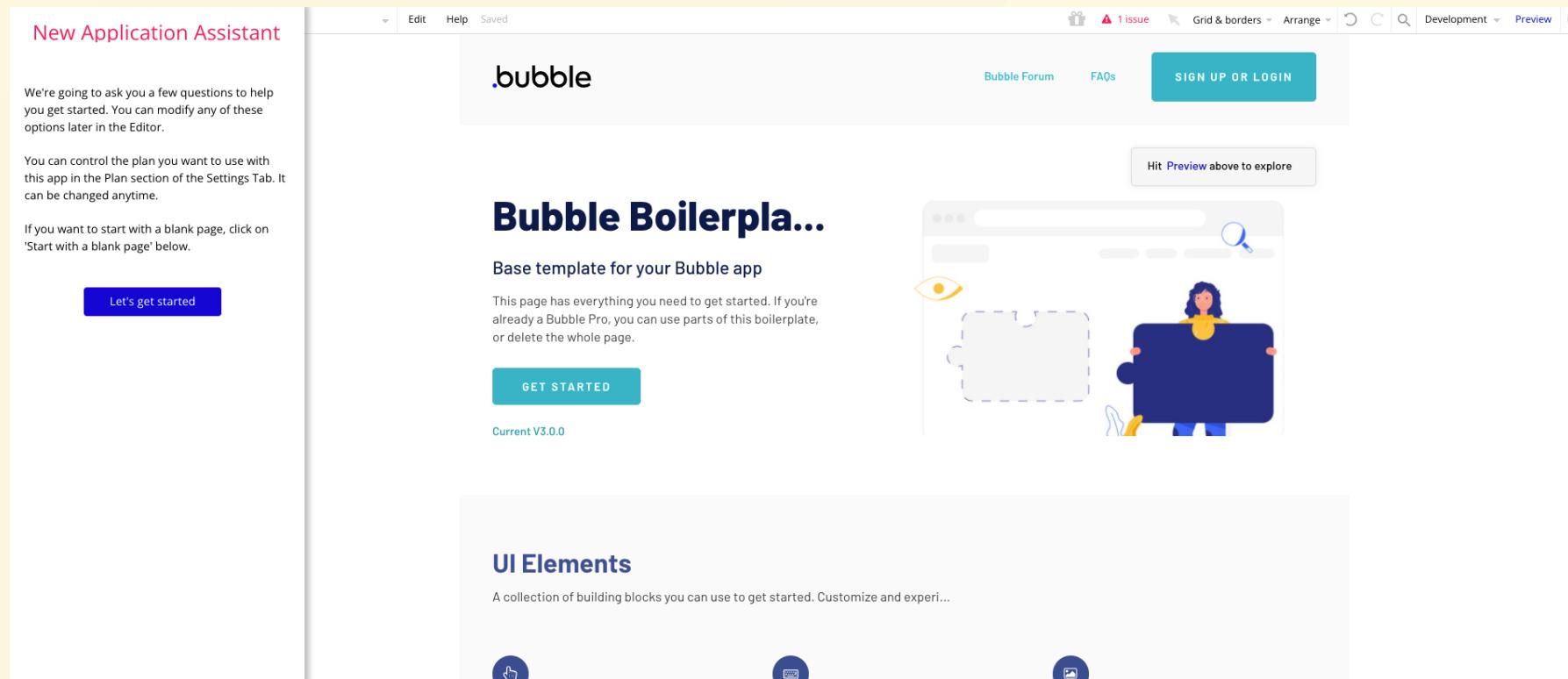
- するとエディタ画面が表示され、チュートリアルが始まりますが、これは後からでも実施できるので閉じます
- 画面右上のアイコンをクリックし "My applications" 選択



- Welcome ページが表示されるので、画面下までスクロールして "Create a new app" クリック

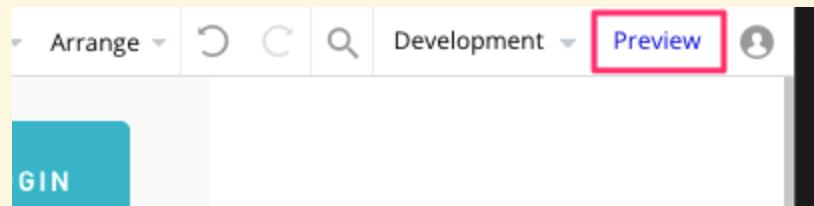


- このような画面が表示されたら OK です
- これは Bubble が自動生成したテンプレートが初期状態として表示されています
- 左パネルの New Application Assistant が表示された方は、下の "Close the assistant" を選択



まずは Bubble のテンプレートを動かしてみよう

- 画面右上の Preview ボタンをクリックしてください



- プレビュー画面が起動します

The screenshot shows the preview interface for a Bubble app. At the top, there's a navigation bar with the 'bubble' logo, 'Bubble Forum', 'FAQs', and a 'SIGN UP OR LOGIN' button. Below the navigation, the title 'Bubble Boilerplate' is displayed in large, bold, dark blue font. Underneath the title, it says 'Base template for your Bubble app'. A text block explains that the page contains everything needed to get started, mentioning Bubble Pro users can use parts of the boilerplate or delete the whole page. A prominent 'GET STARTED' button is located below this text. To the right of the text, there's a cartoon illustration of a person holding a large blue puzzle piece, with a magnifying glass icon above them. Below the illustration, the text 'Current V3.0.0' is shown. Further down, there's a section titled 'UI Elements' with a sub-section 'Buttons and Actions' featuring a hand icon, 'Data Collection' with a database icon, and 'Images and Graphic elements' with a camera icon. Each section has a brief description and a link to more examples. At the bottom, there are buttons for 'Debugger', 'Normal' (which is highlighted in blue), 'Slow', and 'Step-by-step'. On the far right, there are buttons for 'index', 'Inspect', and a refresh icon.

Bubble Boilerplate

Base template for your Bubble app

This page has everything you need to get started. If you're already a Bubble Pro, you can use parts of this boilerplate, or delete the whole page.

[GET STARTED](#)

Current V3.0.0

UI Elements

A collection of building blocks you can use to get started. Customize and experiment!

Buttons and Actions

Buttons are used for navigation, actions, triggers, and more. Set `rollover_states`, `active_states`, and `start/edit_workflows`.

Data Collection

Form fields and inputs are essential parts of collecting data from your users. Here are a few examples to get you started.

Images and Graphic elements

Bring your site to life with visual in-page elements and images. You can use static uploaded visuals or dynamic data driven ones.

Debugger [Normal](#) [Slow](#) [Step-by-step](#)

index [Inspect](#)

Sign up してみよう

- 画面右上の "SIGN UP OR LOGIN" をクリック
- モーダルが表示されるので Email / Password / Confirm password を入力して "SIGN UP" クリック
 - 入力した Email と Password はメモしておいてください(後で使います)

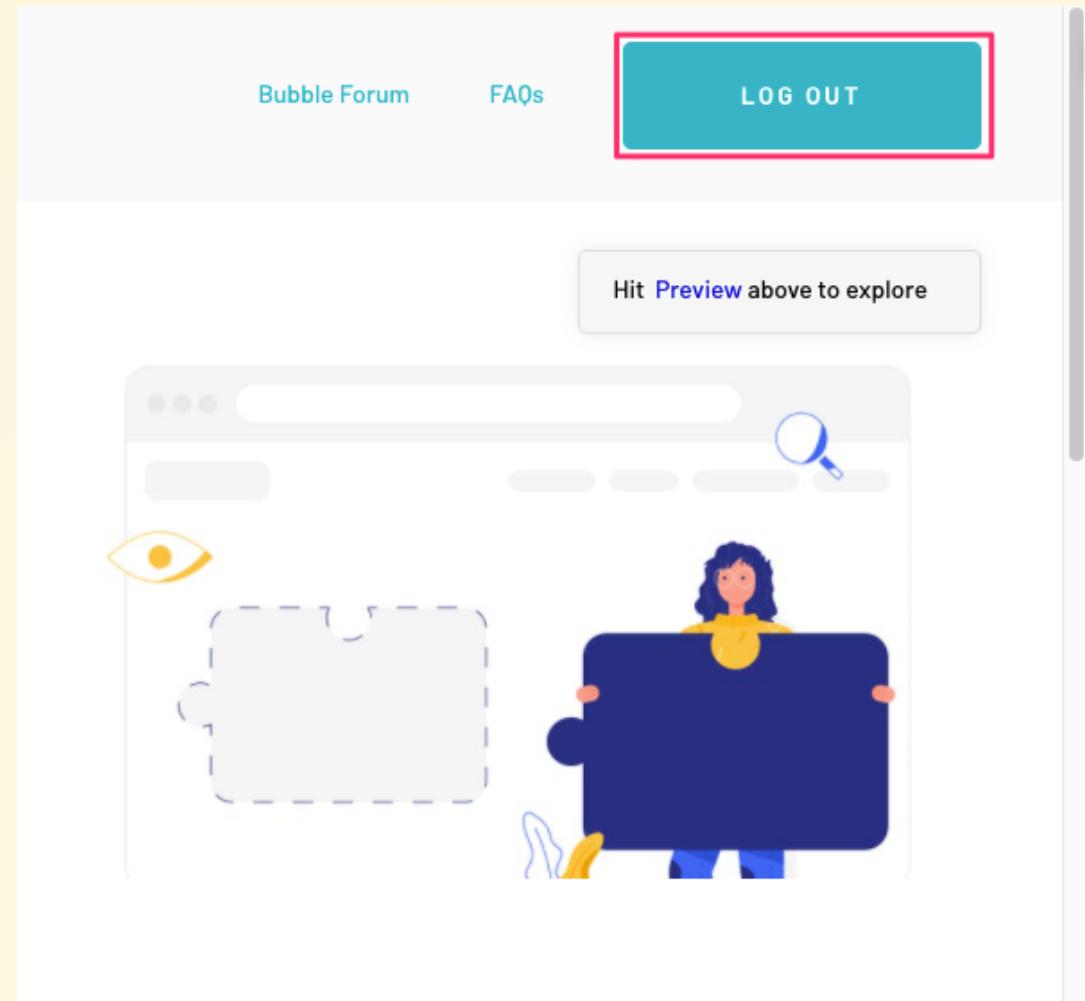
Sign up

By signing up, I agree to the Privacy Policy and the Terms of Services.

SIGN UP

OR LOGIN

- すると画面右上のボタンが "LOG OUT" になったと思います
- つまりアカウント登録が行われ、ログイン状態になったということです

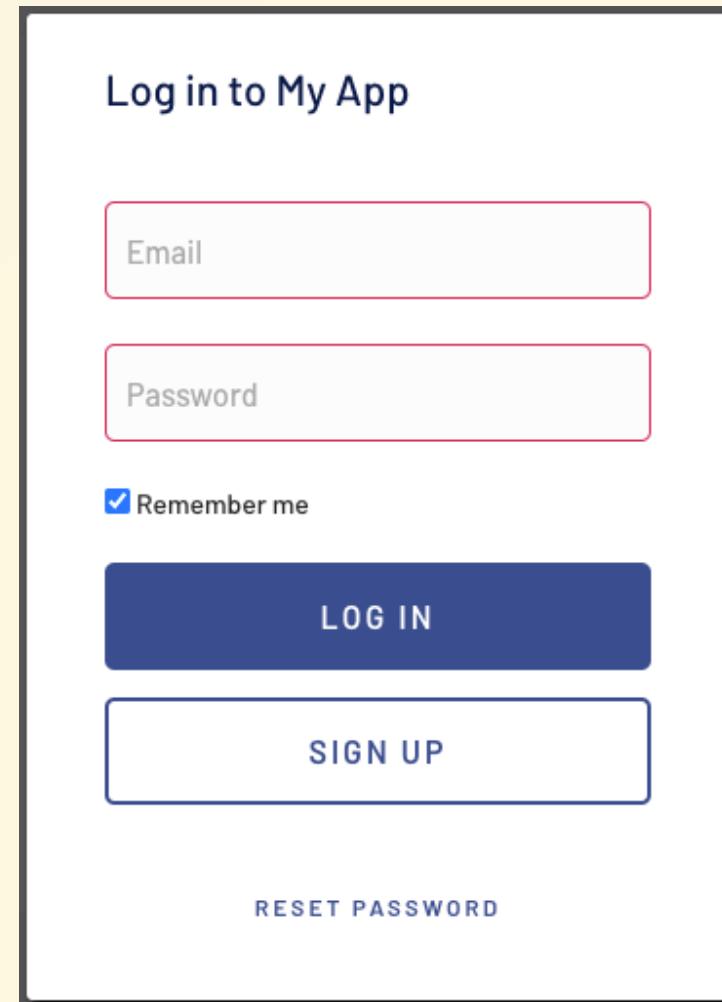


Login してみよう

- 画面右上の "LOG OUT" をクリックして一度ログアウトします
- 再度右上の "SIGN UP OR LOGIN" をクリック
- モーダルの下にある "OR LOGIN" をクリック

The image shows a login modal window titled "Log in to My App". It contains two input fields: "Email" and "Password", both outlined in red. Below the fields is a checkbox labeled "Remember me" with a checked blue box. At the bottom are three buttons: a large dark blue "LOG IN" button, a white "SIGN UP" button with blue text, and a smaller "RESET PASSWORD" link at the bottom right.

- ログイン用のモーダルが表示されるので、そこで先程入力したメールアドレスとパスワードを入力
- 最後に "LOG IN" をクリック
- 成功すると画面右上のボタンが "LOG OUT" になったと思います
- これでサインアップとサインインが動いていることが確認できます

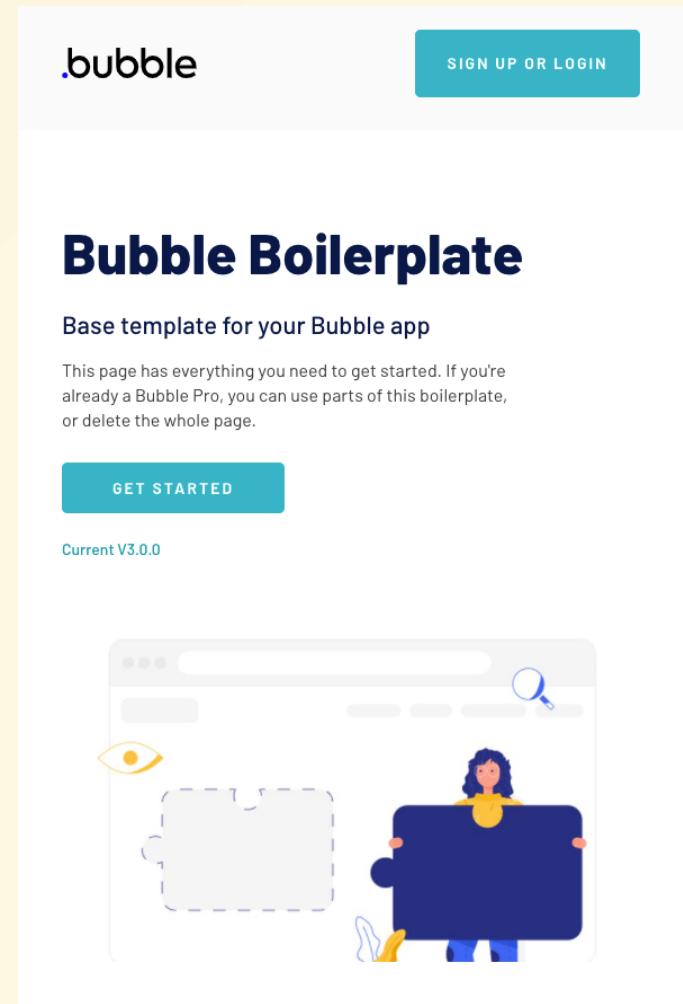


The image shows a login modal window titled "Log in to My App". It contains two input fields: "Email" and "Password", both outlined in red. Below the fields is a checked checkbox labeled "Remember me". A large blue button labeled "LOG IN" is centered below the checkbox. To the right of the "LOG IN" button is a white button labeled "SIGN UP". At the bottom of the modal is a link labeled "RESET PASSWORD".

スマホ対応について

- 冒頭でお伝えした通り、Bubble のアプリは Web アプリケーションの前提となっています
- Bubble のアプリをスマホ対応させる場合、**レスポンシブデザイン**という考え方を取り入れます

- まずは動きを見てみましょう
- ブラウザの横幅を縮めてみてください



- レイアウトが崩れることはなく、自動的に要素の配置が変わり、スマホサイズに適した配置になったと思います
- これがレスポンシブデザインというものです
- Google ではこのように説明されています

ユーザーのデバイスの種類（パソコン、タブレット、モバイル、非視覚的ブラウザ）に関係なく、同じ URL で同じ HTML コードを配信しつつ、画面サイズに応じてレンダリング方法を変更します。

- ポイントとしては下記の点ですね
1. 表示しているページの URL は同じ
 2. 画面サイズに応じてレンダリング方法を自動的に変える

- もう少し噛み砕いて言うと、画面サイズに応じて、要素が伸びる／縮む、折り返す／折り返さない、表示する／表示しないを制御する
- これを各項目レベルで設定してあげることで、自動でレスポンシブデザインに対応させることができます
 - 実際の開発時には、設定してプレビューして動作確認していくと良いと思います
- 後ほど画面操作の中でレスポンシブを実際に設定していきます（まずは概要まで）

Bubble でのアプリ開発の概要説明

次に、Bubble でのアプリ開発の概要を説明していきます

3つのメイン操作

- Bubble では主に 3 つの操作を使い分けていきます
 1. Design
 2. Workflow
 3. Data
- 左パネルからそれぞれを行き来することができます



Design

- ユーザーインターフェースを作るために画面上にコンポーネントを配置するモード
- 左パネルに UI のコンポーネント類、右パネルに実際の画面編集領域

Page: index Pick an element... Edit Help Saved

UI Builder Responsive

Design

Workflow

Data

Styles

Plugins

UI コンポーネント群

Settings

Logs

UI Builder Responsive

Elements tree

Search for new elements...

Visual elements

Text

Button

Icon

Link

Image

Shape

Alert

HTML

Map

Built on Bubble

Install more...

Containers

Group

Repeating Group

Popup

Floating Group

Group Focus

Install more...

Input forms

Input

Multiline Input

Checkbox

.bubble

Bubble Forum FAQs SIGN UP OR LOGIN

Hit [Preview](#) above to explore

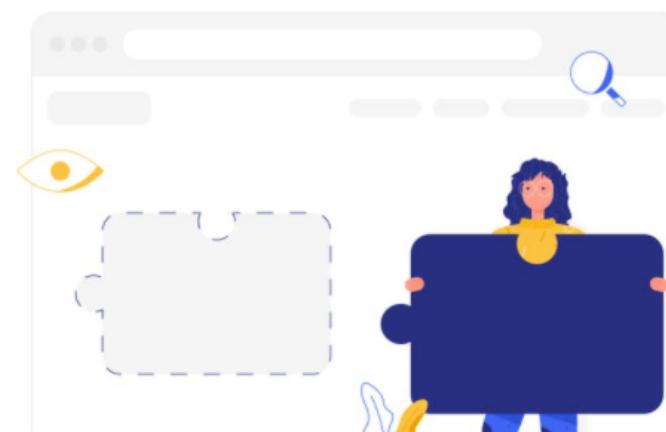
Bubble Boilerpla...

Base template for your Bubble app

This page has everything you need to get started. If you're not already a Bubble Pro, you can use parts of this boilerplate, or delete the whole page.

[GET STARTED](#)

Current V3.0.0

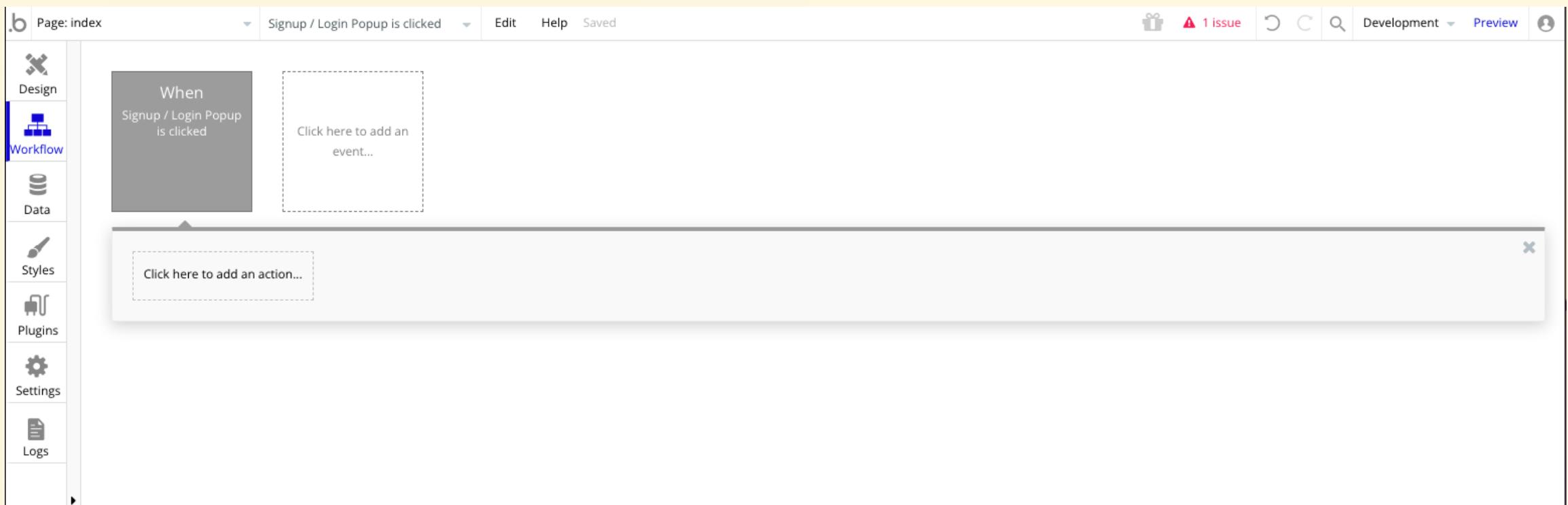


UI Elements

A collection of building blocks you can use to get started. Customize and experi...

Workflow

- 画面に動きをつけるためのモード
 - 例) ボタンが押された時に、画面遷移する
- ここが、少しプログラミングの考え方を必要とする部分となります



Data

- データを定義したり操作したりするモード

The screenshot shows the Bubble Data editor interface. On the left is a vertical sidebar with icons for Design, Workflow, Data (which is selected and highlighted in blue), Styles, Plugins, Settings, and Logs. The main area has a header with tabs: Data types, Privacy, App data, Option sets, and File manager. Below the header, there's a section for Database views with buttons for New view, Primary fields, Search, and a search input field. To the right of the search is a message indicating 1 entry (displaying 1). There are buttons for New entry, Delete (0), Upload, Modify, Export, and Bulk. Further down, there's a table with columns: Email, Created Date, Modified Date, and Slug. A single row is shown with the email 'kyogoku+bubble_test@guildwo...', created date 'Oct 31, 2021 8:51 pm', modified date 'Oct 31, 2021 8:51 pm', and an empty Slug field. At the bottom left is a button labeled 'All Users'.

- "App data" タブを選び、右側から "All Users" のリンクをクリック
- すると、右パネルに登録済みのユーザー一覧のデータが表示され、そこに先程 "Sign up" したユーザが存在するはずです
- これで画面とデータベースが連携していることを確認できましたね

The screenshot shows the Bubble app interface for the 'pet_weight_register' page. The left sidebar has icons for Design, Workflow, Data (which is selected), Styles, Plugins, Settings, and Logs. The main header includes 'Edit', 'Help', 'Saved', a gift icon, '8 issues', and 'Development' with a dropdown. Below the header, tabs for 'Data types', 'Privacy', 'App data' (selected), 'Option sets', and 'File manager' are visible. Under 'Database views', it says 'Application data - All Users - Development version'. It features buttons for 'New view', 'Primary fields', 'Search', '1 entries (displaying 1)', 'New entry', 'Delete (0)', 'Upload', 'Modify', 'Export', and 'Bulk'. A search bar says 'Search view names or data types...'. A table lists one entry: 'Run as → kyogoku+bubble_test@guildwo' with 'Oct 31, 2021 8:51 pm' under 'Created Date' and 'Modified Date'. There are two additional fields: 'Email' and 'Slug'. At the bottom, there are sections for 'All Pets', 'All PetWeightLogs', and 'All Users' (with a red underline).

その他の操作について

- Styles

- スタイルに名前をつけてあげることで、汎用的に使える
- 部品ごとに個別に指定することもできる
- 今回は基本的にありもののスタイルを使っていきますので、今回の講義では扱いません

The screenshot shows the Bluebeam interface with the 'Styles' tab selected in the left sidebar. The main area displays a list of styles:

- Alert - Standard Alert (Default)
- Alert - Warning Alert
- Button - Outline Button
- Button - Primary Button

For the 'Alert - Standard Alert (Default)' style, there is a preview window showing a standard alert message. Below the preview are buttons for 'Find all elements using this style' and 'Define another condition'. The top navigation bar includes links for 'Edit', 'Help', 'Saved', 'Development', 'Preview', and a user profile icon.

- Plugins

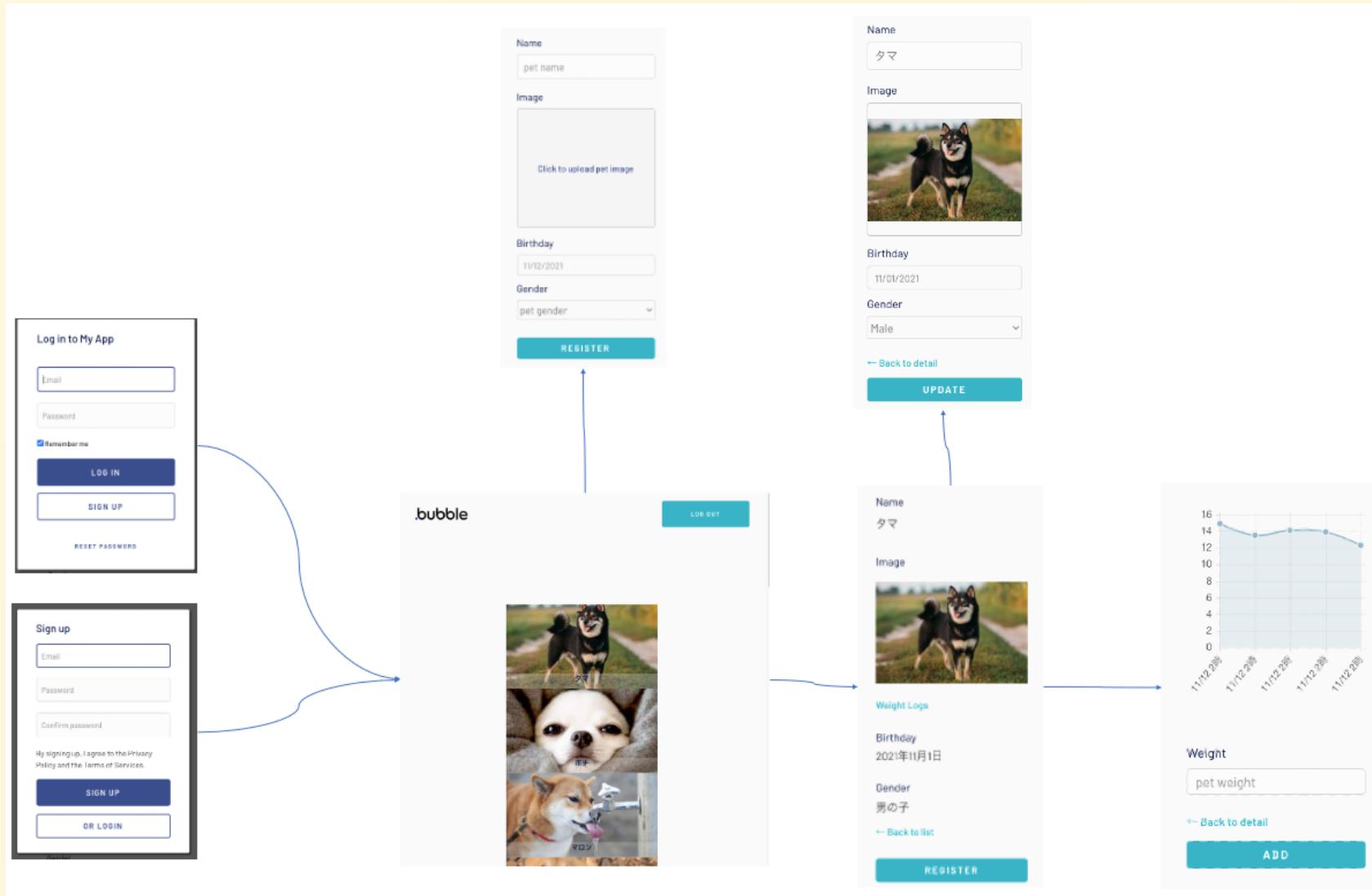
- Adalo で折れ線グラフのコンポーネントを追加したように、Bubble にもアプリケーションを拡張するための様々なコンポーネントが公開されています
- Bubble ではこれをプラグインと呼びます
- Bubble のプラグインには無料 / 有料のものがあるので、お使いになる際には確認しましょう

- Settings
 - Bubble のプランの変更やアカウントに関する操作
 - Bubble が提供しているメッセージを一元管理できます
 - アプリケーションを使うメインユーザ向けにロケール（言語設定）を変更可能
 - このメッセージを変更することで、システム全体でメッセージ管理を統一できます
 - アプリ全体で共通に使えるメッセージ類を変えられるし、デフォルトのメッセージを変えることも可能
 - アプリ固有のメッセージを追加することも可能
 - 今回の講義では扱いません

- Logs

- アプリケーションを動かしたときのログを見ることがあります
- ただ、アプリケーションをプレビュー表示しながらログを見ることもできるので、今回の講義では扱いません

これから作っていく画面の全体像



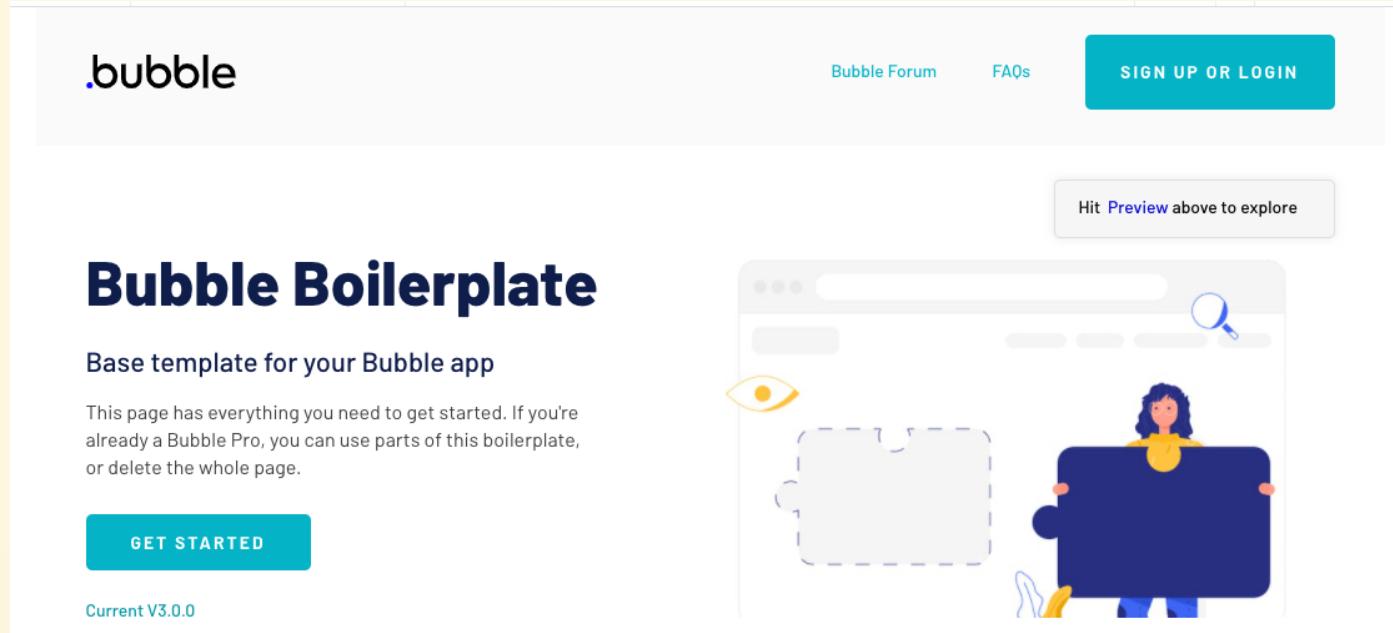
まずはペット登録から一覧表示まで作ってみよう

- Adalo での会で作成した画面要素やデータ構造と同じものを作っていきます
- そのため、Bubble の会では画面とデータベース操作と一緒に作っていきます
- また、認証に関する部分は Bubble で用意されたデフォルトのものを流用します
- まずはペットの登録画面を作成し、そのペットを一覧に表示するところまで作ってみましょう！

ペットの一覧画面の枠だけ用意する

- まず最初に動作確認を使ったページをペットの一覧ページとして使い回します
- そのため、Bubble が自動生成したパーツを一度すべて削除します
- index ページの何か一つの要素を選択したら "Ctrl + A" で全要素を選択します
- そして Delete ボタンをクリックすると画面がまっさらになると思います

要素削除前



The screenshot shows the homepage of the Bubble Boilerplate template. At the top, there's a navigation bar with the '.bubble' logo, 'Bubble Forum', 'FAQs', and a 'SIGN UP OR LOGIN' button. Below the navigation, a callout box says 'Hit [Preview](#) above to explore'. The main title 'Bubble Boilerplate' is displayed in a large, bold, dark blue font. Below it, a subtitle reads 'Base template for your Bubble app'. A text block explains that the page contains everything needed to get started, mentioning Bubble Pro users can customize parts of the template. A 'GET STARTED' button is present. The footer section, titled 'UI Elements', features three categories: 'Buttons and Actions' (with a hand icon), 'Data Collection' (with a database icon), and 'Images and Graphic elements' (with a camera icon). Each category has a brief description and a link to more information.

Bubble Boilerplate

Base template for your Bubble app

This page has everything you need to get started. If you're already a Bubble Pro, you can use parts of this boilerplate, or delete the whole page.

[GET STARTED](#)

Current V3.0.0

UI Elements

A collection of building blocks you can use to get started. Customize and experiment!



Buttons and Actions

Buttons are used for navigation, actions, triggers, and more. Set rollover states, active



Data Collection

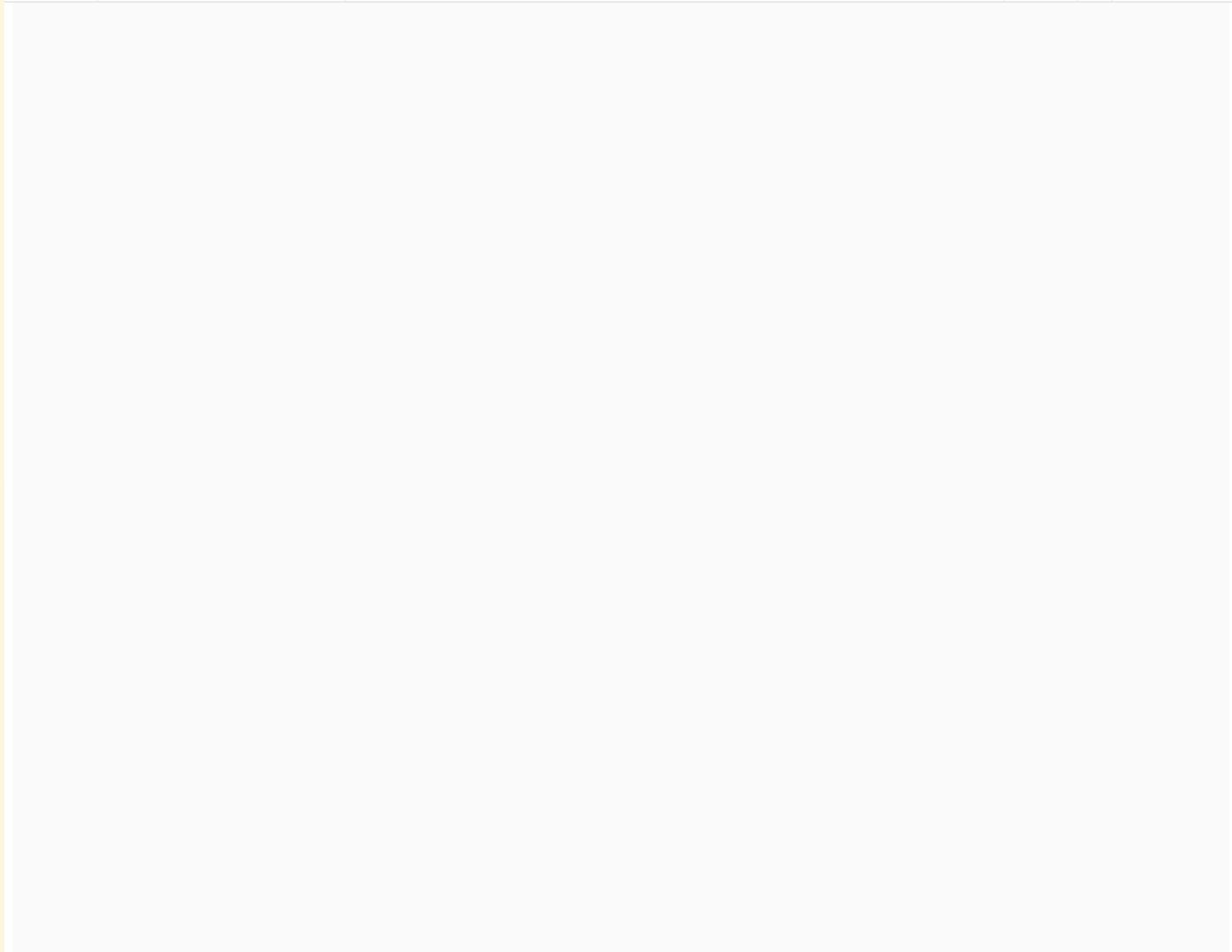
Form fields and inputs are essential parts of collecting data from your users. Here are a few



Images and Graphic elements

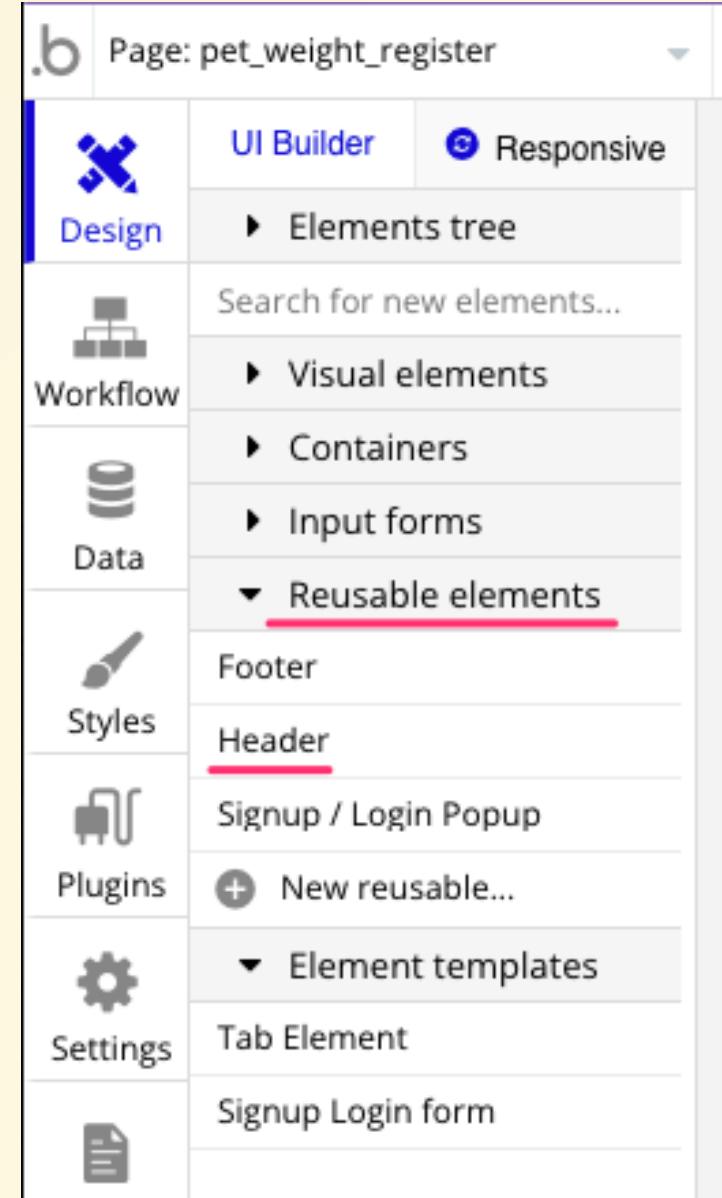
Bring your site to life with visual in-page elements and images. You can use static

要素削除後

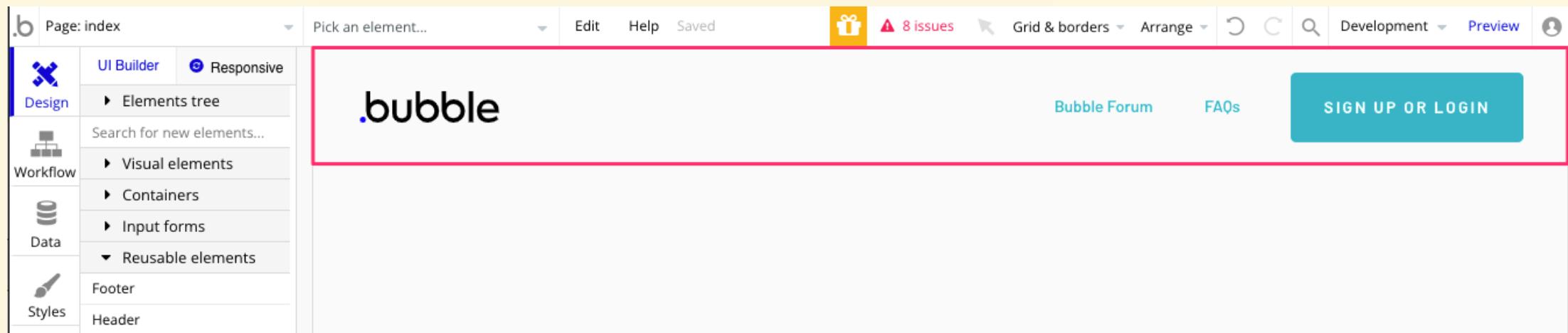


- Bubble のテンプレートに用意されていたヘッダーを使い回します
- Design タブを選択し、"UI Builder" の部分を下にスクロールすると

Reusable elements という項目の中に Header というのがあると思いますので、そちらをクリックします



- そして、右側の描画エリアの上の方で範囲選択（ドラッグ）すると、その部分に共通の Header 要素が追加されます



- 場所がちょうど一番上に配置できない場合は Header 選択しドラッグで画面中央上部に配置します
 - 要素を選んで右クリックし **Center horizontally** を選択すると、要素の横位置を中央に移動します
 - **Center vertically** を選択すると、要素の縦位置を中央に移動します
 - 今後要素を中心寄せたい時などはお使いください
- ヘッダーの中身は後で変更していきます

ペット登録画面を作成していきます

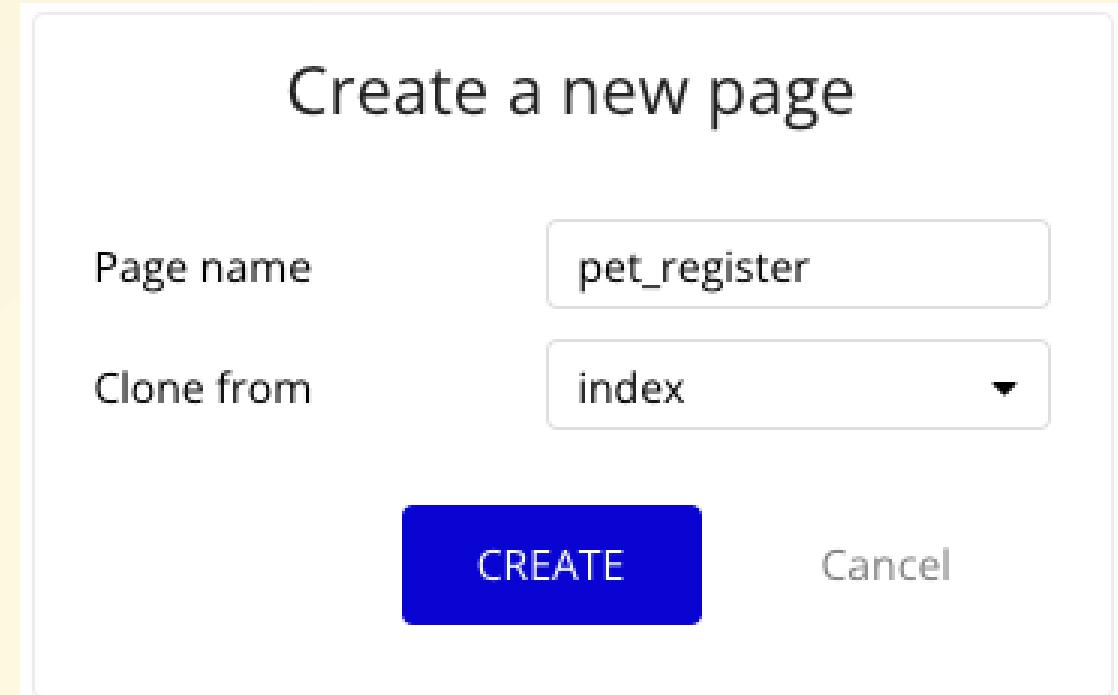
- 画面右上の **Search a page or ~** 部分をクリックするとポップアップが表示されます
- ポップアップの中から **Add a new page...** をクリック



ちなみに...

- ここには現在作成中のアプリケーションの中にある「ページ」や「共通コンポーネント」の一覧が表示されています
- Adalo では 1 つのキャンバスにすべての画面が表示されていましたが、Bubble では 1 つのキャンバスで 1 つの画面を操作していく操作になります
- そのため、画面間の移動はここから変更する必要があるので覚えておきましょう 

- **Add a new page...** をクリックすると新しい画面のポップアップが表示されます
- **Page name** は画面の名前となるので、空白などを含めずに英数字だけで入力します
 - 今回は **pet_register** としましょう



- `Clone from` は似たような画面を作りたい時に、コピー元となる画面を選択してその画面のコピーを作ることができます

Create a new page

Page name

Clone from ▾

CREATE Cancel

- 例えば登録画面と編集画面などは画面要素がほとんど同じになるので、登録画面を作った後に編集画面を作る時に "Clone from" に登録画面を選択すると、開発工数の削減につながります

Create a new page

Page name

Clone from ▾

CREATE Cancel

- 今回は折角なので先ほどヘッダーだけを用意した index を選択してコピーしましょう
- 必要な情報を入力したら "CREATE" ボタンをクリック

Create a new page

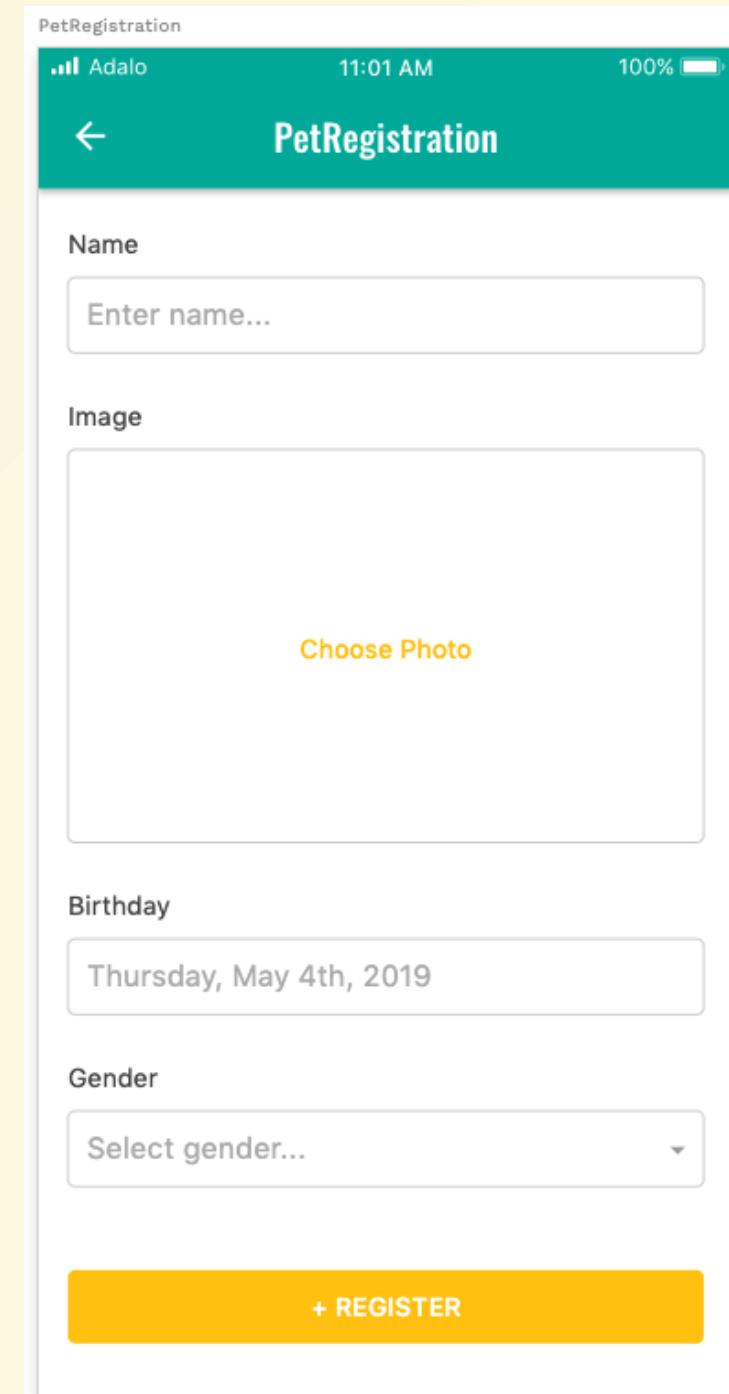
Page name

Clone from ▼

CREATE Cancel

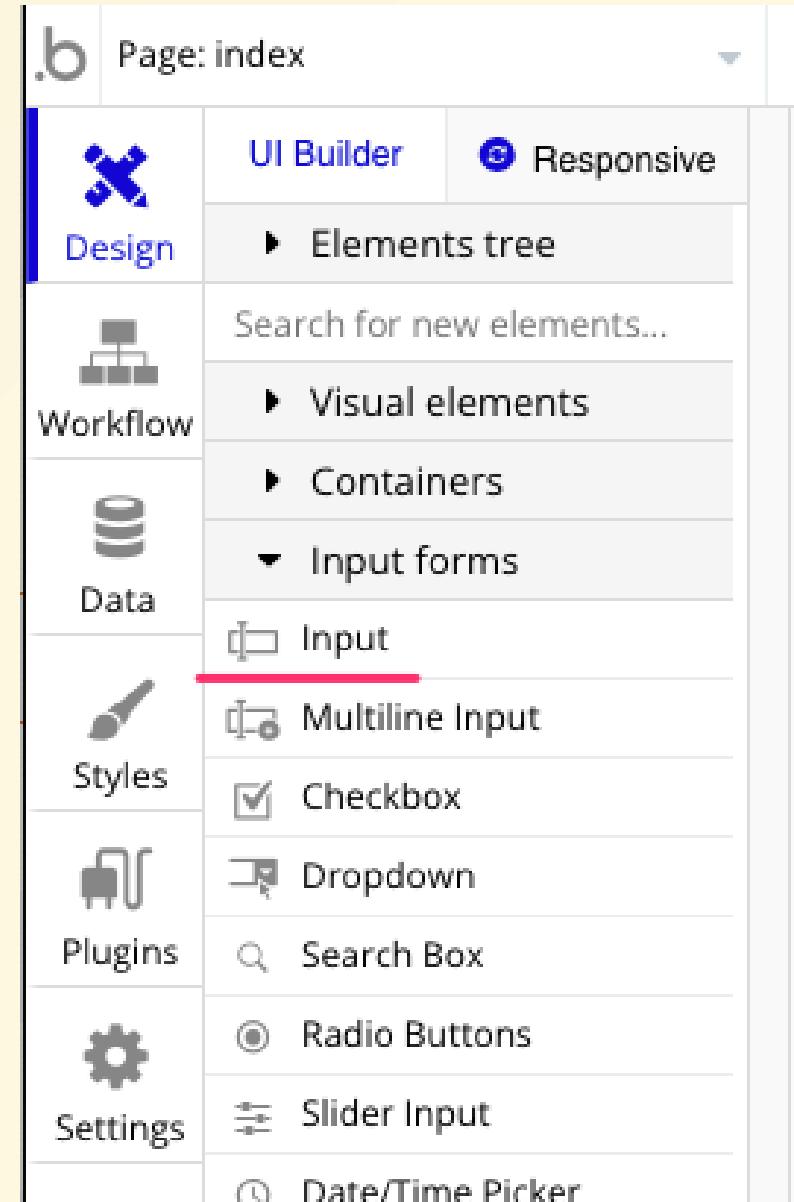
ペット登録画面を組み立てる

- Adalo で作成したペット登録画面のイメージを参考に画面を作ってみましょう

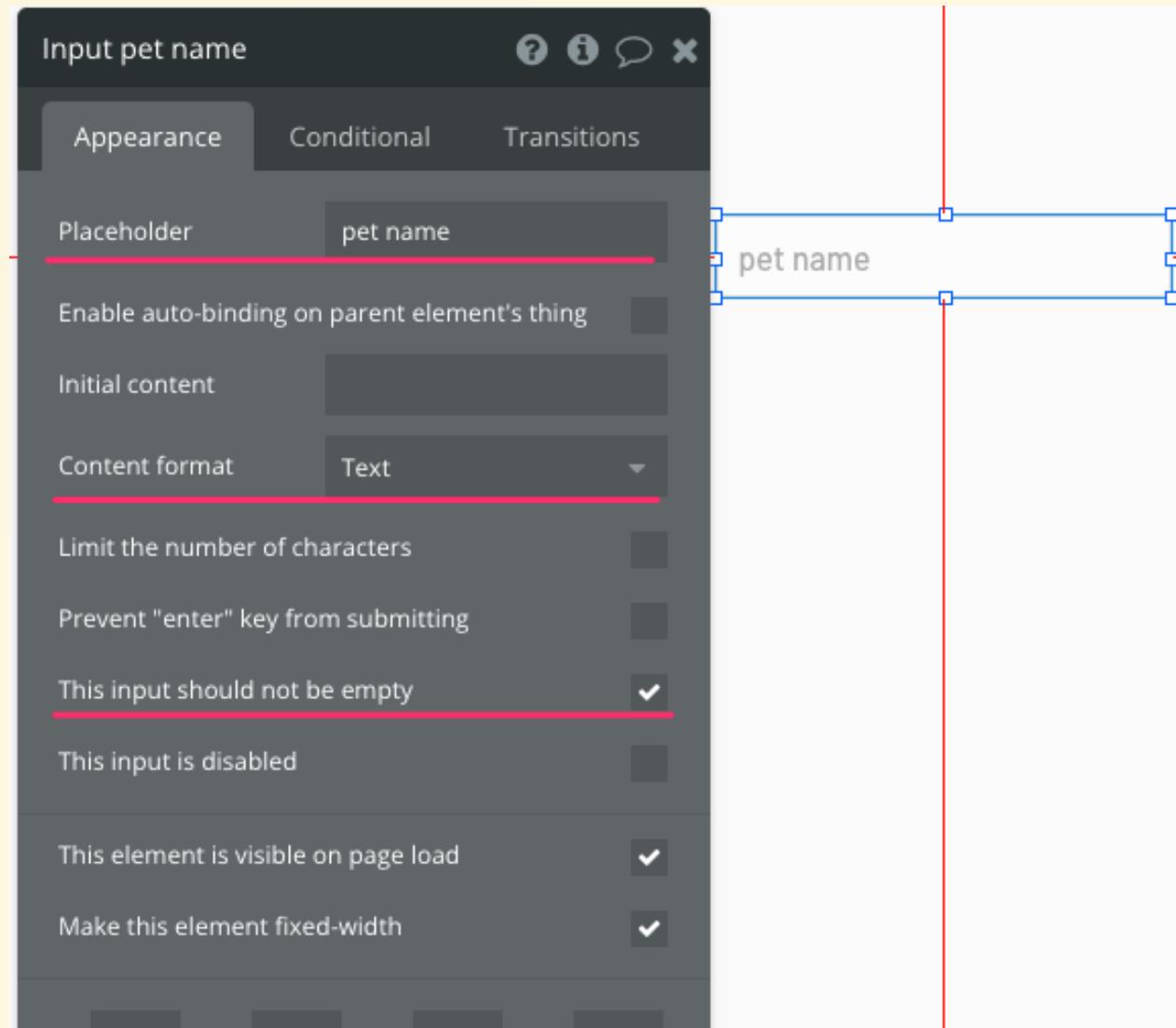


ペットの名前

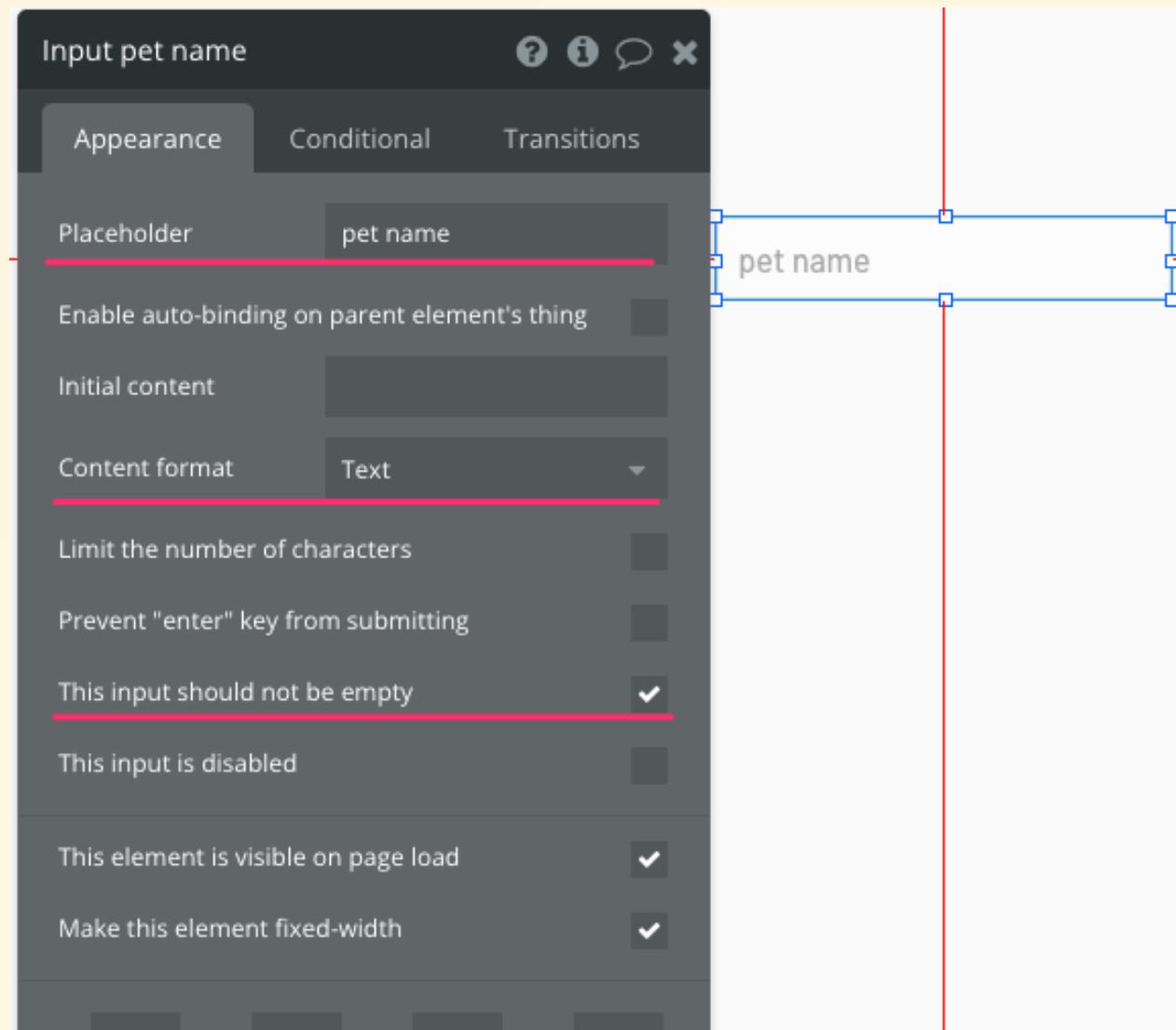
- まずはペットの名前を入力するテキストボックスを配置します
- 左パネルの "UI Builder" から **Input forms** の中にある **Input** を選択
- そして右パネル上で配置したい場所でドラッグをすると入力欄を表示されます



- 要素をダブルクリックすると要素の様々な情報を設定できるダイアログが表示されますので、必要な項目を設定していきます
- Placeholder はテキストボックスが未入力状態の時に表示される補助文言です
 - 今回は「pet name」とします



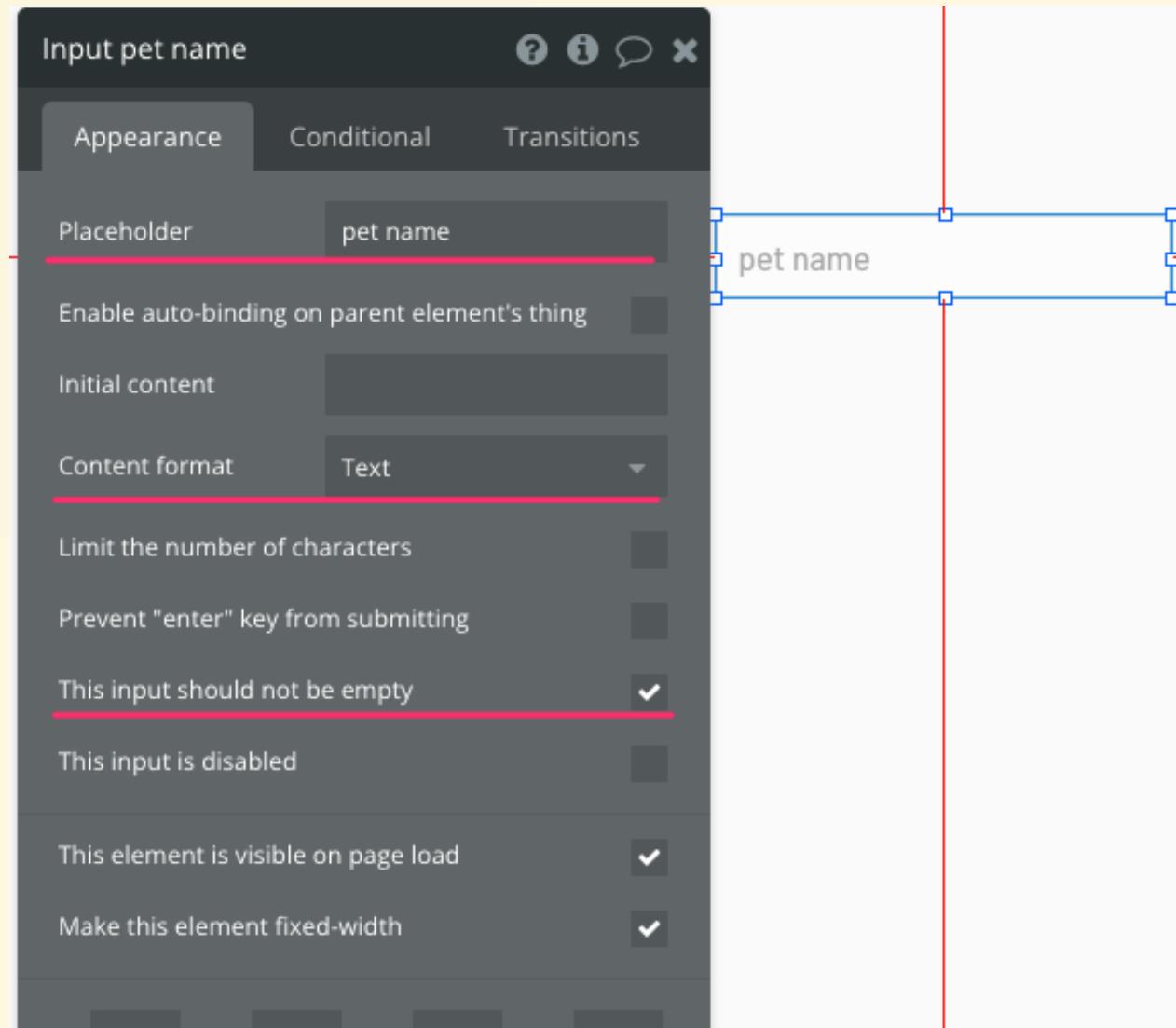
- Content format はテキストボックスに入力できる値の形式を指定できます
 - 今回は文字列の入力となるので "Text" のままとします



- This input should not be empty

は入力必須にするかどうかを指定できます

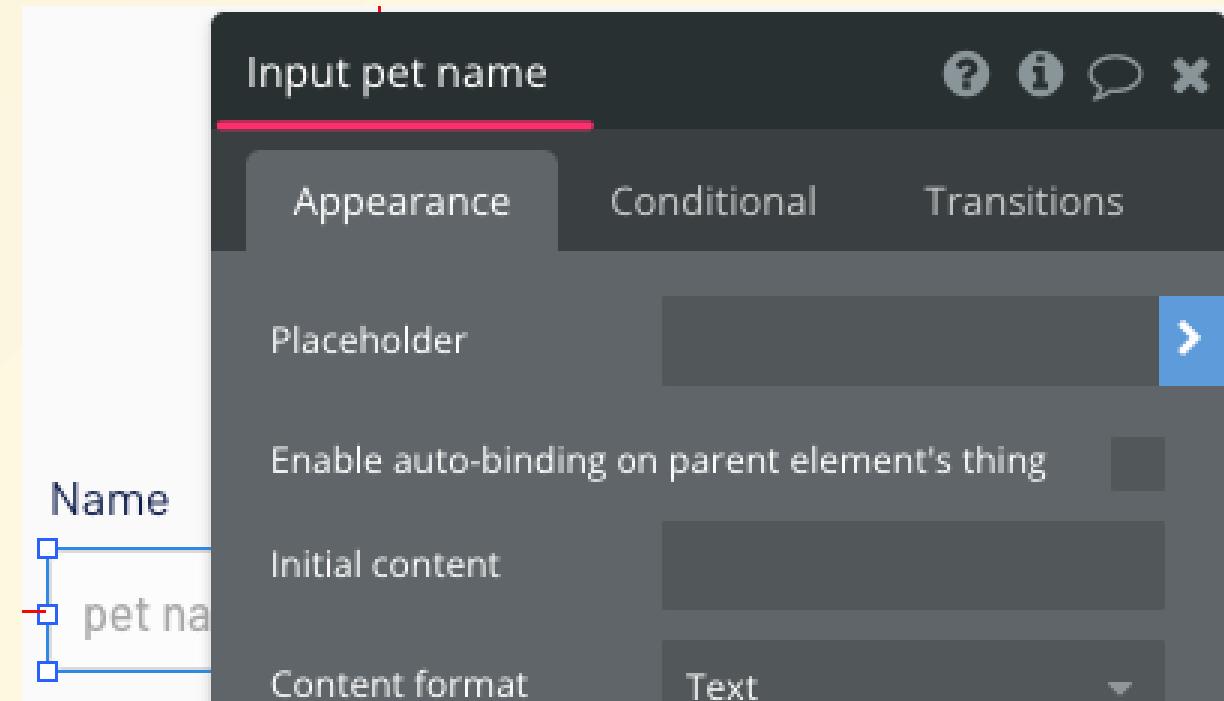
- 今回は必須としたいのでチェックをつけます



- まずはここまで設定したら OK です
 - ほかにも細かく指定ができますが、ここでは説明を省略します
 - 気になる方は見てみてください 

- もう一つだけ設定する項目があります
- ダイアログ先頭の部分をクリックすると、このテキストボックスの要素に対して名前を付けることができます
- これは後々ワークフローを定義するときに便利になるのでここでは

Input pet name と指定しましょう



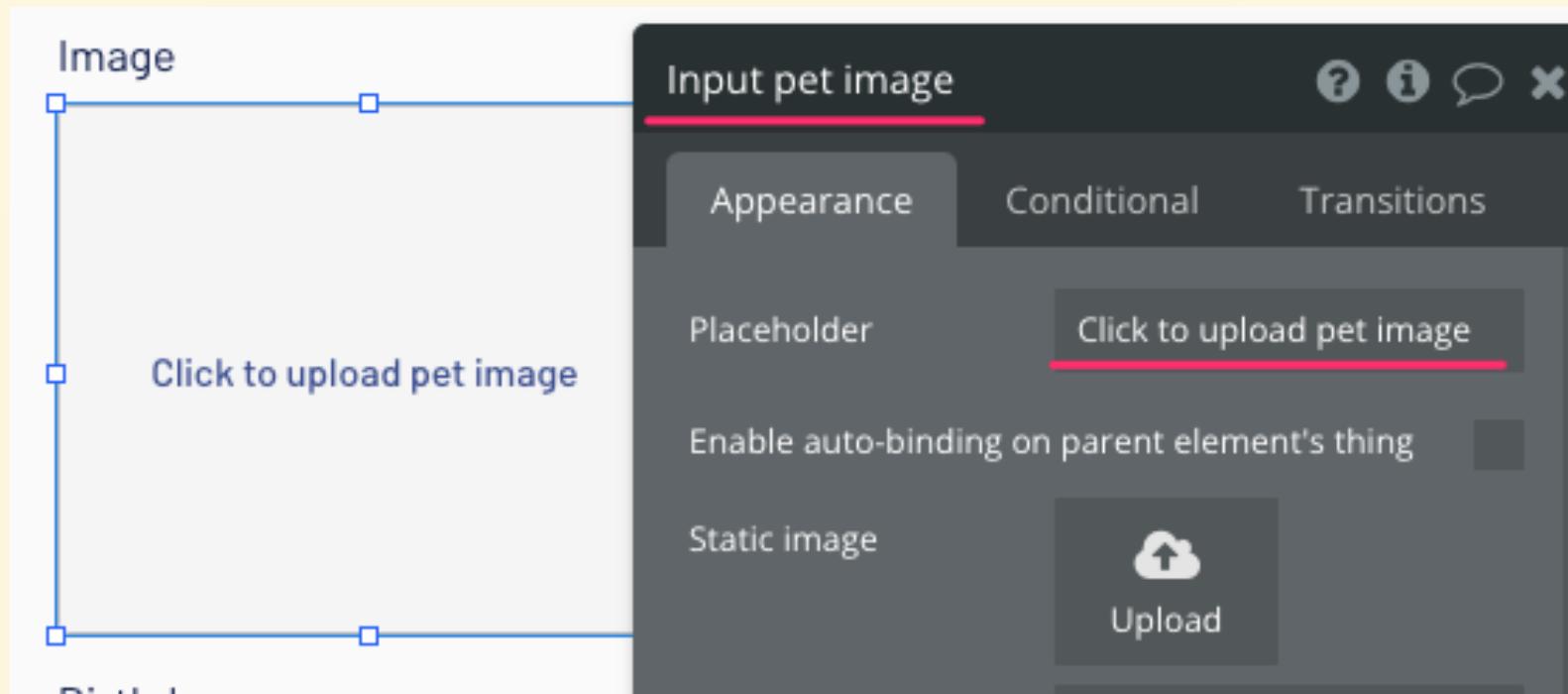
ペットの画像

- 次にペットの画像をアップロードする要素を配置します
- 左パネルから Picture Uploader を選択し、右パネル上でドラッグして要素を配置します

The screenshot shows the UI Builder interface with the following structure:

- UI Builder** (selected)
- Responsive**
- Design** (selected): Elements tree, Search for new elements...
- Workflow**: Visual elements, Containers
- Data**: Input forms
 - Input
 - Multiline Input
- Styles**: Checkbox
- Plugins**: Dropdown, Search Box
- Settings**: Radio Buttons, Slider Input
- Logs**: Date/Time Picker, Picture Uploader (highlighted with a red underline)

- この要素も同じくダブルクリックから要素の情報を設定します
 - 今回は Placeholder に「Click to upload pet image」と入力します
 - 要素の名前として Input pet image を指定します



ペットの誕生日

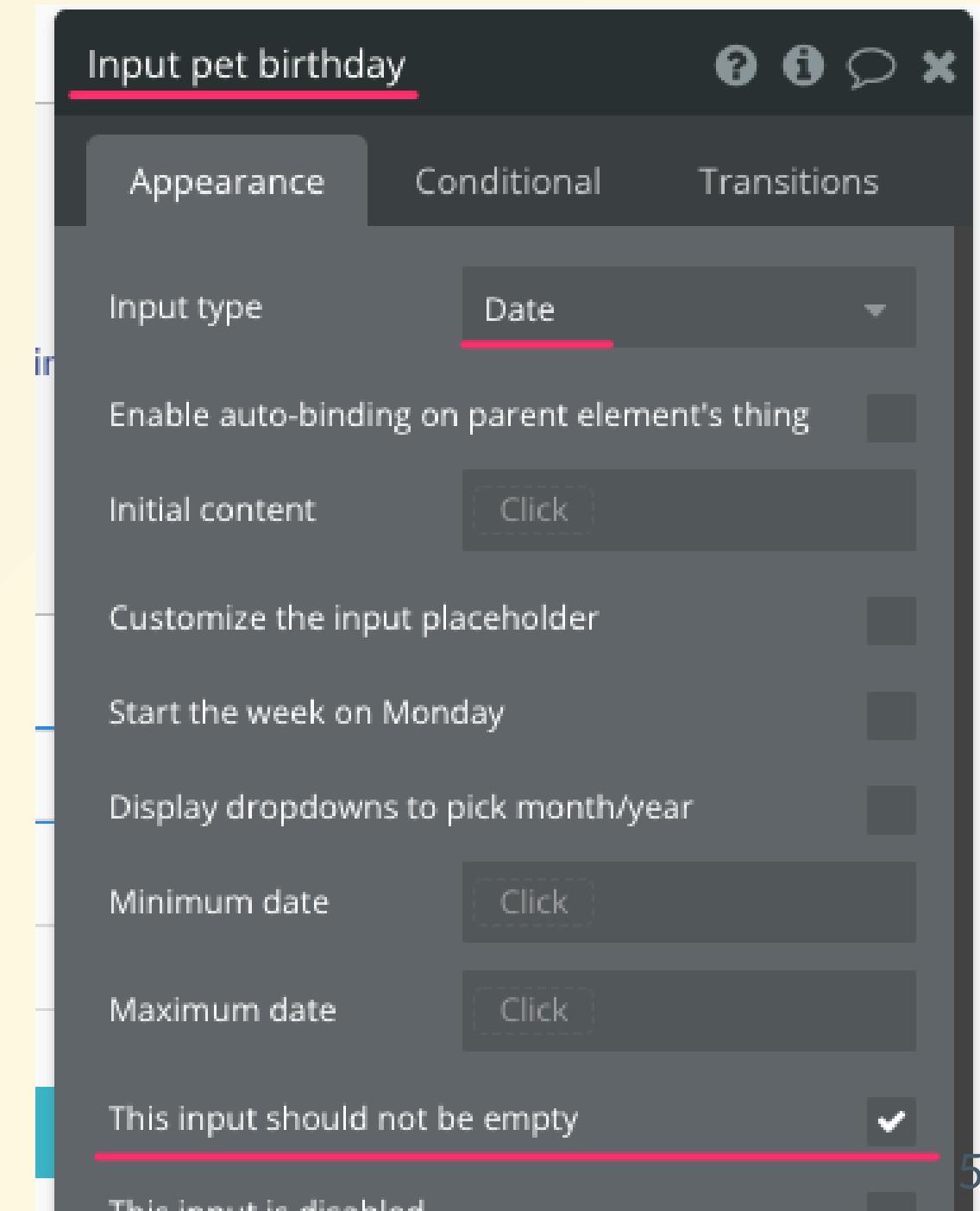
- 次にペットの誕生日を入力するための要素を配置します
- 左パネルから **Date/Time Picker** を選択し、右パネル上でドラッグして要素を配置します

The screenshot shows the UI Builder interface with the following details:

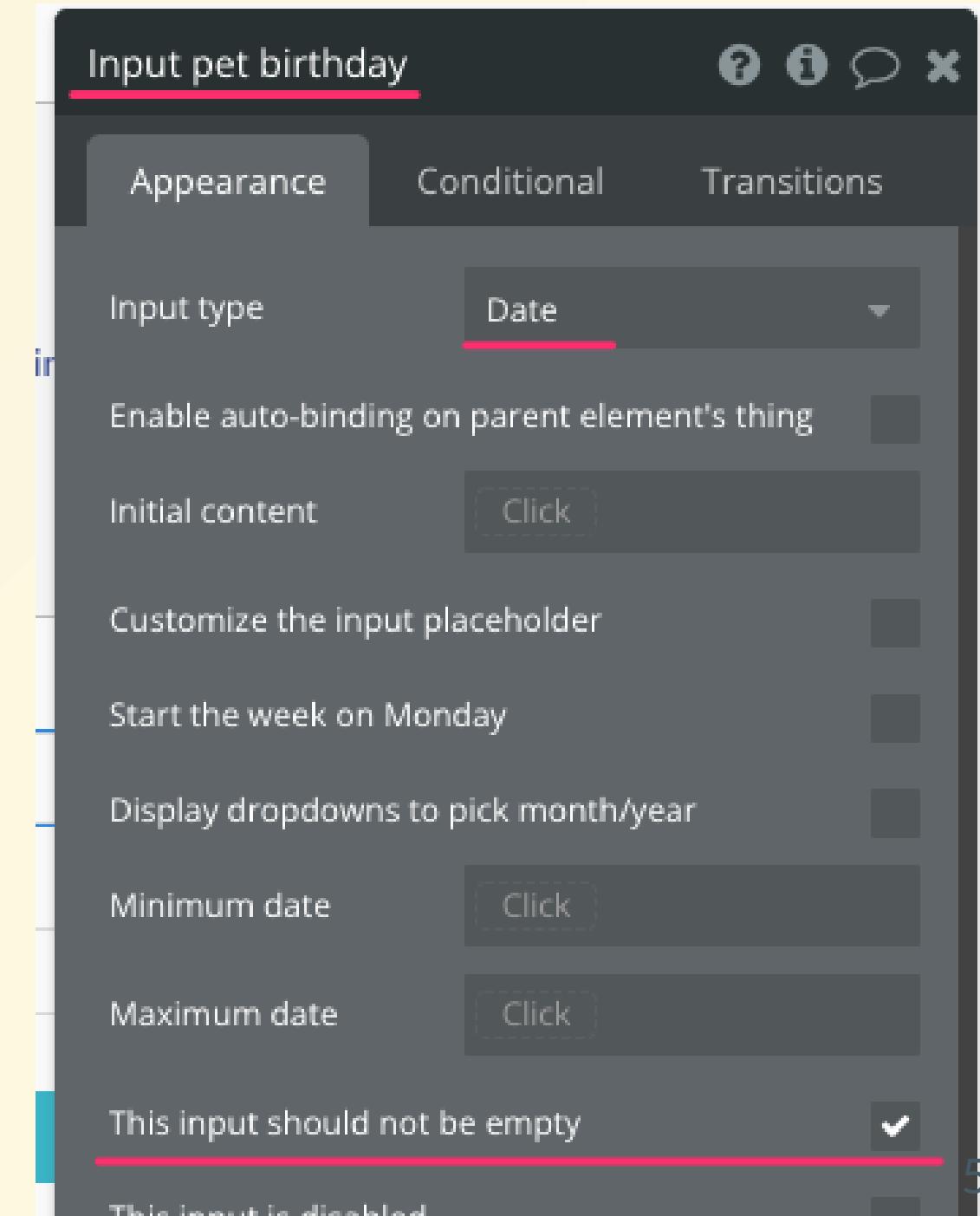
- Page:** pet_register
- UI Builder:** Selected
- Responsive:** Selected
- Design:** Selected tab, showing the Elements tree and a search bar.
- Workflow:** Shows Visual elements and Containers.
- Data:** Shows Input forms, including Input and Multiline Input.
- Styles:** Shows Checkbox.
- Plugins:** Shows Dropdown, Search Box, Radio Buttons, Slider Input, Date/Time Picker, and Picture Uploader.
- Settings:** Shows Log.
- Logs:** Shows Log.

The **Date/Time Picker** element is highlighted with a red underline in the Plugins section.

- こちらも要素をダブルクリックして要素の詳細を設定します
 - **Input type** に Date を選択することで日付入力のみとなります
 - ここで Date & Time を選択することで日付と時刻の入力が可能です
 - 今回は Date とします



- This input should not be empty
はチェックをつけて必須にします
- 要素の名前として Input pet birthday を指定します



ペットの性別

- 最後にペットの性別を選択するための要素を配置します
- 左パネルから **Dropdown** を選択し、右パネル上でドラッグして要素を配置します

.b Page: pet_register

UI Builder Responsive

Design Elements tree

Workflow Search for new elements...

Data Visual elements

Containers

Input forms

Input

Multiline Input

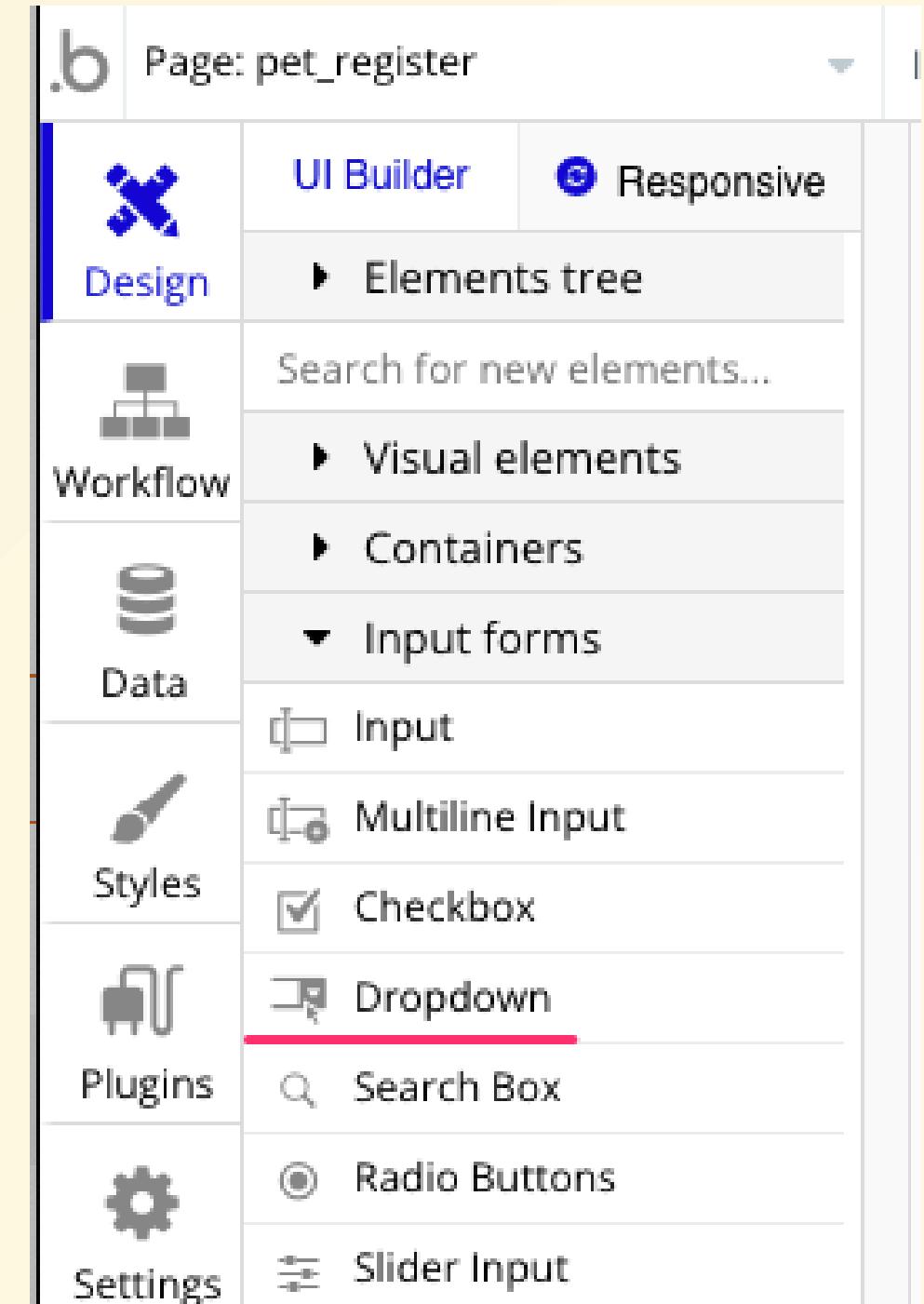
Checkbox

Dropdown

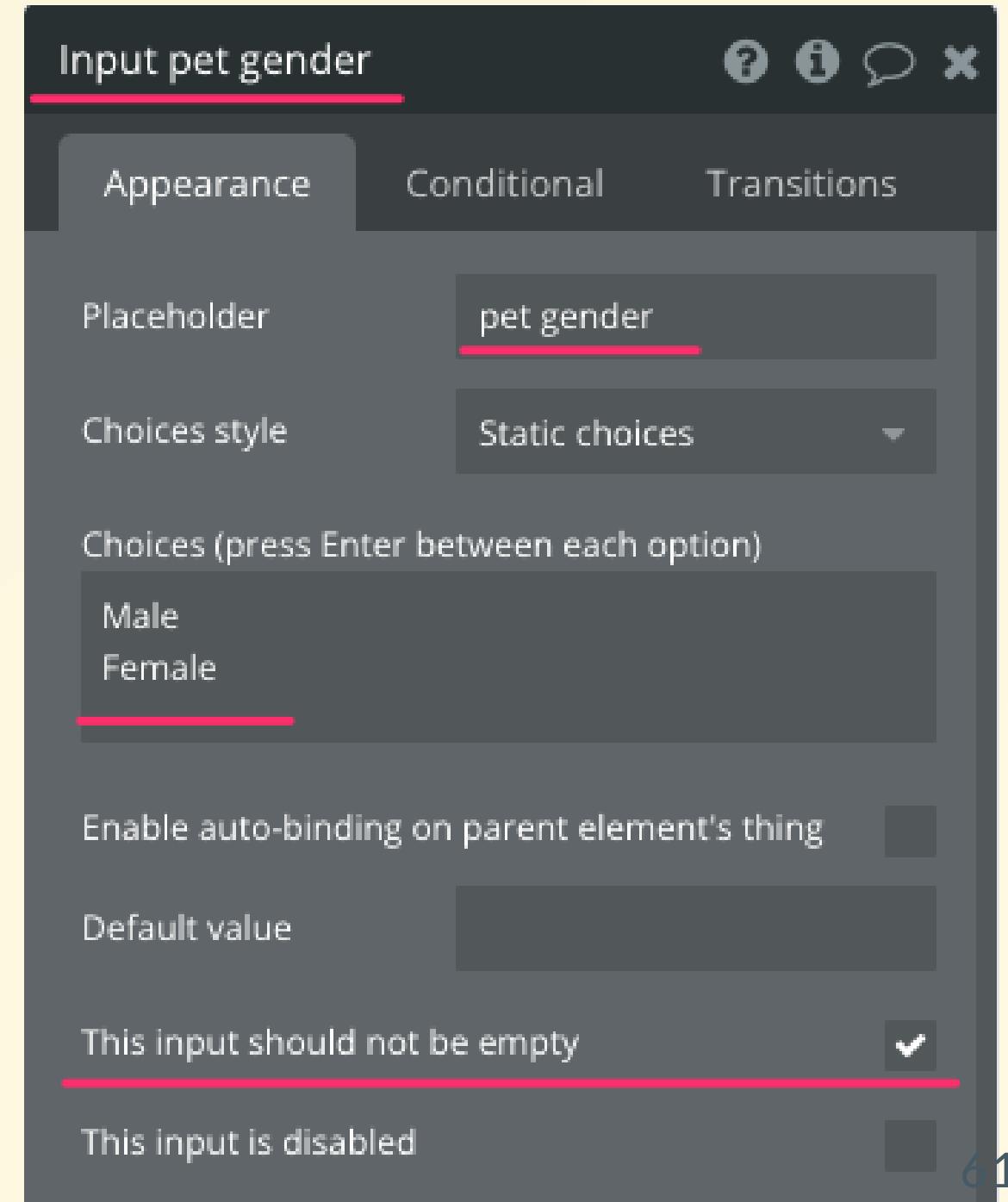
Search Box

Radio Buttons

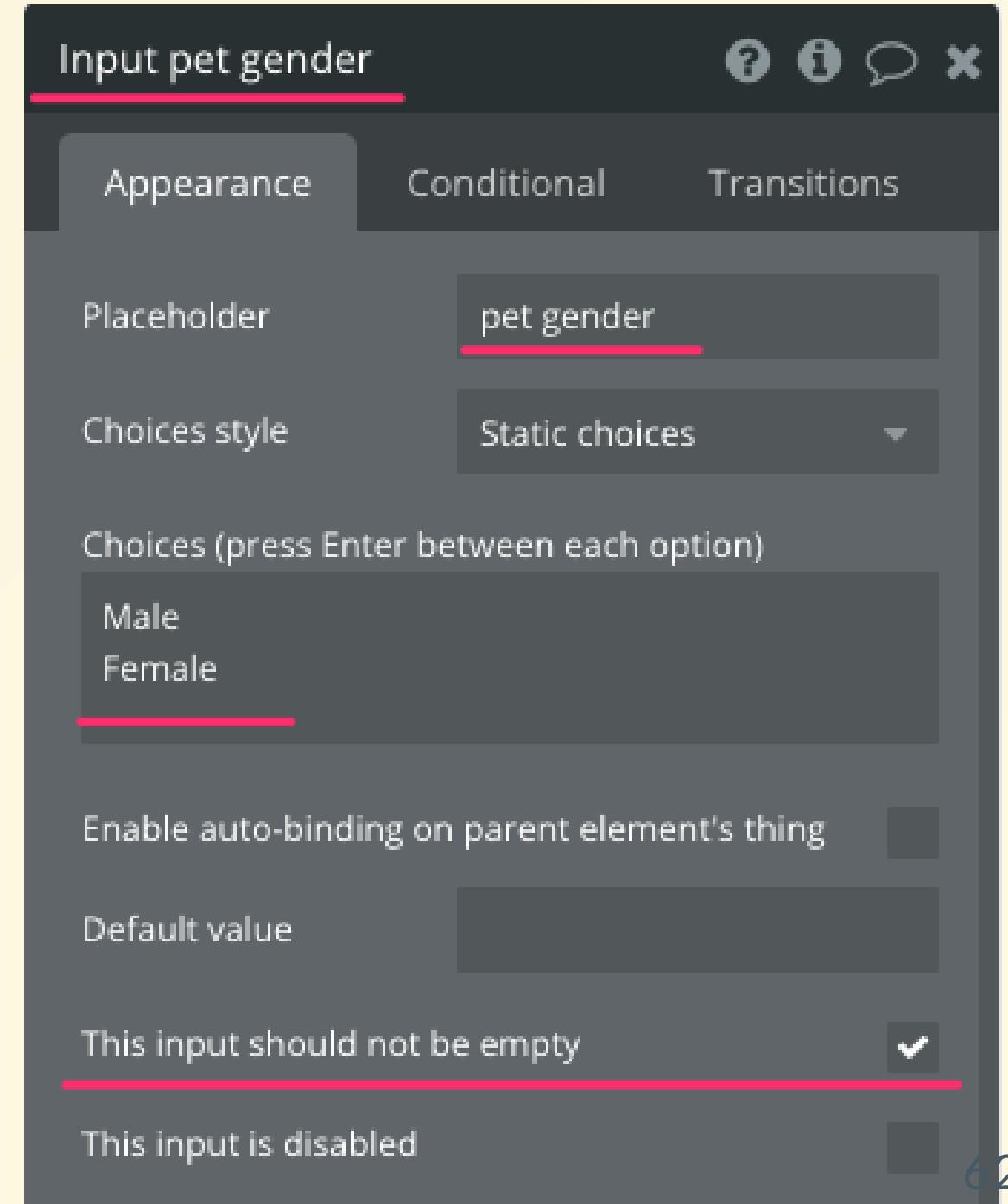
Slider Input



- こちらも要素をダブルクリックして要素の詳細を設定します
 - Placeholder に「pet gender」と入力します
 - Choices に "Male" と入力したら Enter キーで改行し、次の行に "Female" と入力します

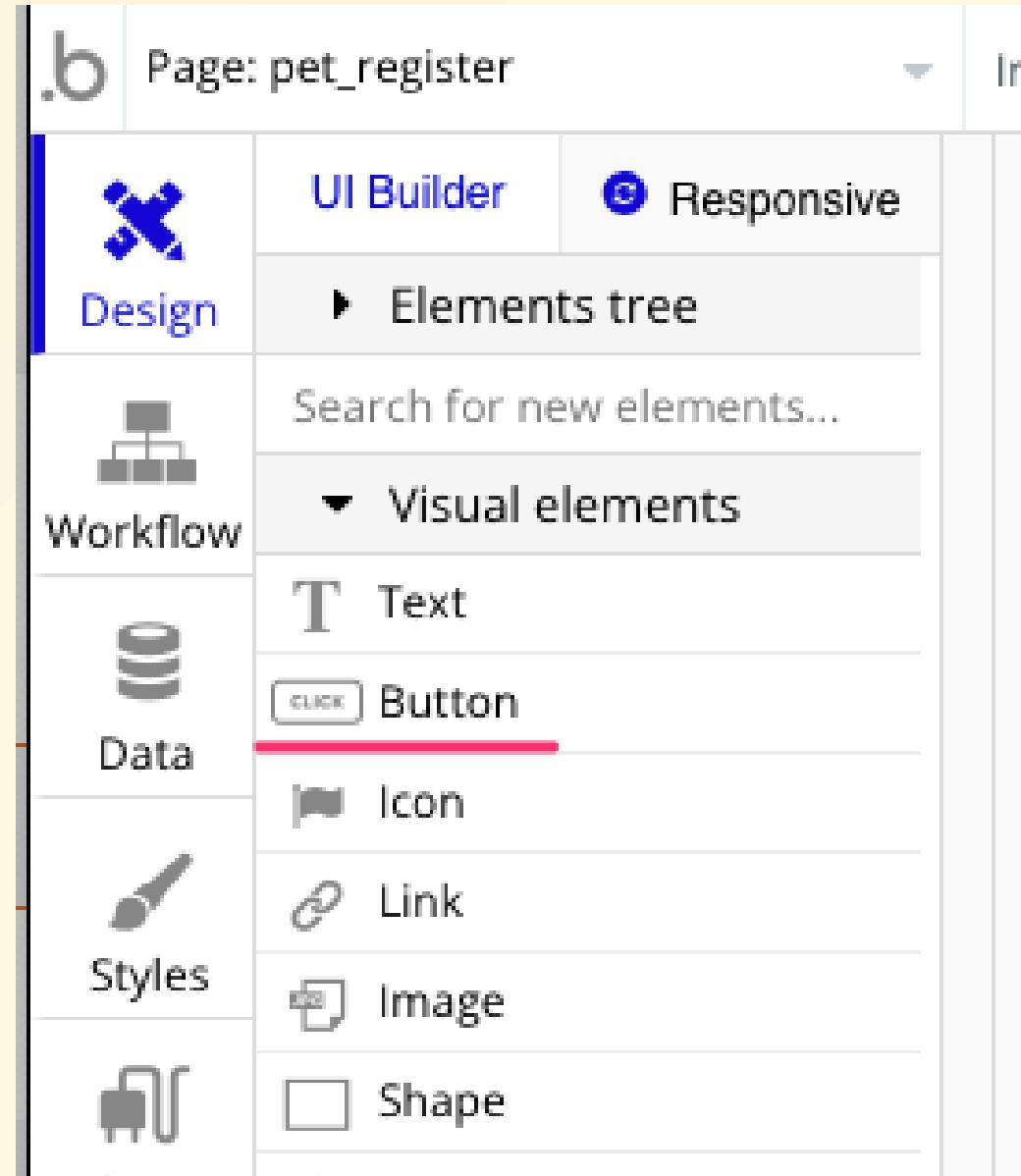


- This input should not be empty
はチェックをつけて必須にします
- 要素の名前として Input pet gender を指定します

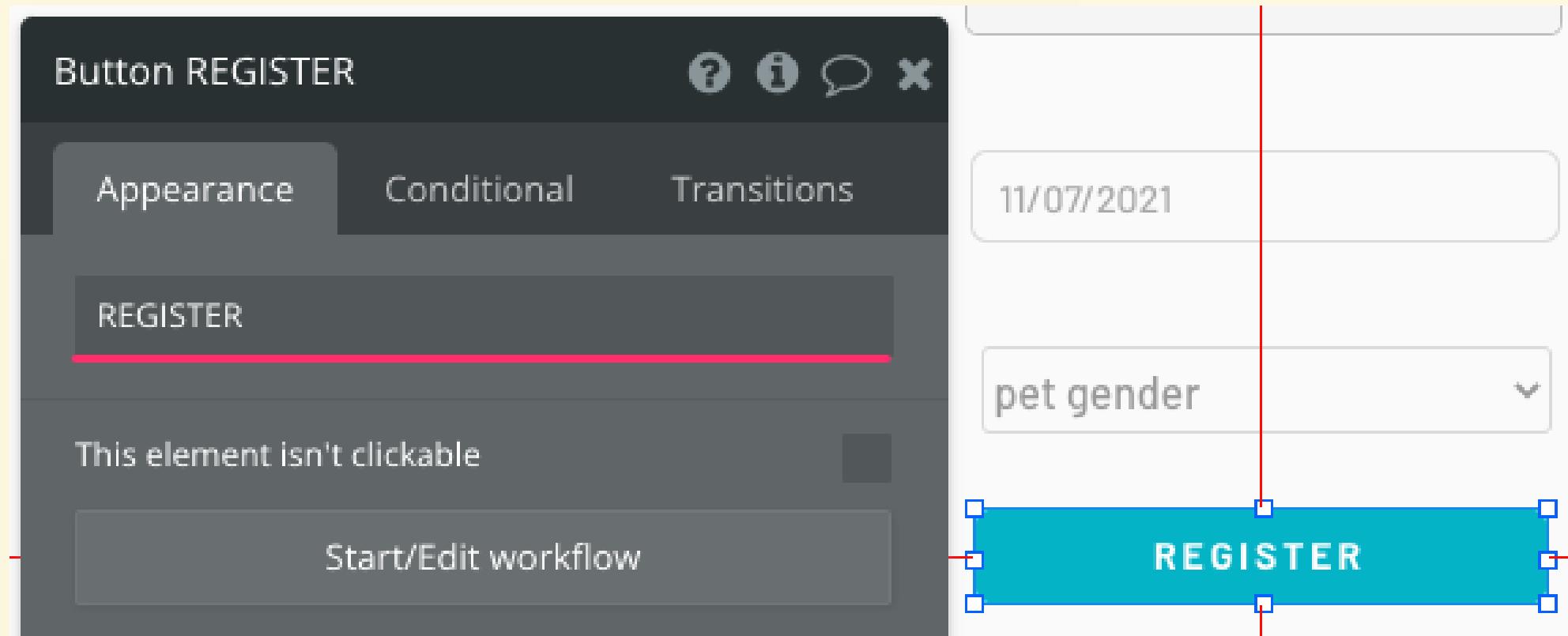


登録ボタン

- ここまでペット情報の入力ができたら登録ボタンを配置します
- 左パネルの **Visual elements** の中にある **Button** を選択し、右パネルでドラッグして要素を配置します

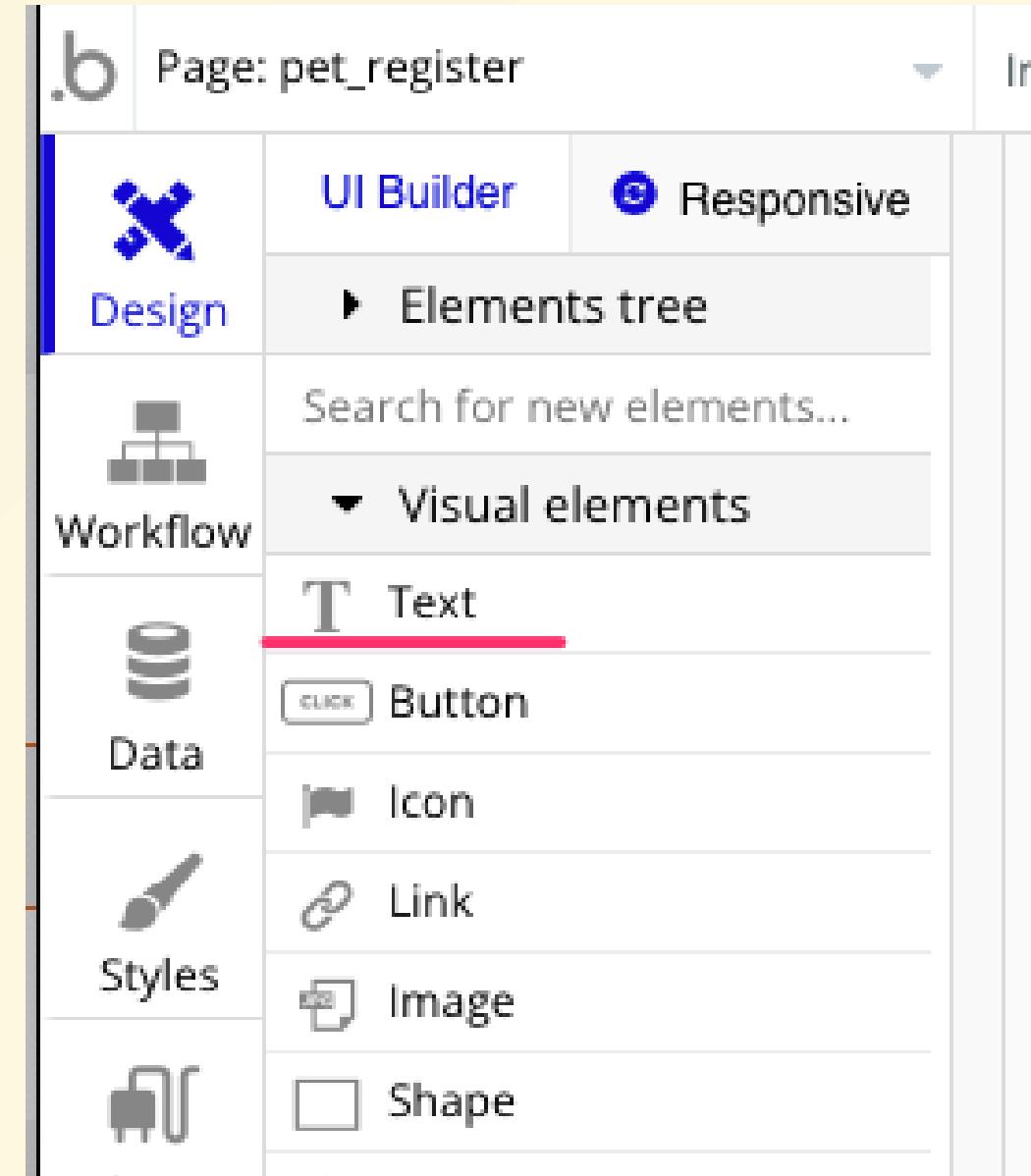


- こちらも要素をダブルクリックして要素の詳細を設定します
 - Appearance タブの上部にある「...edit me...」と入力されている部分を編集して「REGISTER」と入力します

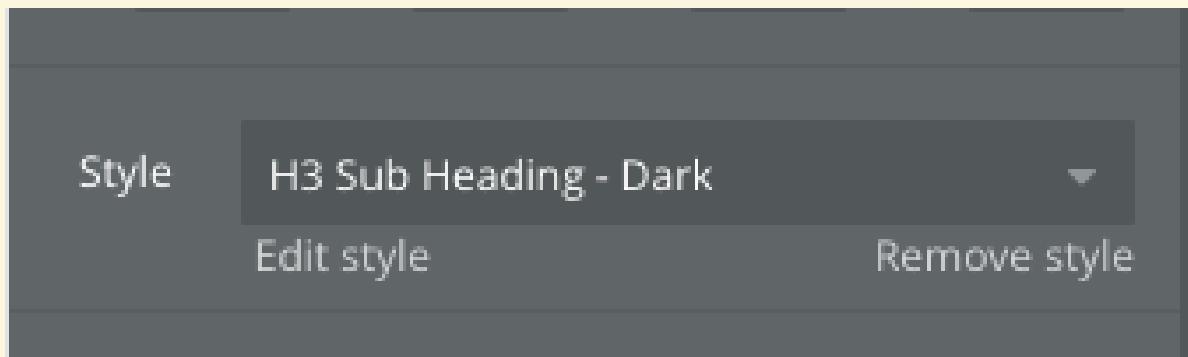


各入力要素にラベルをつけましょう

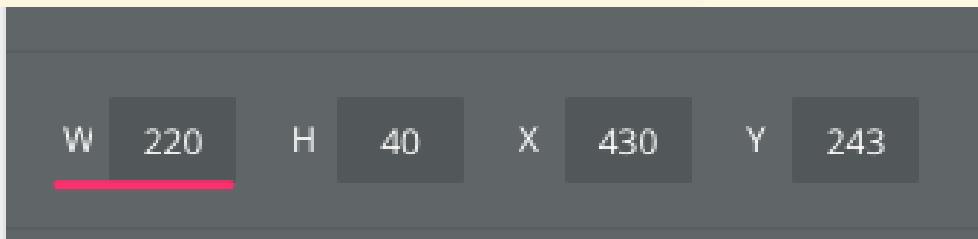
- Adalo の時の画面のよう
に、各要素の左上にラベル
を用意しましょう
- 左パネルの
Visual elements の中にあ
る **Text** を選択し、右パネ
ルでドラッグして要素を配
置します



- 配置する際、高さは 30px 以上になるように配置してください
- そうしないと、要素の中身が正しく表示されないためです！
- こちらも要素をダブルクリックして要素の詳細を設定します
 - Appearance タブの上部にある「...edit me...」と入力されている部分を編集して「Name」と入力します
 - ダイアログ中断にある **Style** に **H3 Sub Heading - Dark** を選択します



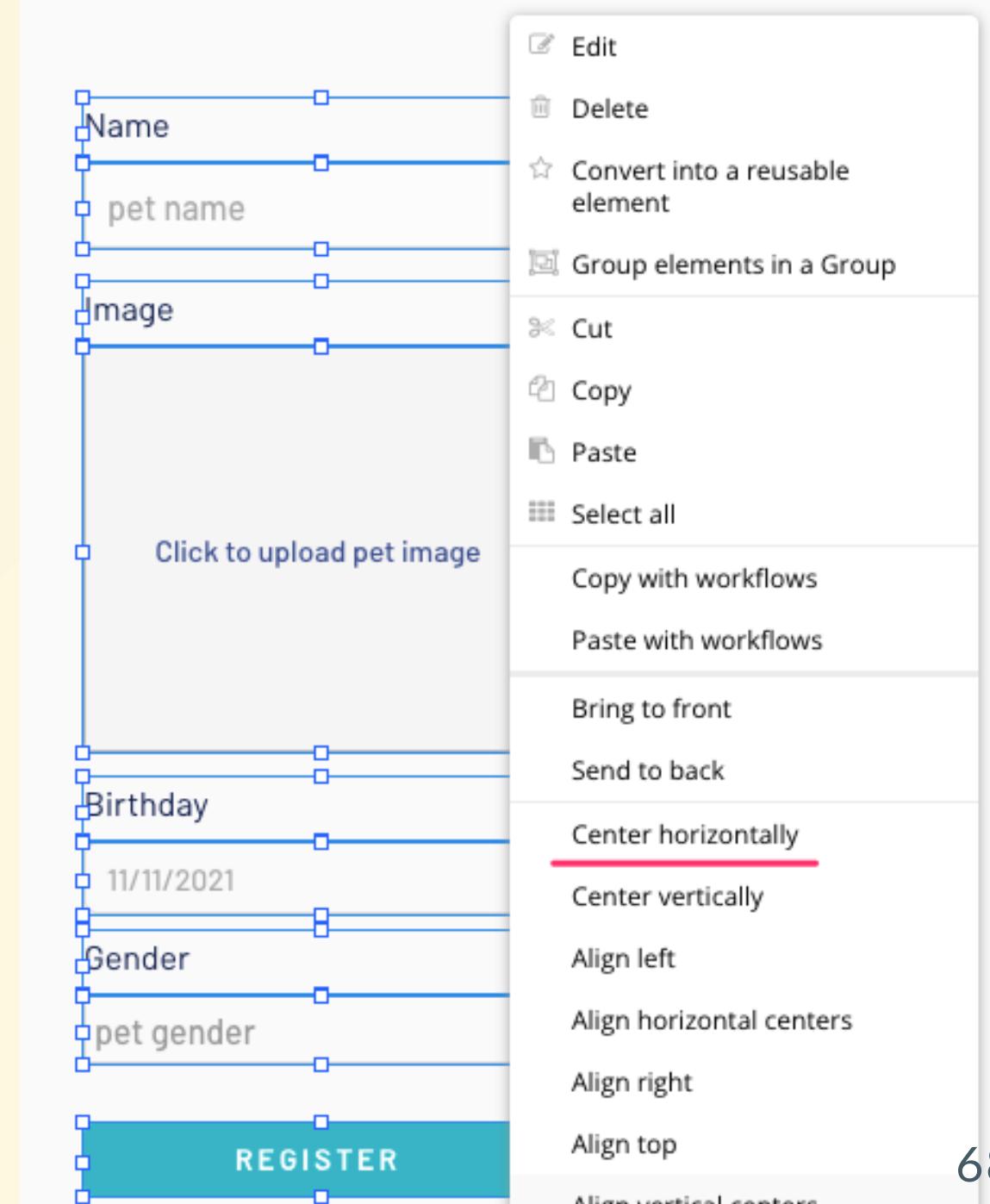
- 同じ要領でその他のラベルもつけてみましょう
- 同じ手順でもよいですが、先ほど作った Name のラベルをコピーして使い回すことで手間を省けます
 - コピー (Ctrl + C) してペースト (Ctrl + P) し、ラベルの内容だけ書き換える
- 最後にすべての要素の幅を 220px に統一してください
 - 各要素をダブルクリックしたときのダイアログから **W** (Width) の値を 220 に統一します
 - これは画面表示したときの見栄えを整えるためです



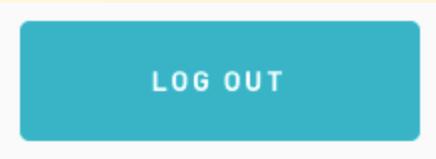
最後に要素を整理してプレビュー

- 右パネルに配置した (Header を除く) すべての要素をドラッグで選択し右クリックから

Center horizontally を選択し、入力フォームの横位置を中央に統一します



- ここまで出来たら画面右上にある **Preview** をクリックしてプレビューを表示して動きを見てみましょう！
- この時、ログイン中の状態であることを確認してください！
 - これは、この後の一覧表示と連携するときに必要な情報となるためです



LOG OUT

- ヘッダーの右上のボタンが
- すべての要素が表示されていますか？
- すべての入力フォームに値を入力できますか？
- 画像を指定すると表示されますか？

The image displays two side-by-side screenshots of a mobile application's pet registration screen. Both screenshots show a white background with black text and input fields.

Left Screenshot (Empty Form):

- Name: pet name
- Image: Click to upload pet image
- Birthday: 11/12/2021
- Gender: pet gender
- Buttons: A large teal "REGISTER" button at the bottom.

Right Screenshot (Filled Form):

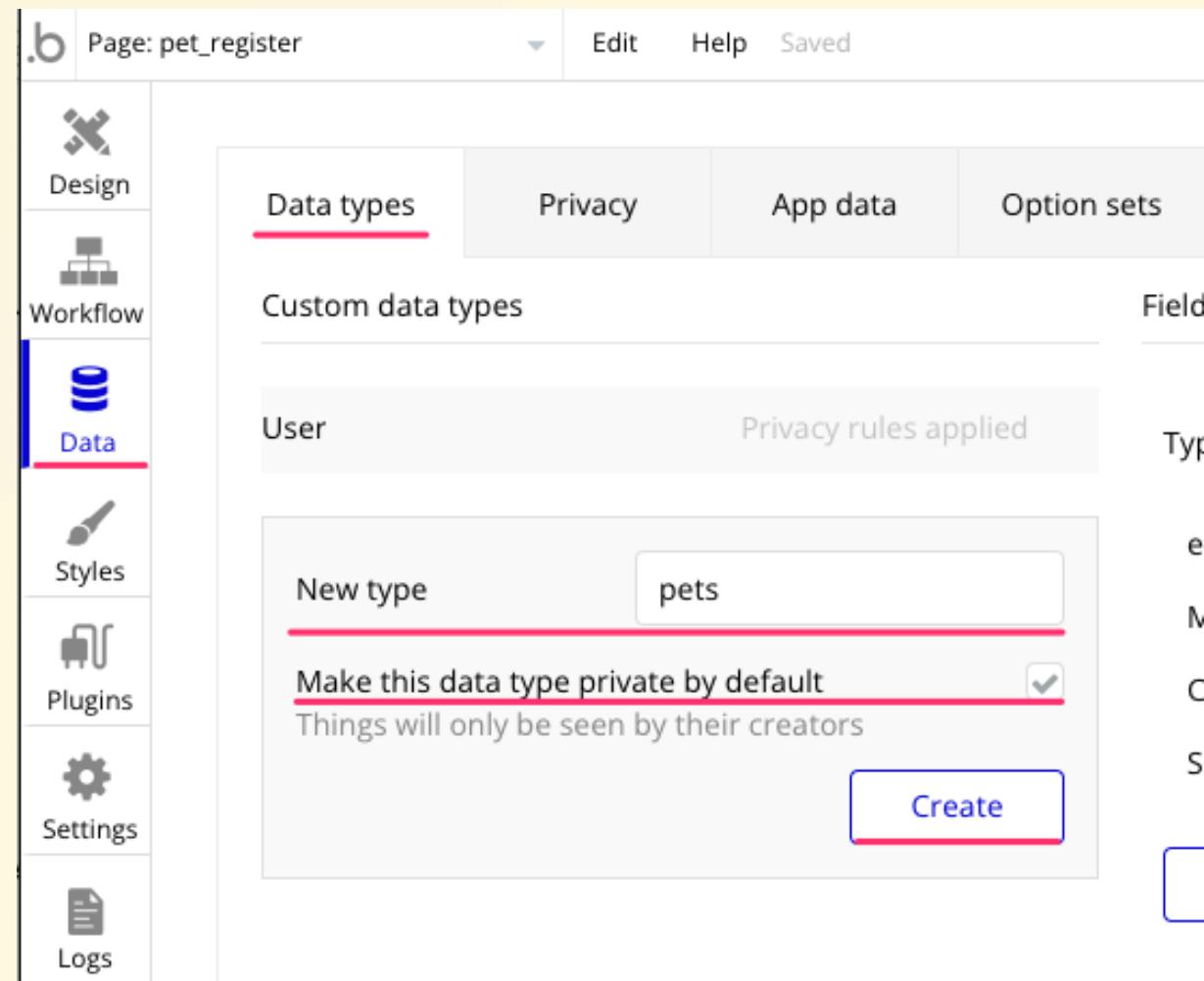
- Name: コロロ
- Image: An image of a Shiba Inu dog standing on a bed of fallen leaves.
- Birthday: 10/20/2020
- Gender: Male
- Buttons: A large teal "REGISTER" button at the bottom.

データベースとつなげてみよう

- ここまで画面レイアウトが整ったので、いよいよデータベースとつなげてみましょう

まずはペットの情報を保存するための箱を用意します

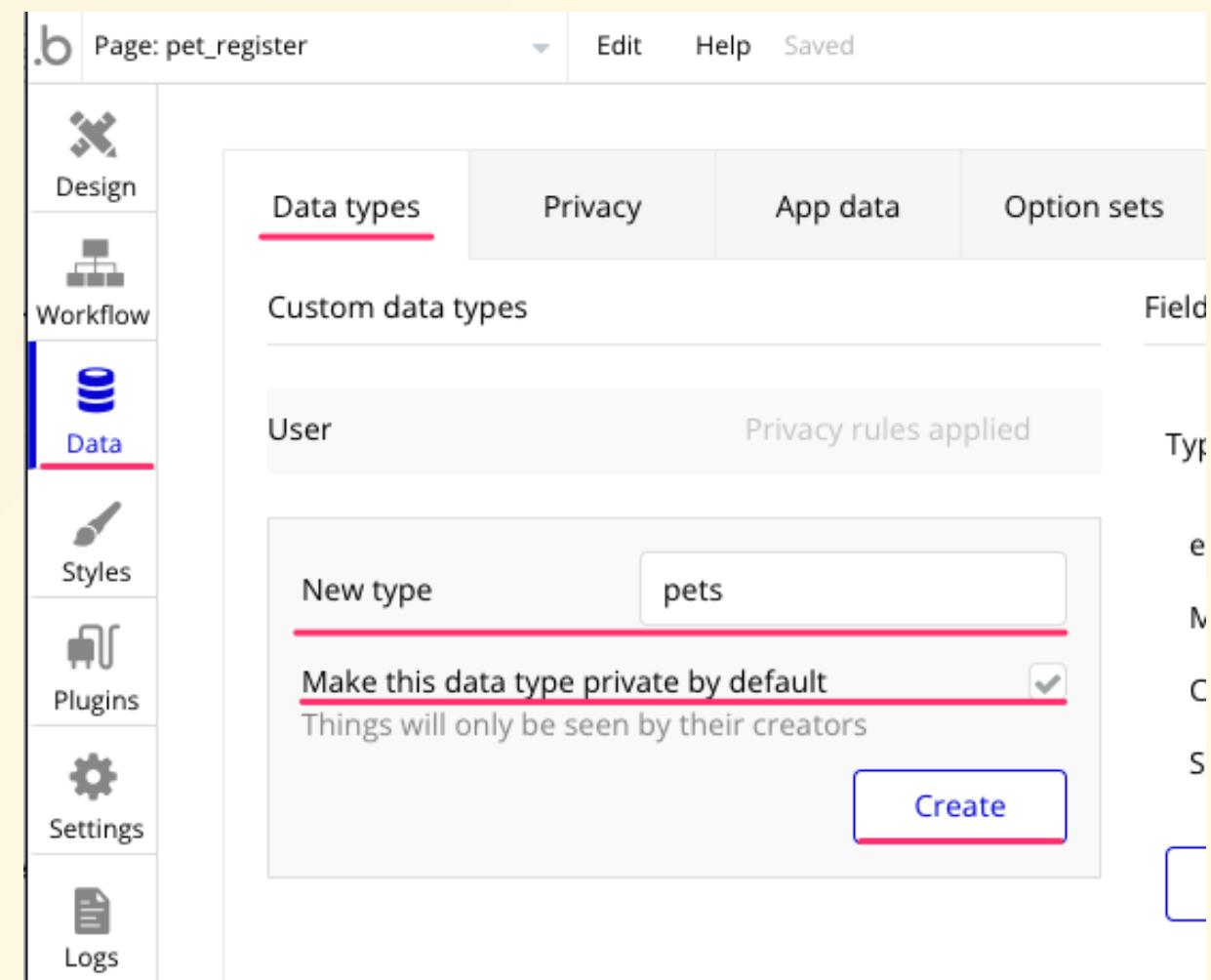
- 左パネルのタブから Data を選択
- すると Data types タブが有効になっていると思うのでその下にある Custom data types の中から New type と書かれたテキストボックスに「Pets」と入力してください



- そして

Make this data type
private by default

のチェックをつけて **Create**
ボタンをクリックします



ちなみに...

- Adalo と Bubble ではデータベースに関する各要素の呼び方が一部異なります
- Type: データの型（箱）を定義したもの
 - Adalo でいうところの Collection
- Field: データの型を表現するための要素
 - Adalo でいうところの Property

- 作成した Pets の type に対して要素 (field) を追加していきます
- Custome data types から Pets を選択
- 右側の Fields for type pets の下にある Create a new field をクリック

The screenshot shows the Low Platform interface for managing custom data types. The left sidebar has icons for Design, Workflow, Data (which is selected), Styles, Plugins, Settings, and Logs. The top navigation bar includes 'Edit', 'Help', 'Saved', and various status indicators like '8 issues'. The main area is titled 'Fields for type Pets'. It lists fields for the 'Pets' type, including 'Birthday' (date), 'Gender' (text), 'Image' (image), 'Name' (text), 'Creator' (User), 'Modified Date' (date), 'Created Date' (date), and 'Slug' (text). A 'Create a new field' button is at the bottom. On the left, there's a 'New type' section with a 'Create' button and a note about making it private by default.

Type name	Field Name	Type	Default Value	Action
Pets	Birthday	date		... trash
Pets	Gender	text		... trash
Pets	Image	image	Upload	... trash
Pets	Name	text		... trash
Pets	Creator	User	Built-in field	
Pets	Modified Date	date	Built-in field	
Pets	Created Date	date	Built-in field	
Pets	Slug	text	Built-in field	

- **Field name** には要素の名前を入力します

Create a new field

Field name

Field type

This field is a list (multiple entries)

CREATE Cancel

- **Field type** には要素の種類（テキスト、数字、日付など）を選択します

Create a new field

Field name

Field type

This field is a list (multiple entries)

CREATE Cancel

- Adalo では Relationship から type 同士の関連を指定できました
- Bubble では **Field type** の選択肢の中に、登録されている type も表示されているので、そこで関連付けたい type を選択することで Adalo と同様のことが実現できます

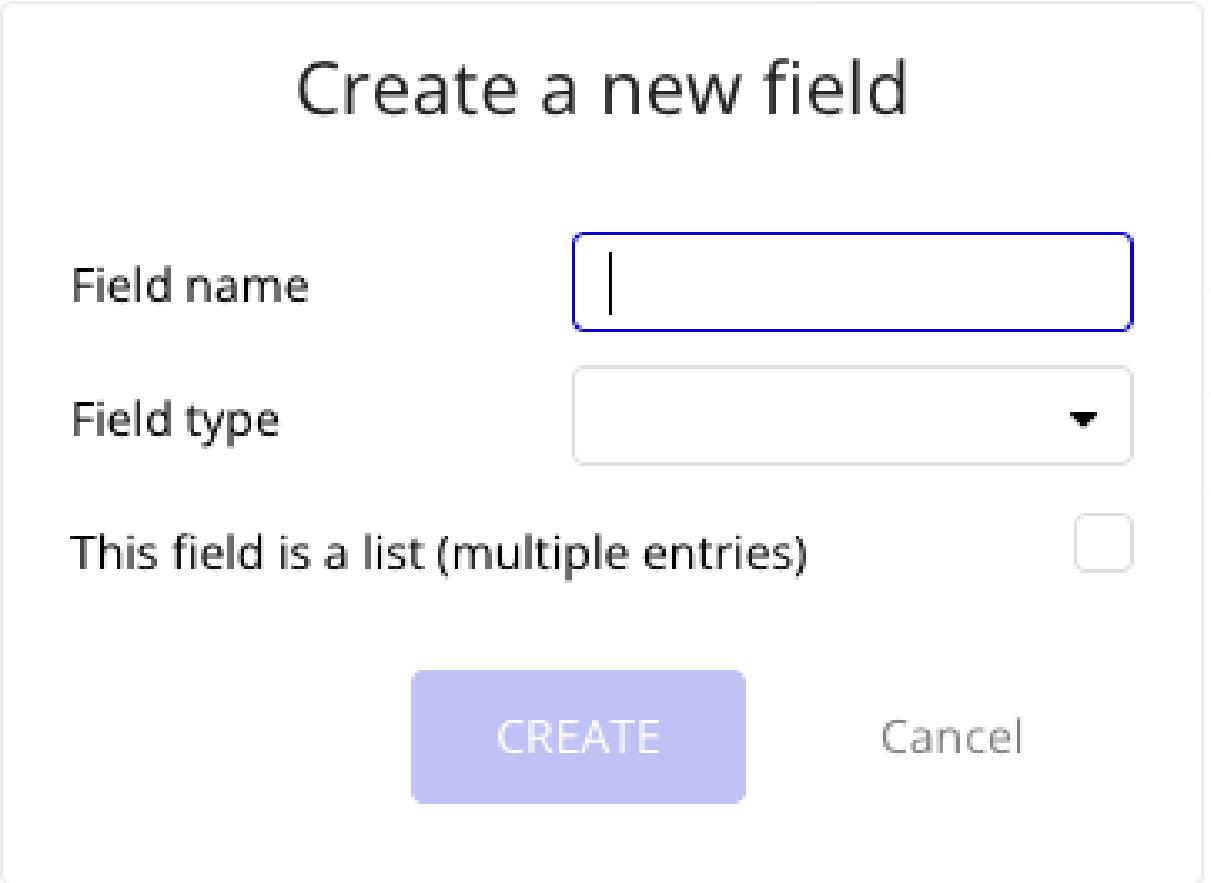
Create a new field

Field name

Field type

This field is a list (multiple entries)

CREATE Cancel



それでは Pets type に必要な field を追加していきましょう

- Name: text
- Image: image
- Birthday: date
- Gender: text

Fields for type Pets

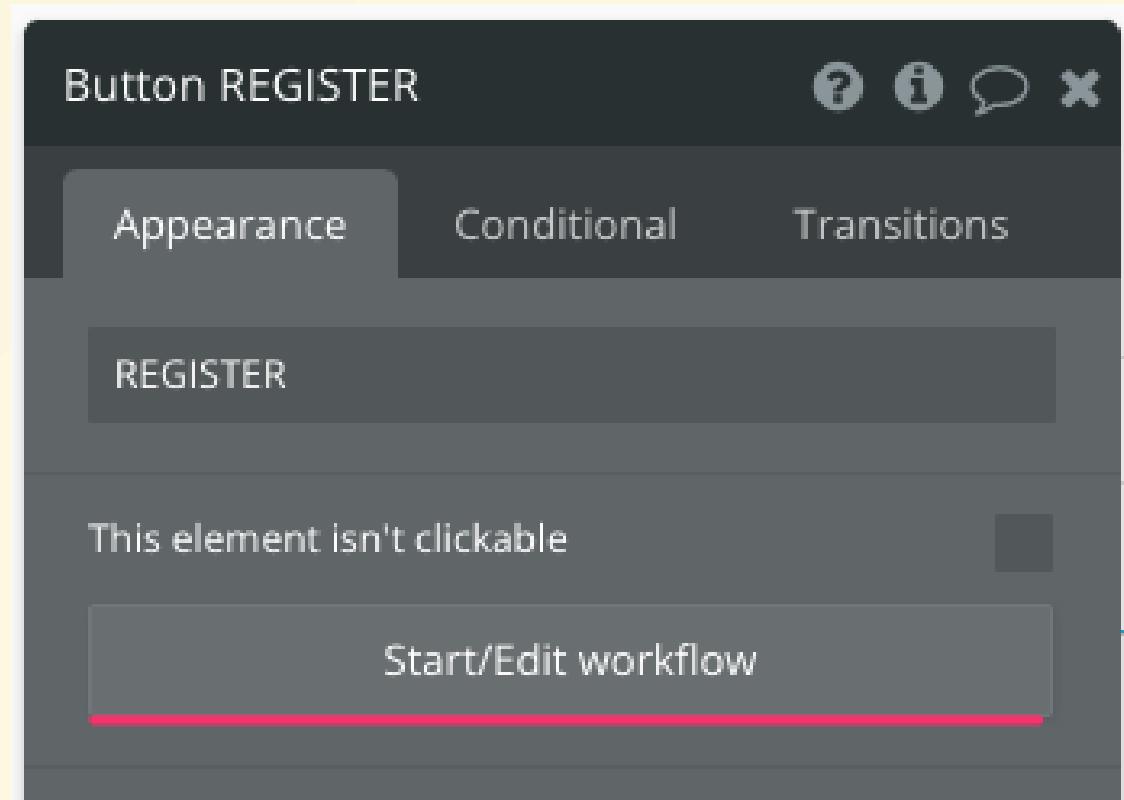
Type name	Pets	
Birthday	date	default
Gender	text	default
Image	image	default Upload
Name	text	default
Creator	User	Built-in field
Modified Date	date	Built-in field
Created Date	date	Built-in field
Slug	text	Built-in field

[Create a new field](#)

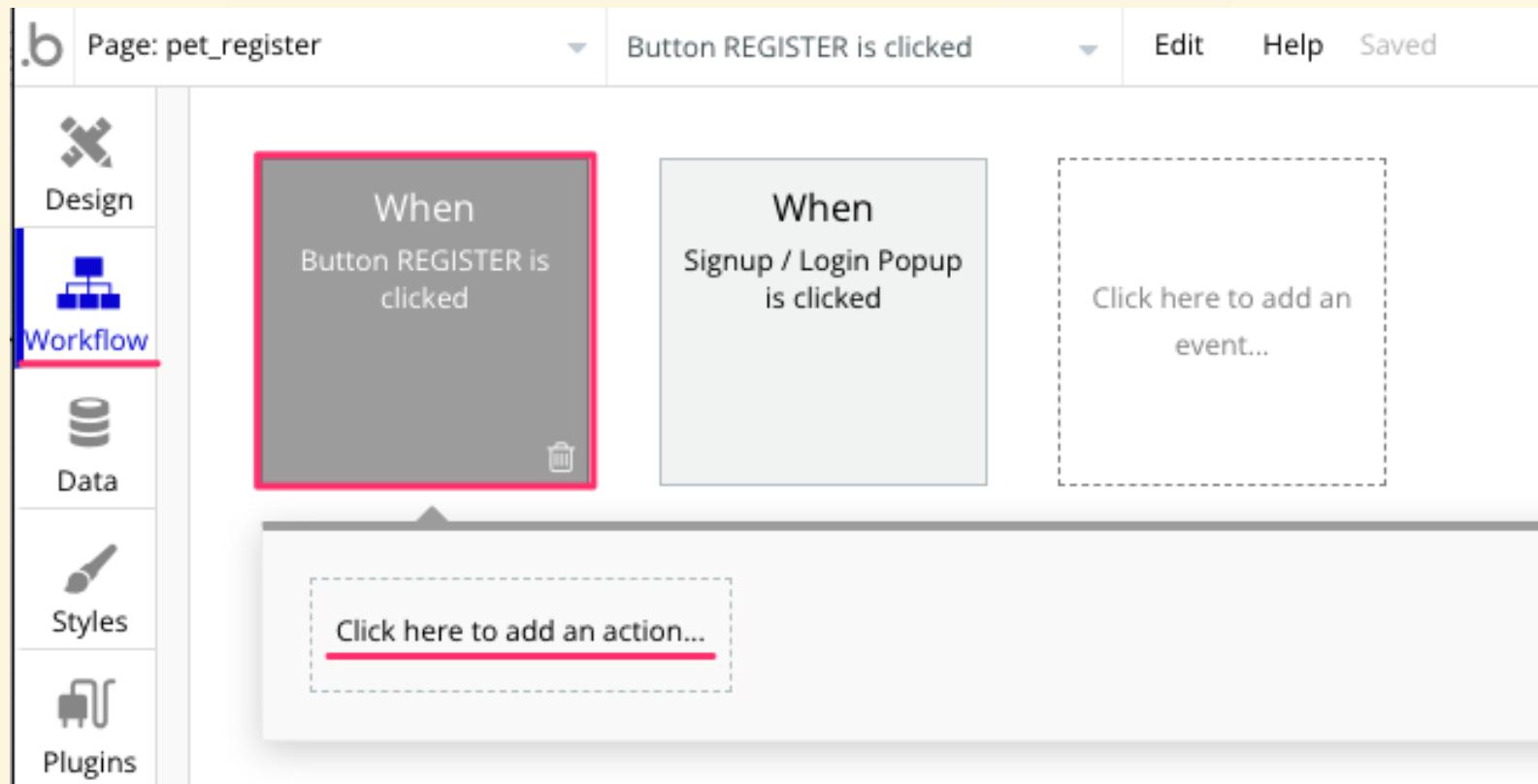
登録ボタンをクリックしたらペットを登録できるよう にする

- ペットの情報を格納するための箱が準備だったので、いよいよ登録ボタンに押したときにデータベースに保存する動きをつけていきます
- Bubble では動きを設定するのはすべて Workflow タブで操作をしていきます
- いきなり Workflow タブから設定も出来ますが、今回は動きをつける元となるボタンからワークフローの設定をしていきます

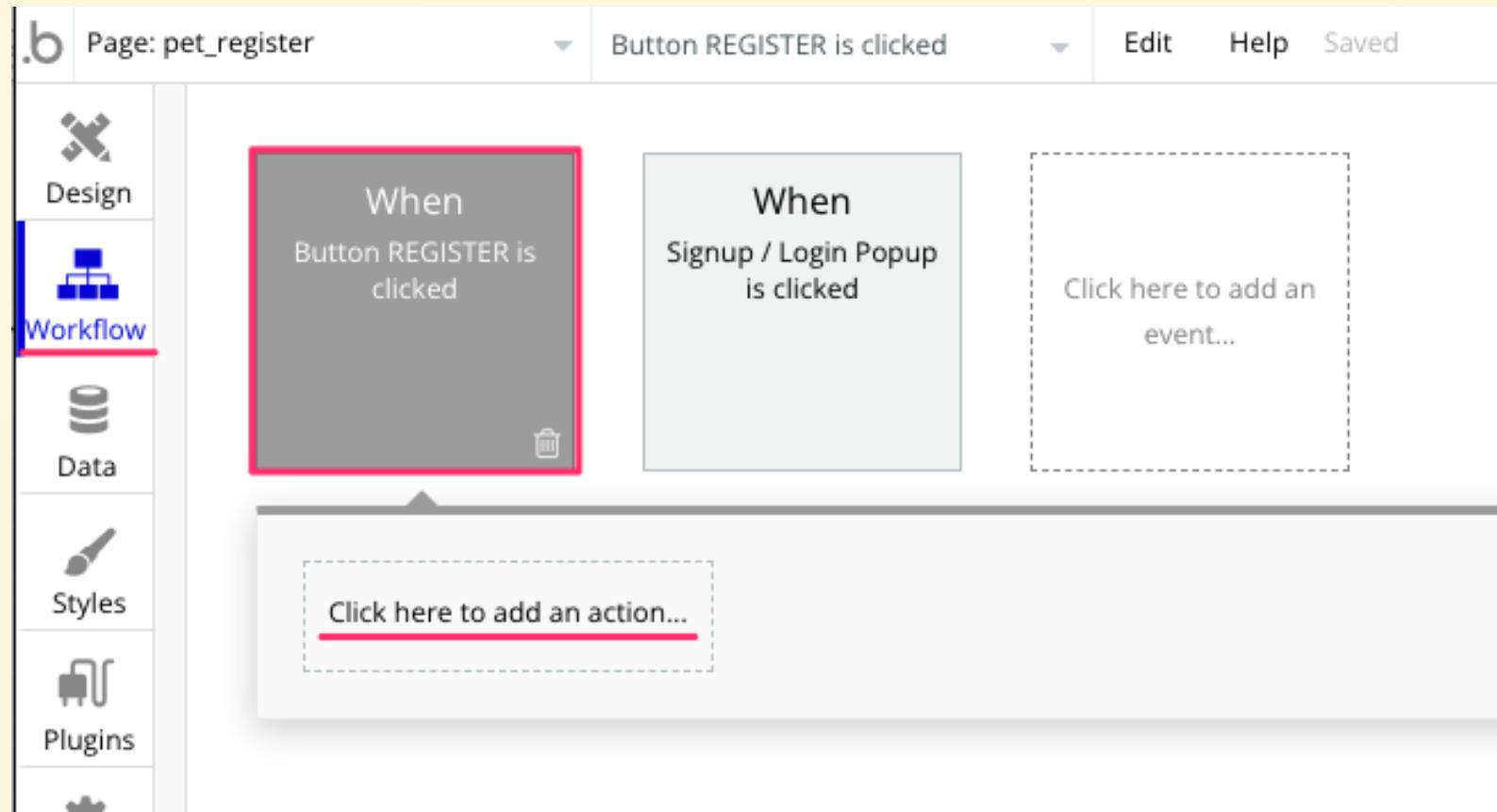
- 左パネルから Design タブを選択
- 右パネルからペット登録画面の "REGISTER" のボタンをダブルクリックします
- "Appearance" タブにある Start/Edit workflow をクリックします



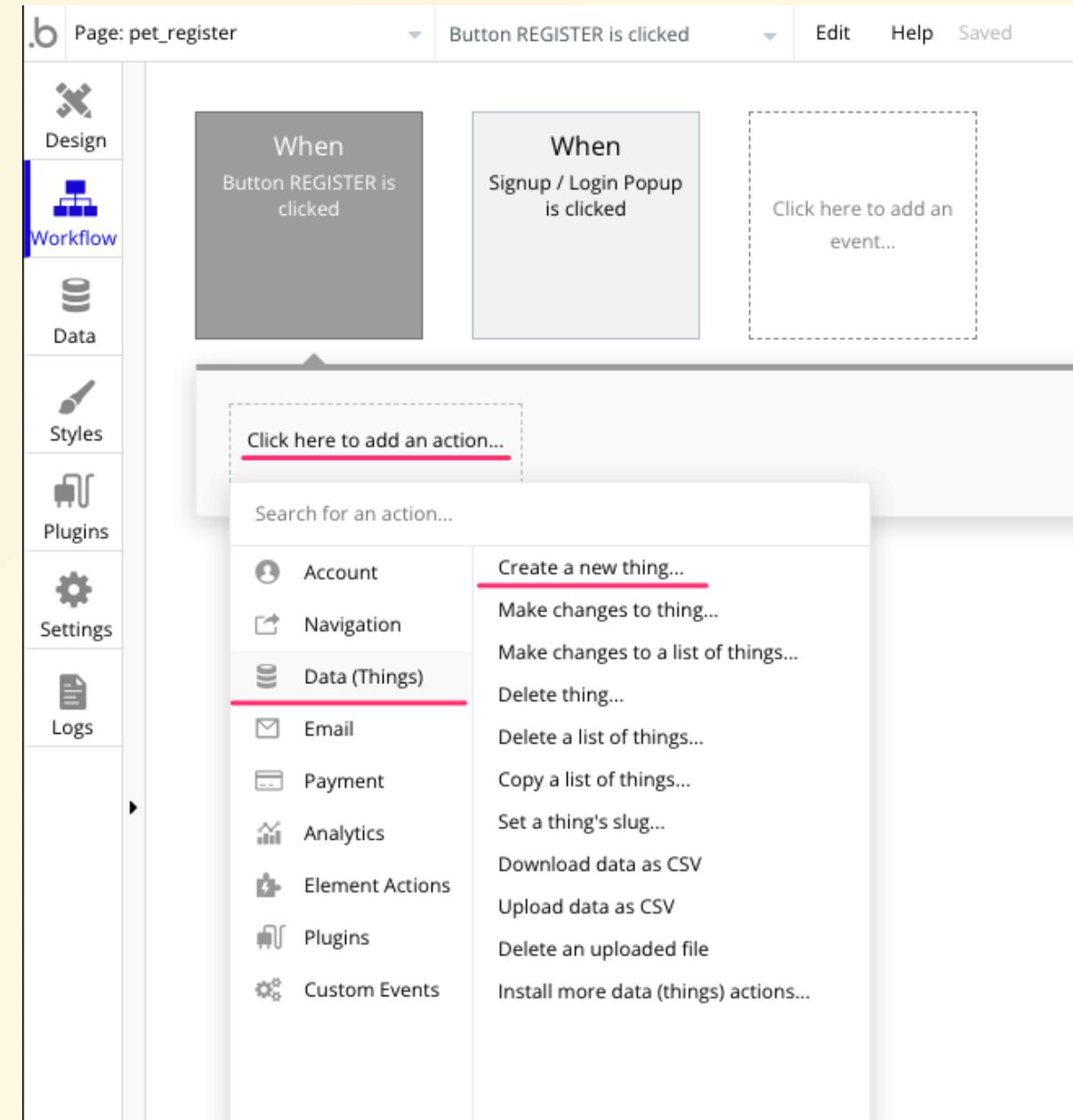
- すると Workflow タブに切り替わり、When の部分に
Button REGISTER is clicked と表示されていると思います
 - これは文字通り "REGISTER" ボタンを押したときのワークフロー定義になります



- なので、その下にある **Click here to add an action...** を選択して、ボタン押下時の振る舞いを設定していきます



- Click here to add an action...
- を選択すると、様々なアクションを指定できるポップアップが表示されます
- 今回のようにデータベースに関する操作（アクション）の場合 Data(Things) を選びます

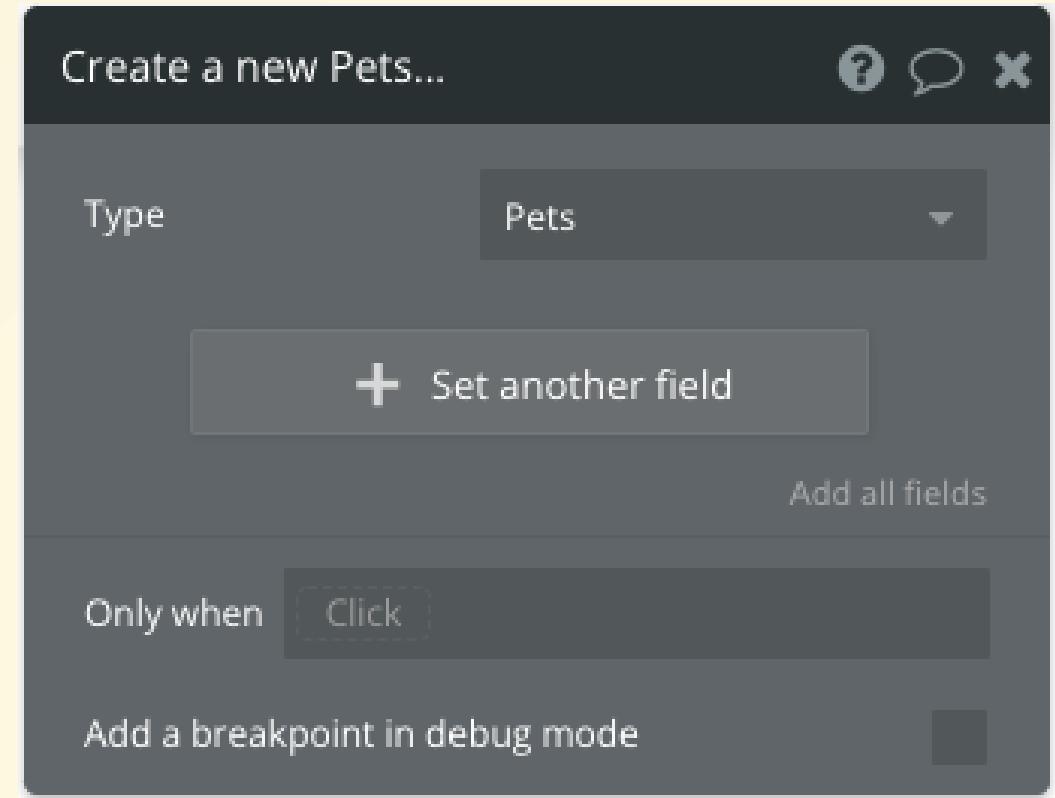


- すると下位要素がさらに表示されると思いますので、その中から **Create a new thing...** を選択します

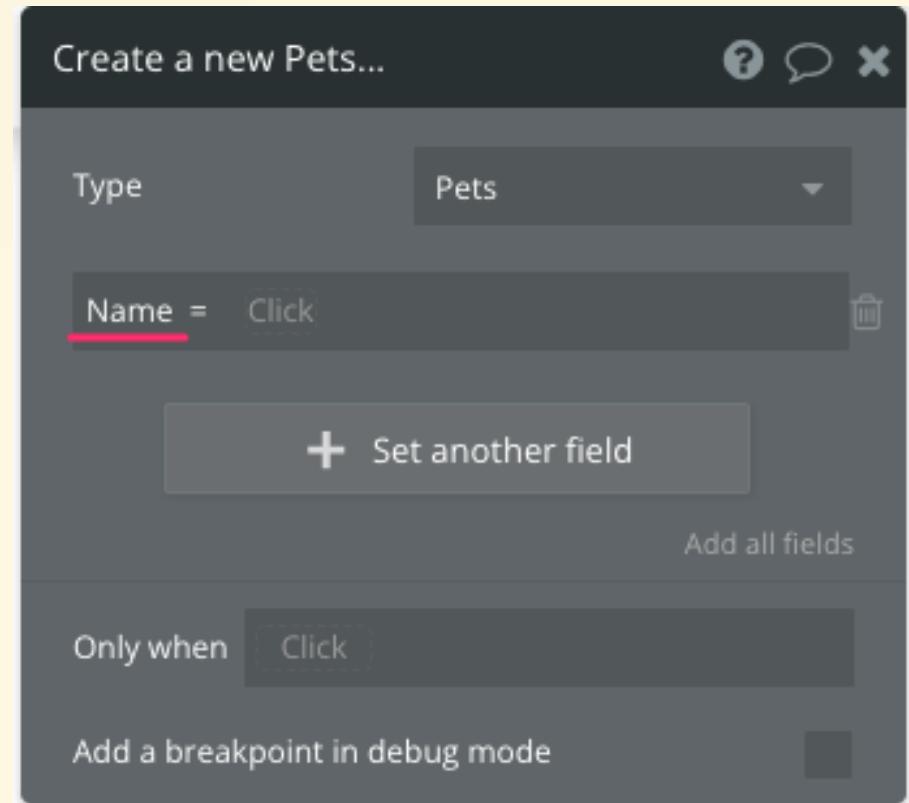
The screenshot shows the Bubble app's workflow editor. At the top, it says "Page: pet_register" and "Button REGISTER is clicked". The top navigation bar includes "Edit", "Help", and "Saved". On the left, there's a sidebar with icons for "Design", "Workflow" (which is selected), "Data", "Styles", "Plugins", "Settings", and "Logs". The main area has two "When" blocks: "Button REGISTER is clicked" (grey) and "Signup / Login Popup is clicked" (white). Below them is a red-bordered box labeled "Click here to add an action...". A search bar says "Search for an action...". A list of actions is shown in two columns:

Account	Create a new thing...
Navigation	Make changes to thing...
Data (Things)	Make changes to a list of things...
Email	Delete thing...
Payment	Delete a list of things...
Analytics	Copy a list of things...
Element Actions	Set a thing's slug...
Plugins	Download data as CSV
Custom Events	Upload data as CSV
	Delete an uploaded file
	Install more data (things) actions...

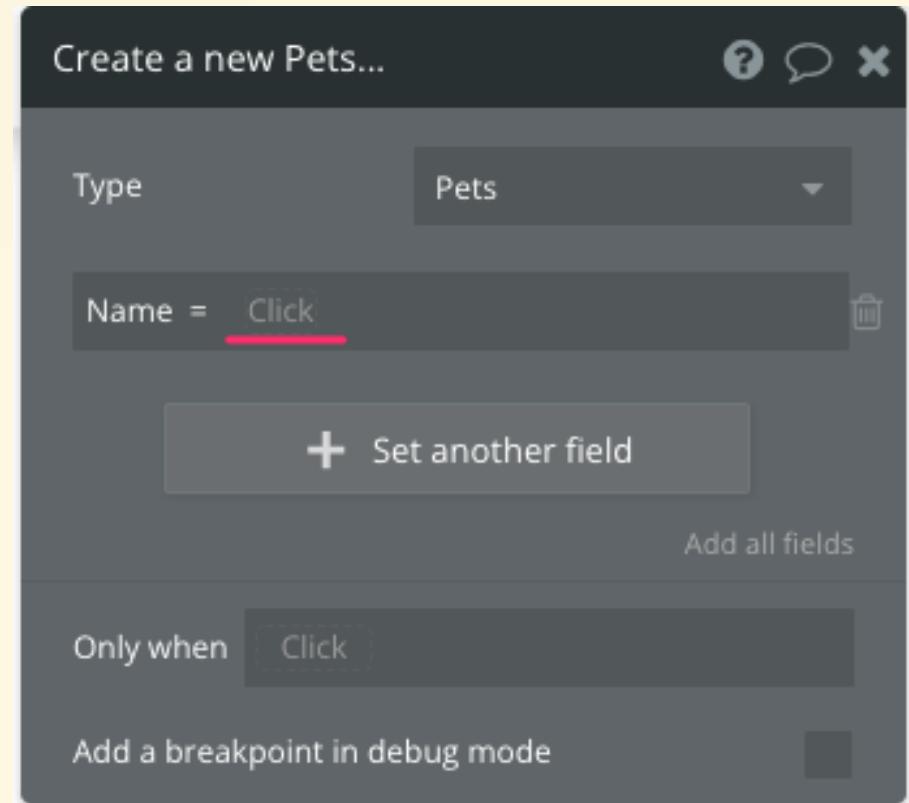
- **Create a new Pets...** のダイアログが表示されるので、今回登録するタイプを指定します
 - 今回は Pets ですね
- すると **Set another field** というボタンが表示されるので、タイトルの通り画面で入力された項目を Pets の "field" にセットしていきます



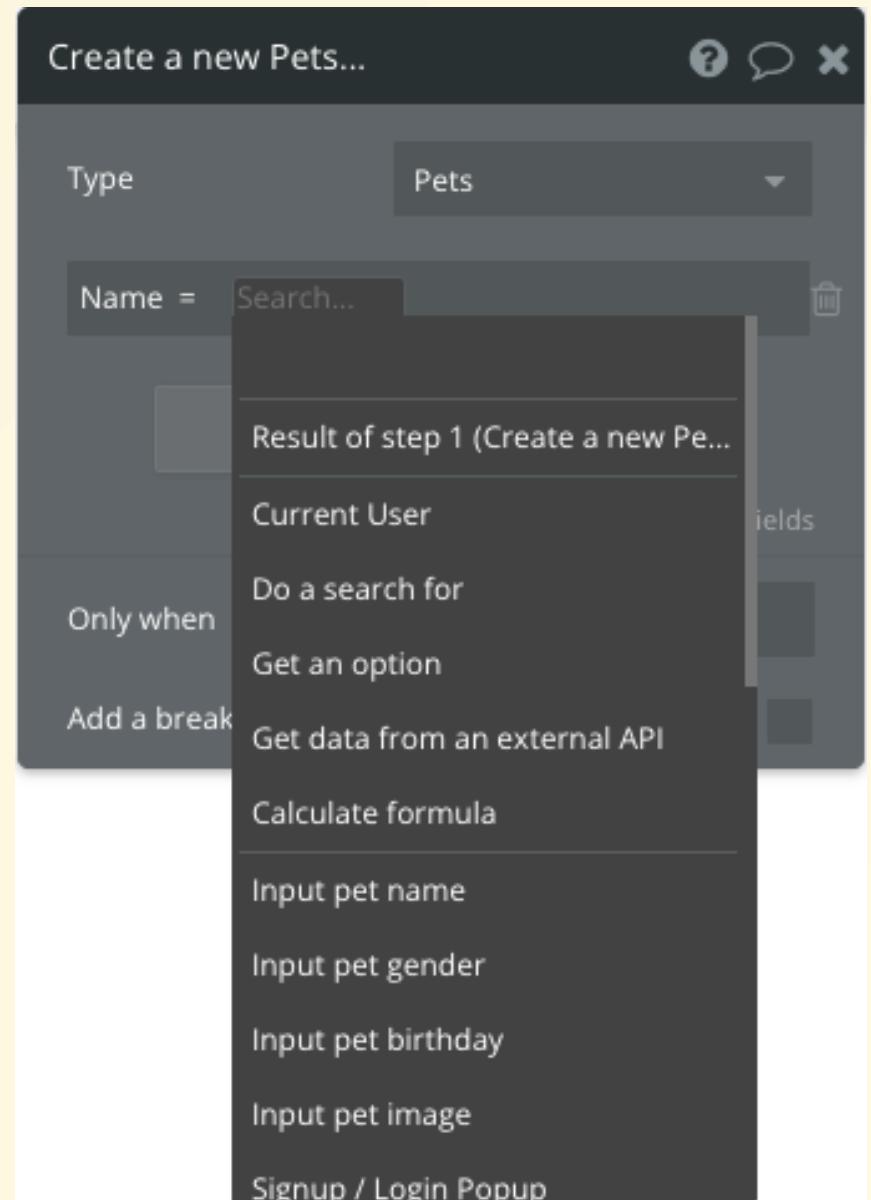
- ここでは Pets に保存したい項目の "field" 名を左辺に指定し、右辺に実際にその "field" に保存する値を指定します
 - まず、左辺に Name を選択します



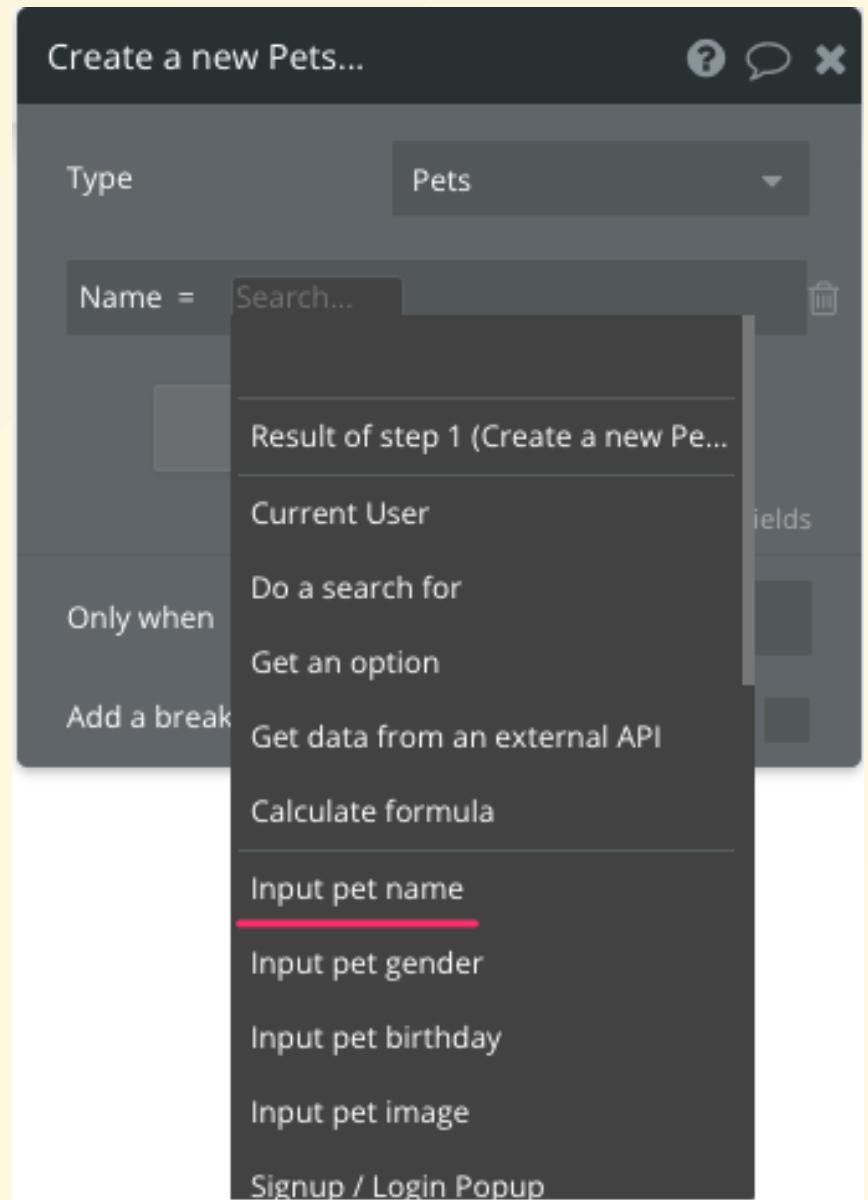
- 左辺を指定すると = に挟まれて今度は右辺に Click というのが表示されると思います



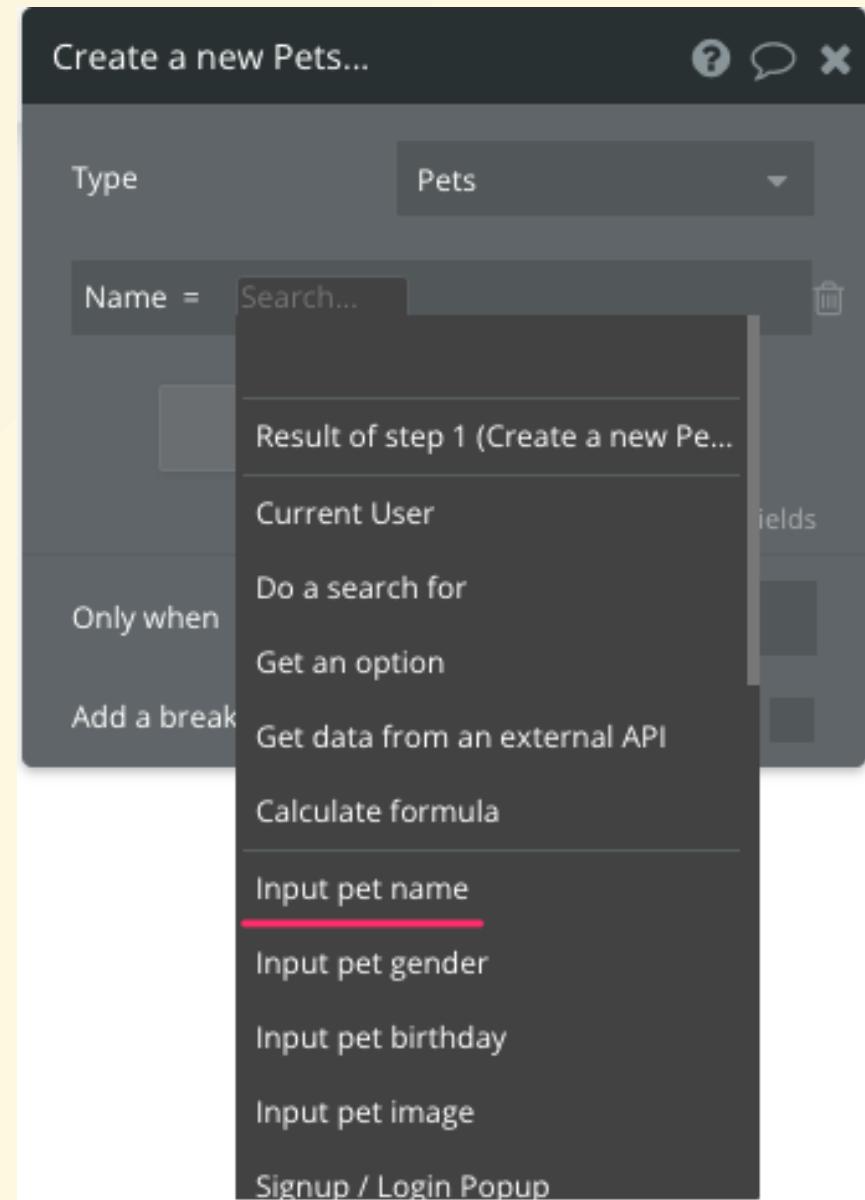
- ここをクリックすると何やらいろいろな選択肢が出てきます
- 詳細はここでは割愛しますが、左辺に選択されたフィールドに対して設定する対象を指定します



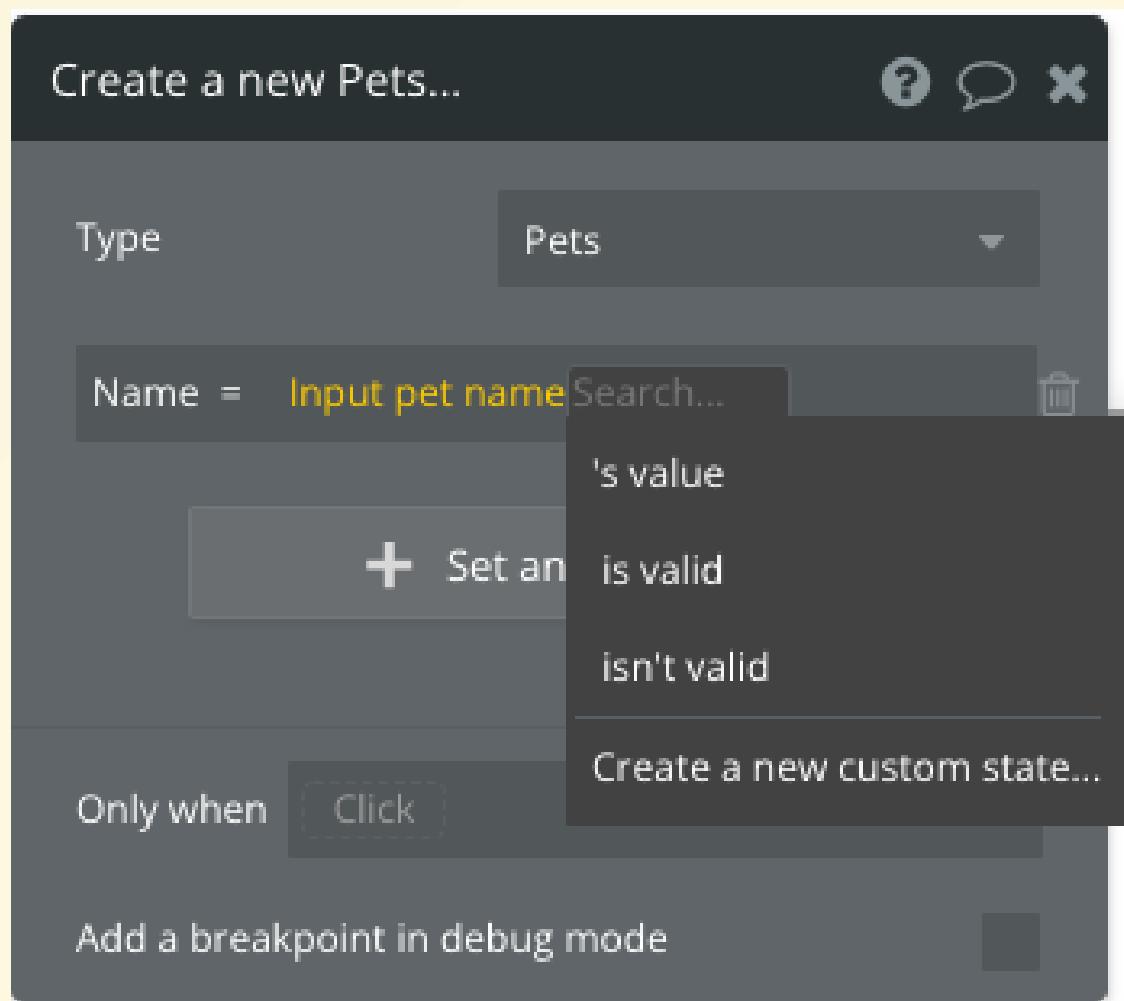
- 今回は画面で入力されたペットの名前になるので
Input pet name という項目を選択します



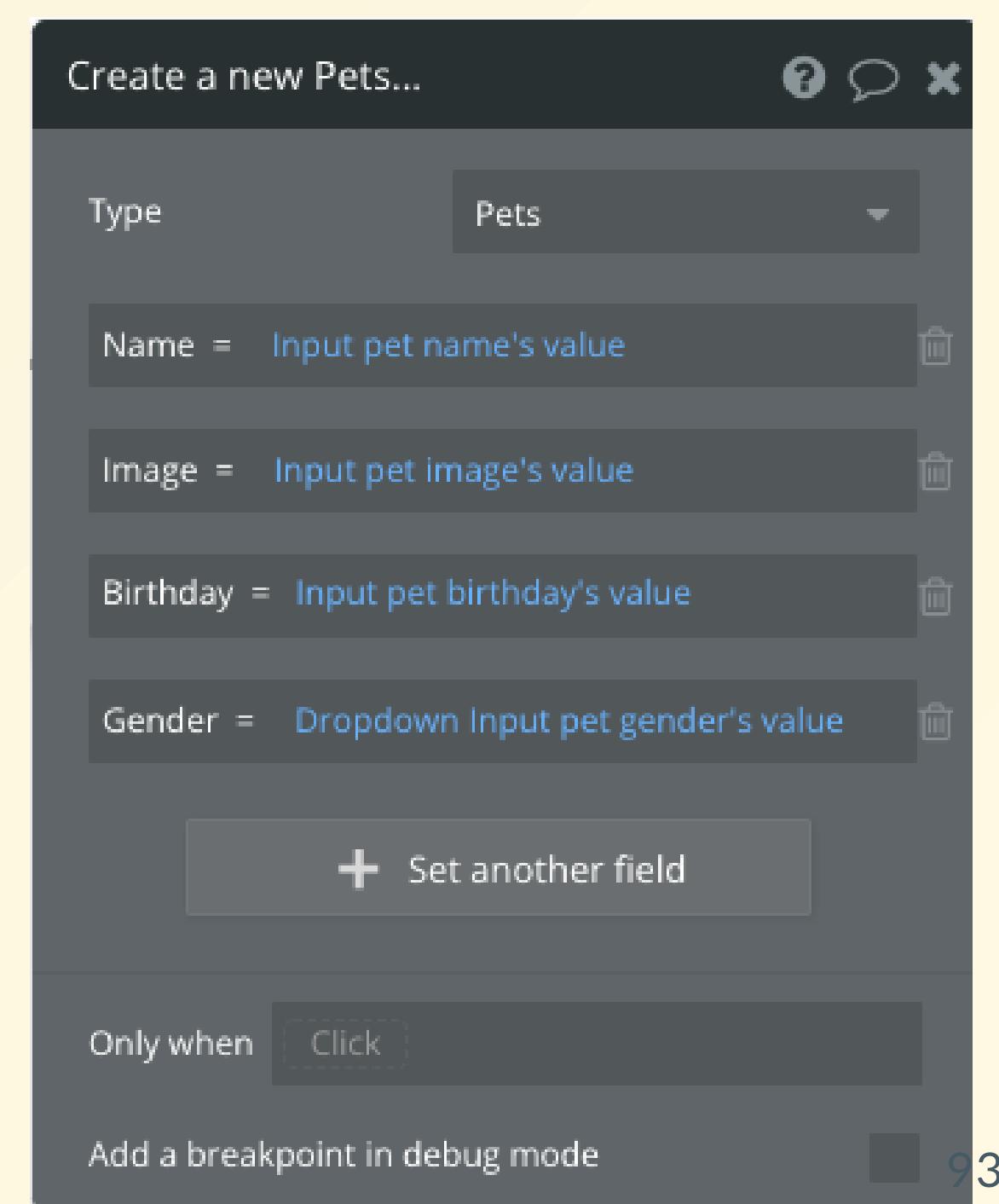
- この **Input pet name** という
のは画面を作るときに入力
要素に対して名前を付けた
ものがここに表示されてい
ます
- こうすることで、例えば1
つの画面に複数の入力要素
が存在したときにも、それ
らを一意に識別しやすくな
ります



- `Input pet name` を選択するとさらに選択肢が表示されます
- ここでは画面で入力されたペットの名前の「値そのものの」を使いたいので `'s value` を選択します
 - その他の選択肢である `is valid` と `isn't valid` は入力された値が正しいかどうかの判定結果を使いたい時に利用します



- すべて指定するとこんな感じ



動かしてみよう！

- ここまで設定したら実際にプレビューでペットが登録されるかどうかを確認してみましょう
 - ちなみに "Ctrl + P" をクリックするとプレビュー表示ができます（ショートカットキー）
- 入力要素をすべて埋めて "REGISTER" ボタンをクリックしてみましょう
- 画面としては特に何も起こらないですが、実際にデータが登録されたかどうか見てみましょう

- ・プレビューを閉じ Data タブを開いて App data を選択、そして左パネル All Pets を選択したときに、先ほど画面から入力された情報が保存されていることを確認してください

The screenshot shows the Airtable interface with the following details:

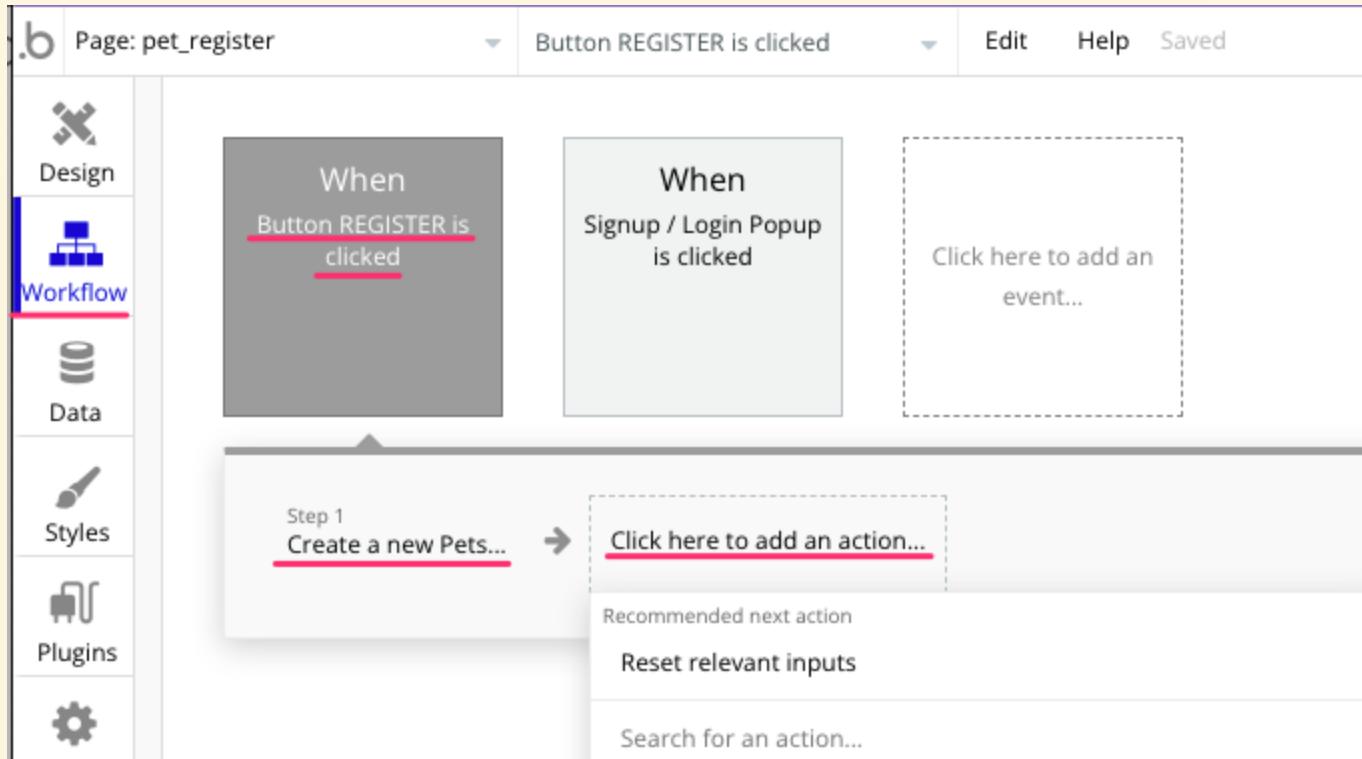
- Page:** pet_register
- Menu:** Edit, Help, Saved
- Left Sidebar (Data Tab Selected):** Design, Workflow, Data (highlighted), Styles, Plugins, Settings, Logs.
- Top Navigation:** Data types, Privacy, App data, Option sets, File manager.
- Section Headers:** Database views, Application data - All Pets - Development version, Copy a.
- Action Buttons:** New view, Primary fields, Search, 1 entries (displaying 1), New entry, Delete (0), Upload.
- Search Bar:** Search view names or data types...
- Data View:** A table titled "All Pets" with columns: Birthday, Gender, Image, Name, Created Date. One entry is displayed:

	Birthday	Gender	Image	Name	Created Date
<input type="checkbox"/>	Nov 1, 2021 12:00 am	Male		たま	Nov 8, 2021 10:41 pm
- Buttons:** All Pets (highlighted), All Users.

- これで画面で入力した内容をデータベースに保存するところまで出来ました！ 
- 続いて、画面遷移を作り込んでみます

画面遷移してみよう

- 左パネルから Workflow タブをクリック
- 先ほど設定した "REGISTER" ボタンをクリックした時の振る舞いが表示されていると思います

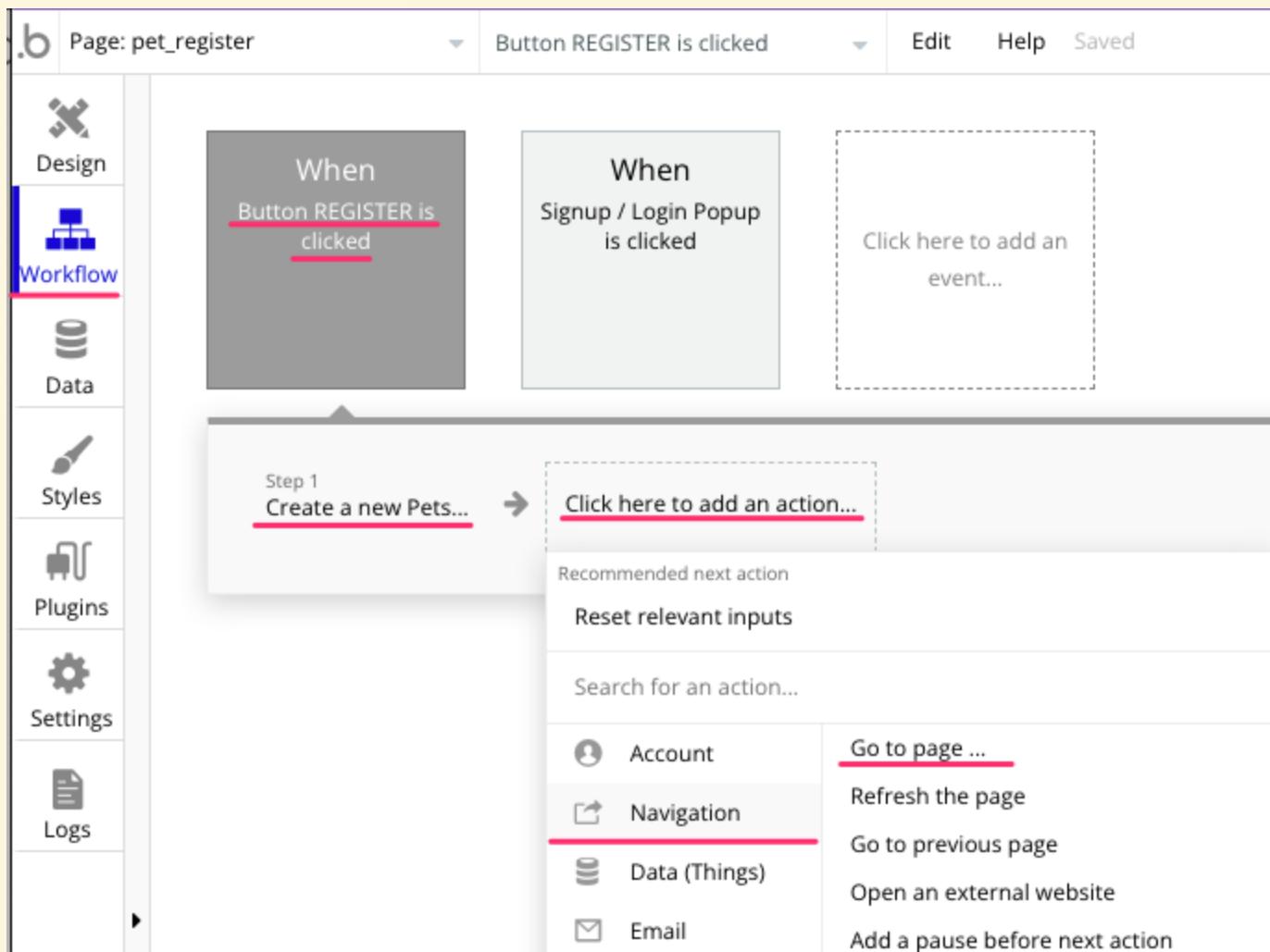


- ここで "Create a new Pets..." の右にある
Click here to add an action... をクリック

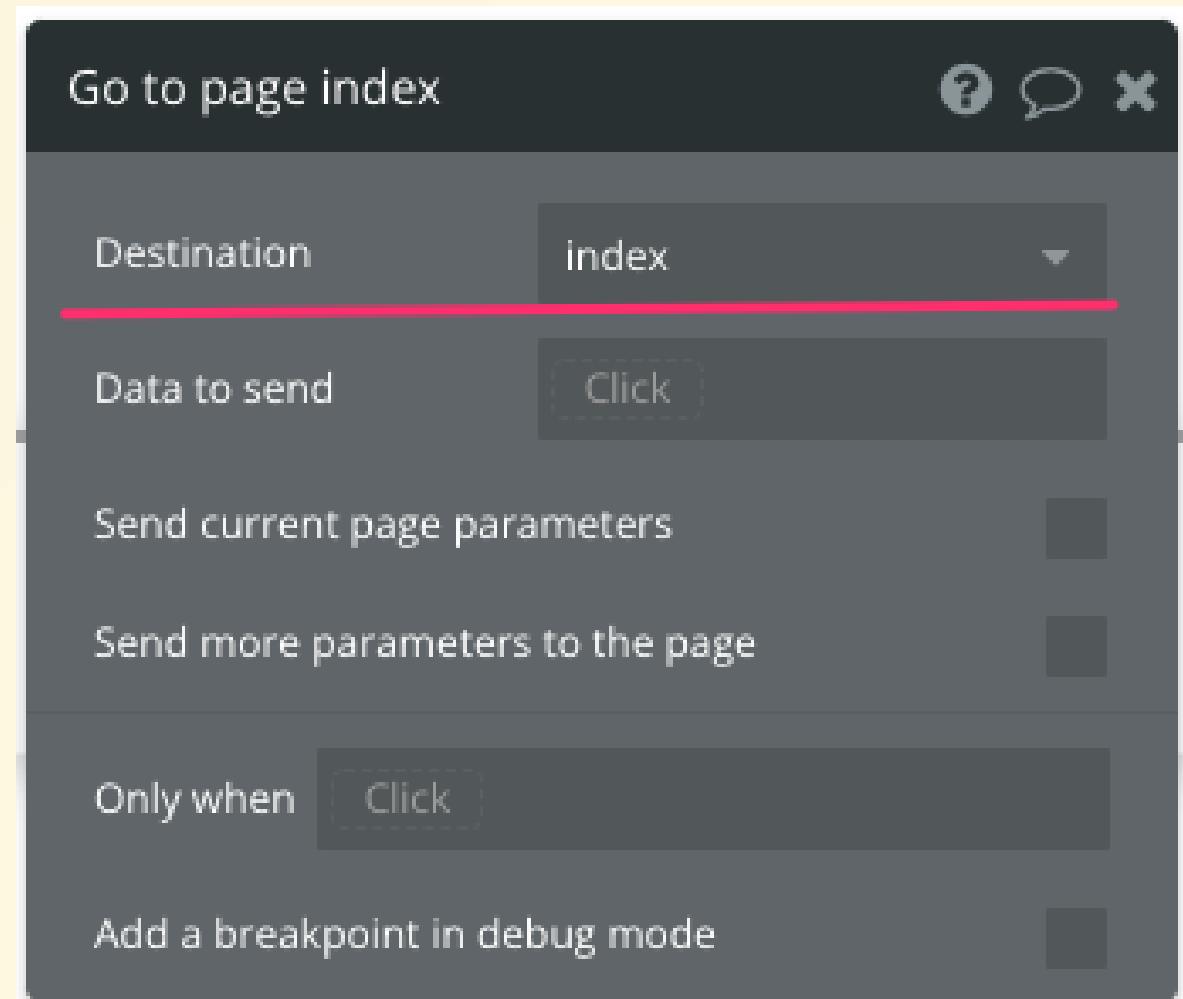
The screenshot shows a workflow editor interface for a page named 'pet_register'. The top navigation bar includes 'Edit', 'Help', and 'Saved' buttons. On the left, a vertical sidebar lists 'Design', 'Workflow' (which is selected), 'Data', 'Styles', 'Plugins', and 'Logs'. The main workspace displays three event blocks: 'When Button REGISTER is clicked', 'When Signup / Login Popup is clicked', and a dashed box labeled 'Click here to add an event...'. Below these, a step labeled 'Step 1 Create a new Pets...' has a red underline. To its right is a callout box with the text 'Click here to add an action...' underlined in red, which is also highlighted with a yellow background. A tooltip below it says 'Recommended next action' and 'Reset relevant inputs'. A search bar is labeled 'Search for an action...'. A list of actions is shown, with 'Navigation' underlined in red and 'Go to page ...' also underlined in red.

Action Type	Action Name
Account	Go to page ...
Navigation	Refresh the page
Data (Things)	Go to previous page
Email	Open an external website
	Add a pause before next action

- 今度は **Navigation** を選択し、下位要素として **Go to page...** をクリック



- するとダイアログが表示されるので "Destination" に遷移したい画面を選択します
- 今回はペット一覧画面へ遷移したいので **index** を選択します



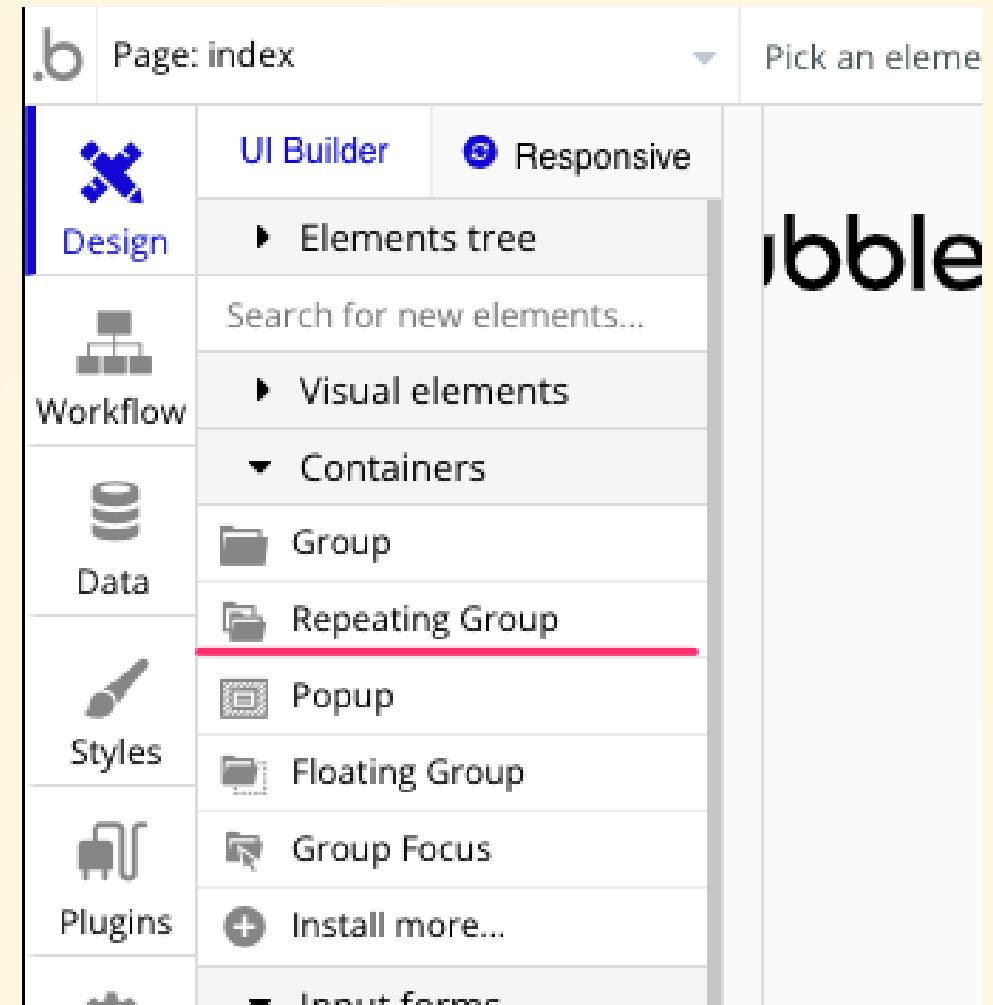
- それではプレビューしてみましょう
- ペット登録画面が表示されたと思うので、先ほどと同じようにペット情報を入力し "REGISTER" ボタンをクリックします
- すると、まっさらなページに遷移したと思います
 - まだペット一覧画面を作っていないので今は問題ありません
- これで画面遷移もマスターしました！ 

ペットの一覧表示

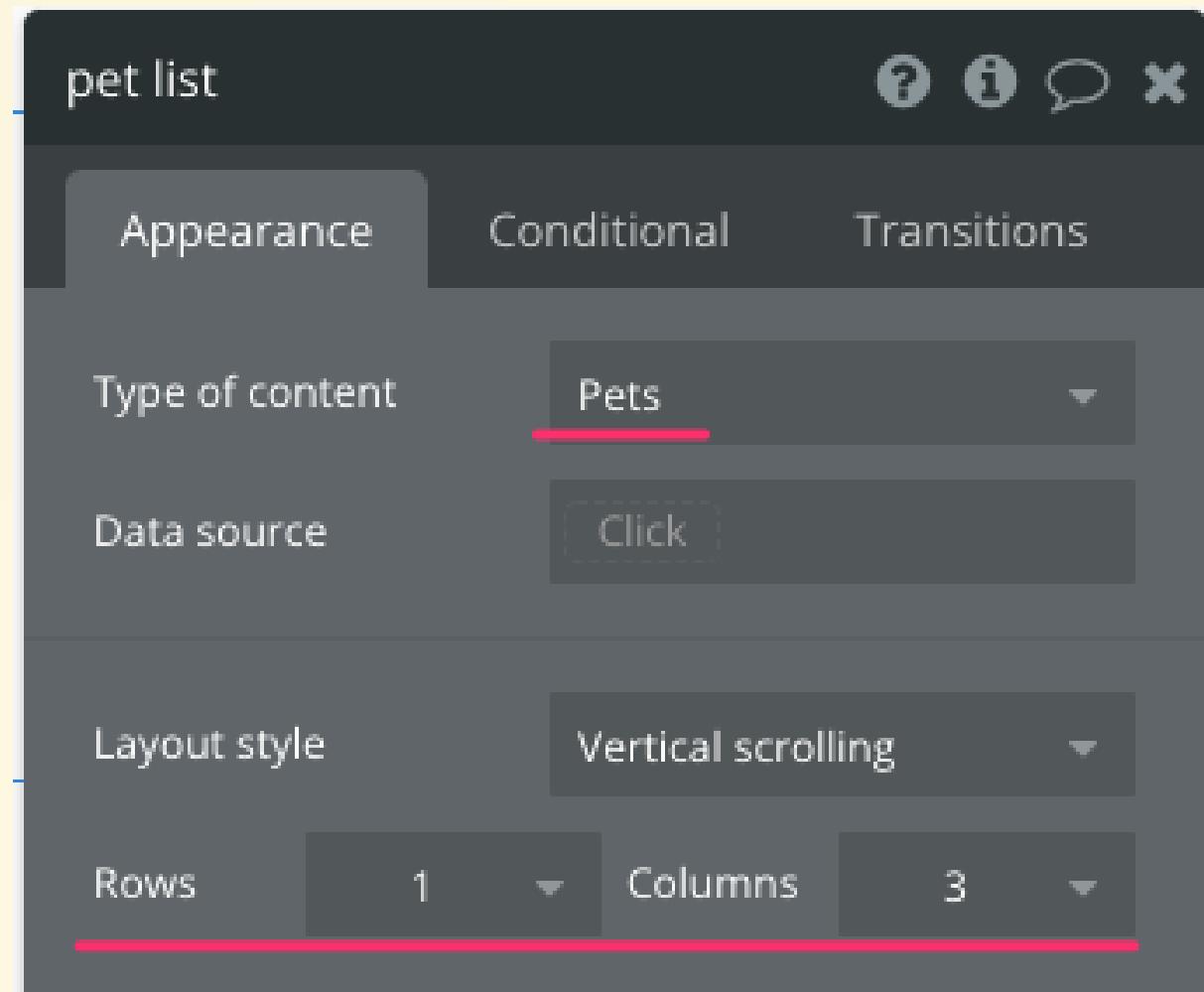
- 次にペットの一覧表示を作ってみましょう

The screenshot shows a user interface for managing pets. At the top left is the logo '.bubble'. On the right side, there are links for 'Bubble Forum' and 'FAQs', and a teal-colored 'LOG OUT' button. Below this header, there is a grid of three images of dogs. Each image has a name label underneath it. From left to right, the images are: a black and tan Shiba Inu named 'タマ' (Tamama), a white Chihuahua named 'ポチ' (Pochi), and a tan and white Shiba Inu named 'マロン' (Maron). The background of the page is light yellow.

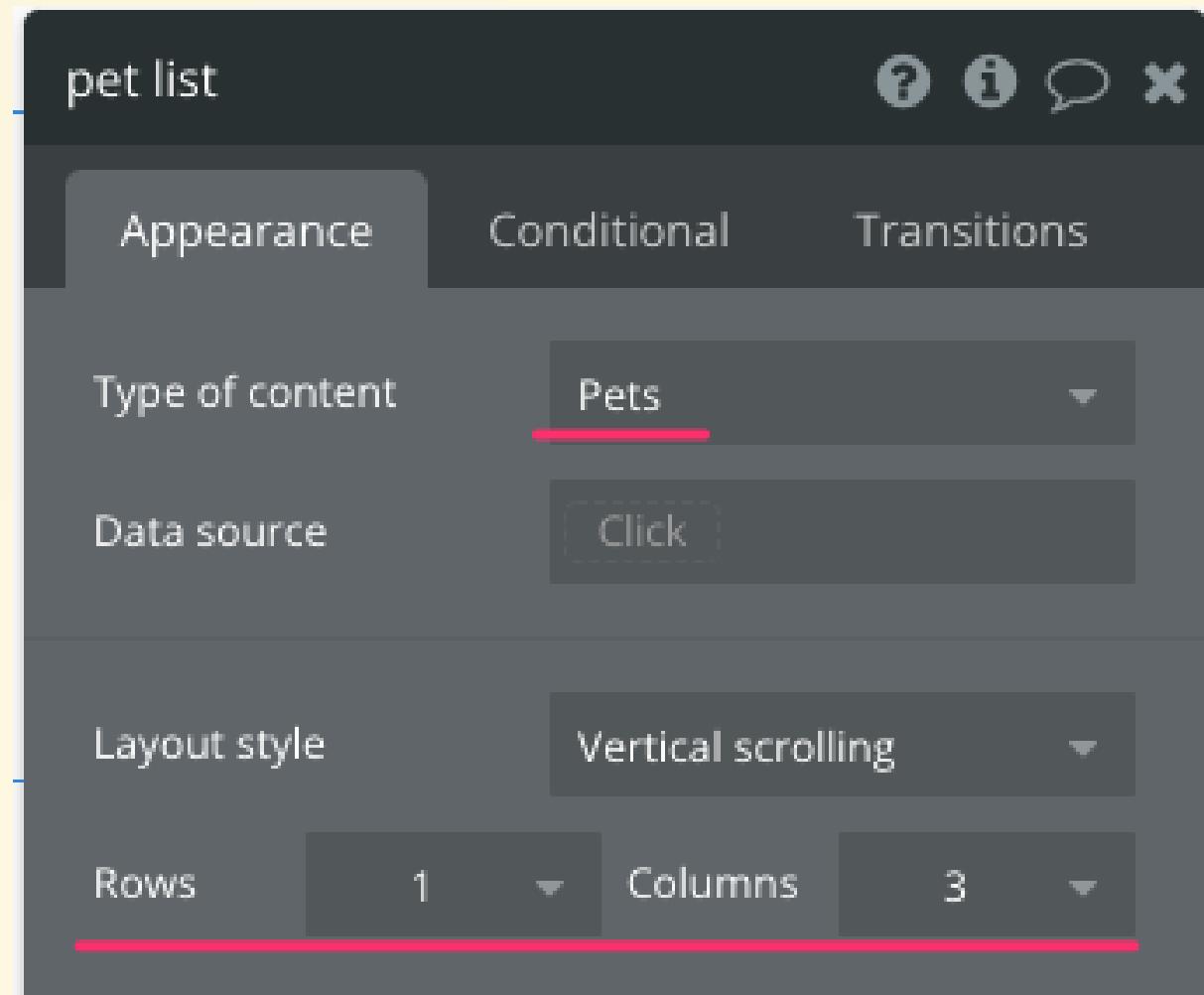
- 左上のページ一覧から **index** を選択
- 同じ要素を繰り返し表示する場合、**Containers** の中から **Repeating Group** を選択
- 右パネルのキャンバスにドラッグします



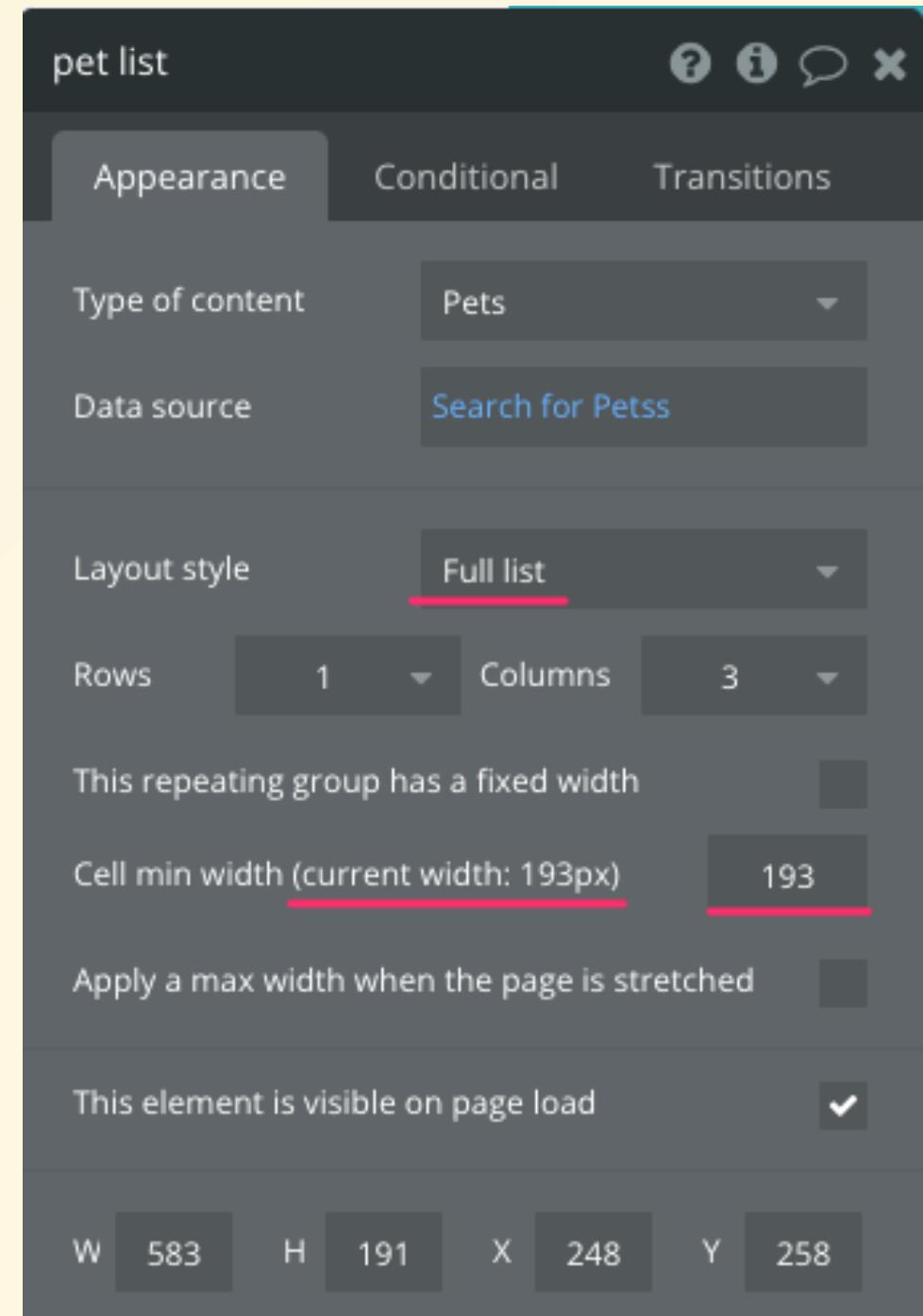
- 要素の詳細設定popupアップが表示されるので設定していきます
- 一番上の要素名に `pet list` を入力
- `Type of content` には、繰り返し表示するデータのタイプを指定します
 - 今回は `Pets` ですね



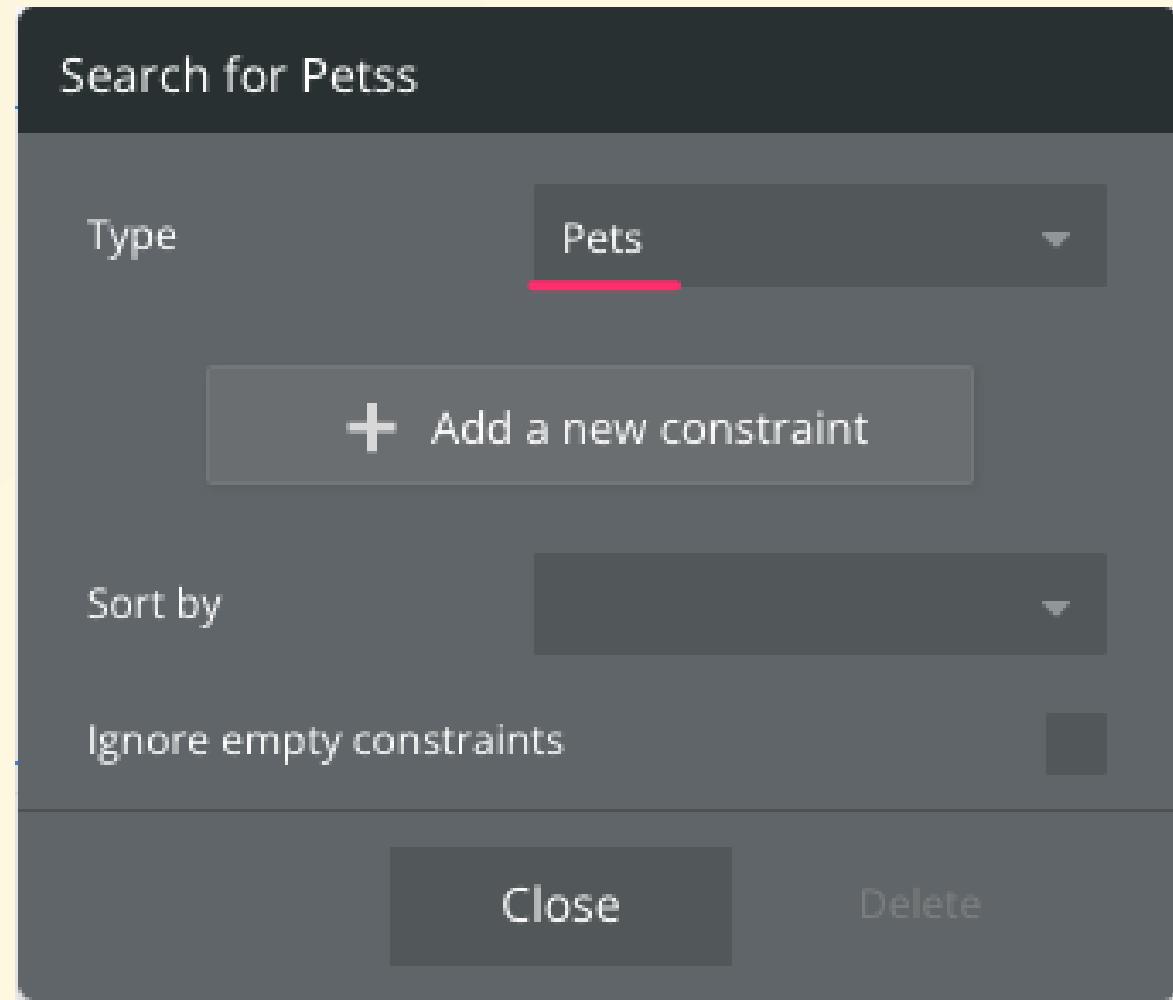
- **Rows** はデフォルトで表示する際の行数
 - データが複数あれば自動的に複数行になるためここでは **1** とします
- **Columns** は 1 行あたりの列数
 - 今回は **3** とします



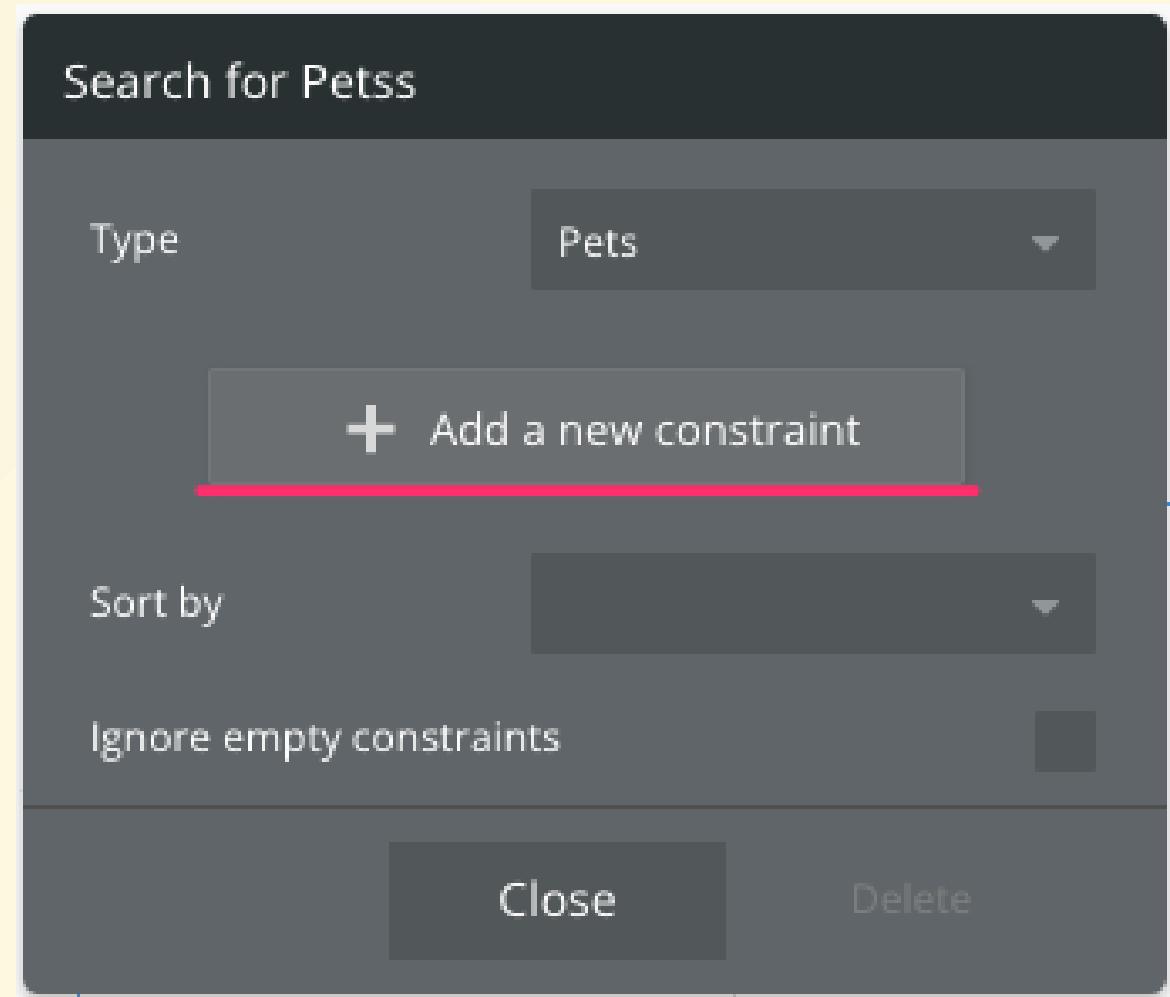
- もし 4 件以上のペットデータがあったときに、一覧を自動的に広げるために "Layout style" に **Full list** を選択しておきます
- また、それぞれのセルの中で画像の横幅をいっぱいに表示したいので Cell min width に括弧書きで記載されている current width と同じ値を設定しておきます（画像の例だと 193）



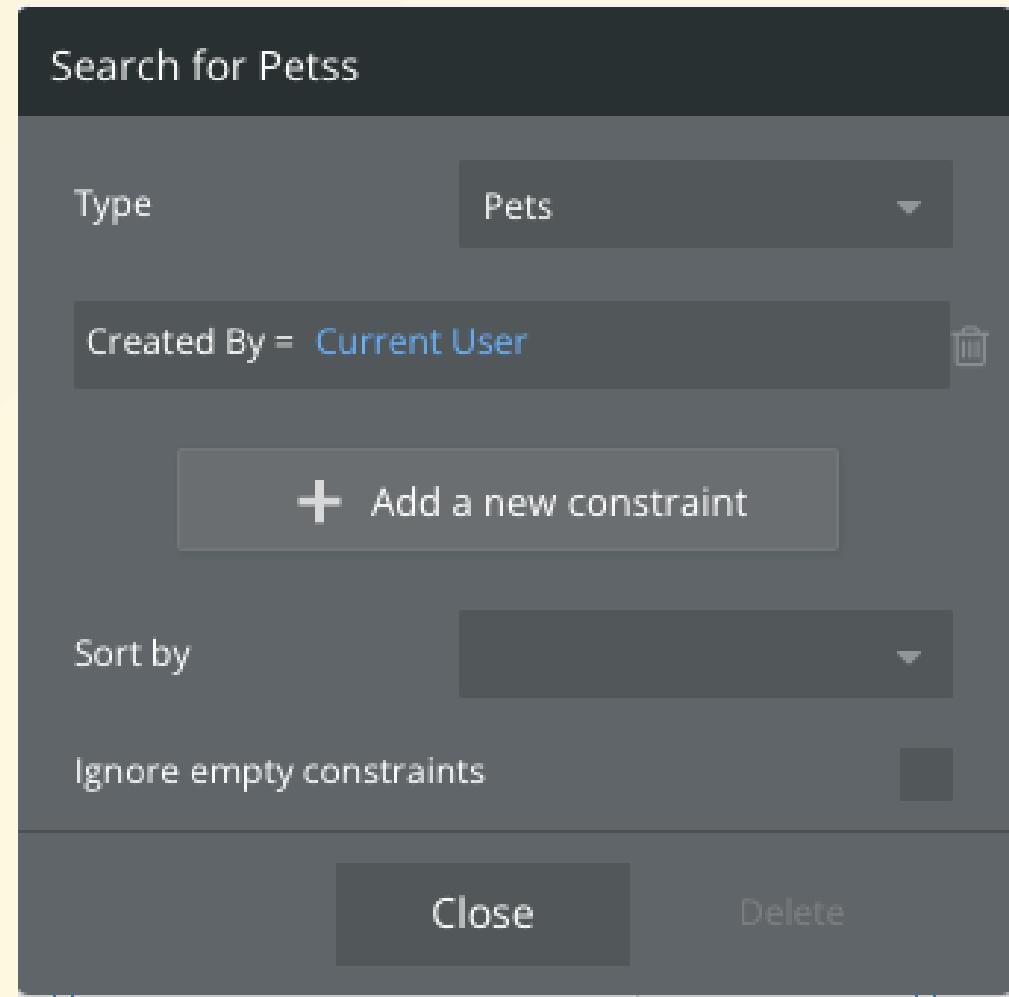
- そして **Data source** の Click から **Do a search for** をクリック
 - これは一覧表示対象のデータ元を指定するためのものです
- 別のポップアップが表示されますので Type に **Pets** を指定します



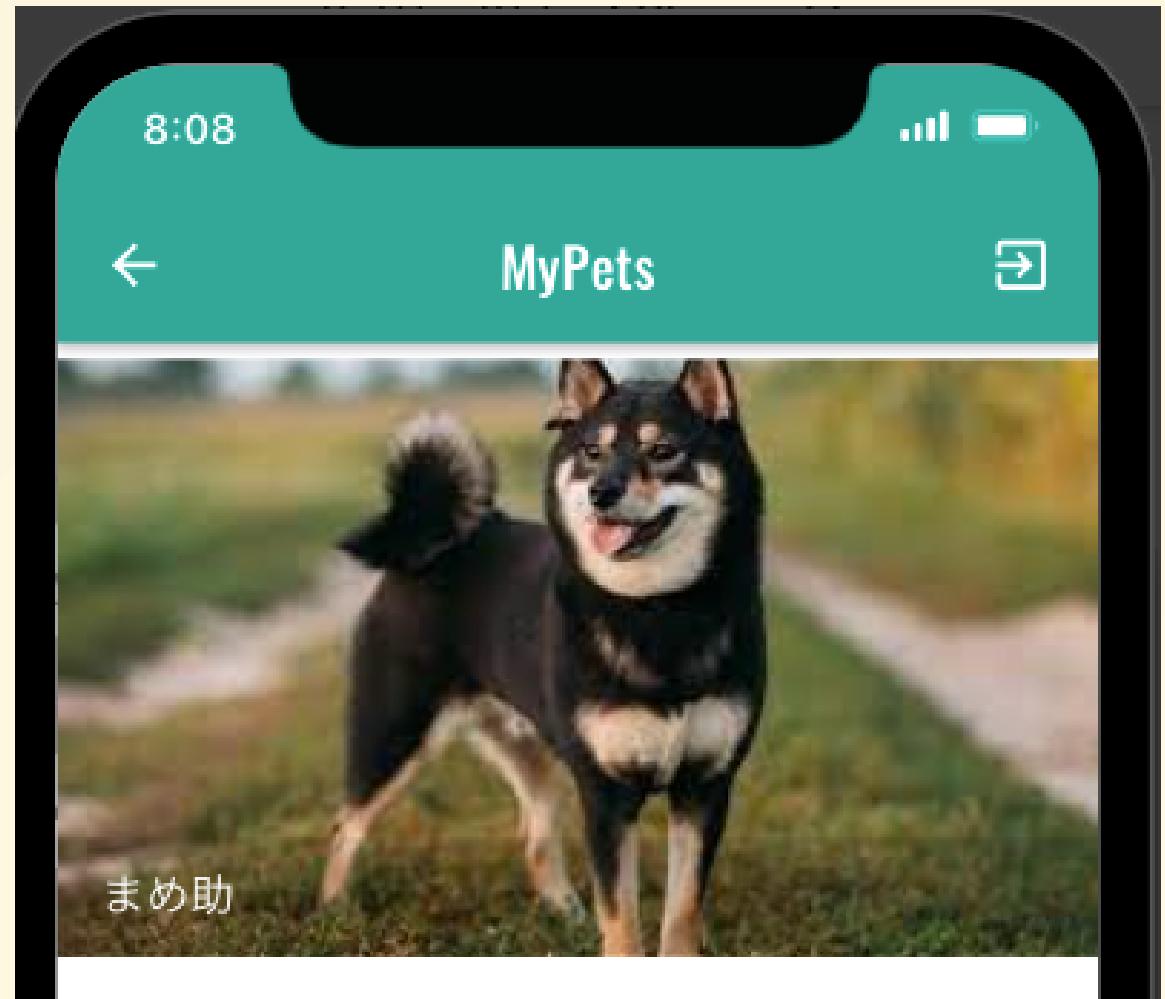
- これだけだと、未ログイン状態でもペットの一覧が表示できてしまうため、ログインユーザが登録したペットだけを表示するように条件を追加していきます
- "Add a new constraint" をクリックし、さらに条件を追加します
 - どんな設定内容になりますか？



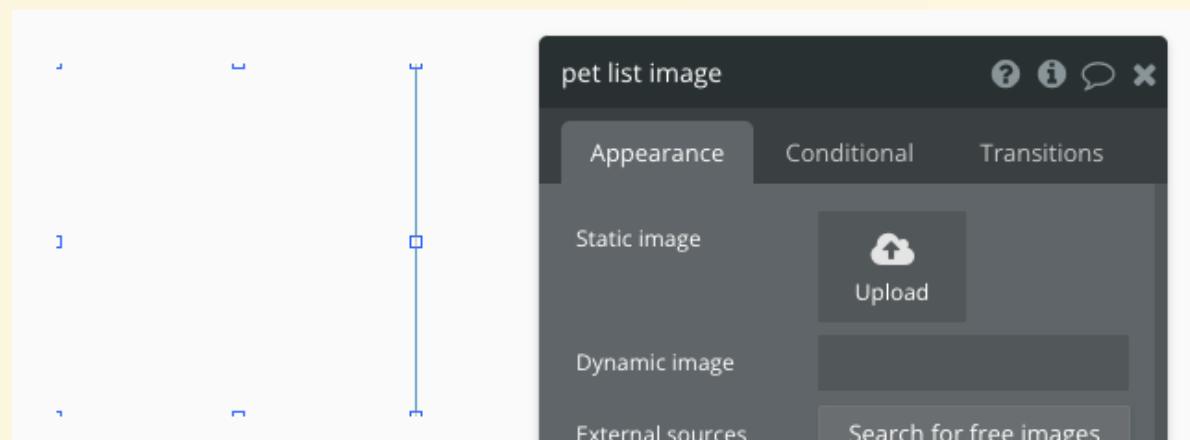
- 設定する条件としては「ペットの作成者 (Created By) が "Current User" と同一であること」



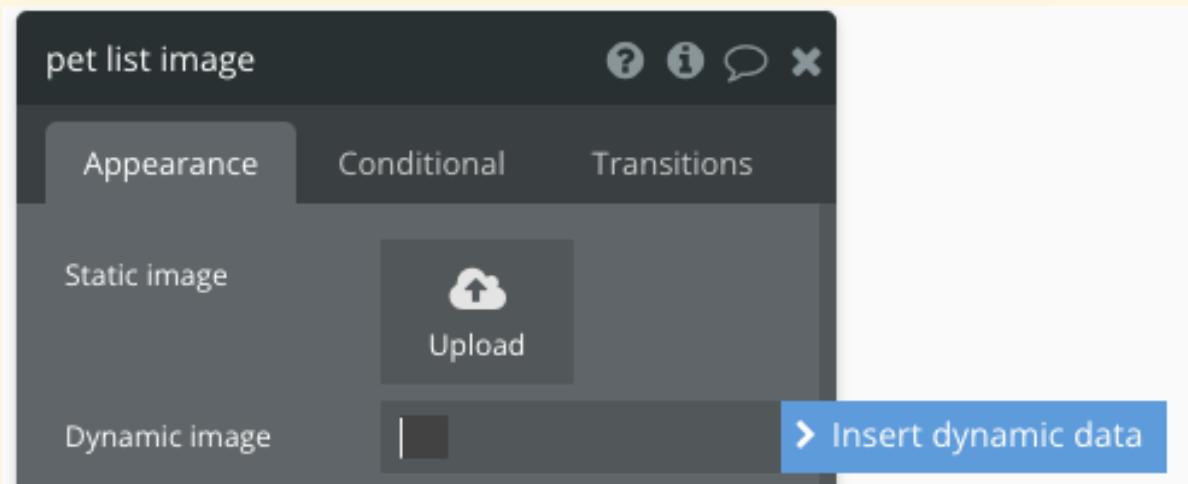
- 次に繰り返し表示する内容を設定していきます
- Adalo の時のように、画像と名前を表示してみましょう



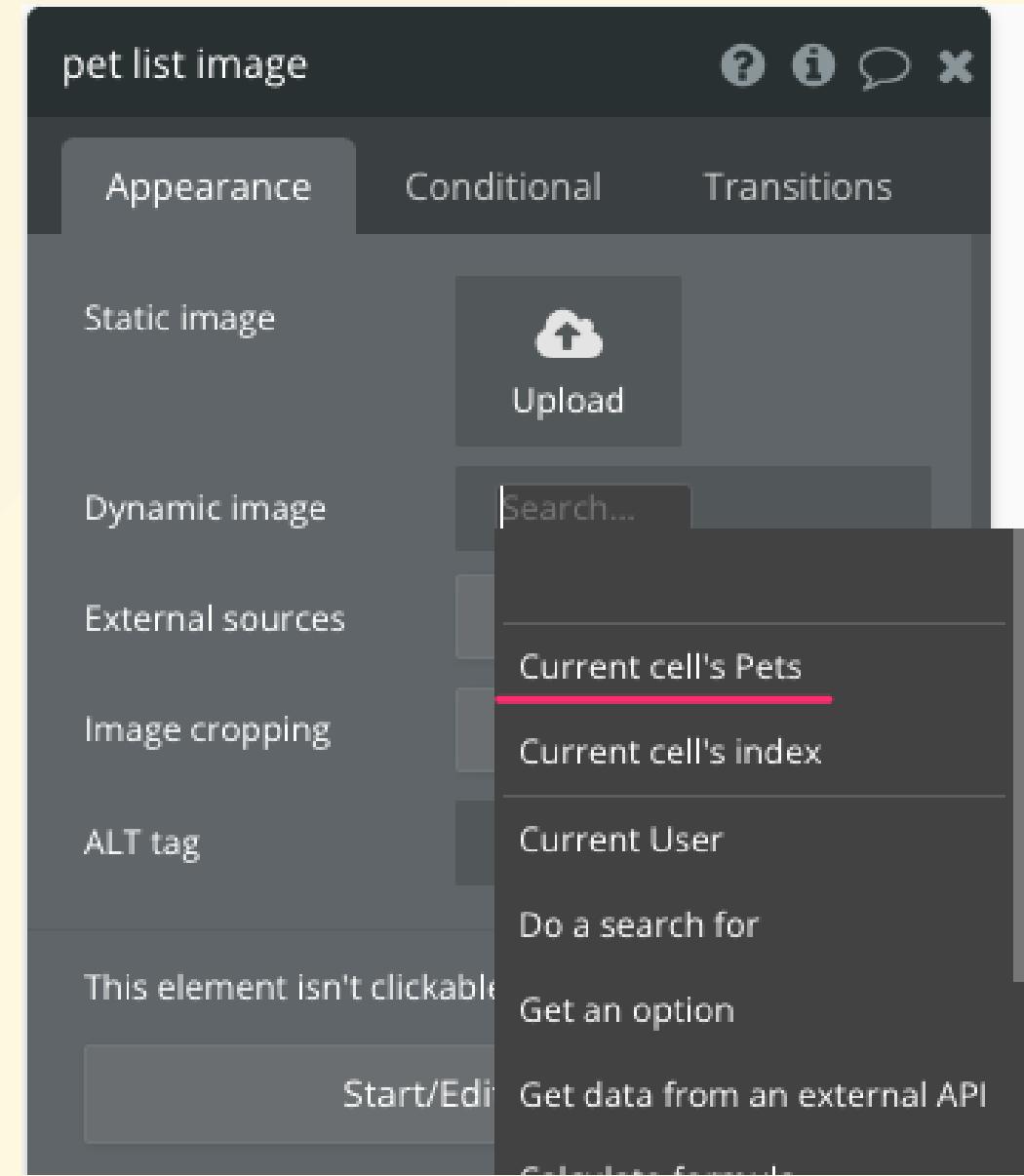
- まず **Visual elements** から **Image** を選択し、右パネルにドラッグ
 - このとき先ほど配置した "Repeating Group" の中に含める形でドラッグすると、後で移動する手間が無くなります
 - 後から Repeating Group に含めることも可能
 - 配置した後に Repeating Group の中いっぱいにドラッグして広げます
- 要素名に **pet list image** と名付けましょう



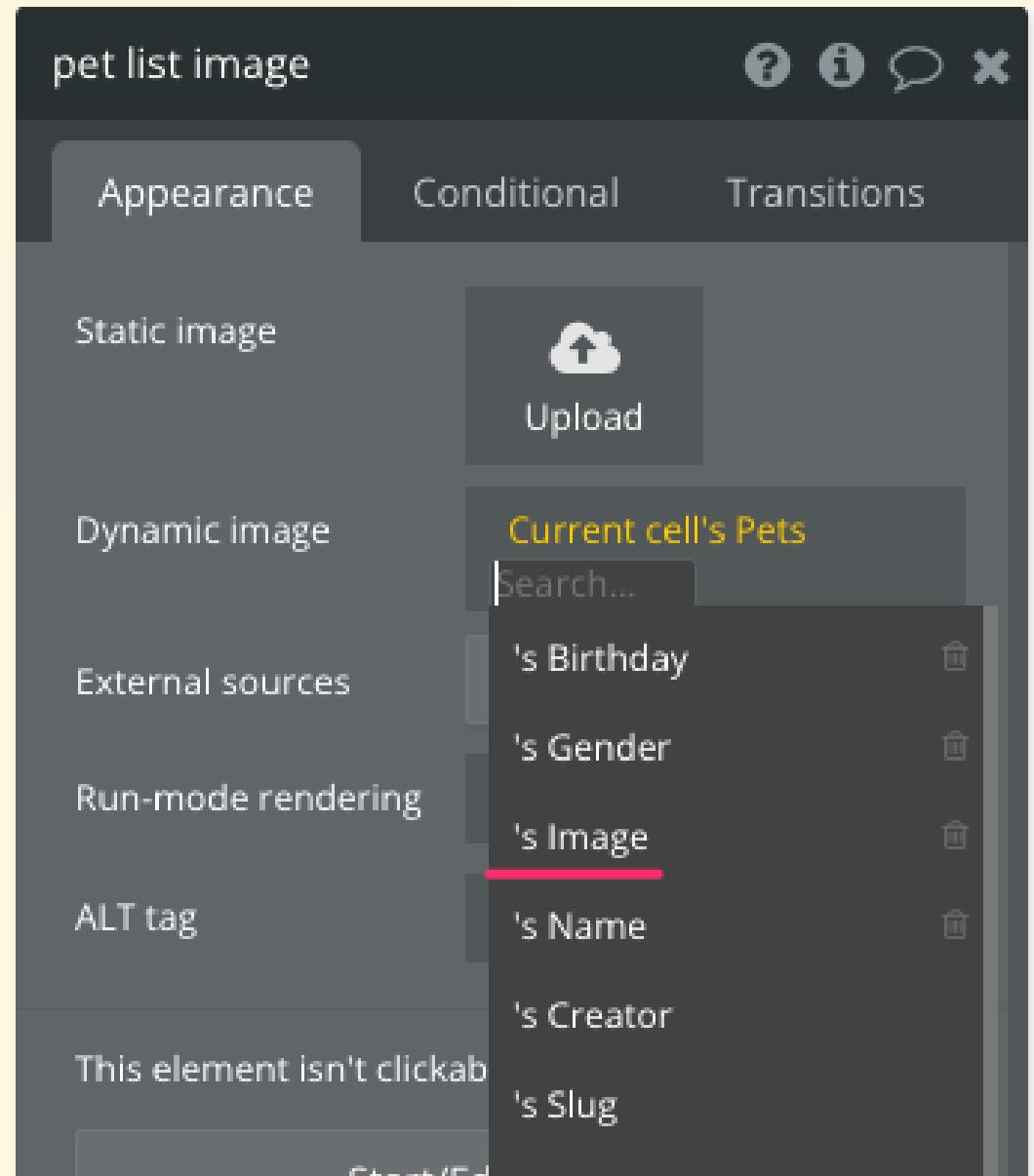
- 要素を配置したら、表示される詳細設定のポップアップから "Dynamic image" を選択
- すると **Insert dynamic data** というボタン（のようなもの）が表示されるのでクリック
 - これは固定の値や画像を表示するのではなく、文字通り動的に値や画像を出したい時に利用する機能となります



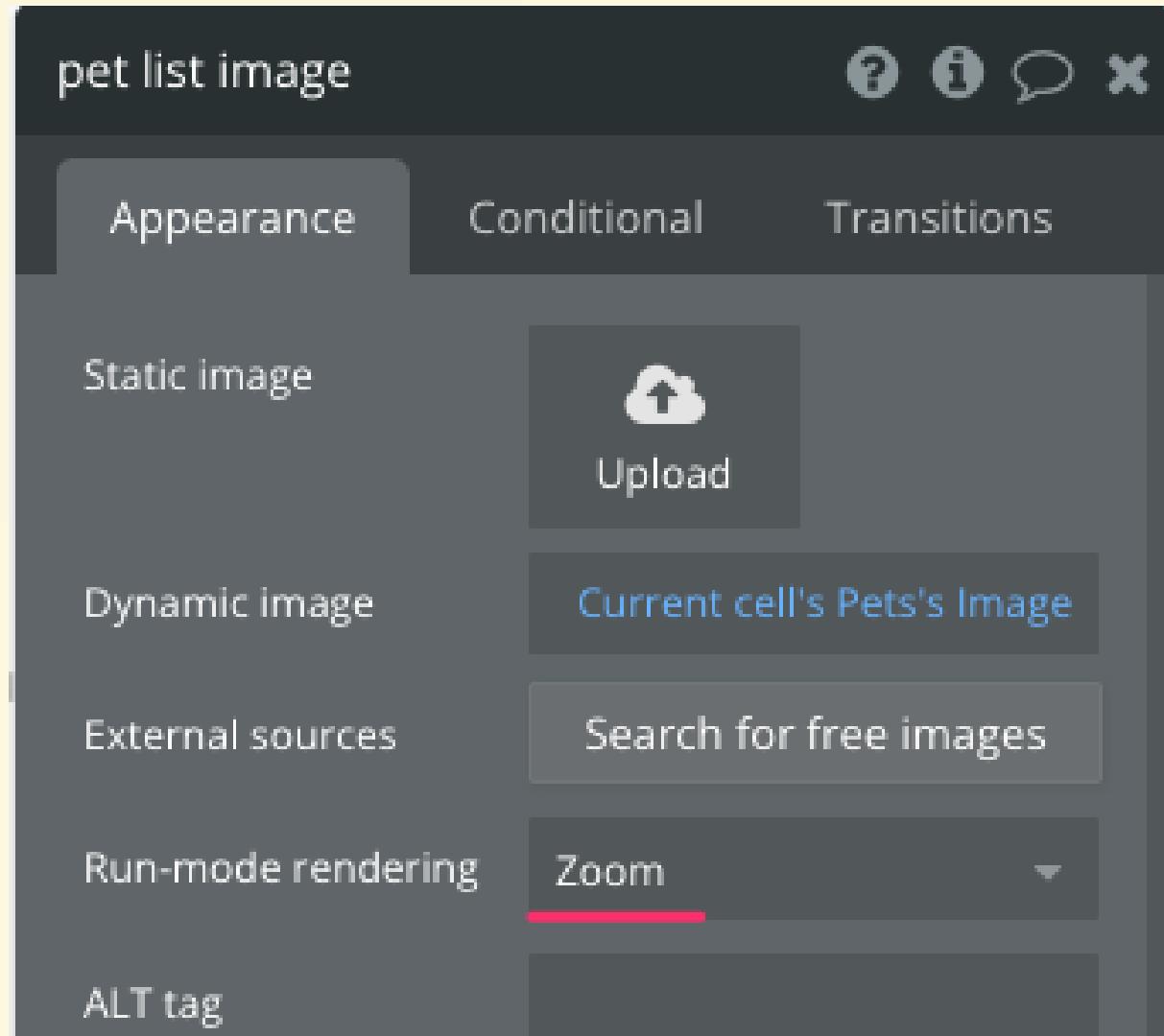
- ・クリックすると、プルダウンが表示されますので、そこから **Current cell's Pets** をクリック
 - これは文字通り現在のセルのペットのデータを使いたい場合に利用できます



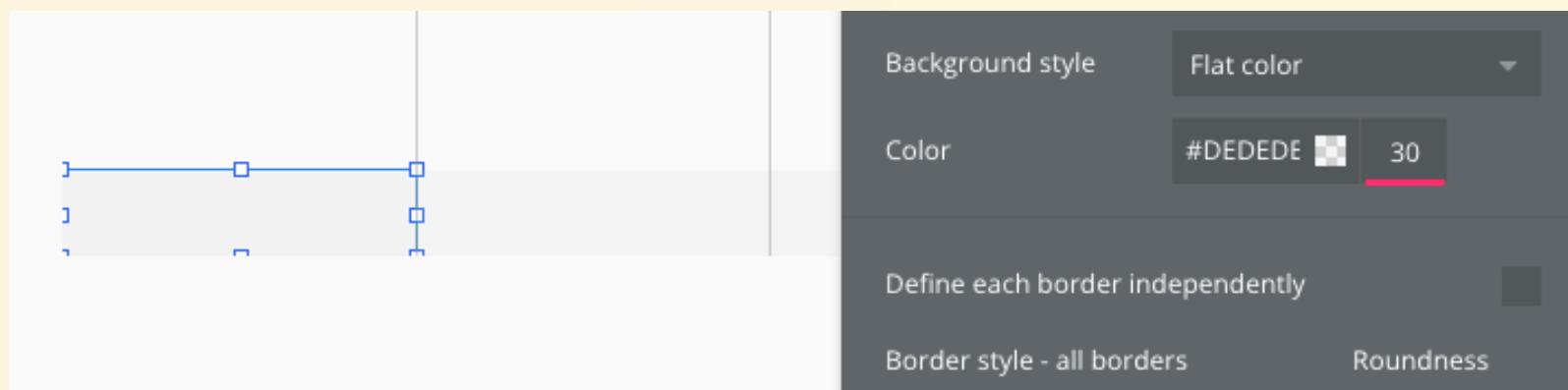
- **Current cell's Pets** を選ぶとさらにプルダウンが表示され、Pets タイプが持っている field が表示されますので **'s Image** をクリック
 - これで、現在のセルのペットの画像が表示する、という命令になるため、ペットデータが複数存在する場合に、それぞれのセルにペットの画像が並ぶようになります



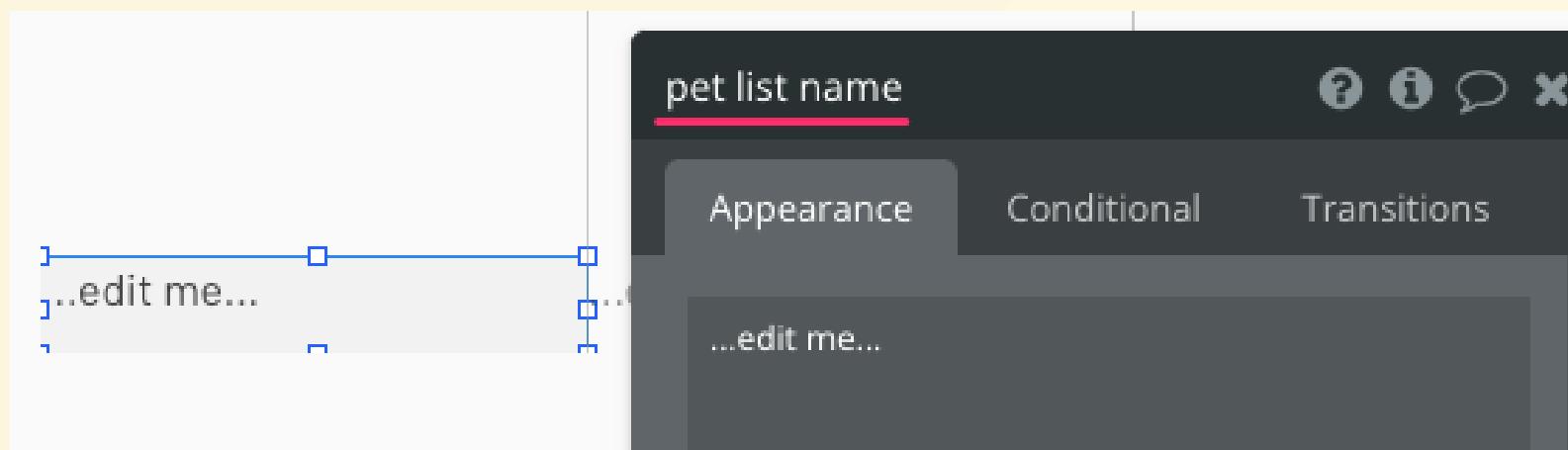
- 先ほどセルの横幅いっぱいに画像を表示する設定にしたのですが、画像要素の "Run-mode rendering" の値を `Streth` から `Zoom` に変えておきましょう
 - こうしないと、セルの中で画像の縦横比率を自動で変えられてしまうため、拡大表示に変更します



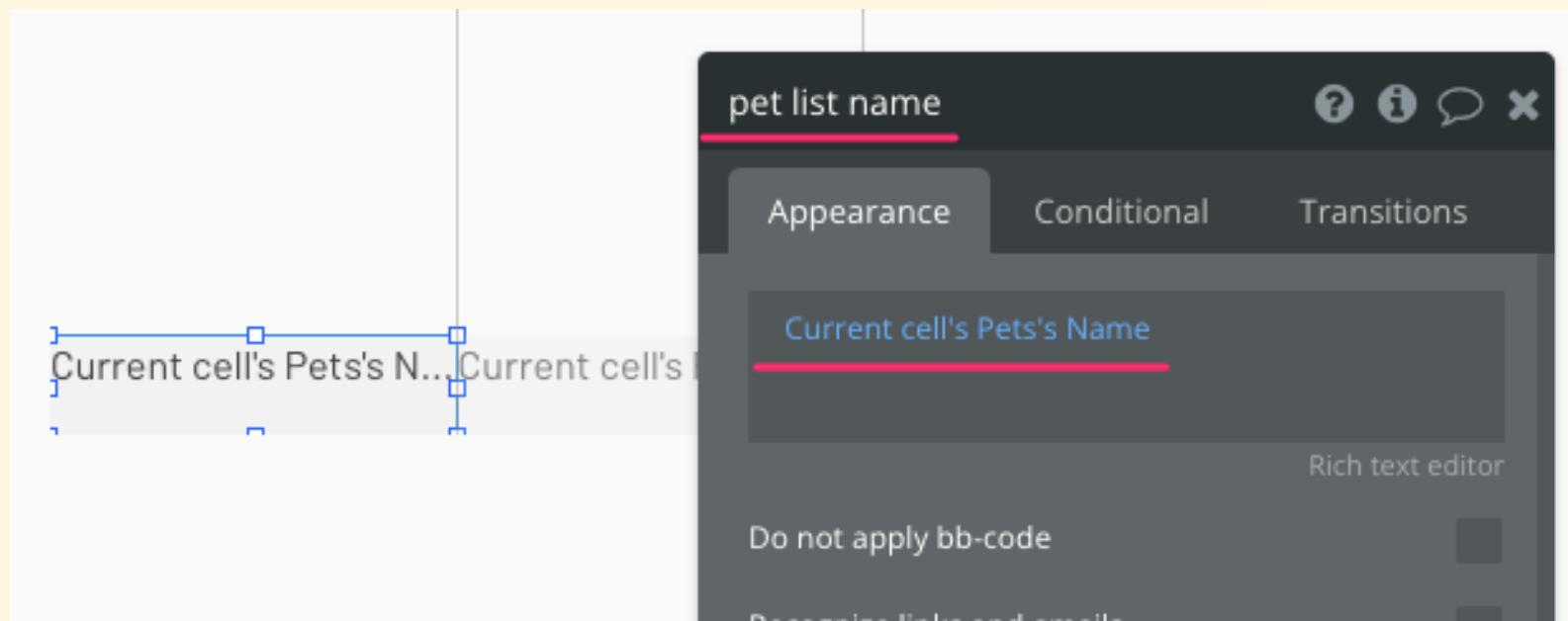
- 続いて画像下に表示する名前の背景を設定しましょう
- Visual elements から Shape を選択し、右パネルにドラッグ
 - こちらも "Repeating Group" の中に含める形でドラッグ
 - 配置した後に Repeating Group の下部に移動して横幅いっぱいにドラッグして広げます
- 配置した Shape の Color の横の数値を 30 にします
 - これは画像の透過率となり 100 が非透明、0 が透明となります



- 最後にペットの名前を表示します
- Visual elements から Text を選択し、先ほど配置した Shape の上にドラッグ
 - こちらも Repeating Group の中に含める形でドラッグ
 - 配置した後に Shape と同じサイズにドラッグして広げます
- 要素名に pet list name と名付けましょう



- ・ペットの画像と同様、ペットの名前も Dynamic data を使います
- ・**...edit me...** となっている部分をクリックすると、画像の時と同様 **Insert dynamic data** が表示されるので
そこから **Current cell's Pets** --> **'s Name** を選択



ここまで出来たらプレビューしてみましょう

- ペットの一覧として、ペットの画像とペットの名前が表示されていますか？

.bubble

Bubble Forum FAQs

LOG OUT

タマ

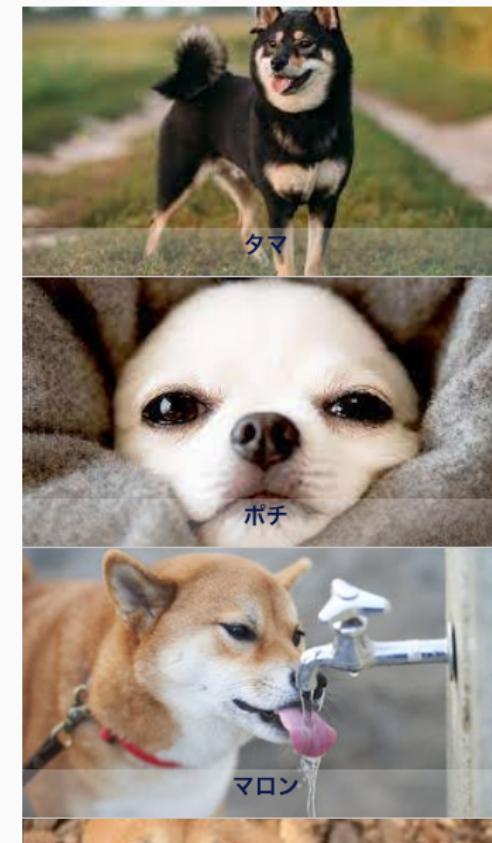
ボチ

マロン

- ちなみに Repeating Group はレスポンシブに対応しているため、画面の横幅を変えると自動的に画像のサイズが変わると思います！

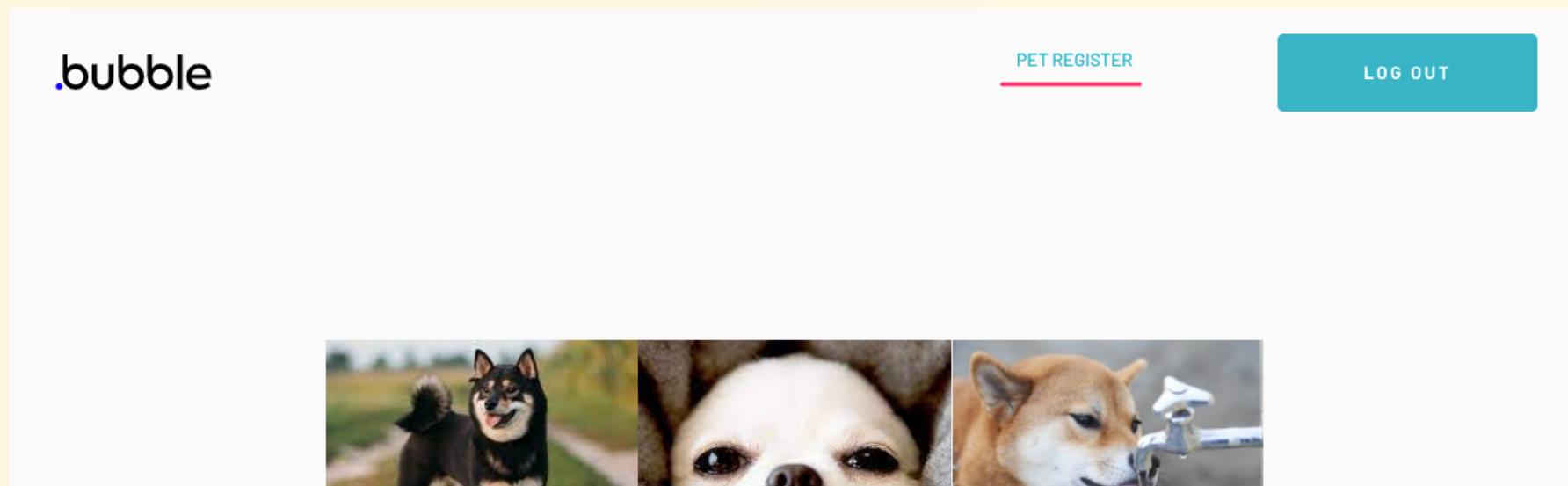
.bubble

LOG OUT

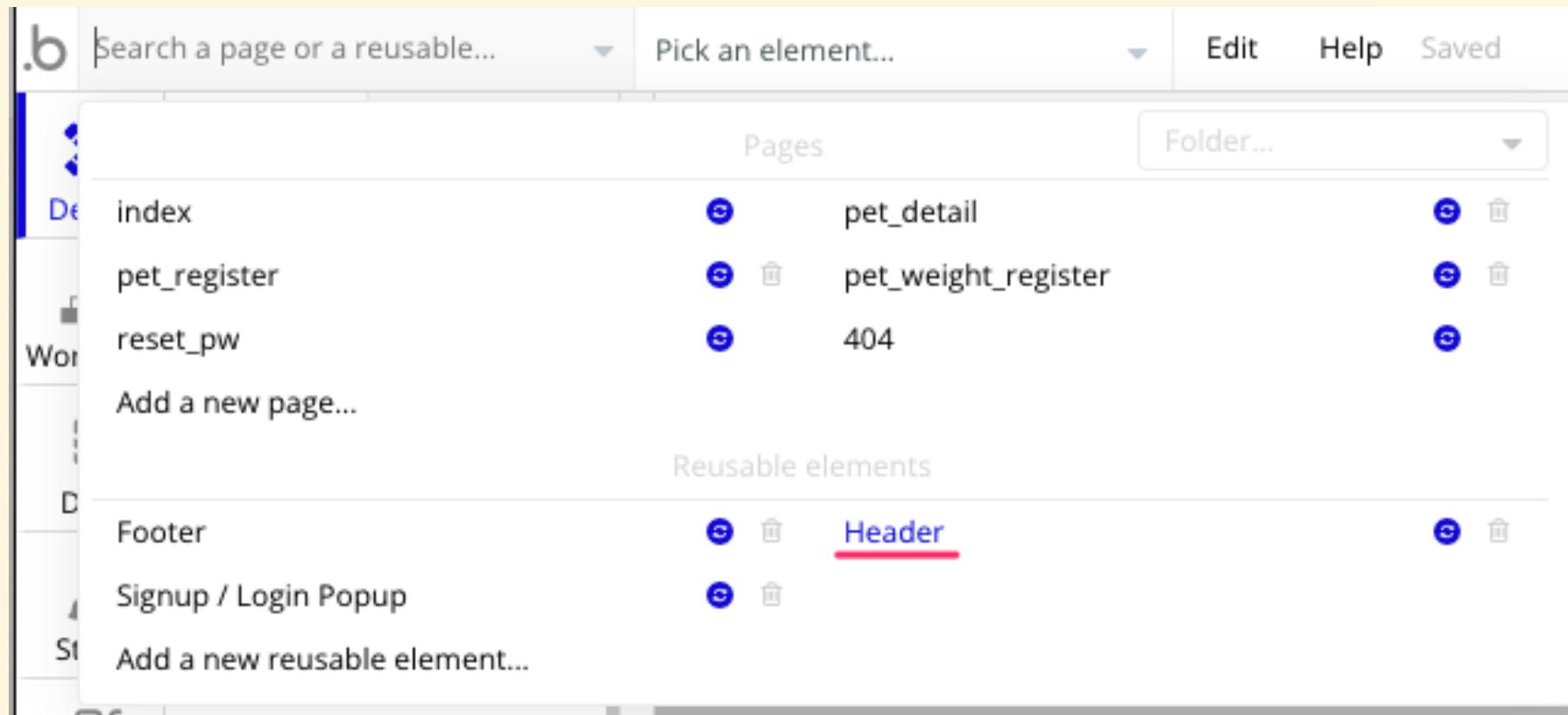


ペット一覧から登録画面の導線を用意しましょう

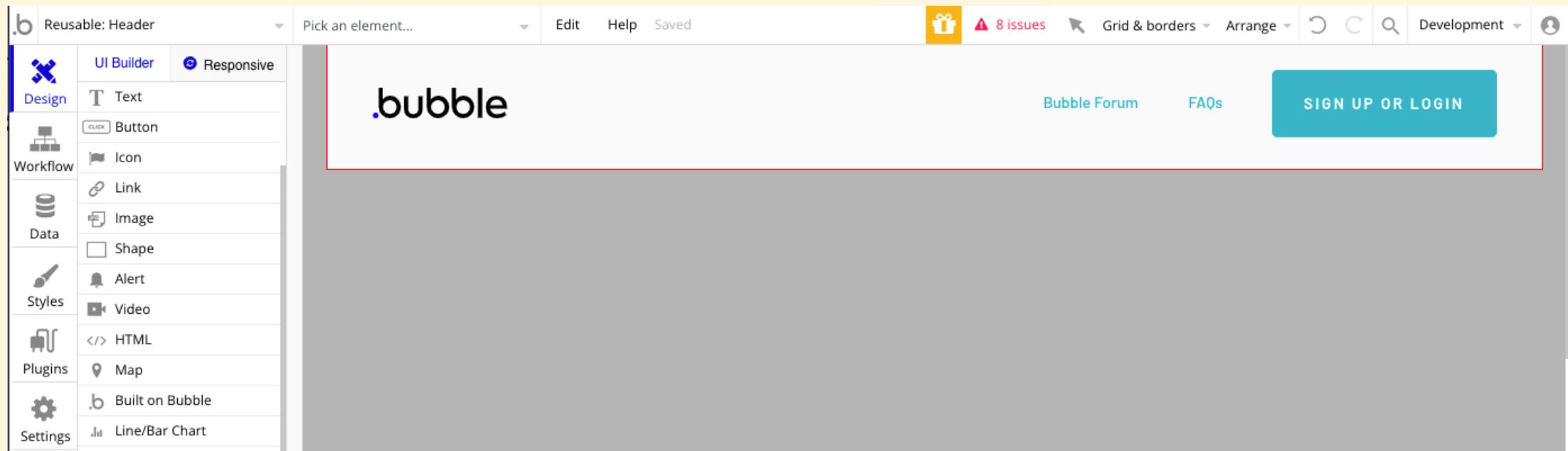
- ペットの一覧画面が出来たので、一覧画面から登録画面への導線を用意しましょう
- この導線は Header からいつでも遷移出来るように導線を用意してみましょう



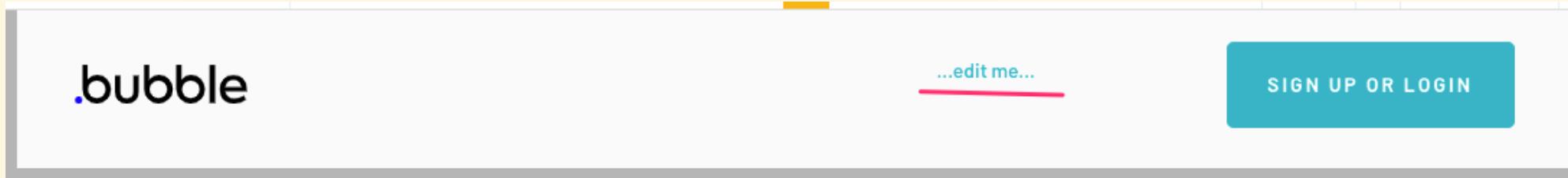
- 左上のメニューをクリックし、その中から Header を選択します



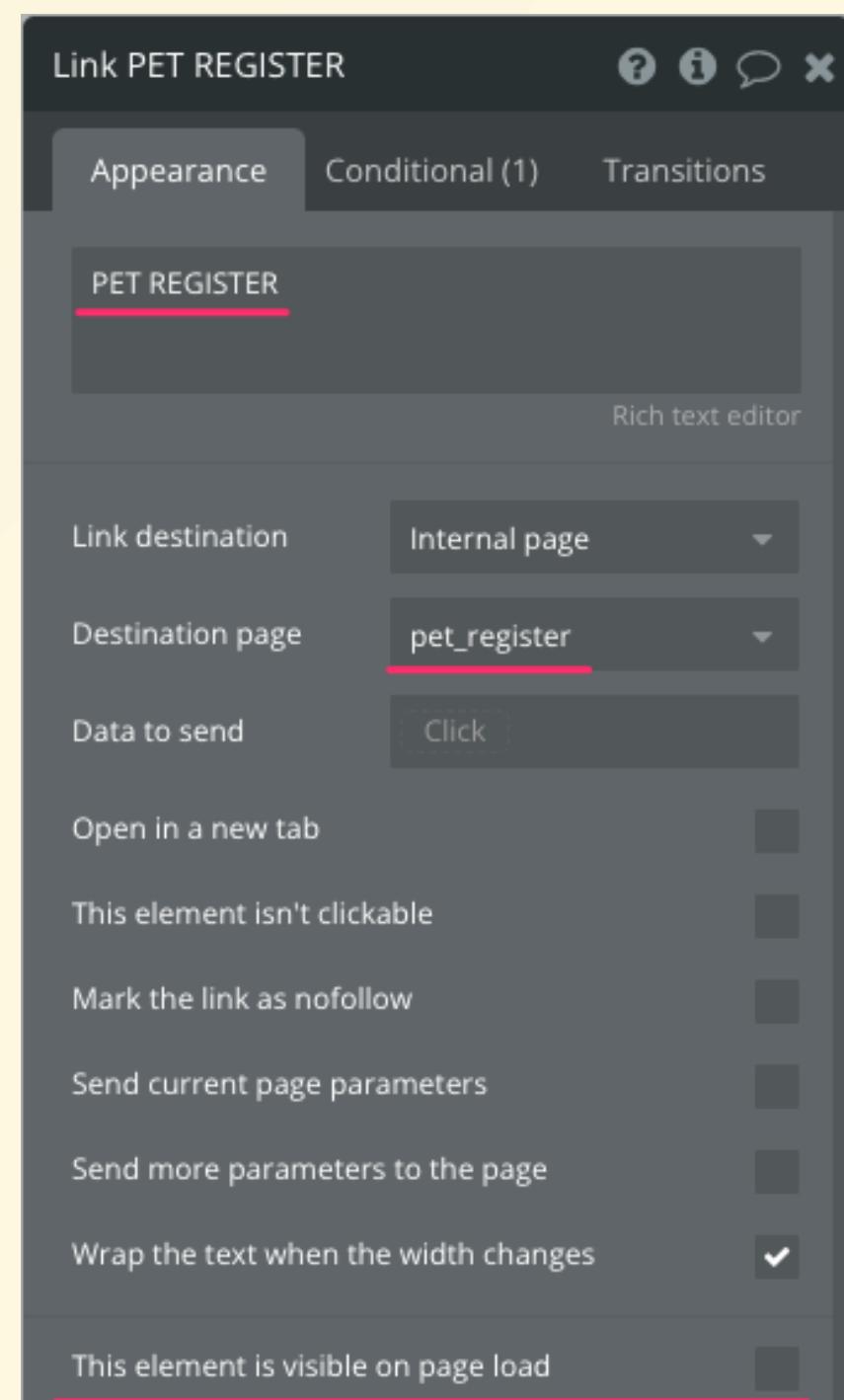
- するとヘッダー部分の要素が右パネルに表示されるので、ここを編集していきます



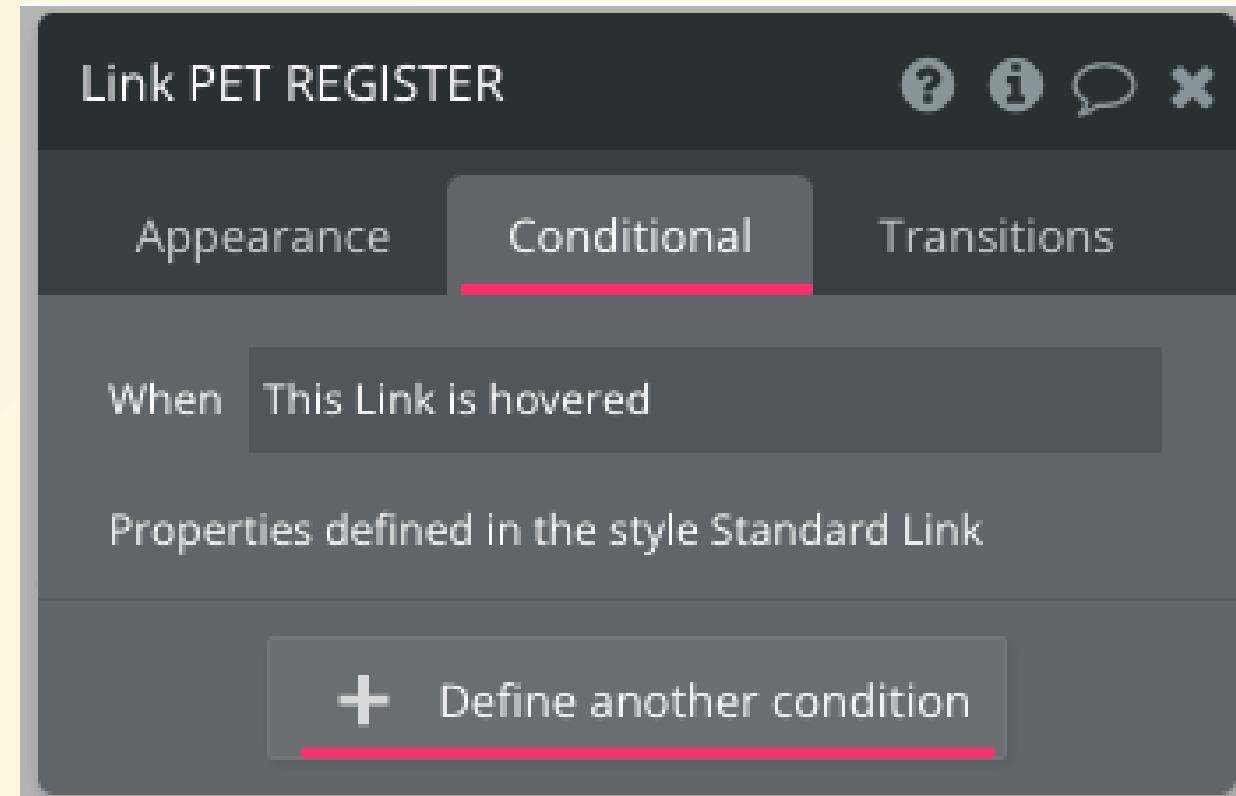
- まず元々ある **Bubble Forum** と **FAQs** のリンクは不要なので削除します
- そして **Visual elements** の中から **Link** を選択してヘッダー領域にドラッグします



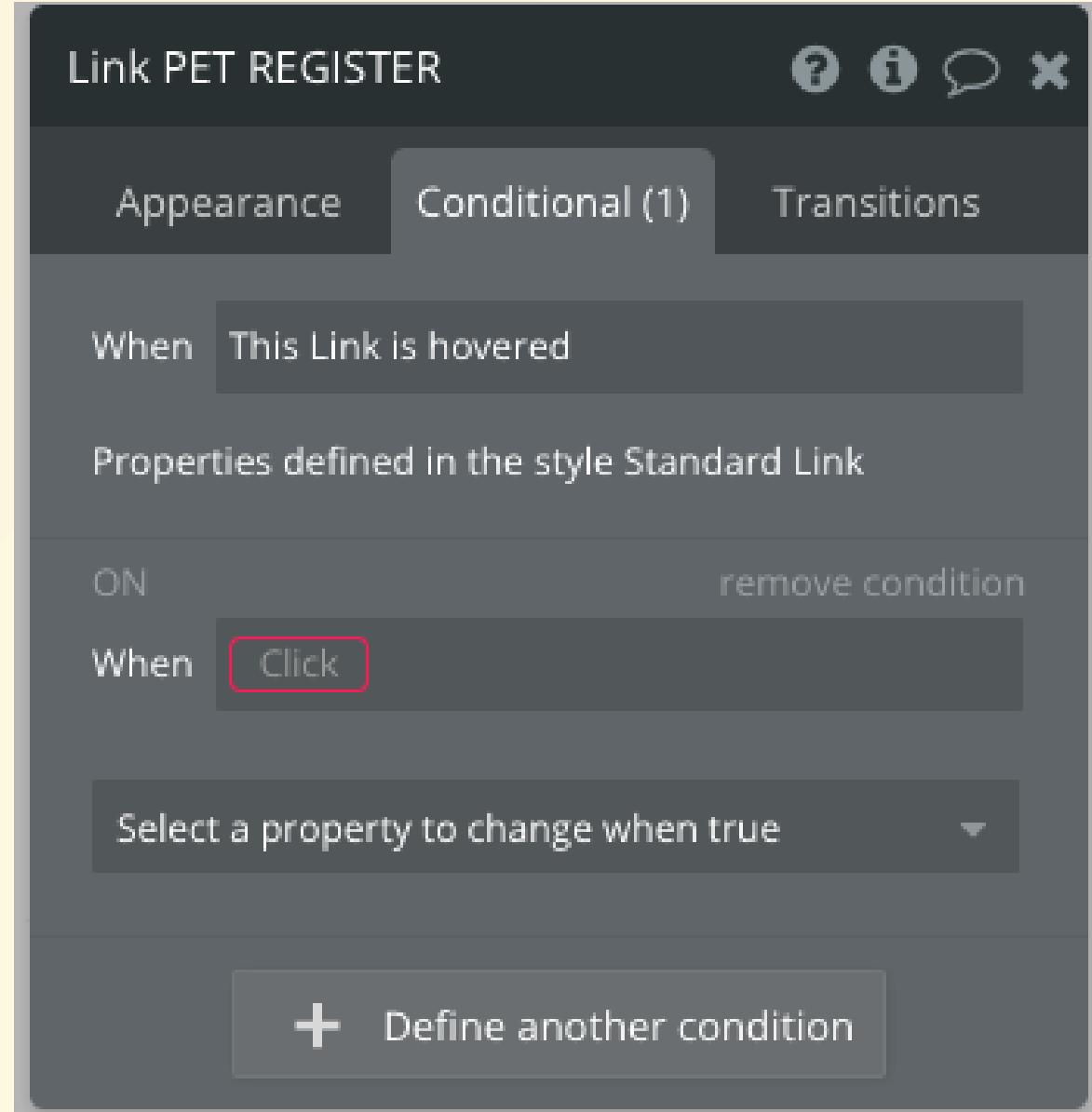
- リンクの名前は "PET REGISTER" としましょう
- "Destination page" として
遷移先は `pet_register` を選択
- "This element is visible on page load" のチェックを外します
 - このチェックを外す、要素は常に非表示となります
 - この理由はこのあと説明しますね



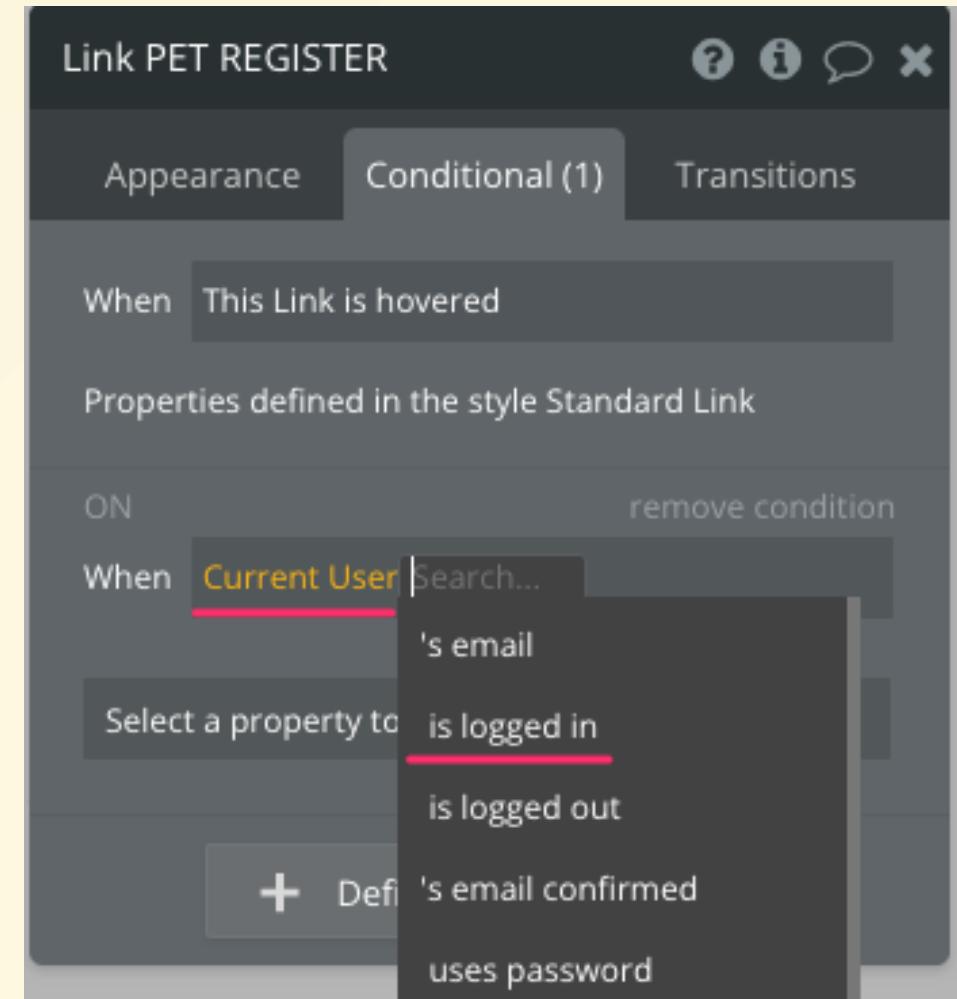
- このペット登録のリンクは、ユーザがログインしているときだけ表示させたいので、その制御を設定していきます
- 同じダイアログから **Conditional** タブを選択し **Define another condition** をクリック



- ここでは、このリンク自体に対して簡単な振る舞いを定義することができます
- 今回の場合、ユーザがログインしているときだけ表示させたいので、まず When のところで、表示させたいときの条件を設定します

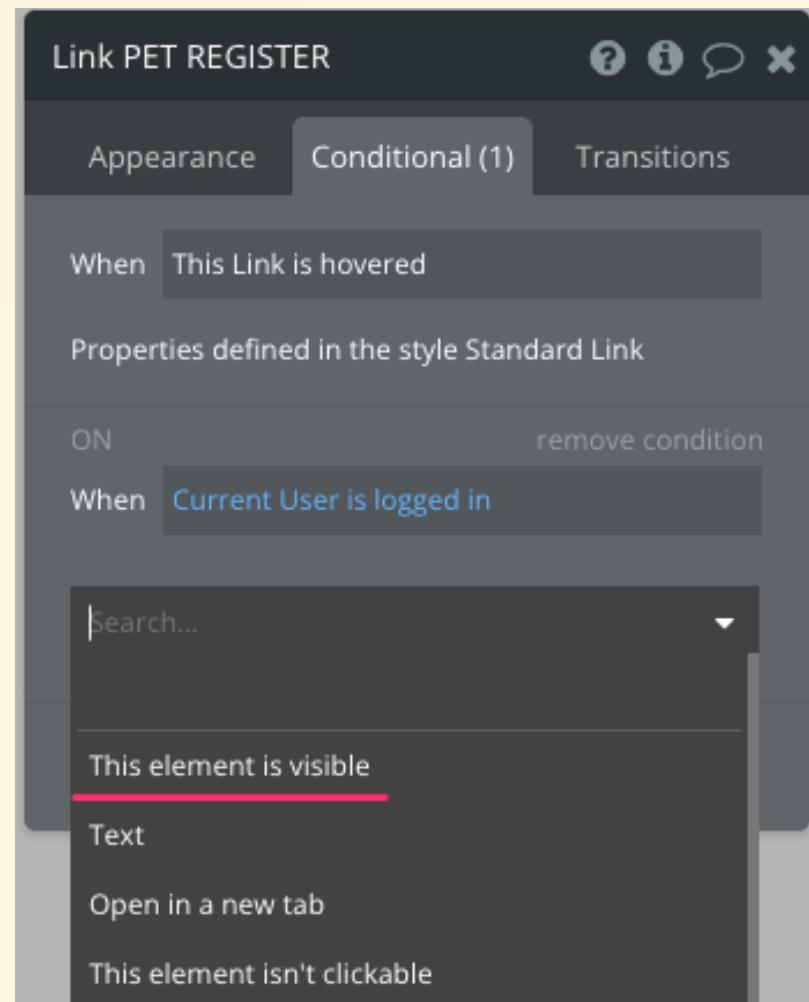


- Click から Current User を選択し、次に is logged in を選択することで、「現在のユーザがログインしている場合」という条件を指定できます



- 続いて、When を満たしたときの振る舞いを設定します

- Select a property to change when true のプルダウンの中から This element is visible を選択します
- これは文字通り、この要素を表示する、という振る舞いになります
 - 選択したら、その振る舞いにチェックを付けます



ではプレビューしてみましょう！

- 一度 index へ戻り、プレビューを実行
- 現在はログイン済みの状態のはずなので、"PET REGISTER" のリンクが表示されていると思います
- そこから右上の LOG OUT をクリックすると、"PET REGISTER" のリンクが非表示になりましたか？

ペット詳細画面を作 ってみよう

- ・ 次にペット詳細画面を作つ
ていきます
- ・ ここでのポイントは、一覧
画面で選択されたペットの
情報の受け渡し部分ですね

Name

タマ

Image



Weight Logs

Birthday

2021年11月1日

Gender

男の子

[← Back to list](#)

REGISTER

まずは詳細画面を新たに用意

- 左上から "Add a new page..." を選択
- "Page name" は `pet_detail`
- "Clone from" は画面構成が似ているので `pet_register` を選びましょう

- クローン（コピー）してきてたままだと、登録画面と同じになってしまって見直していく

削除前

Name

Image

Birthday

Gender

REGISTER

- まずは入力項目として用意した入力要素をすべて削除します

削除後

Name

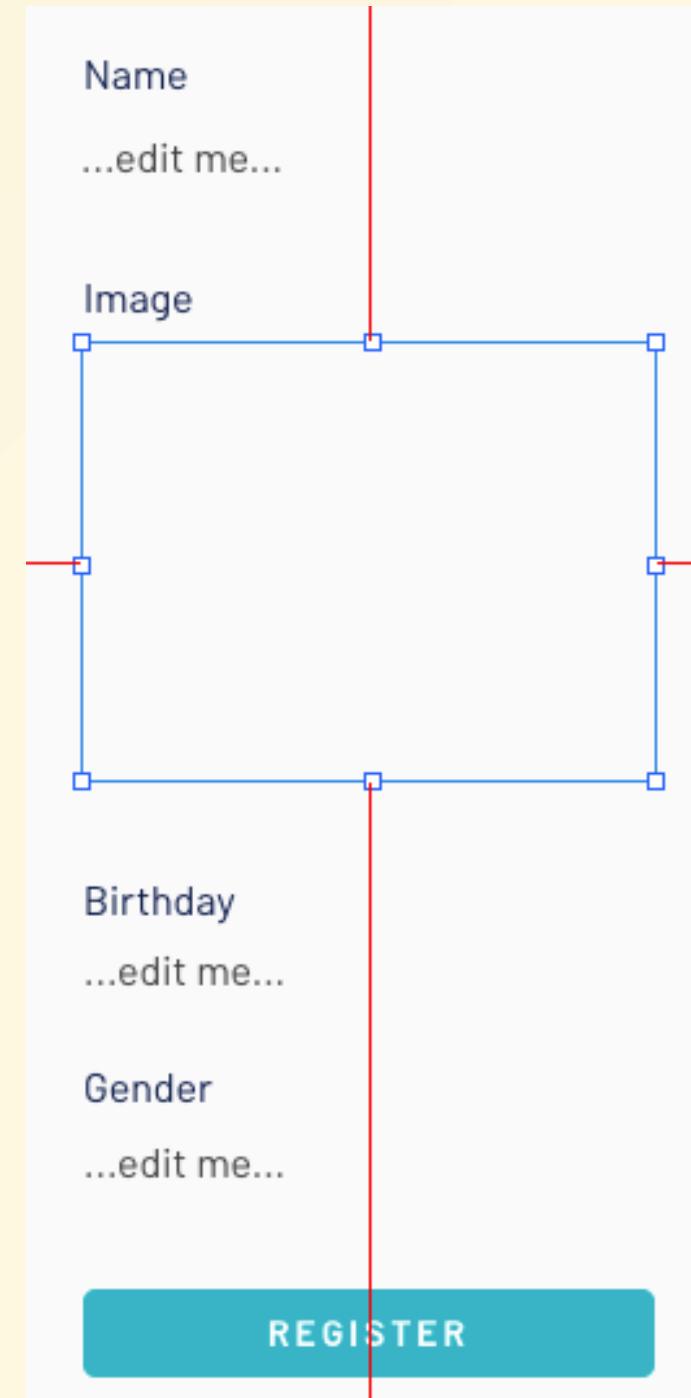
Image

Birthday

Gender

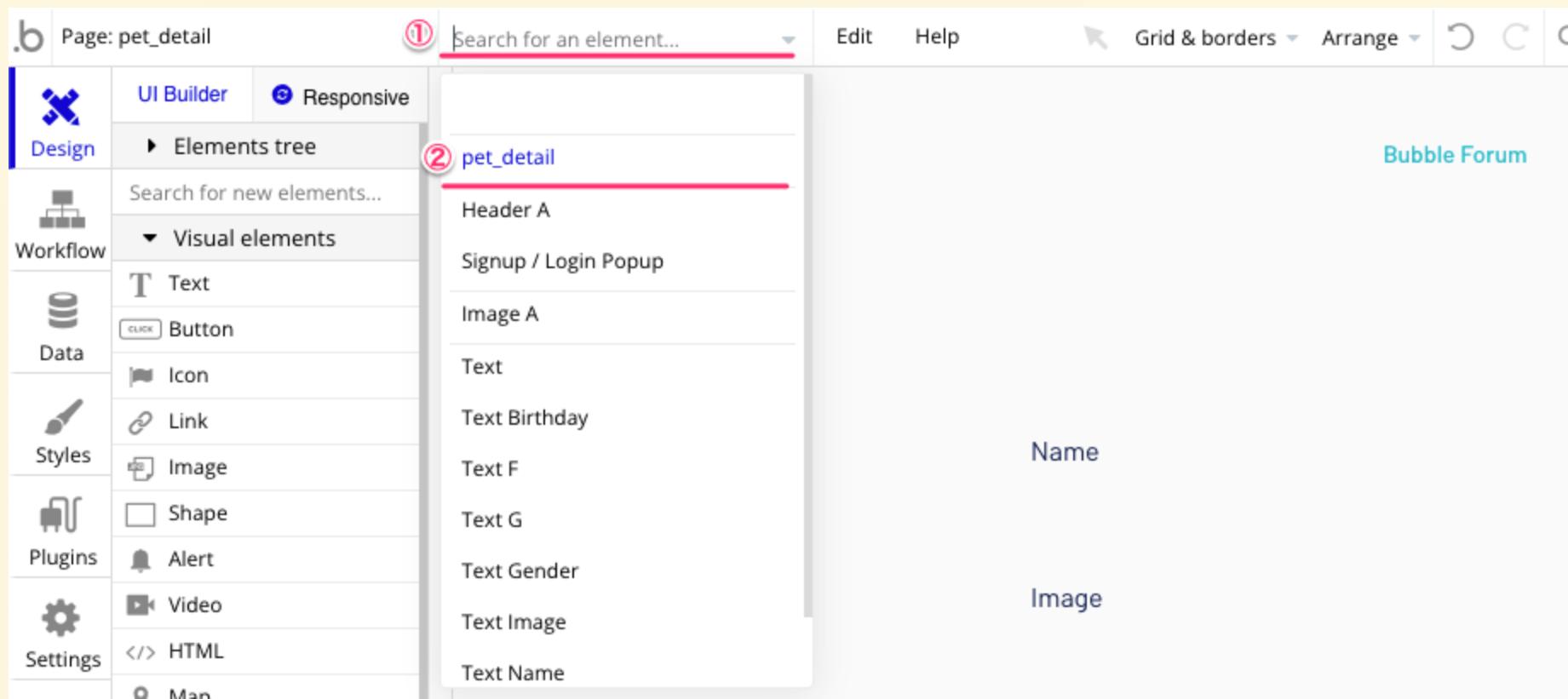
REGISTER

- 続いて表示用の要素を配置していきます
- "Visual elements" からそれぞれの要素ドラッグしてみましょう
 - Name: Text
 - Image: Image
 - Birthday: Text
 - Gender: Text
- 要素の幅 (w) はすべて 220px としてください

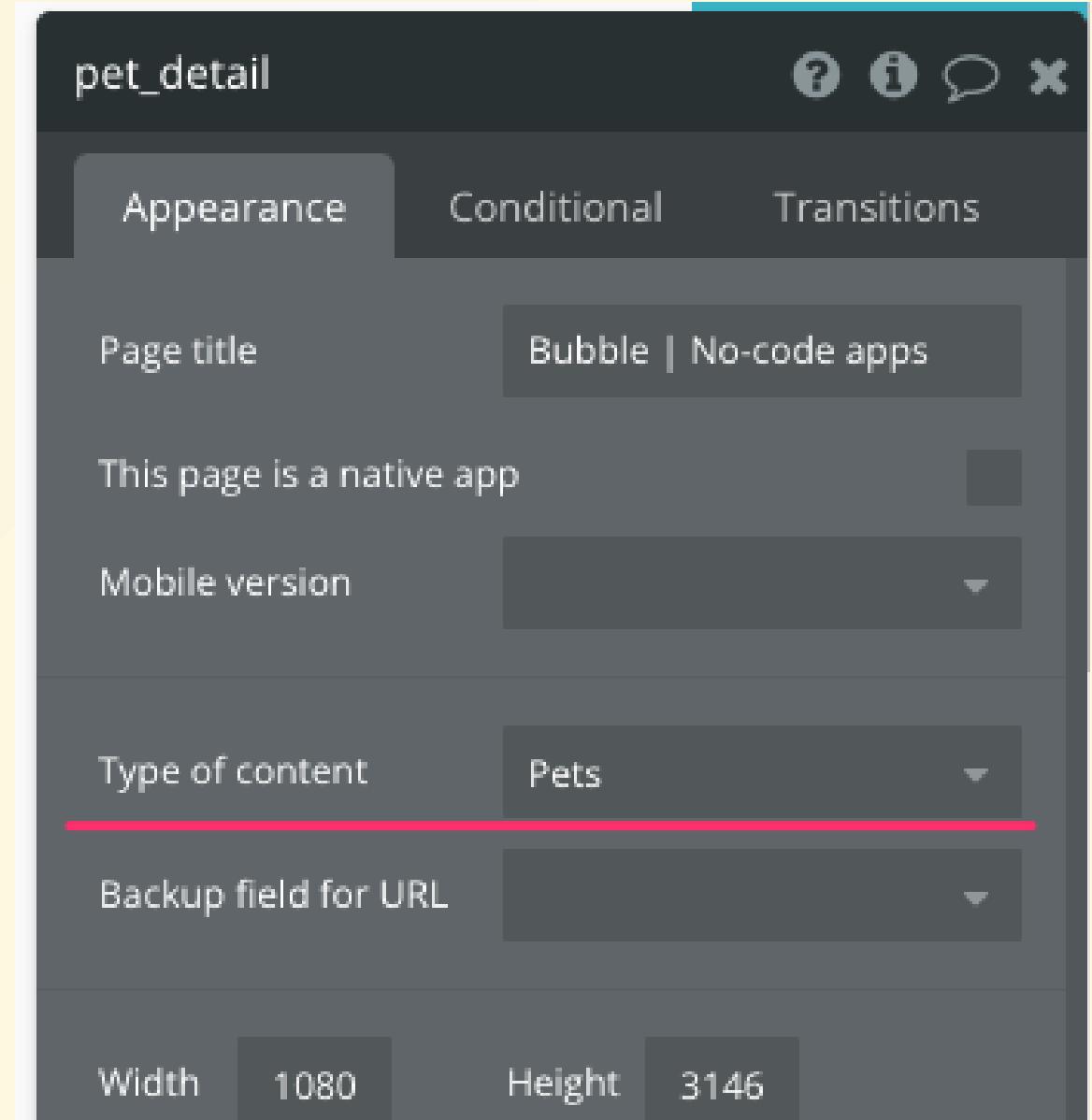


- 要素の準備が出来たら実際にデータベースから参照する値を定義していきます
- まず最初にこの画面自体の詳細設定用ダイアログを表示します

- 右パネルの左上の部分から `pet_detail` を選択すると、ダイアログが表示されます
 - 要素がいくつも重なっているときなどはここから要素を選ぶことも出来て便利です！



- その中から
Type of content という項目があるので、そこで Pets を指定します
- こうすることで、この画面を表示する元となるデータベースのタイプが何なのかを指定でき、各項目にはそのタイプのどのフィールドを使うのかを指定するだけで済みます

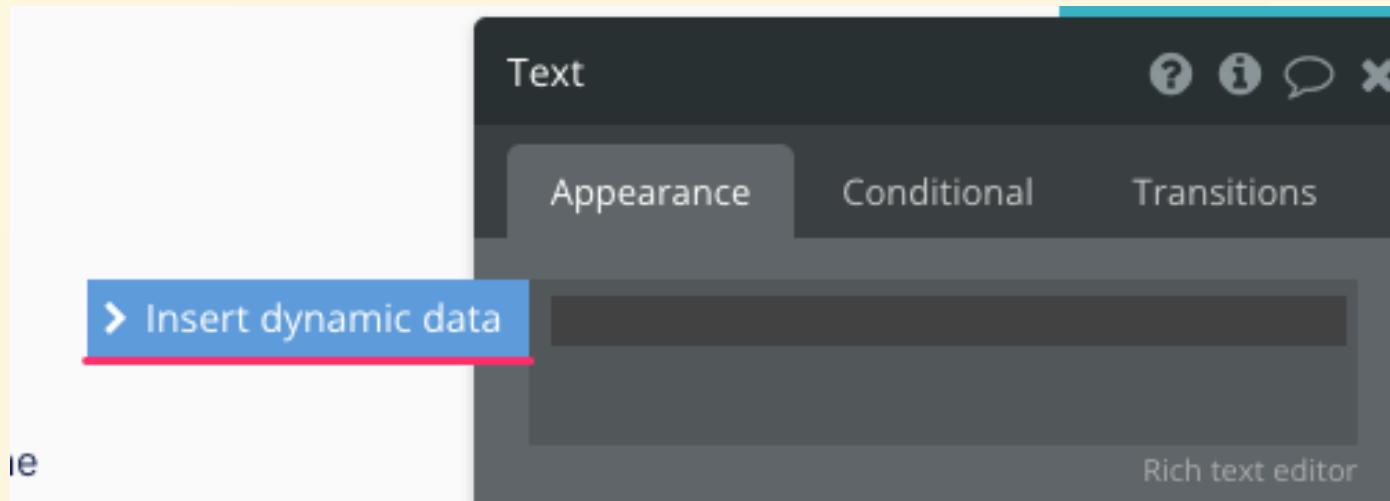


- ちなみに、タイプは指定できましたが、具体的にどのペットの情報なのか（ポチなのかタマなのか）は一覧画面から遷移するときに指定しますが、これは後ほど設定しましょう

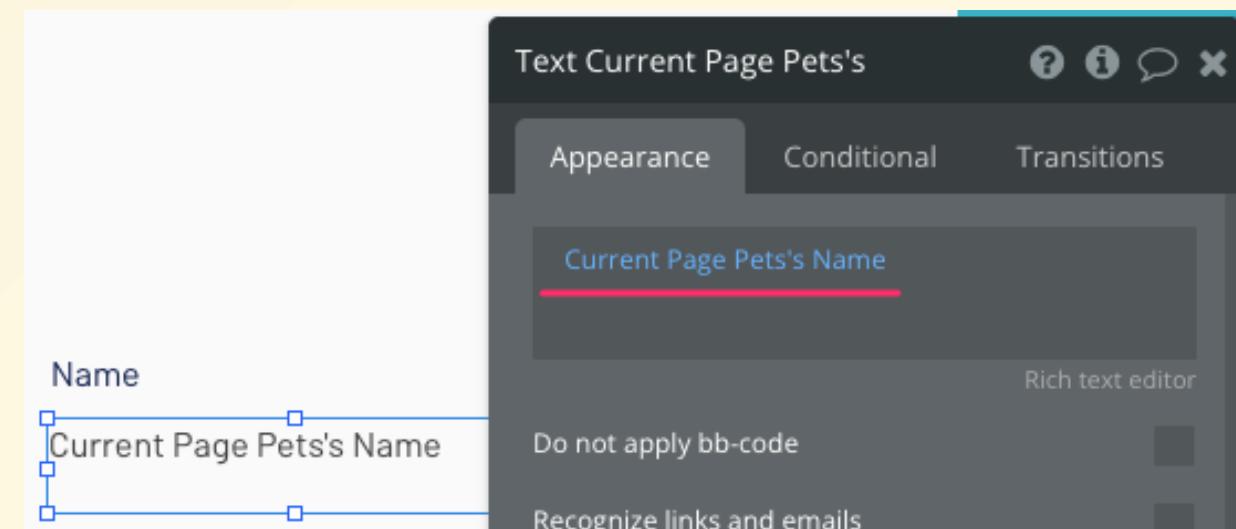
- それでは表示用の要素と Pets タイプのフィールドを紐付けていきます
- まずは Name からやってみましょう
- 今回のように、データベースから取得した値を動的に表示したい場合、どうやるか覚えていましたか？



- そうです！ "Dynamic data" を使います！
- Name の要素をダブルクリックし **...edit me...** と書かれた部分をクリックすると **Insert dynamic data** というボタンが表示されるのでクリック



- すると、プルダウンが表示され、その中から
Current Page Pets's をクリック
 - これは文字通り現在のページに割り当てられたペットの情報を指しています
- すると Pets タイプが持っているフィールドが表示されるので 's Name を選択



- 同じ要領で Image / Birthday / Gender の Dynamic data を設定してみましょう
- 設定が済んだらペット詳細に関するすべての要素を選択して中央寄せにします
 - 要素を選択した状態から右クリック --> Center horizontally

Name

Current Page Pets's Name

Image

Current Page Pets's Image

Birthday

Current Page Pets's Birthday

Gender

Current Page Pets's Gender

REGISTER

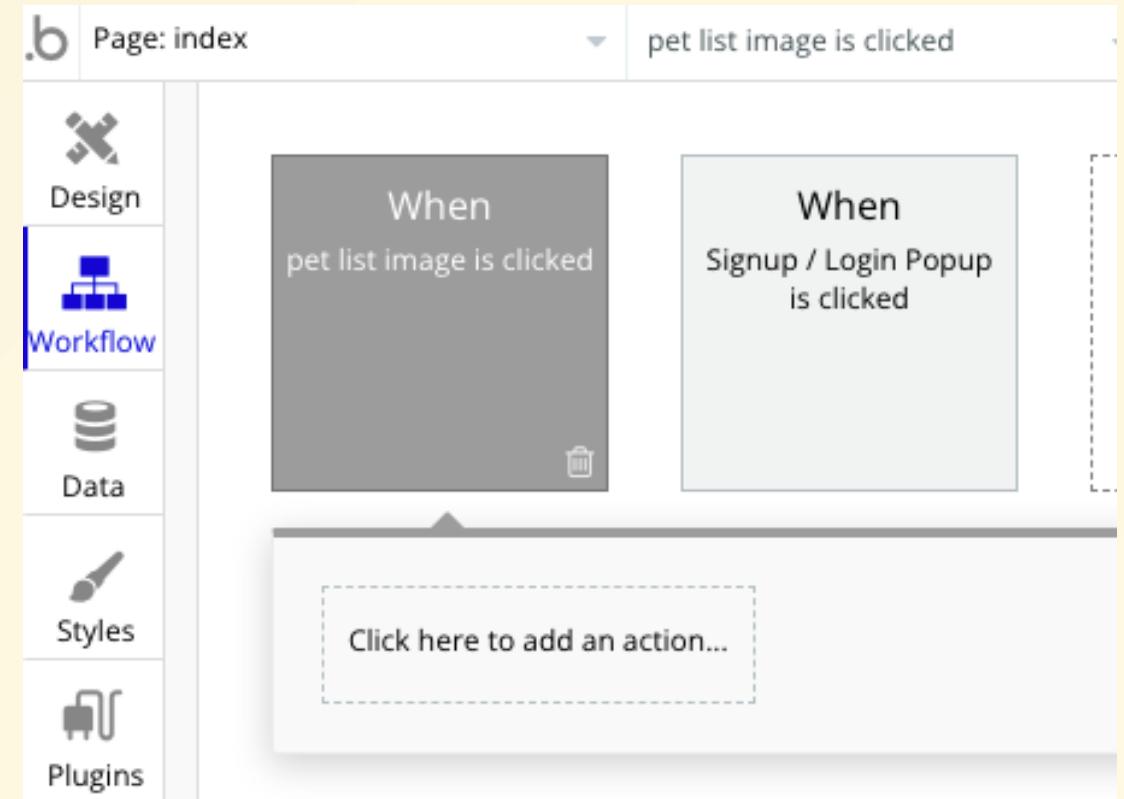
- 設定が済んだらプレビューしてみましょう
- 正しく値が表示されませんよね？
- これはまだ一覧画面からどのペットの情報なのか？を指定していないからです
 - **Lorem ipsum** と出ているのはよくあるダミーコンテンツのイディオムを Bubble が自動的に出しています

Name	
Image	
Birthday	Nov 10, 2021 10:15 pm
Gender	
REGISTER	

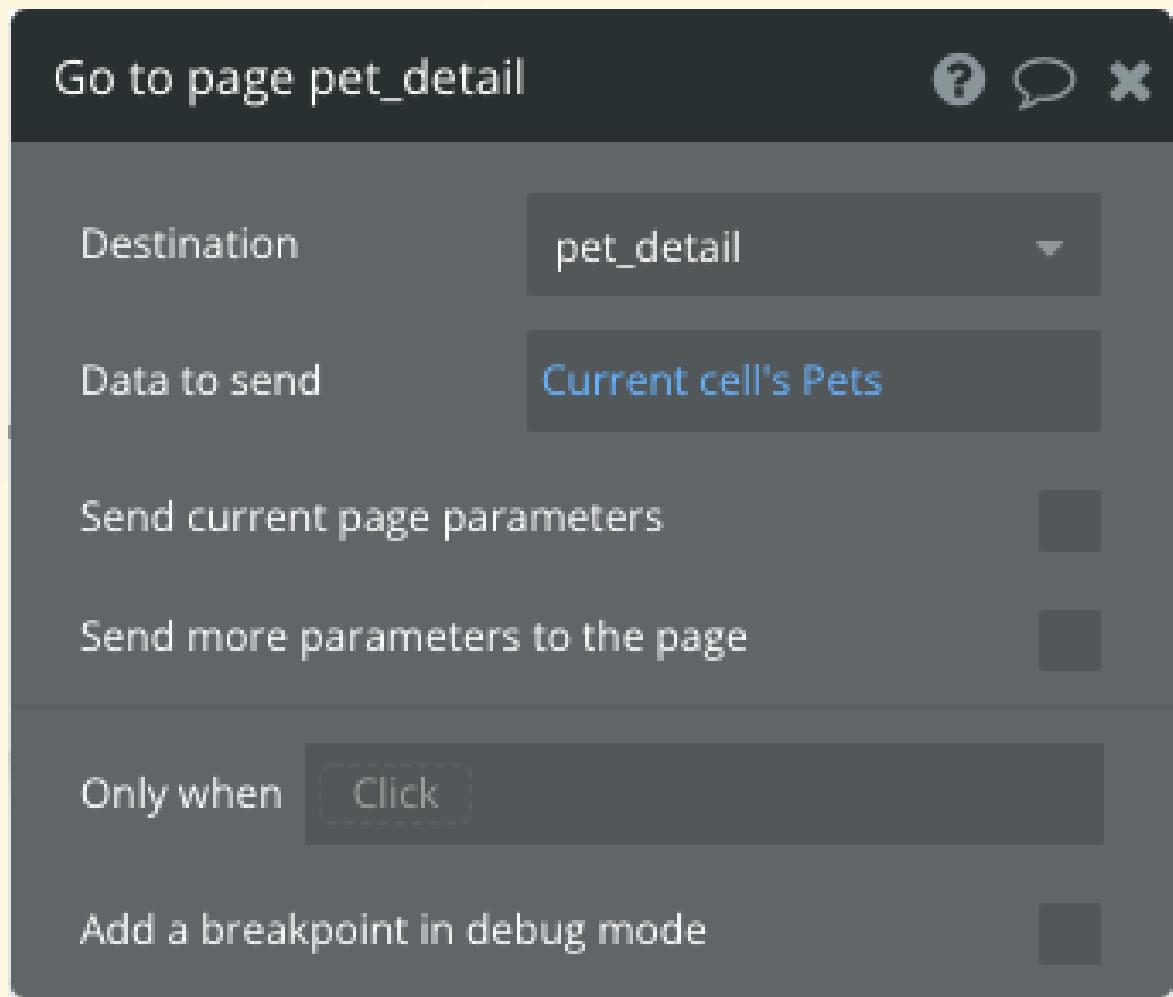
それでは一覧画面と詳細画面をつなげてみましょう

- 左上のメニューから index ページに切り替えます
- 画面操作のイメージとしては、ペット一覧画面で表示されているペットの画像をクリックしたら、そのペットの詳細画面へ遷移させたいですね
- なので、一覧画面のペット画像に対してワークフローを設定していきます

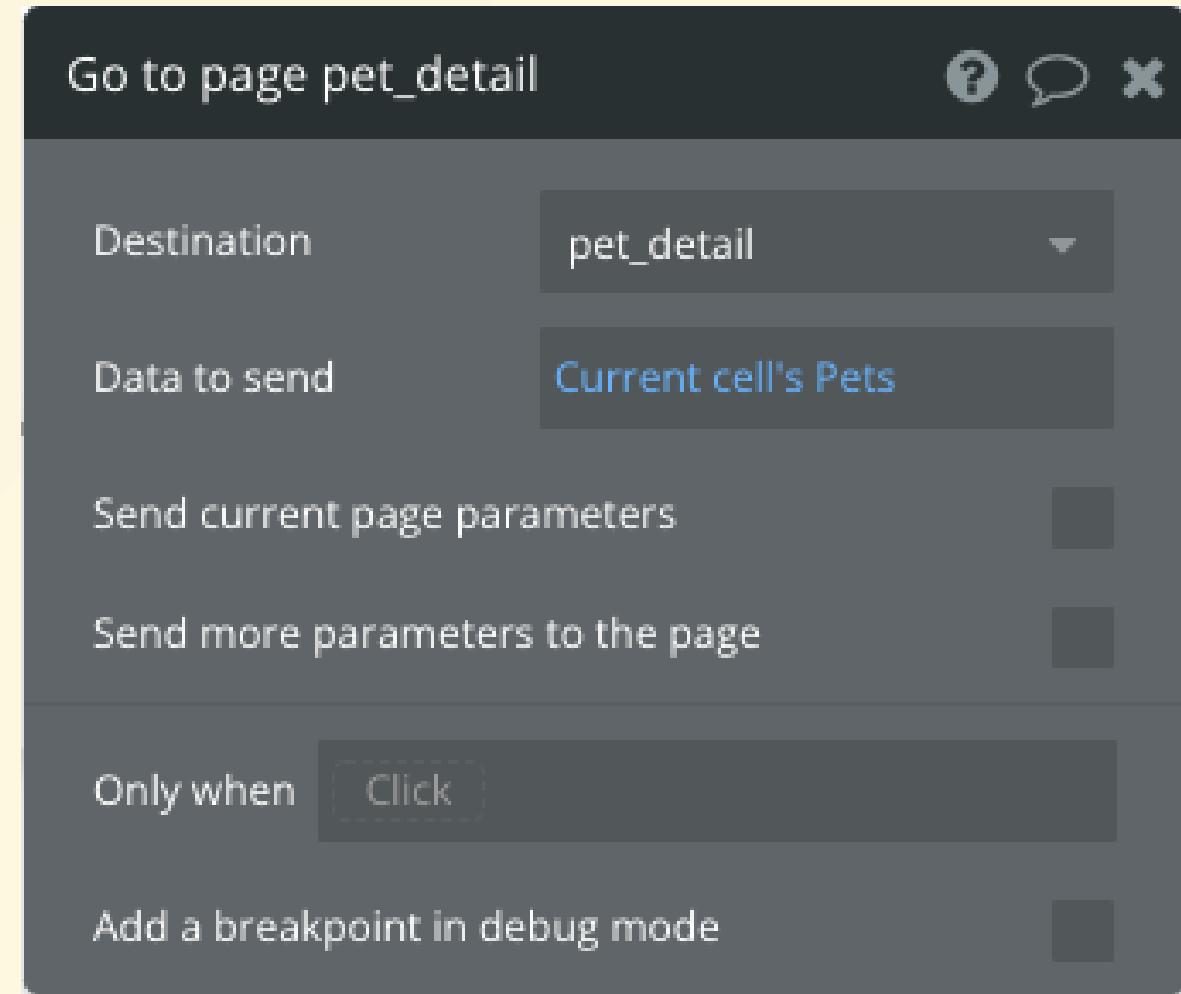
- ・ペット画像をダブルクリックし **Start/Edit workflow** をクリックします
- ・すると "When pet list image is clicked" の箱に対して振る舞いを設定する前の状態になっているかと思います



- なので "Click here to add an action" から Navigation -> Go to page を選択
- ダイアログが表示されるので Destination (遷移先) には pet_detail を指定



- そして **Data to send** に
Current cell's Pets を指定
 - これでペット詳細画面へ遷移するときに、現在のセルのペット情報を遷移先へ送る、という指定ができました



- それではプレビューしてみましょう
- 一覧画面でペットの画像を選択すると、そのペットの詳細画面が表示されることを確認できましたか？

Appendix

- ここで表示のアドバイス
- Birthday と Gender の表示が少し味気ないので、表示の書式を変えてみましょう

Name

まめ助

Image



Birthday

Nov 11, 2021 12:00 am

Gender

Male

REGISTER

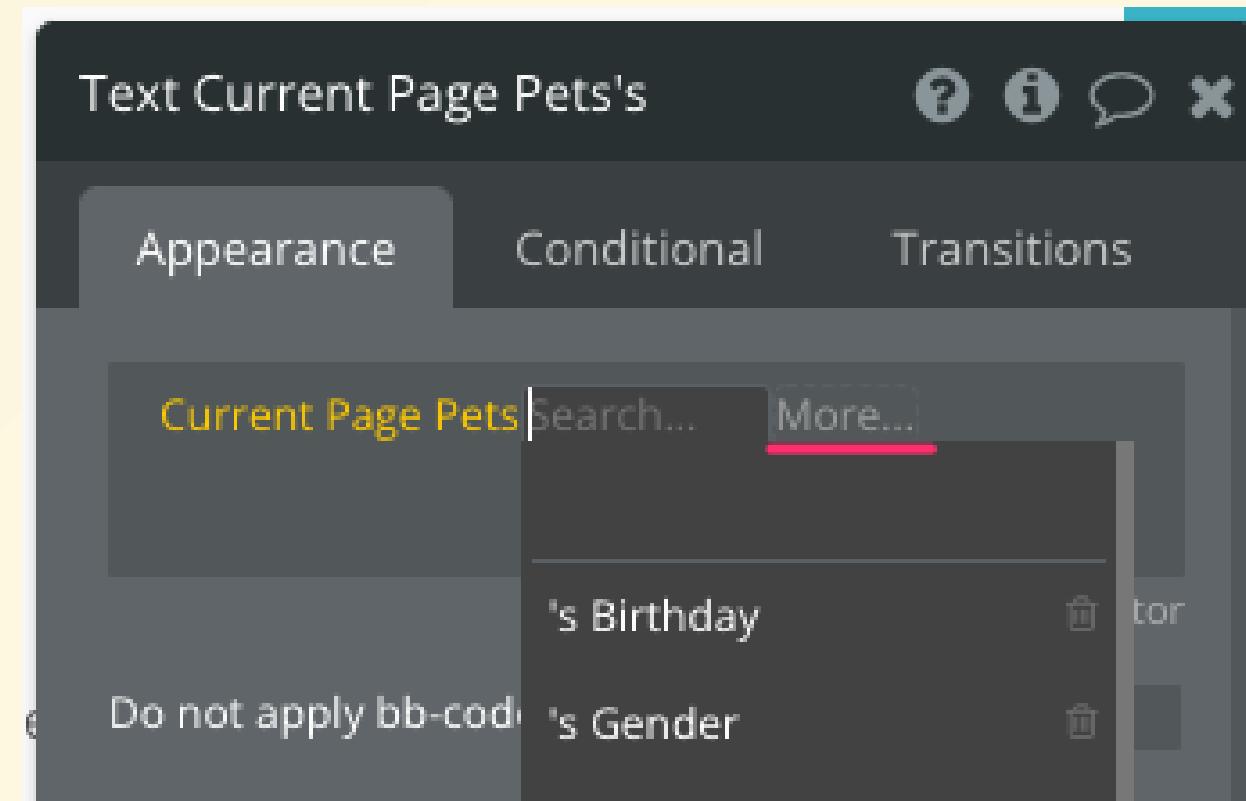
Appendix1: 誕生日の書式を日本風に変えてみます

- pet_detail 画面を Design タブで開きます
- そして Birthday の表示要素をダブルクリックし、詳細設定用ダイアログを表示します

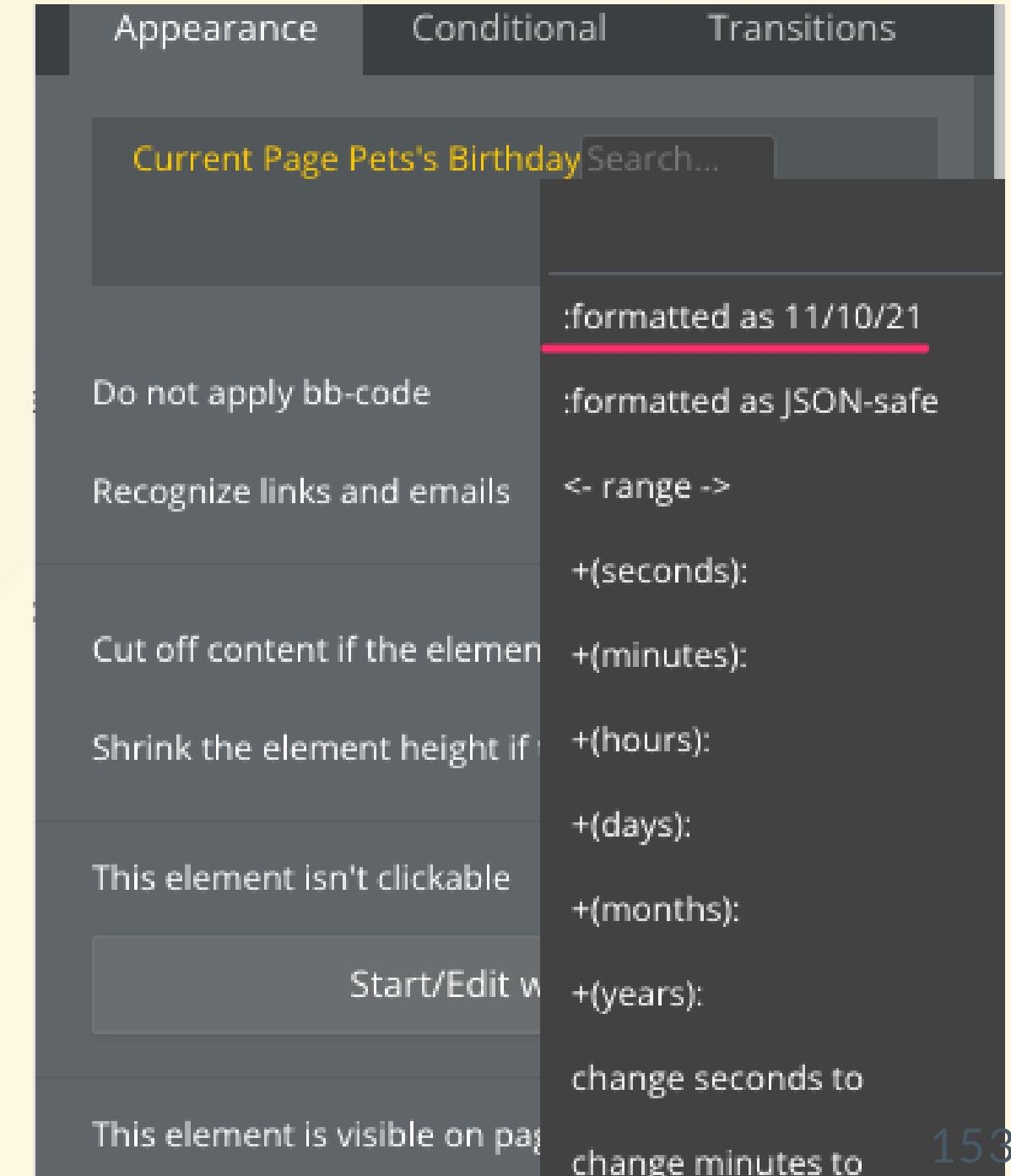
- そして

Current Page Pets's
Birthday

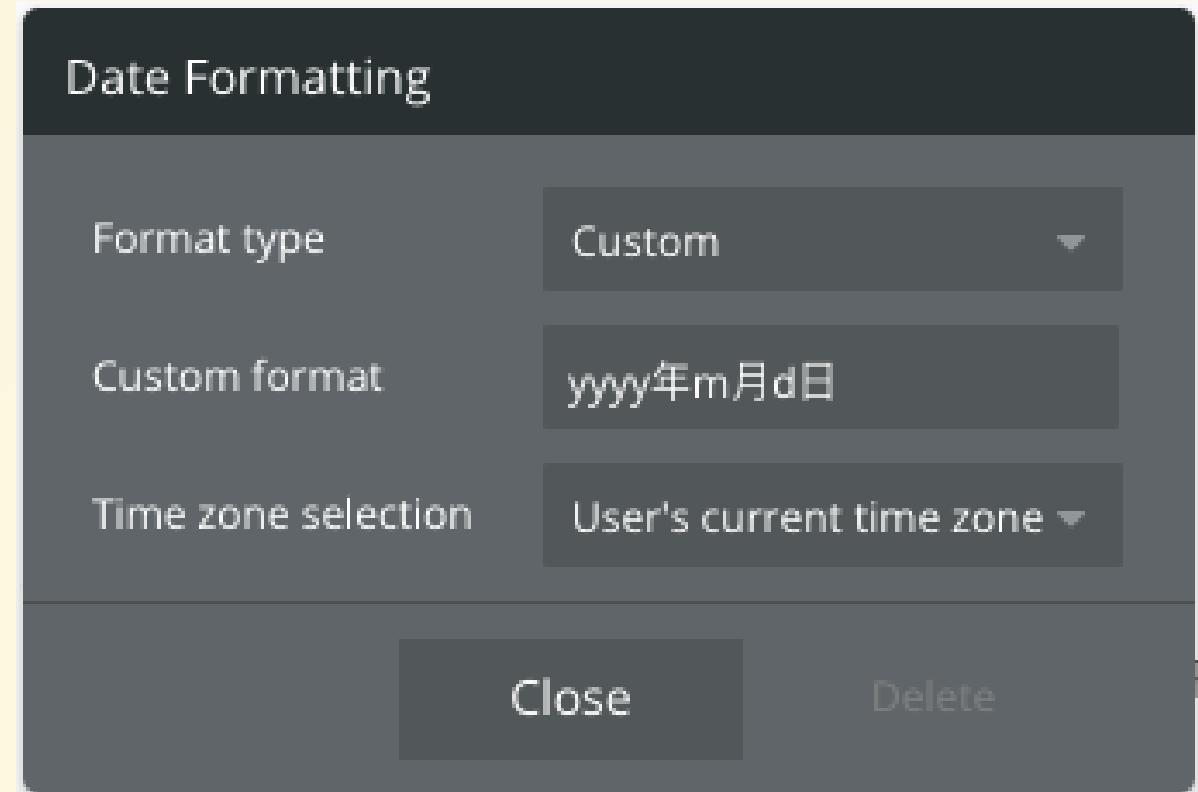
の Birthday の部分をクリックすると、後に More という項目が表示されると思しますので、それをクリックします



- すると Birthday の値の表示フォーマットをさらに指定できるようになります
- 今回は一番上の :formatted as DD/MM/YY をクリックします
 - DD/MM/YY には本日日付が入っていると思います



- すると、さらに隣に Date Formatting のダイアログが表示されると思います
- ここでフォーマットの細かな指定が可能です
- 今回は Format type に Custom を選択し、Custom format に yyyy年m月d日 を指定します



- それではプレビューしてみましょう
 - 一度 index ページを開いた上で、プレビュー実行
- ペット一覧からペット詳細を開いてみると Birthday の日付が指定した書式になっていると思います

Name

まめ助

Image



Birthday

2021年11月11日

Gender

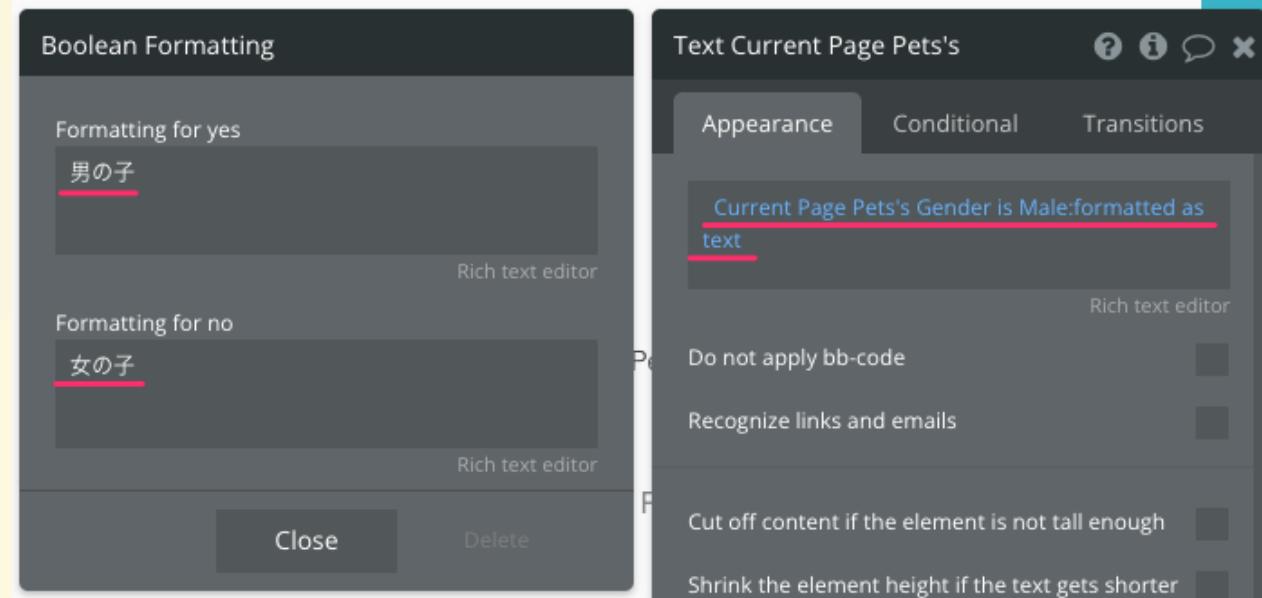
Male

REGISTER

Appendix2: 性別のラベルを変えてみよう

- pet_detail 画面を Design タブで開きます
- そして Gender の表示要素をダブルクリックし、詳細設定用ダイアログを表示します
- 先ほどと同じ要領で **Current Page Pets's Gender** の Gender の部分をクリックし **More** をクリックし、続けて下記のように入力します

- **is** を選択、**Male** と手入力してエンターキークリック、**:formatted as text** を選択
- すると **Boolean Formatting** というダイアログが表示されるので下記の通り入力します
 - **Formatting for yes:** 男の子
 - **Formatting for no:** 女の子



- これは下記の処理を書いています

現在のペットの Gender (性別) が Male だった場合には「男の子」、そうでない場合は「女の子」を表示する

ここで少しプログラミング的な要素が出てきましたね！

- それでは index ページから プレビュー実行してみましょう
- ペット一覧からペット詳細を開いてみると Gender の値が「男の子」「女の子」になっていると思います

Name

まめ助

Image



Birthday

2021年11月11日

Gender

男の子

REGISTER

Appendix3: 詳細画面から一覧画面への導線を設けてみよう

- 詳細画面から一覧へ戻るための導線を用意してみましょう
- これまでの講義を踏まえるときっとイメージは沸いていると思うので、完成イメージとヒントだけ記載しておきます

ヒント

- 使用する要素は
Visual elements の Link
- Link 要素の場合、画面遷移
は Workflow ではなく、要
素自身に "Destination
page" を設定できます
- 完成イメージ

Name

まめ助

Image



Birthday

2021年11月11日

Gender

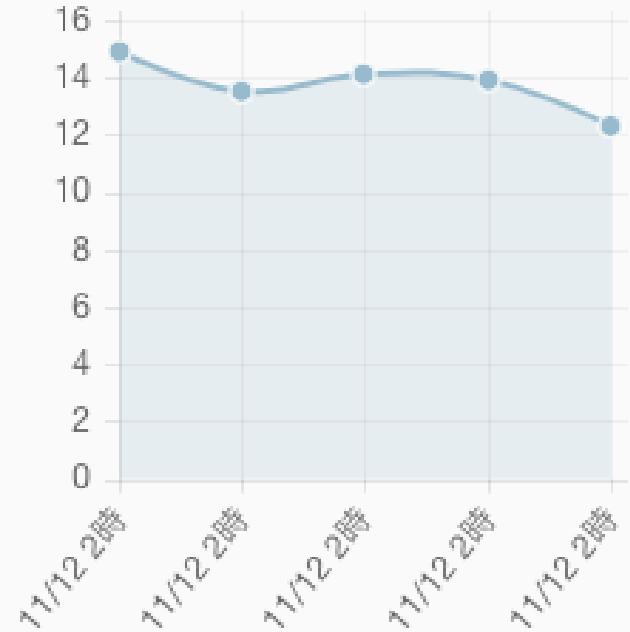
男の子

[← Back to list](#)

REGISTER

ペットの体重を管理 しよう

- 続いてペットの体重管理を
見てきましょう



Weight

pet weight

← Back to detail

ADD

- まずは Type を用意していきます
- Data タブから New type として **PetWeightLogs** を入力して Create します
- 続いて field を設定していきます

- 画面から入力されたペットの体重
 - WeightKg: number
- どのペットの体重かを紐付ける
 - pet: Pets

The screenshot shows a software interface for managing data types. On the left, there's a vertical sidebar with icons for Design, Workflow, Data (which is selected), Styles, Plugins, and Settings. The main area has tabs for Data types, Privacy, App data, Option sets, and File manager. Under 'Data types', there's a section for 'Custom data types'. A table lists existing data types: 'Pets' (Publicly visible), 'PetWeightLogs' (Publicly visible, currently selected and highlighted in red), and 'User' (Privacy rules applied). To the right, under 'Fields for type PetWeightLogs', there are several fields: 'Type name' (PetWeightLogs), 'pet' (type 'Pets'), 'WeightKg' (type 'number'), 'Creator' (type 'User'), and 'Modified Date' (type 'date'). There are also sections for 'default' and 'Built-in field'.

- データベースの準備が出来たので画面を設定していきましょう
- まずはペットの体重管理画面を新たに用意します
- Add a new page からページ名 `pet_weight_register` でコピー元は `pet_register` を選択して CREATE

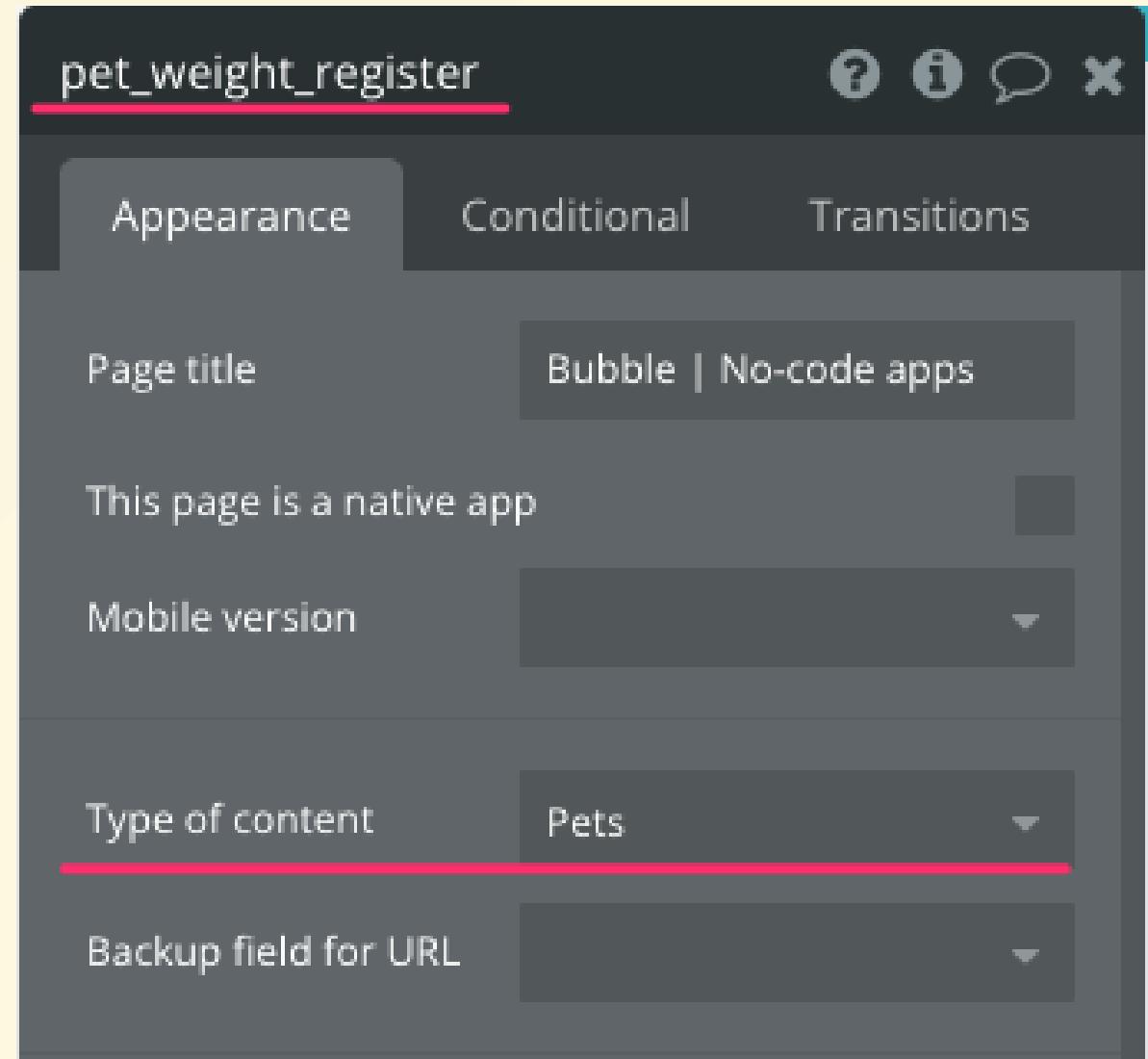
Create a new page

Page name

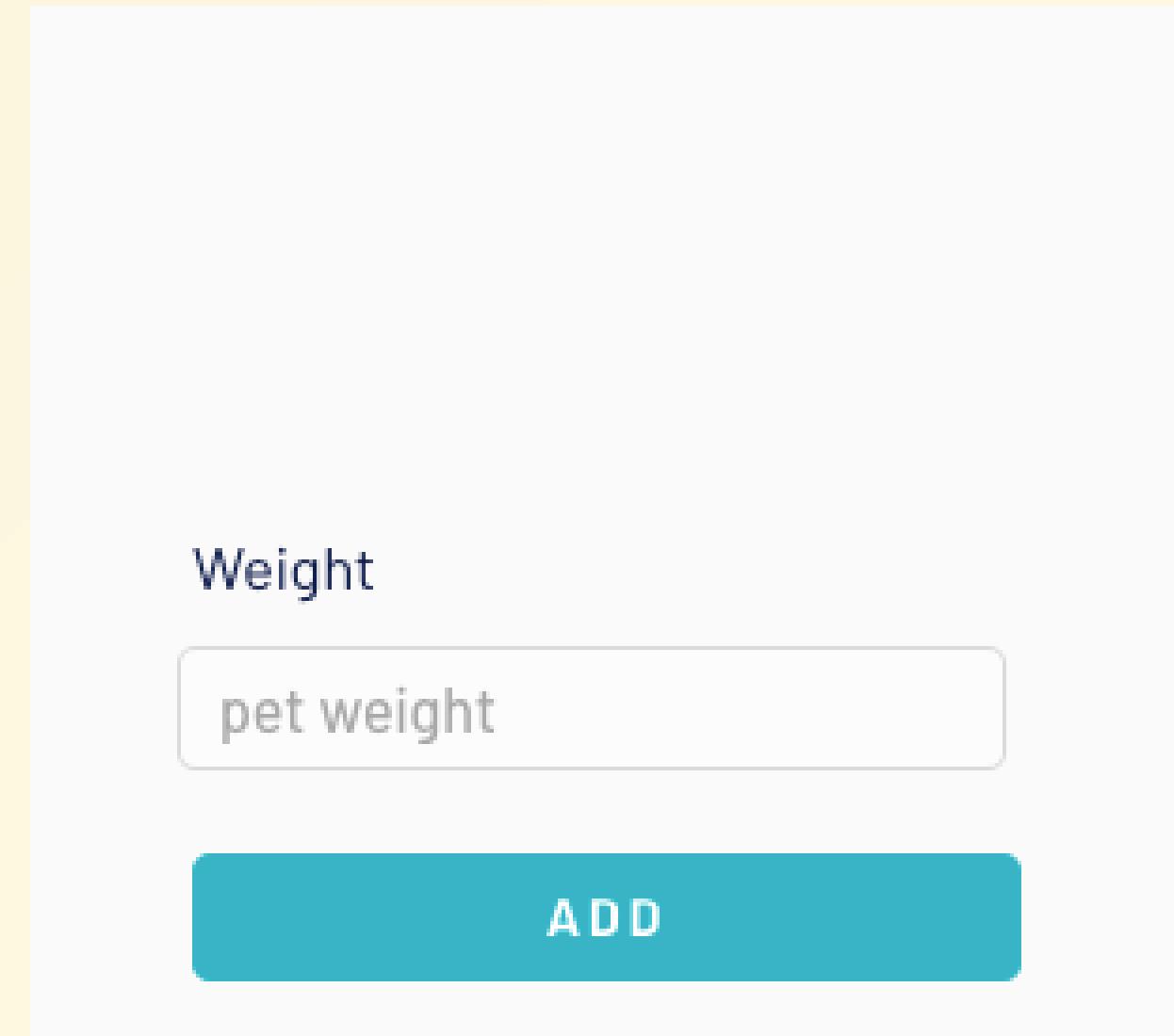
Clone from ▾

CREATE Cancel

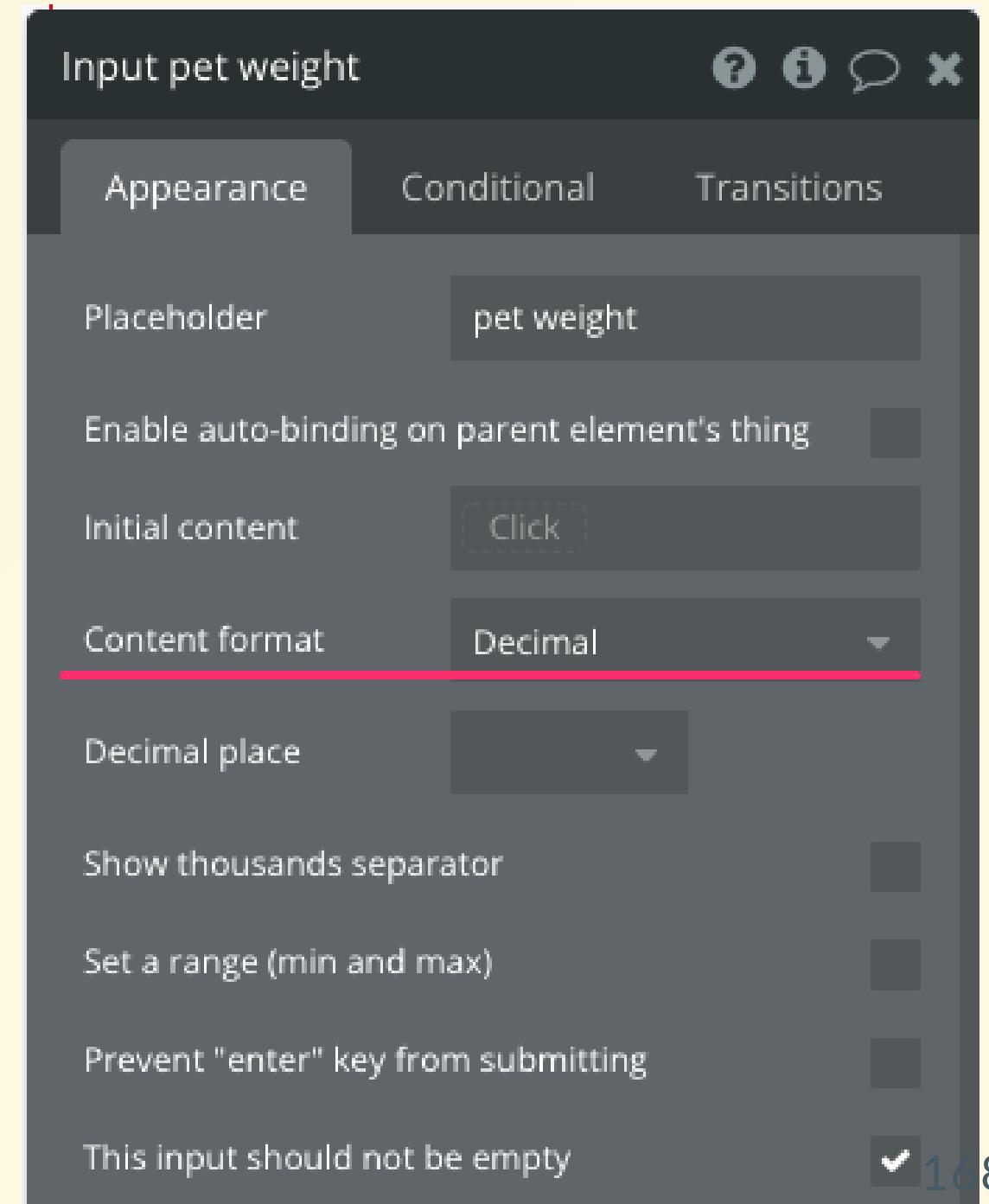
- この画面も、ペット詳細から遷移する前提であるため pet_weight_register 画面に対する詳細設定ダイアログから、Type of content に Pets を指定します



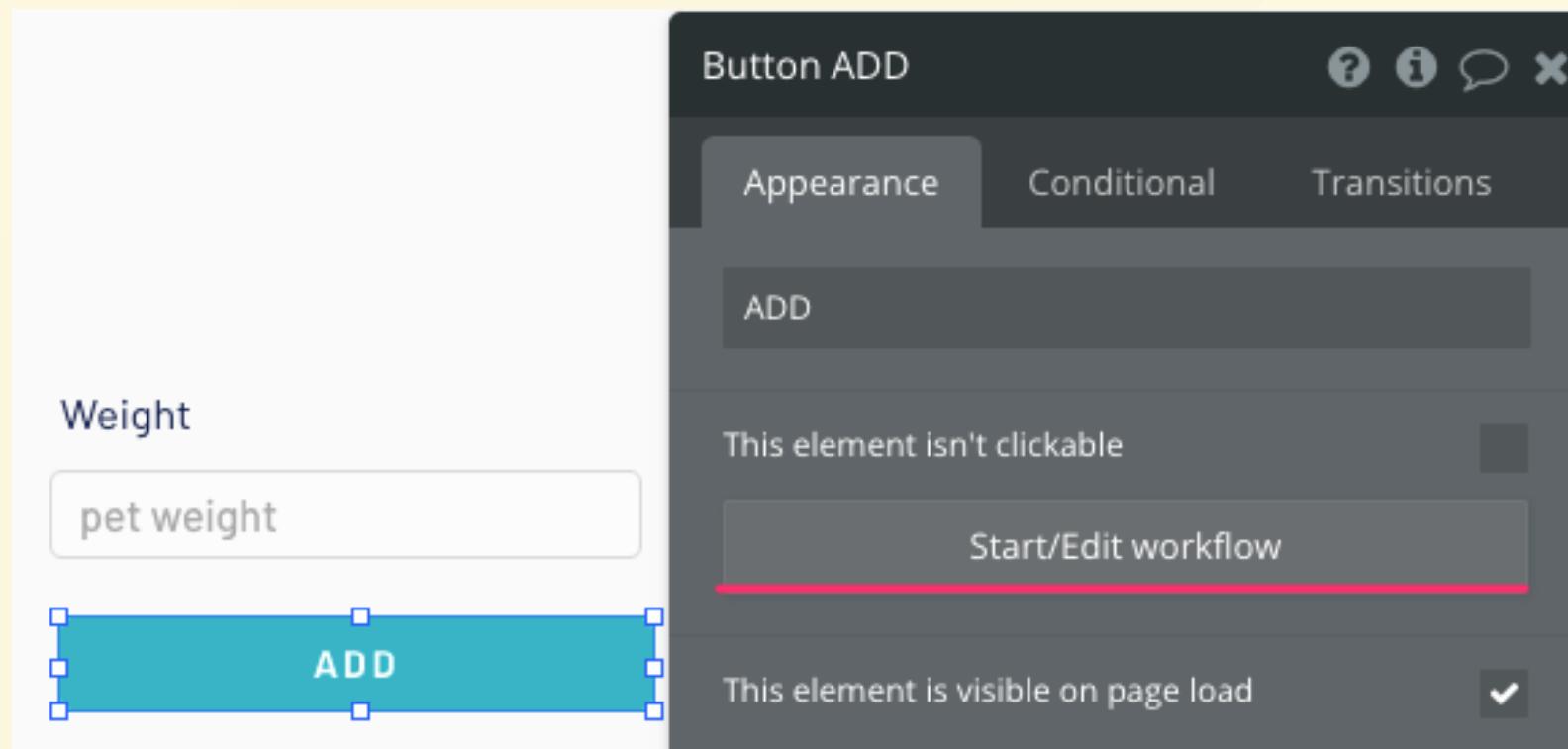
- コピー元のペット登録画面の要素が残っているので、不要なものは削除しつつ参考画像のような要素を配置します
- ただし、グラフ描画エリアは現時点では配置しません



- このとき、体重の入力要素について、今回は小数点付き数字だけの入力を許容したいので、その設定だけ行います
- ペットの体重入力要素をダブルクリックし
Content format に Decimal を選択します
 - ちなみに Integer は小数点なしの整数入力の場合に選択します



- 次に、体重保存のボタンクリックに対してワークフローを設定していきます
- ADD ボタンをダブルクリックし、詳細設定ダイアログから Start/Edit workflow を選択



- すると `When Button ADD is clicked` に対して、すでに振る舞いがいくつか設定されています
- これはページ作成時に `pet_register` をコピーして作ったため、そのときのワークフロー設定が引き継がれています
- 今回は新たに振る舞いを定義していくので、既存のワークフローは削除します
- Step1 / Step2 の振る舞いを選択すると、右下に `delete` と表示されているので、そこから振る舞いを削除します



- それではワークフローを設定していきましょう
- まずやりたいのは、「画面で入力された体重の値を、先ほど用意した PetWeightLogs に保存する」ですね
- 今までの講義内容を踏まえて設定してみてください！
- 次のページに解説を載せているので、まずはやってみましょう！

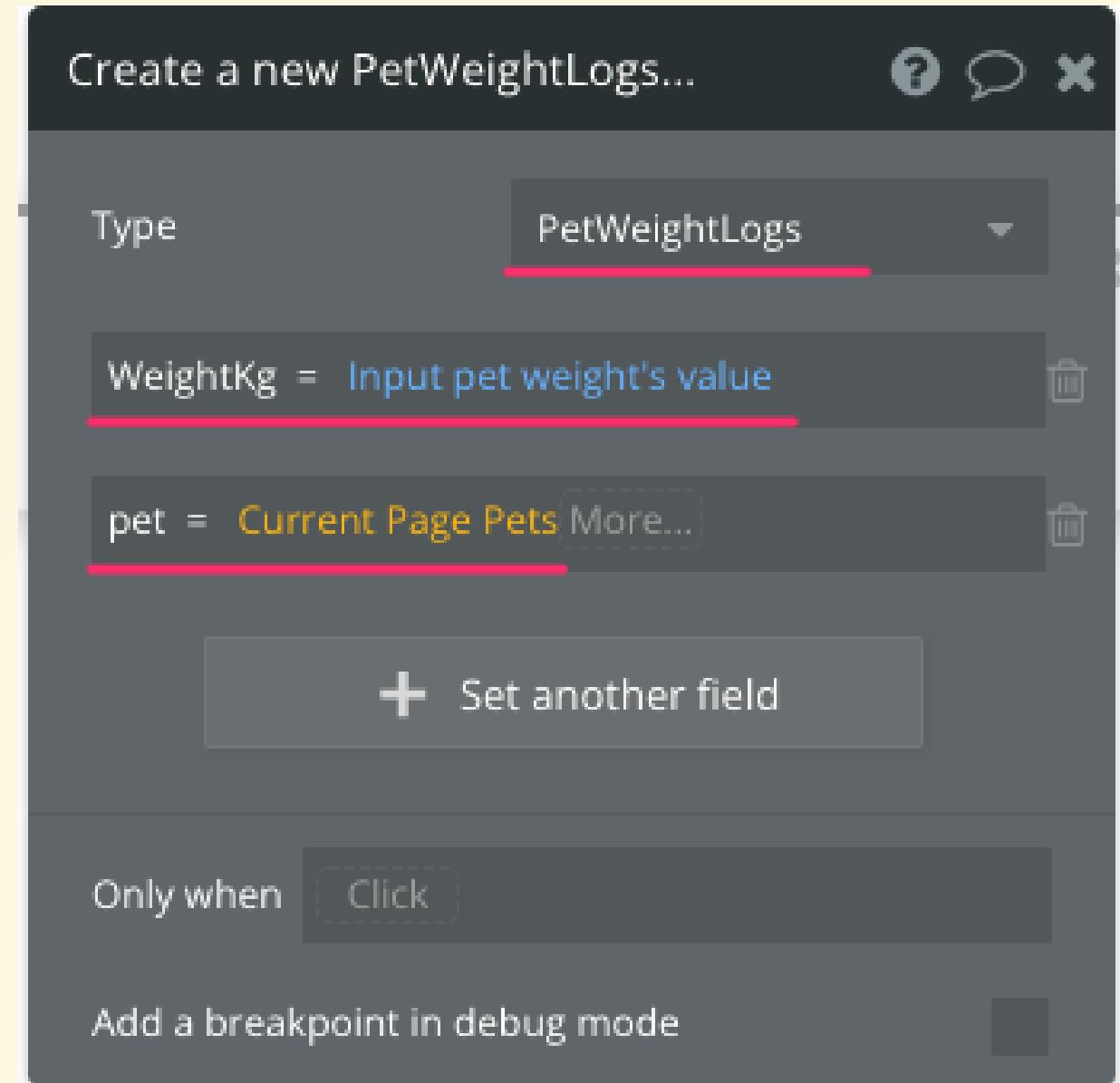


- まず設定するアクションとしては Data(Thing) --> Create a new thing... ですね

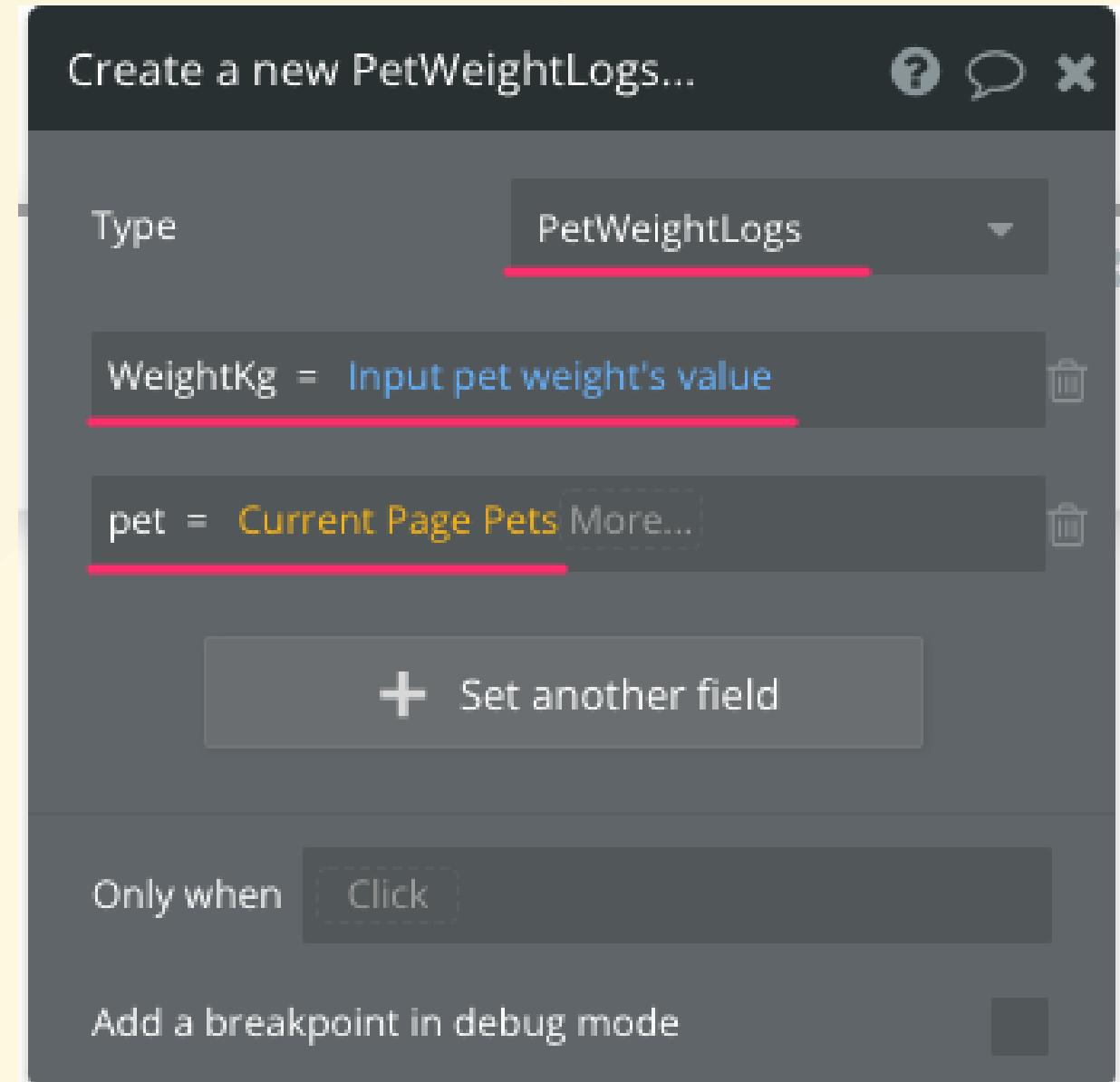
Search for an action...

Account	Create a new thing...
Navigation	Make changes to thing...
Data (Things)	Make changes to a list of things...
Email	Delete thing...
Payment	Delete a list of things...
Analytics	Copy a list of things...
Element Actions	Set a thing's slug...
Plugins	Download data as CSV
	Upload data as CSV
	Delete an uploaded file

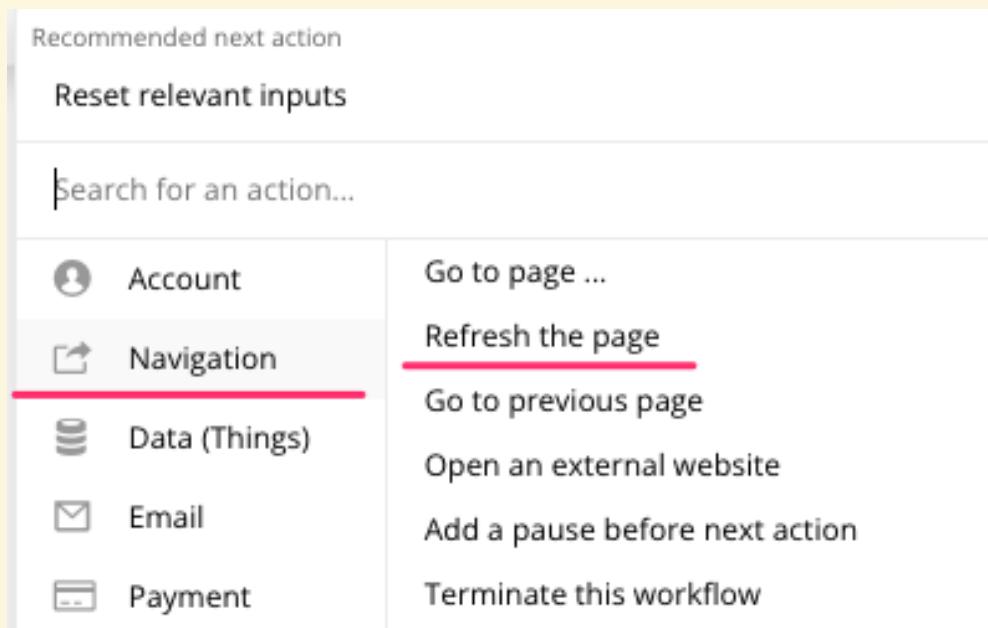
- そして、登録するタイプとフィールドを指定します
- 今回登録するのはペットの体重なので Type は PetWeightLogs



- 設定するフィールドは `WeightKg` と `pet` ですね
- `pet` フィールドを設定しないと、どのペットの体重なのかを判断できないので指定します
- その際に指定するのが `Current Page Pets`、今表示している画面の `Pets` となります



- 登録ができたら画面を再描画して、この後表示するグラフを最新化したいので、登録後に画面を再描画する振る舞いを定義しましょう
- 画面操作に関するアクションは **Navigation** でしたね
- その中のサブメニューを見てみると **Refresh this page** というピッタリのアクションがあるのでこれを選択しましょう



- これでペットの体重登録が出来たので、詳細画面からの導線を用意しましょう
- 詳細画面を開き、Image と Birthday の間にリンクを用意しましょう
- リンクは `Visual elements` の `Link` を使いましょう

- ポイントとしては 2 つ
 - 1. 遷移先を **Destination page** に設定する
 - 2. リンク先に対して現在ページのペット情報を渡してあげる
- 次のページに設定イメージを書いていますのでまずはご自身で設定してみましょう

- こんな感じ！

The screenshot shows a 'Link Weight Logs' dialog box from a web editor. The dialog has tabs for Appearance, Conditional, and Transitions, with Appearance selected. The title 'Weight Logs' is displayed in a rich text editor. Below the editor are three settings: Link destination set to 'Internal page' with a dropdown arrow, Destination page set to 'pet_weight_register' with a dropdown arrow, and Data to send set to 'Current Page Pets'. A red vertical line and the text '240px' are overlaid on the left side of the dialog, indicating the width of the link text.

Link Weight Logs

Appearance Conditional Transitions

Weight Logs

Rich text editor

Link destination Internal page

Destination page pet_weight_register

Data to send Current Page Pets

Weight Logs

Birthday

Current Page Pet's Birthday: f...

Gender

Current Page Pet's Gender is...

240px

- 詳細画面から体重管理への導線が出来たら、逆の導線も用意しておきましょう
- 考え方は詳細画面に設置したリンクと同じで大丈夫です

- 設定できたら一覧画面に戻ってプレビューを実行してみましょう
- 一覧、詳細、体重管理へ遷移した後、Weight に値を入力して ADD を押すと画面が再描画されましたかね？

Weight

pet weight

← Back to detail

ADD

- 念のため、データベースにデータが登録されていることを確認してみましょう
- Data の App data タブから All PetWeightLogs を選択すると左側にデータが登録されていますよね？

The screenshot shows the Bubble app interface with the following details:

- Page:** pet_weight_register
- Tab:** App data (selected)
- View:** All PetWeightLogs
- Database views:** Database views, Application data - All PetWeightLogs - Development version
- Actions:** New view, Primary fields, Search, 2 entries (displaying 2), New entry, Delete (0), Upload, Modify, Export, Bulk
- Fields:** pet, WeightKg, Created Date, Modified Date, Slug, Created By
- Data:** Two entries listed:
 - pet: 1636457887522x519076068, WeightKg: 13.1, Created Date: Nov 11, 2021 9:17 am, Modified Date: Nov 11, 2021 9:17 am, Slug: (empty), Created By: kyogoku+bubble_test@gmail.com
 - pet: 1636457887522x519076068, WeightKg: 12.3, Created Date: Nov 11, 2021 9:13 am, Modified Date: Nov 11, 2021 9:13 am, Slug: (empty), Created By: kyogoku+bubble_test@gmail.com

ペットの体重をグラ フィカルに表示して みよう

- 登録したペットの体重を Adalo の時と同様グラフを使って表示してみましょう



Weight

pet weight

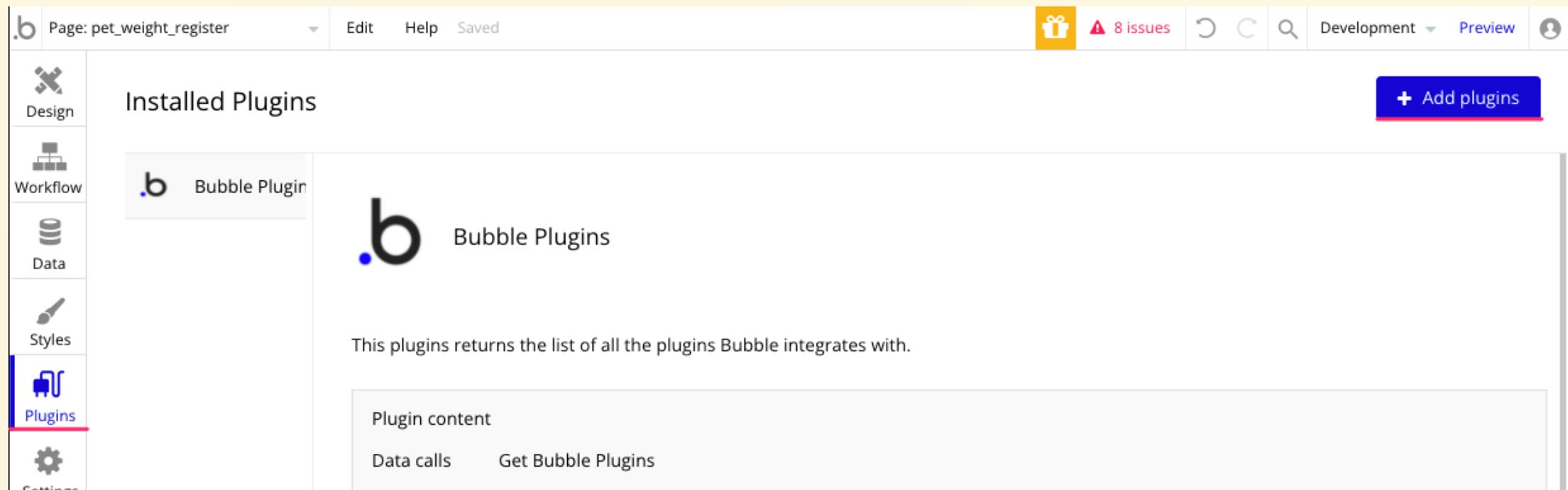
← Back to detail

ADD

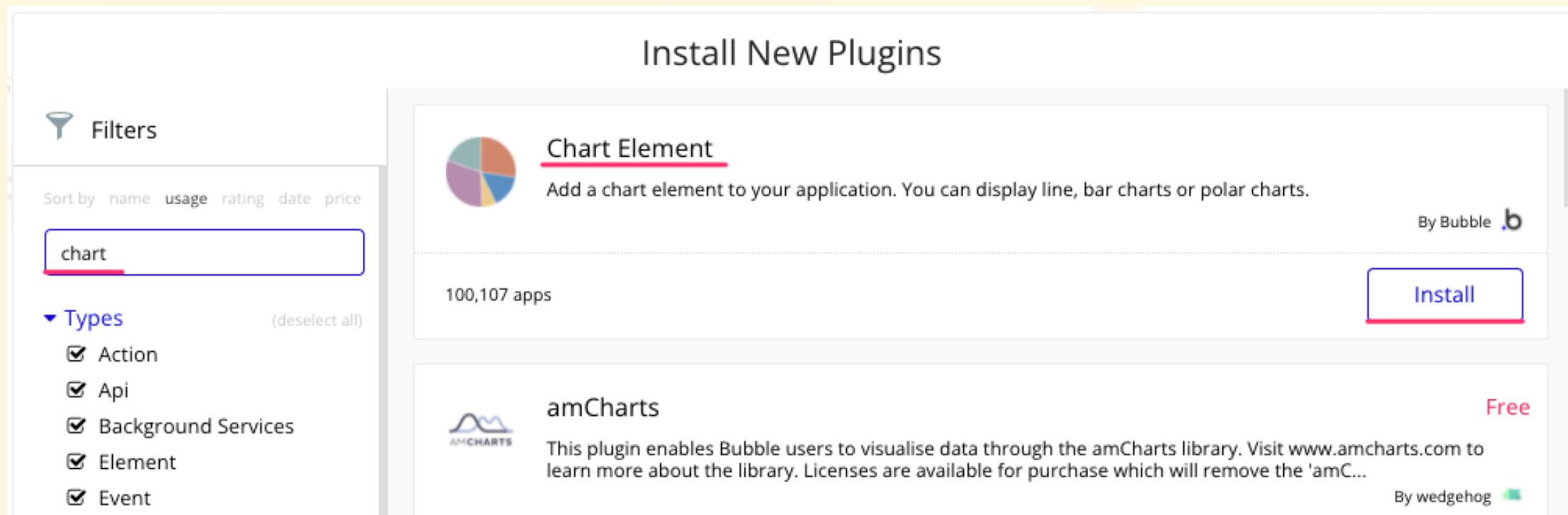
まずはプラグインを導入します

- Bubble でもグラフ表示の機能は標準では搭載していないため、プラグインという形で機能を追加します
- Adalo の時は Component という形でしたが、Bubble では Plugins から導入します

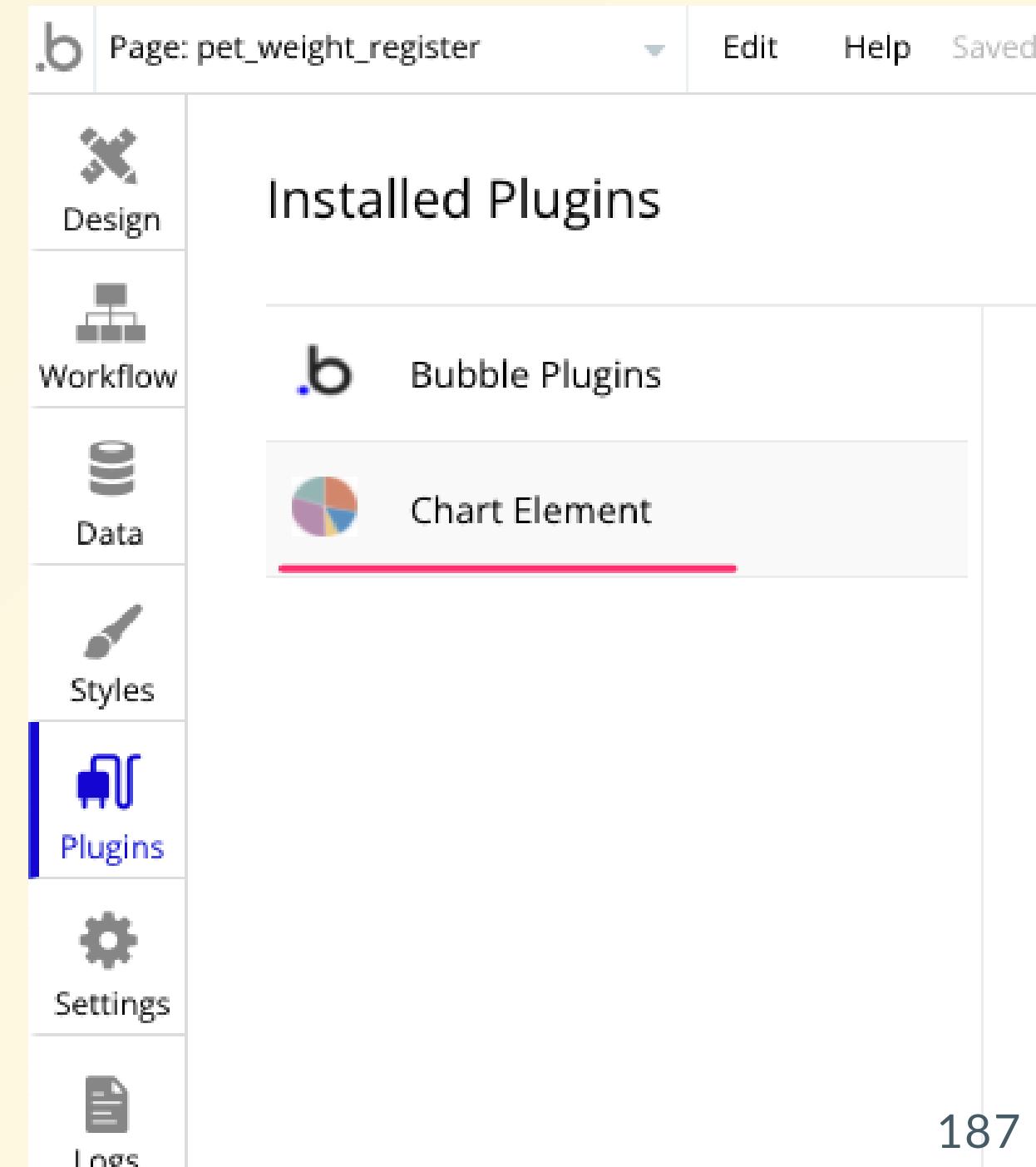
- 左メニューから Plugins を選択し、プラグイン画面を表示します
- そして、右上の **+ Add plugins** を押して追加するプラグインを検索します



- **Install New Plugins** というポップアップが表示されますので、左パネルの検索窓に **chart** と入力します
- すると、右パネルに検索結果が表示されますので、その中から（おそらく一番上にある） **Chart Element** の **Install** ボタンをクリックします



- インストールが完了したら DONE をクリックしてポップアップを閉じます
- Installed Plugins に Chart Element が入っていれば OK デス



- Bubble のプラグインには無料のものと有料のものが存在します
- プラグイン一覧の画面で **Free** と書かれているもの、もしくは **By Bubble** と書かれているものは無料です
 - とくに **By Bubble** と書かれているものは文字通り Bubble の公式が提供しているプラグインのため、何かと信頼できます
- 逆に金額が記載されているプラグインは文字通り有料です
- 様々なプラグインがあるので、導入する際には Free のもので事足りるプラグインがないかをまずは探し、有料のプラグインしかどうしても存在しない場合には有料のプラグインインストールを検討してみても良いでしょう



Simple LineChart from ApexCharts

Free

Easy to use beautiful line chart from the Apex JS library

By Avalan Labs

2,710 apps

All versions 4.0/5.0

Current version 4.0/5.0

Plugin page

Install



Chart JS - Charts and Graphs

\$190 once or \$9/month

24 highly-customizable and beautiful charts! These charts are built on the most popular Javascript chart library (Chart.js) as well as hundreds of hours of development for custom features and perf...

By TechBlocks

2,192 apps

All versions 4.8/5.0

Current version 4.8/5.0

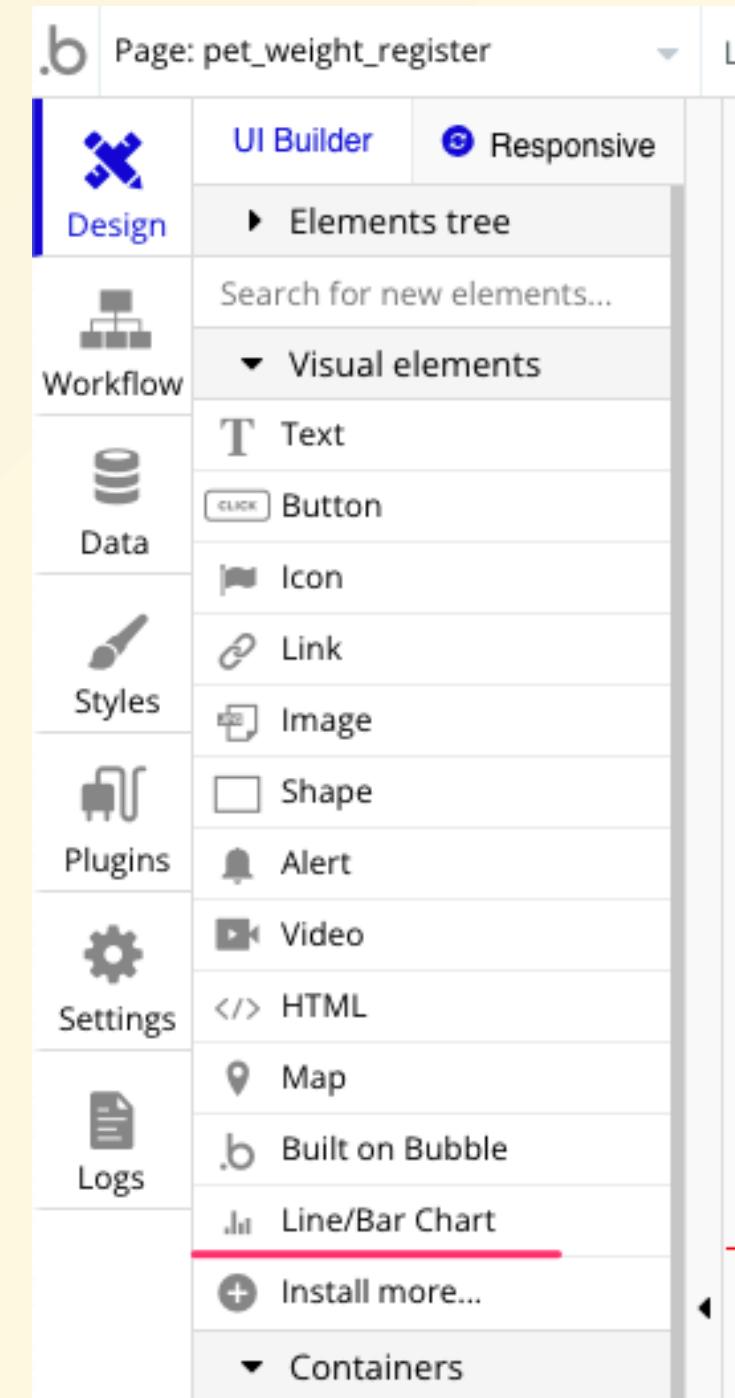
Plugin page

Buy \$190

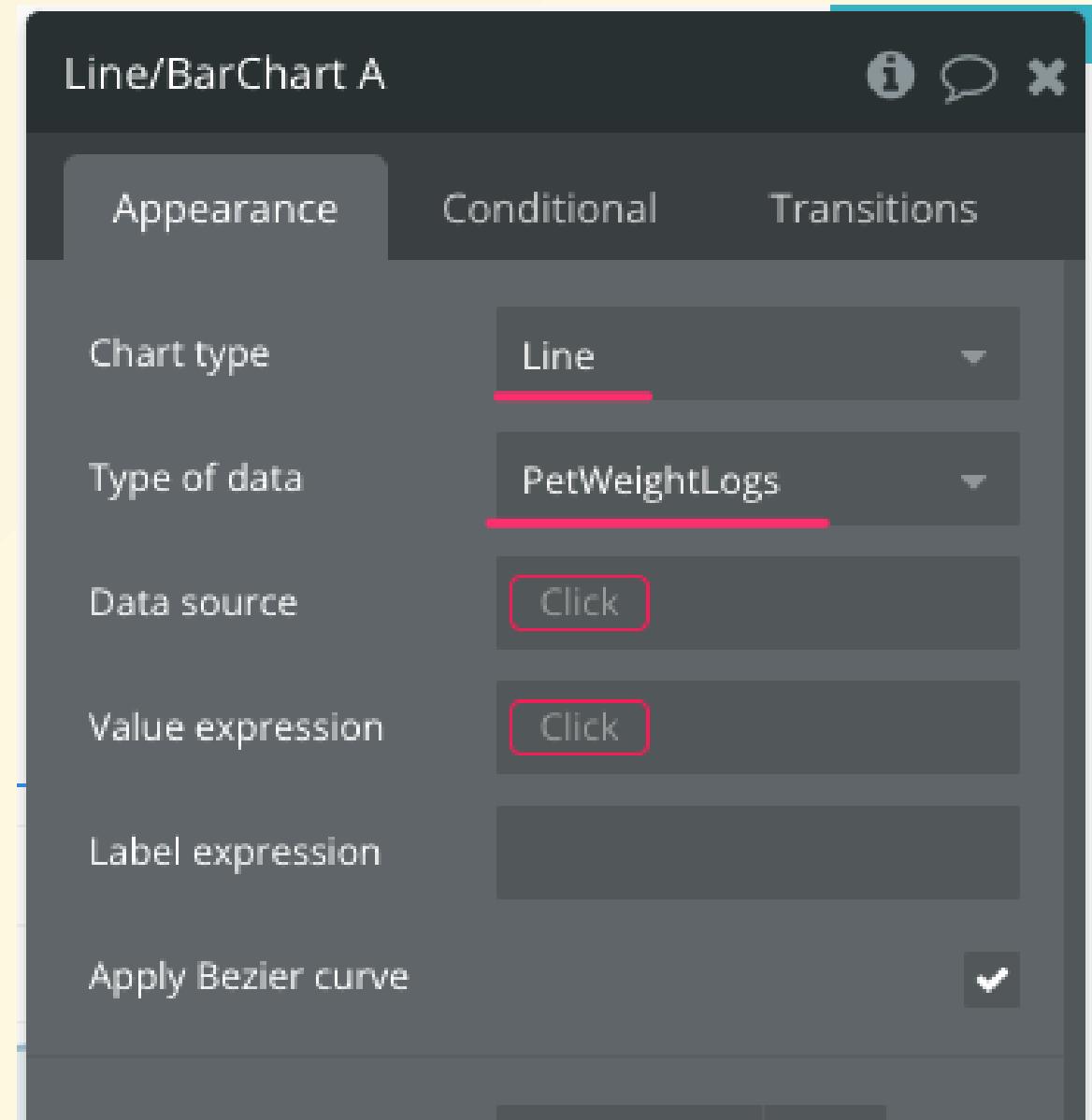
Subscribe

それでは実際にグラフ描画を行っていきましょう

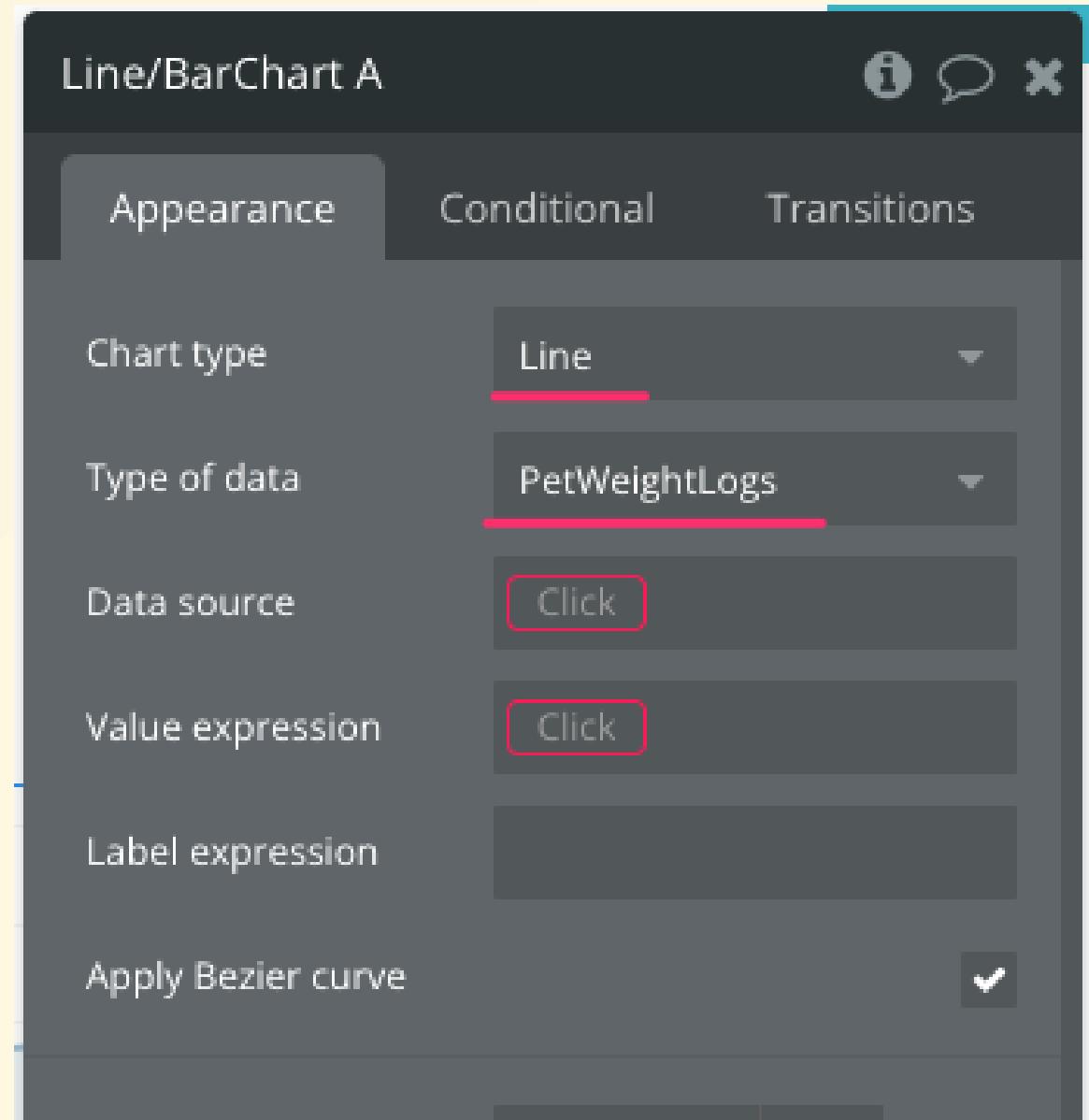
- pet_weight_register ページの Design を開きます
- すると Visual elements の中に Line/Bar Chart という要素が新たに追加されていると思いますので、この要素を選択して急速体重入力の上部にドラッグします



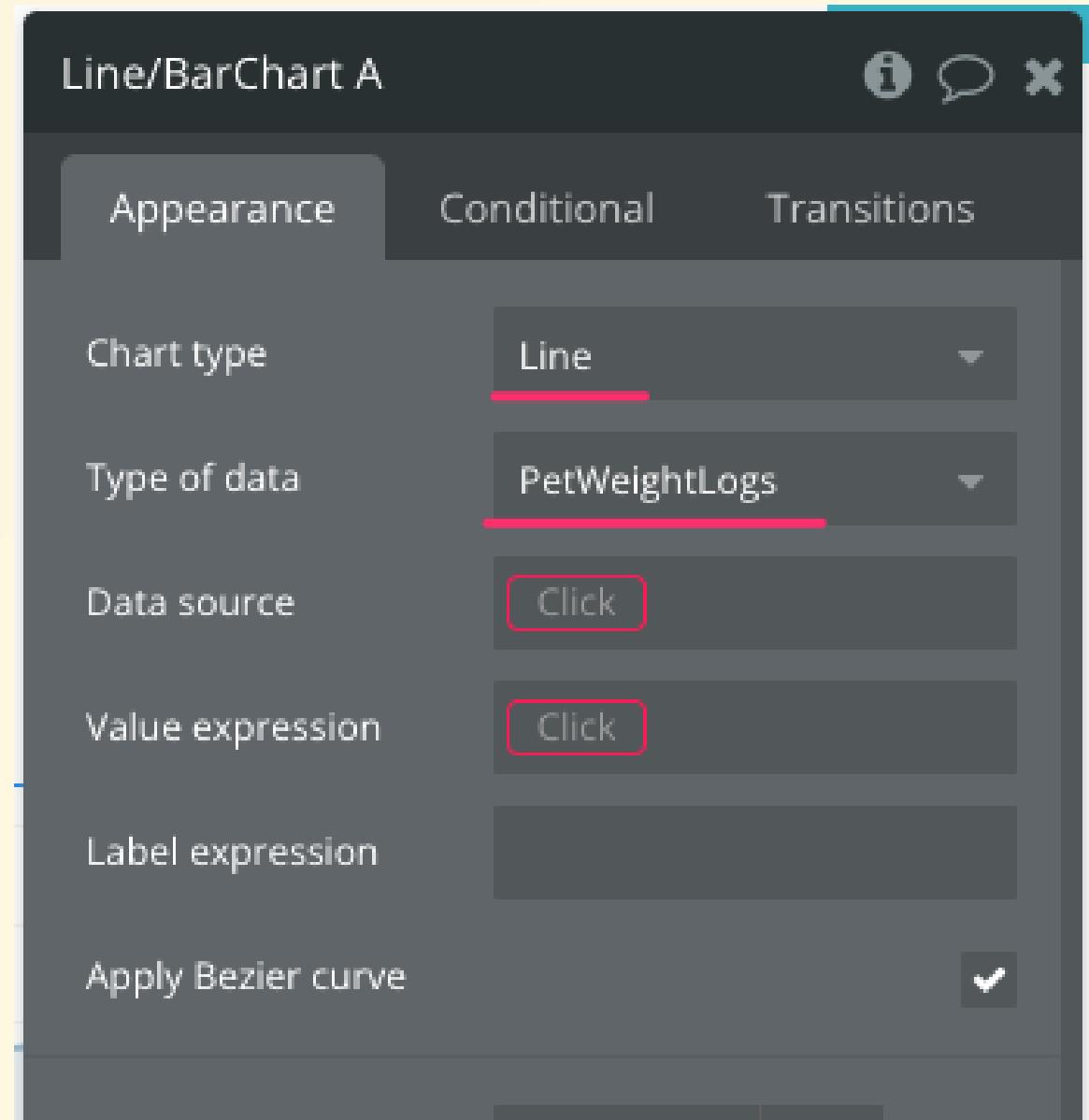
- 描画するとこれまでと同じく詳細設定用のダイアログが表示されますので、各項目を設定していきましょう
- Chart type はグラフの種類を指定します
 - 今回は折れ線グラフなので Line とします



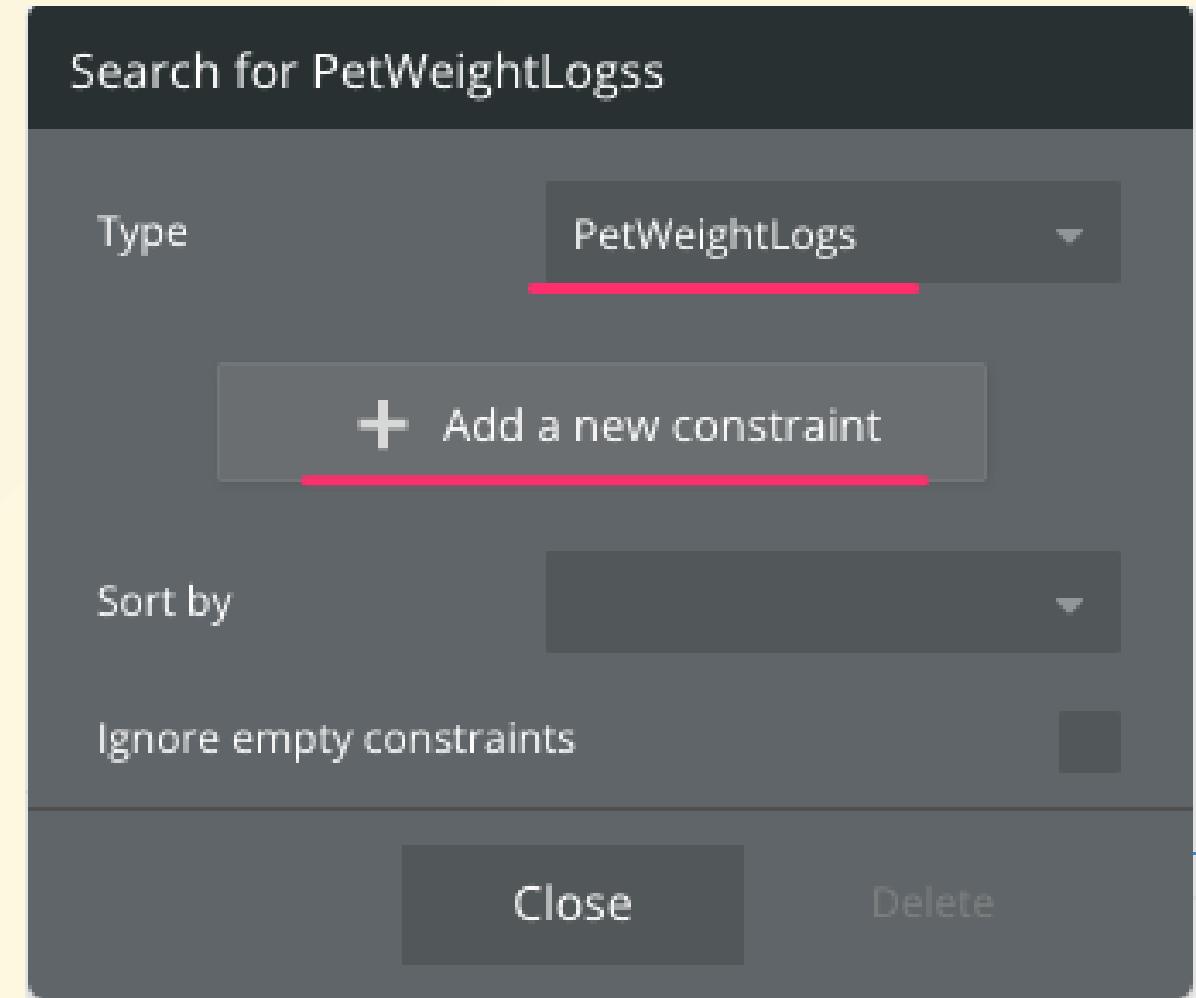
- Type of data はグラフ描画するデータのタイプを指定します
 - 今回はペットの体重表示となるので PetWeightLogs とします



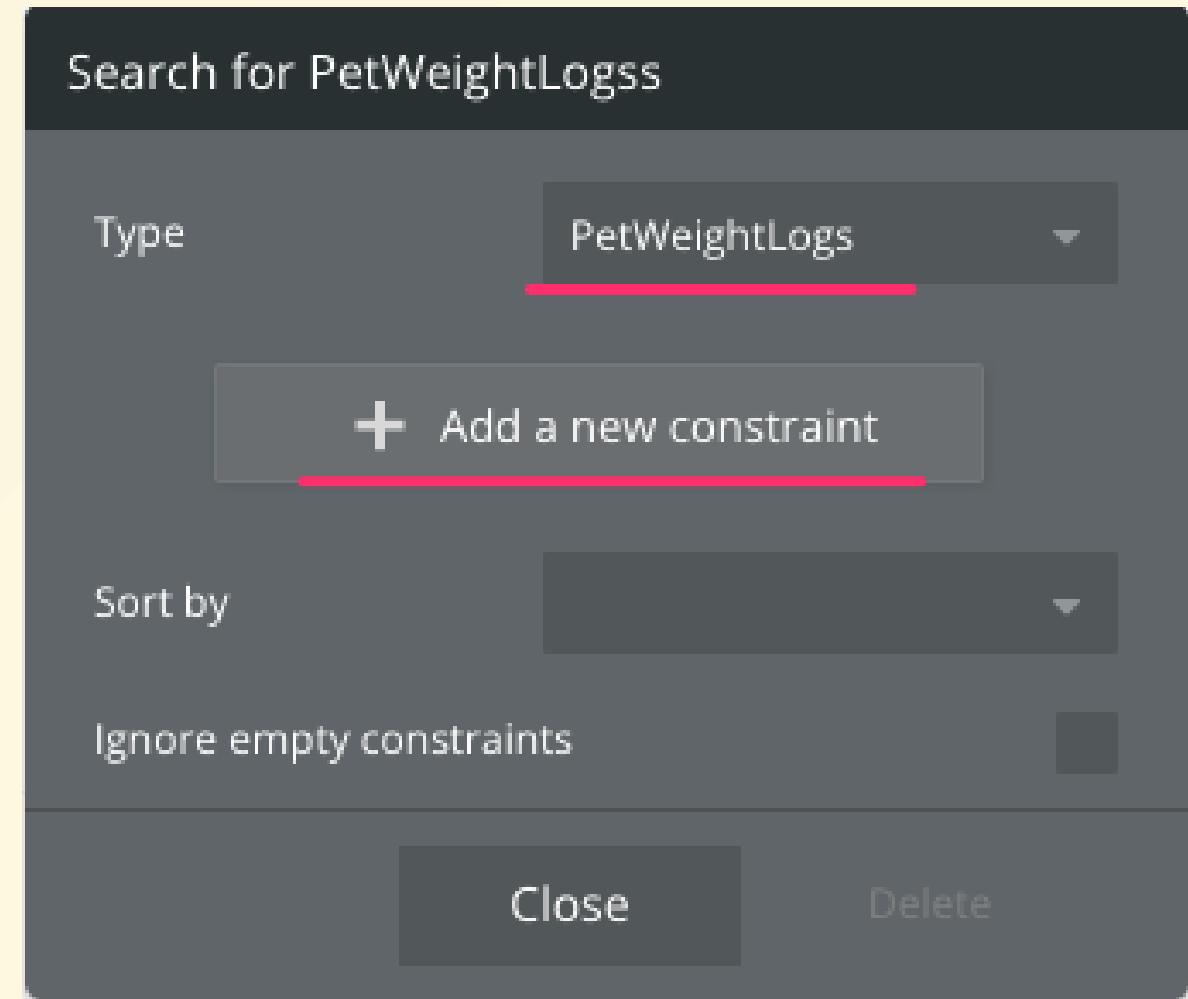
- 続けて **Data source** に対象データの条件を指定します
 - 一覧画面での条件指定と同じように、まずは **Do a search for** を選択



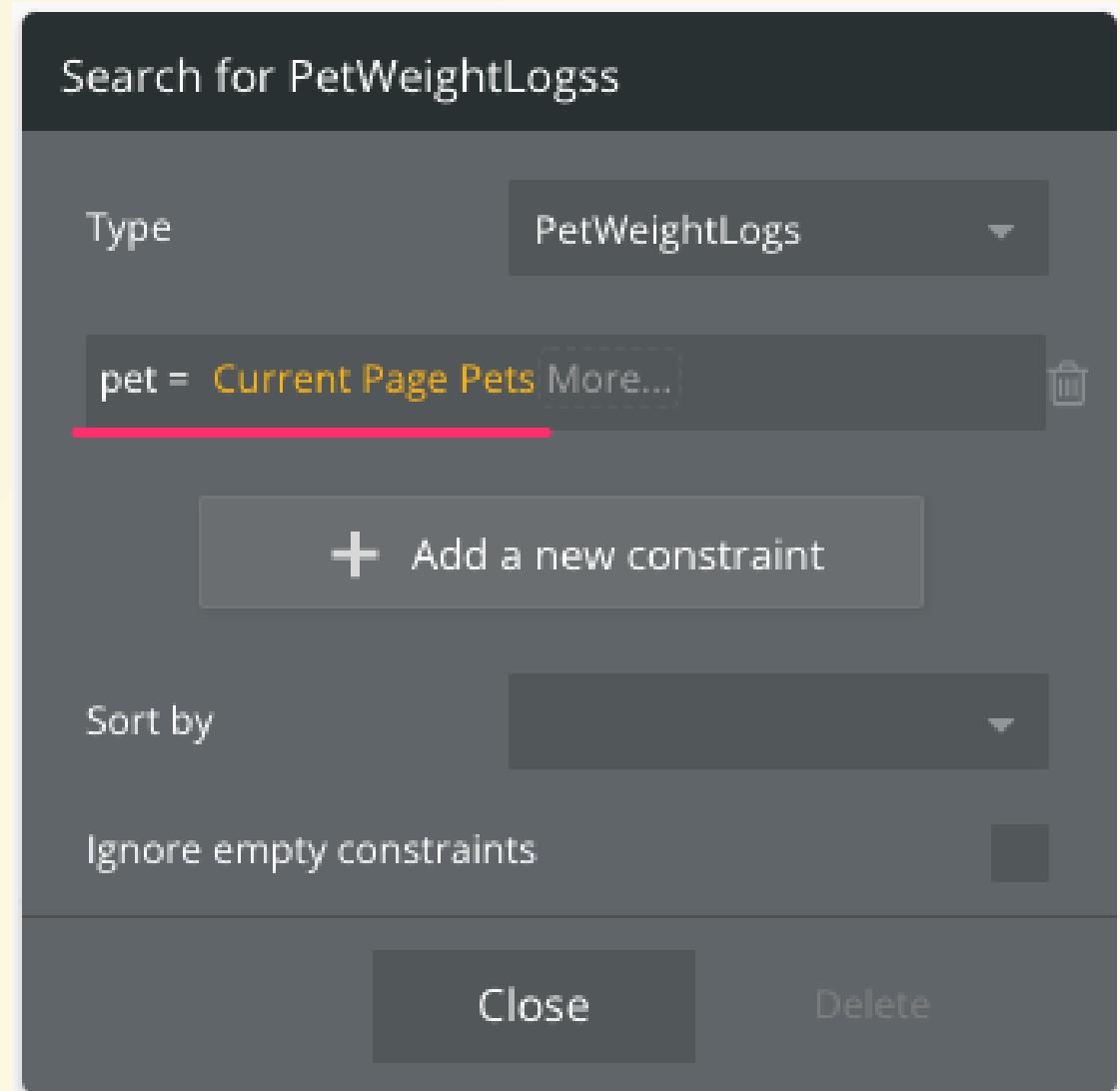
- Do a search for のダイアログがさらに表示されますので
 - Type には対象データの型を指定しますので PetWeightLogs を指定



- これだけだとすべてのペットのすべての体重データが表示されてしまうので絞り込みの条件を指定します
 - **Add a new constraint** をクリックして条件を指定します

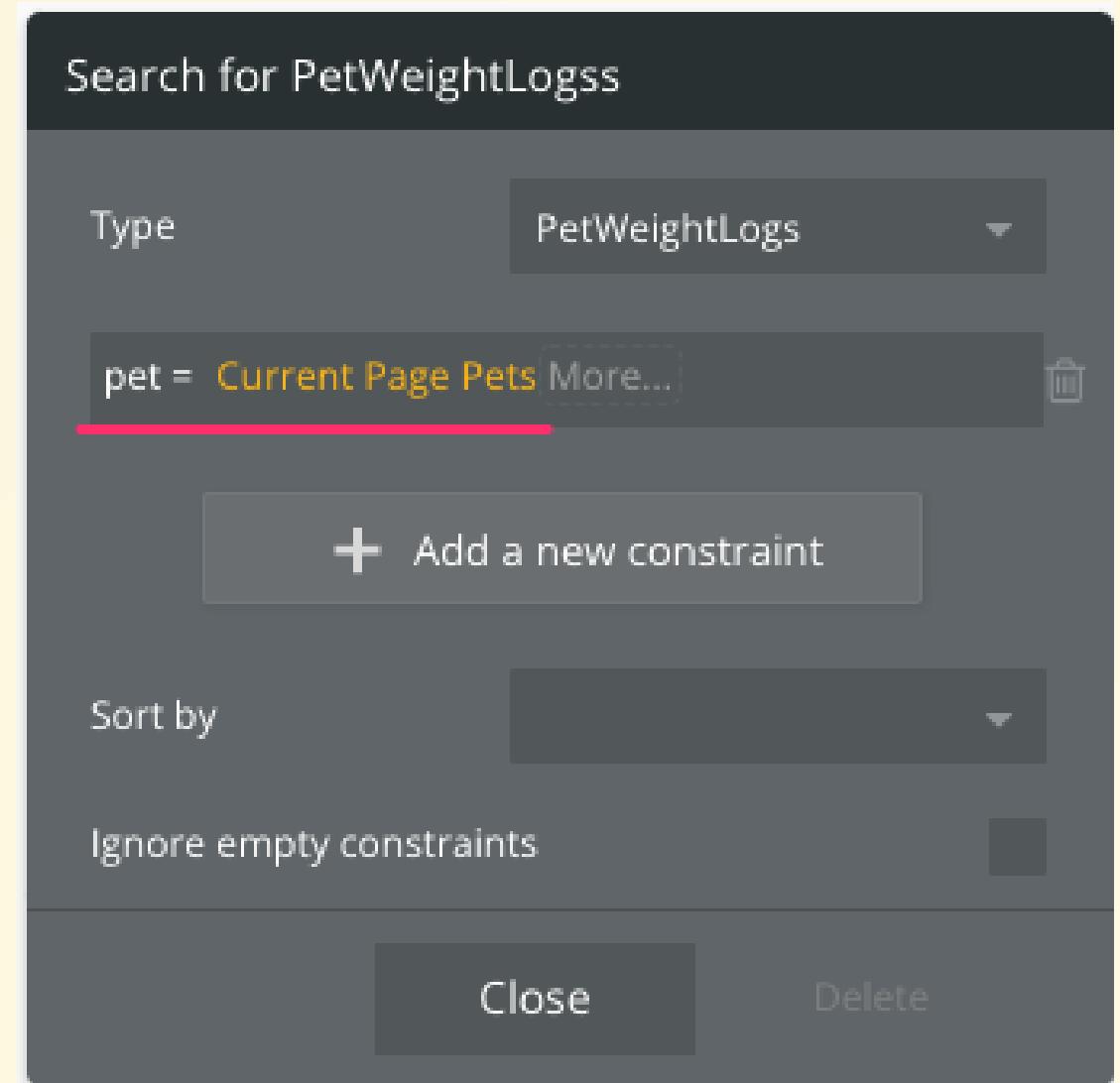


- まずは Click をクリックすると Type に指定した **PetWeightLogs** が持っているフィールドが表示されます
- 今回は **PetWeightLogs** に関する **Pet** の情報が、現在表示しているペットと同じデータを対象とします



- これを条件として表現する
とこのようになりますの
で、それを設定してみまし
ょう

```
pet = Current Page Pets
```



- 最後にデータの並び順を指定します
 - グラフの見せ方として、登録した体重を新しいものの順にしたいと思います



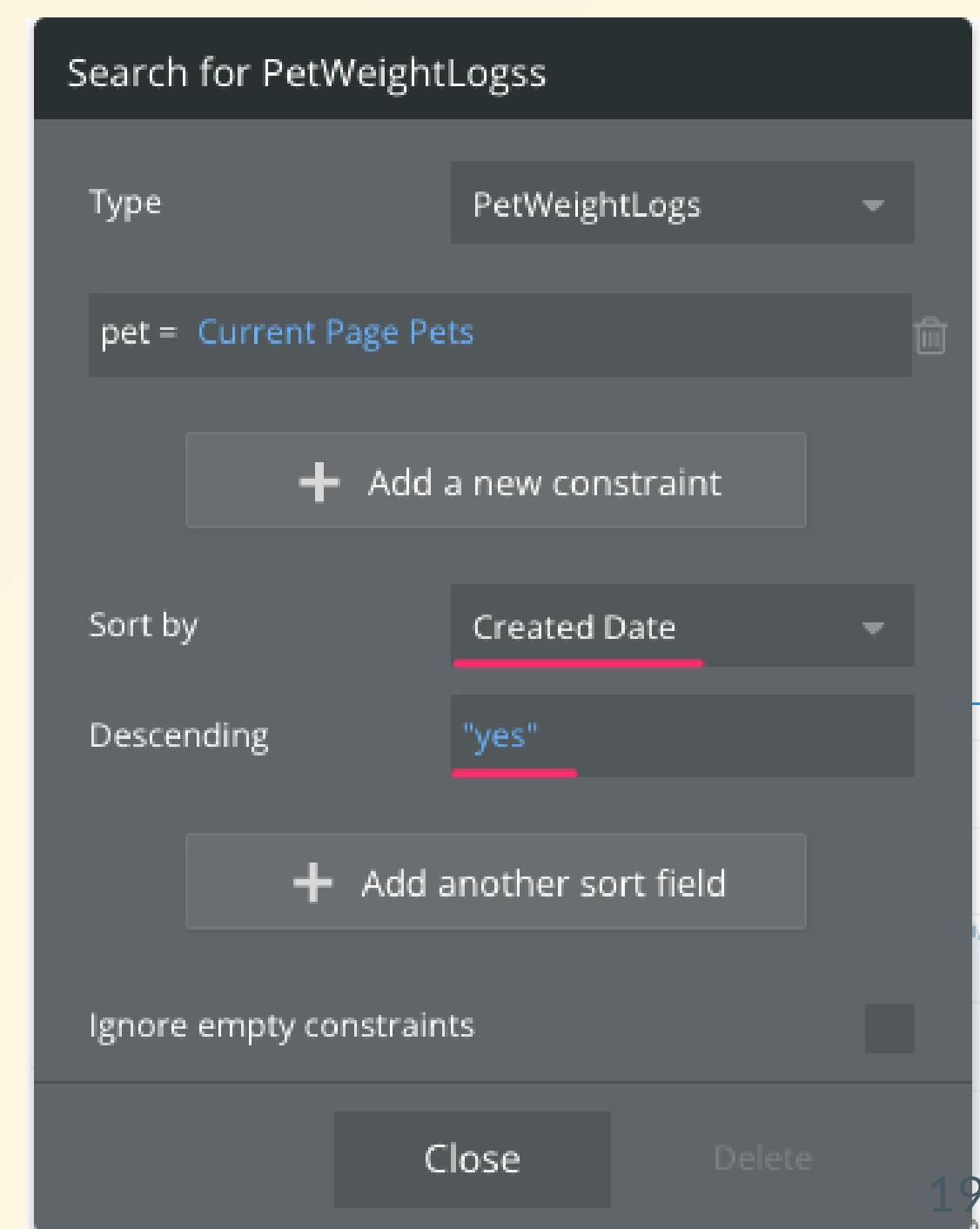
Weight

pet weight

← Back to detail

ADD

- Sort by に Created Date を指定します
 - これで並び順のキーを指定しました
- Descending に "yes" を指定します
 - これでキーの新しいものの順（降順）と指定しました



- 仮に登録日の古いもの順としたい場合、`Descending` に "no" を指定すれば古いもの順（昇順）となります
- 設定完了したら `Close` をクリックしてこのサブダイアログを閉じておきます

- 最後にグラフに描画する X 軸と Y 軸を指定します
- 今回の場合、X 軸は登録日時、Y 軸は体重となります



Weight

pet weight

← Back to detail

ADD

- X 軸の設定は

Label expression

- 今回は

Current point's
Creation Date

とします

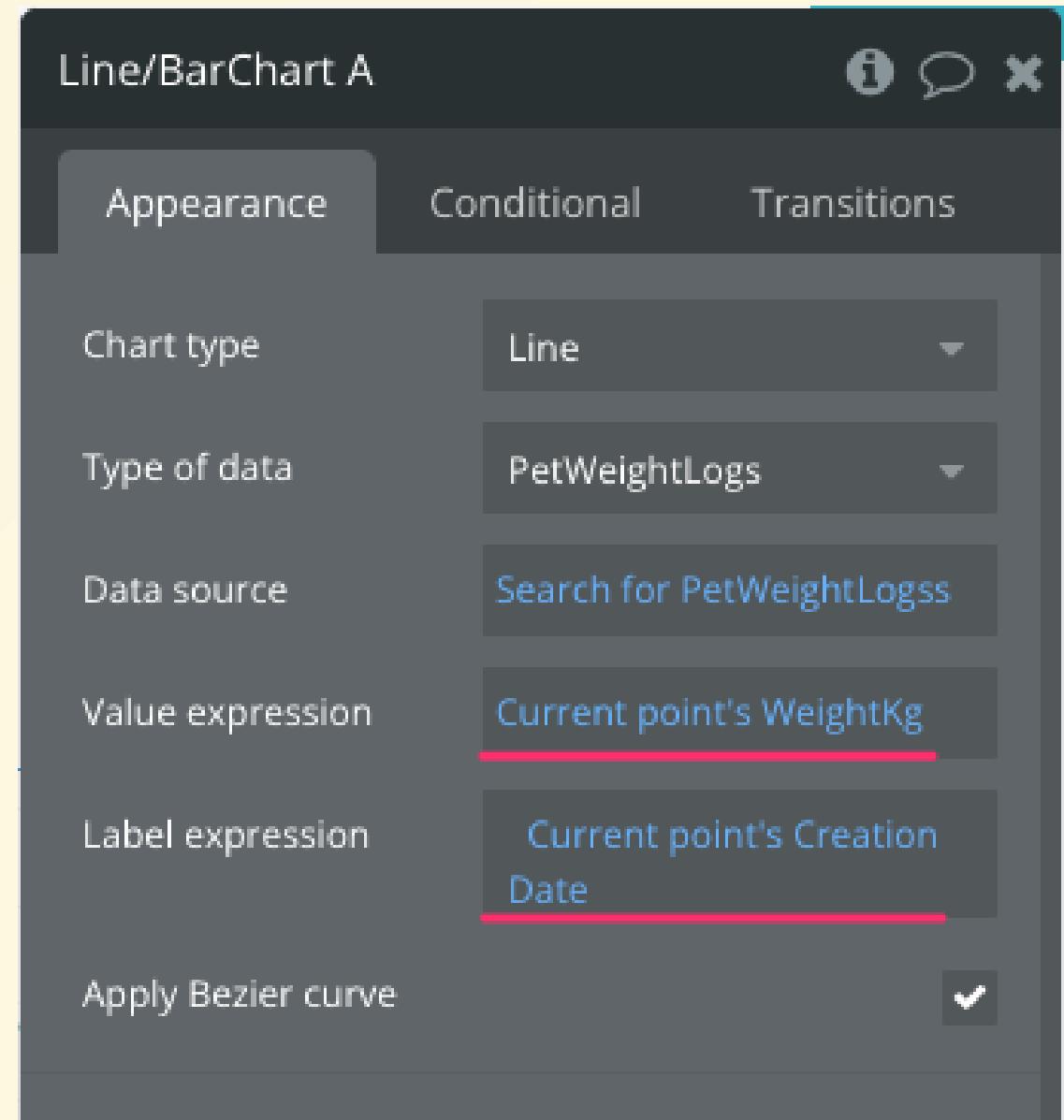
- Y 軸の設定は

Value expression

- 今回は

Current point's
WeightKg

とします



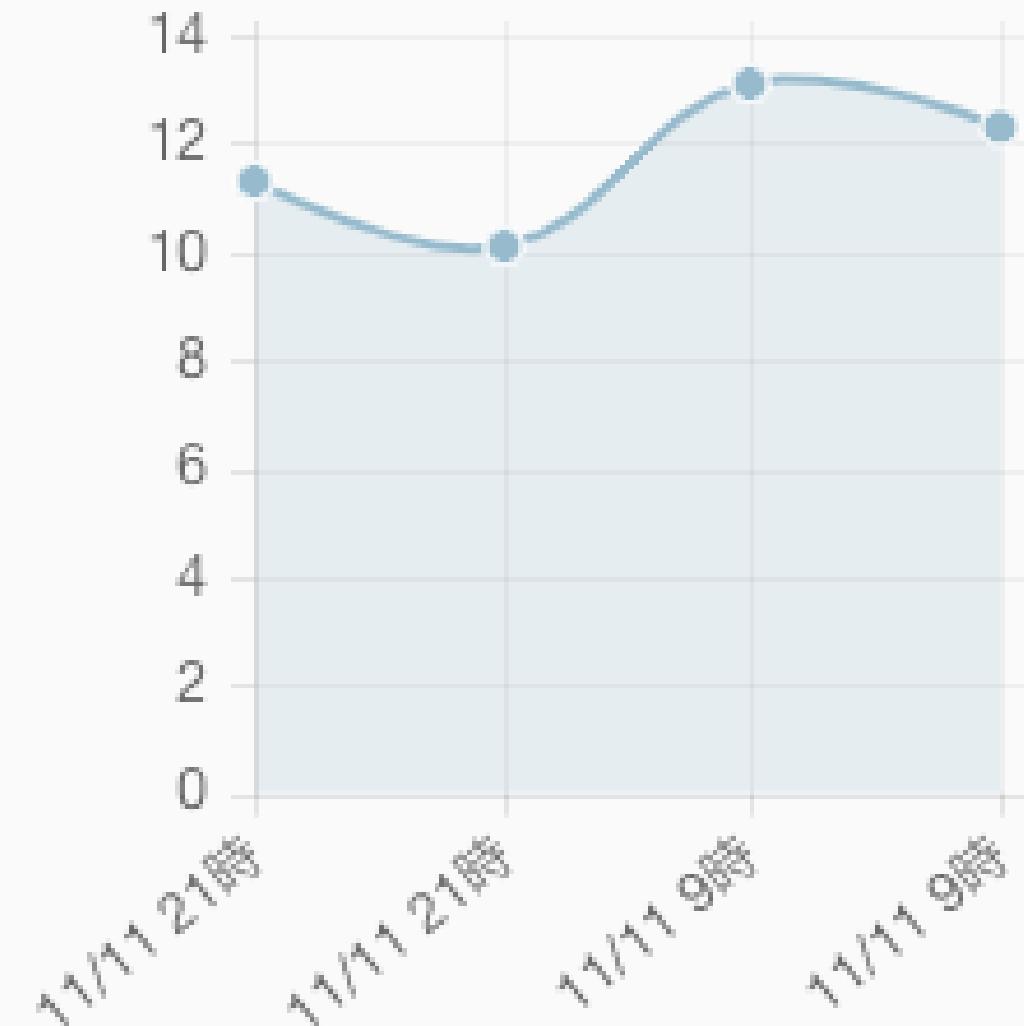
- `Current point` は対象データのタイプで指定した `PetWeightLogs` の各データをプロットした 1 件 1 件のデータを指します

- それでは一覧画面に戻ってプレビューしてみましょう！
- 一覧で選択したペットに対して登録された体重がグラフとして表示されましたか？
- 同じ画面で Weight をどんどん追加するとグラフの内容が変化しますか？



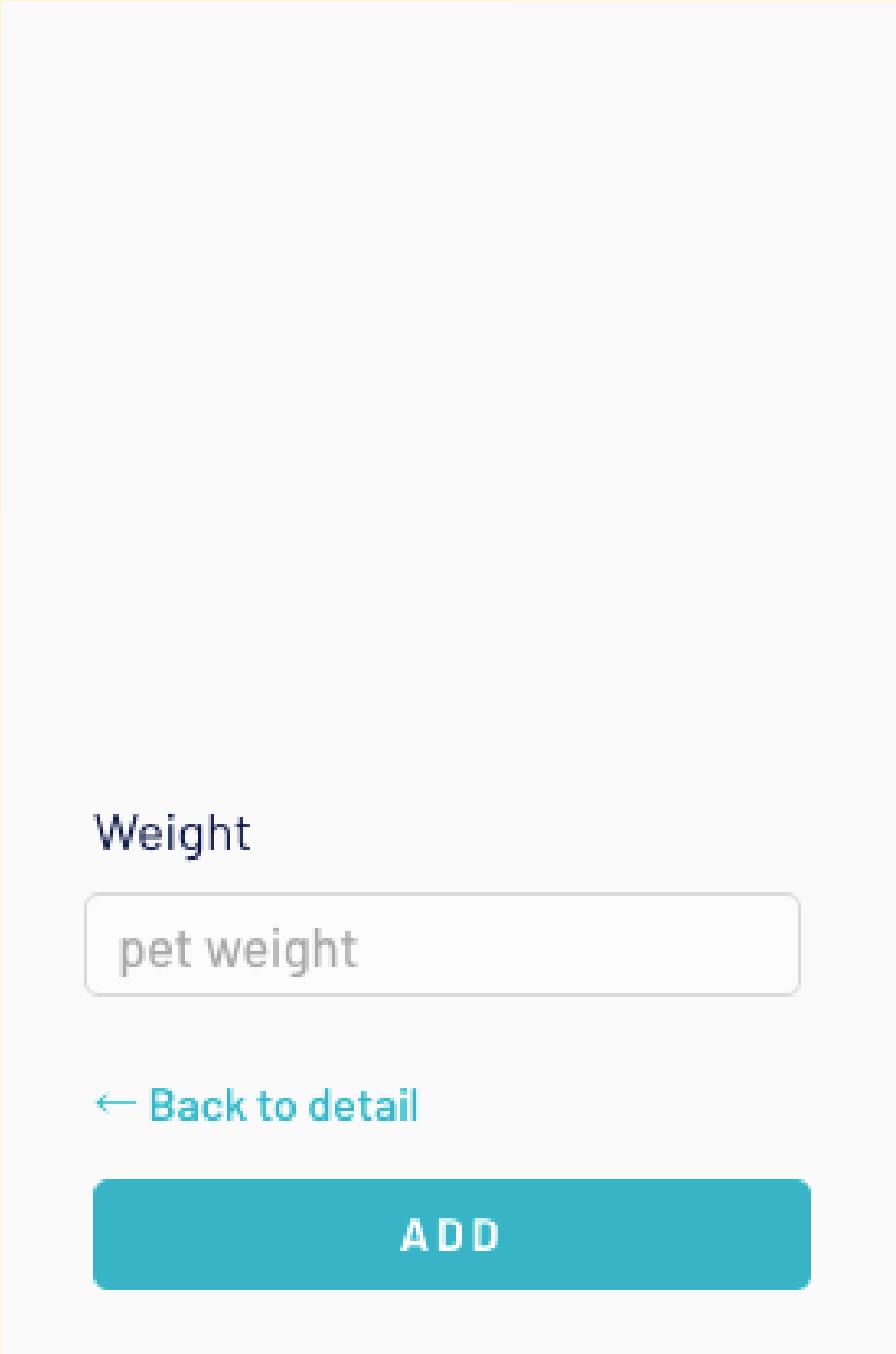
Appendix4: グラフのX軸のラベルを変えてみよう

- 現時点のX軸のラベルは長ったらしい日時の書式になっているので、皆さんでお好きなフォーマットに変えてみてください
- 例えばこんな感じ

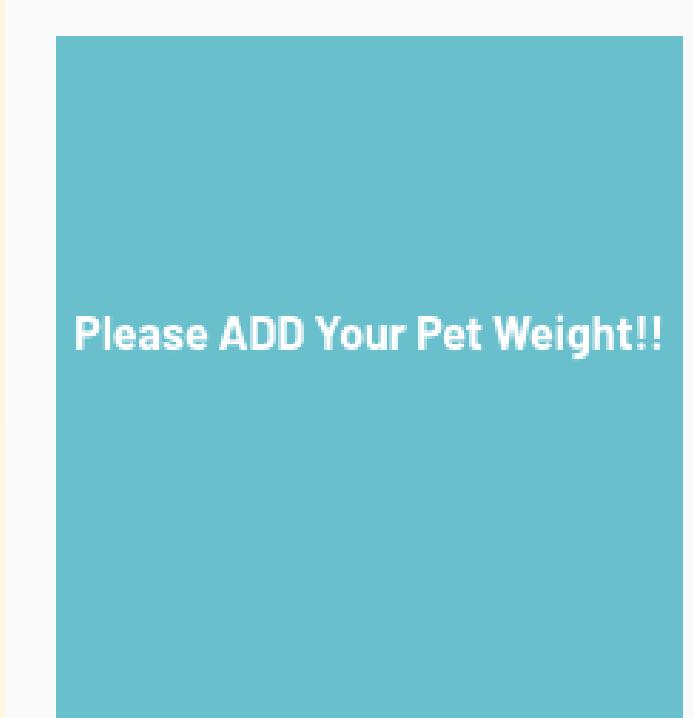


Appendix5: 体重データ が1件もない場合にメ ッセージを出してみよ う

- 現在の開発内容だと、ペットの体重が1件も登録されていない場合、グラフ描画エリアが空白になってしまい、ページとしての見栄えが悪いですね



- そこで、ペットの体重が1件も登録されていない場合に専用のメッセージを表示して、ユーザビリティを向上させてみましょう



Weight

[← Back to detail](#)

[ADD](#)

- 対応イメージとしては、グラフ要素と同じ幅と高さの Shape を同じ位置に用意
 - その上に表示したいメッセージを Text として配置し、Shape と Text をグルーピング
 - グルーピングすることで 1 つの要素として扱うことができます
- その Shape に対して下記を設定します
 - 要素自身は常に非表示とする
 - Conditional で PetWeightLogs の件数が 0 件だった場合に、要素を表示する
- こうすることで、体重が 1 件も登録されていない場合は Shape を表示し、そうでない場合はグラフを表示することができます

- 設定イメージはこんな感じです

The image displays two side-by-side configuration interfaces.

Left Screenshot: Search for PetWeightLogss

- Type: PetWeightLogs
- pet = Current Page Pets
- + Add a new constraint
- Sort by
- Ignore empty constraints
- Close
- Delete

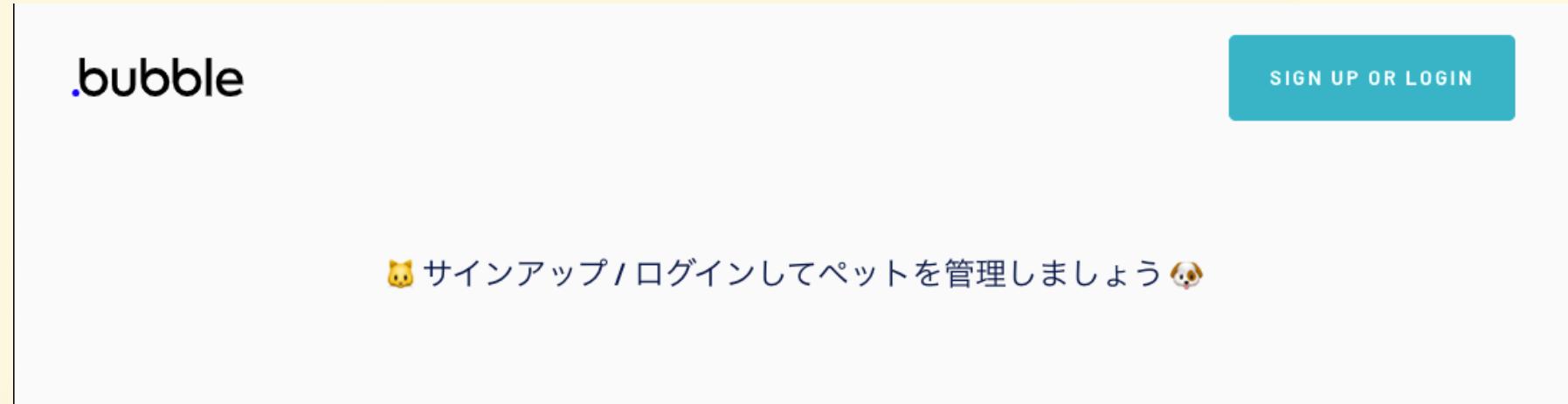
Right Screenshot: Group A

- Appearance
- Conditional (1)
- Transitions
- OFF remove condition
- When Search for PetWeightLogss:count is 0 More...
- This element is visible This element is visible
- Select a property to change when true
- + Define another condition

Appendix6: ペット一覧にメッセージを出し分けてみよう

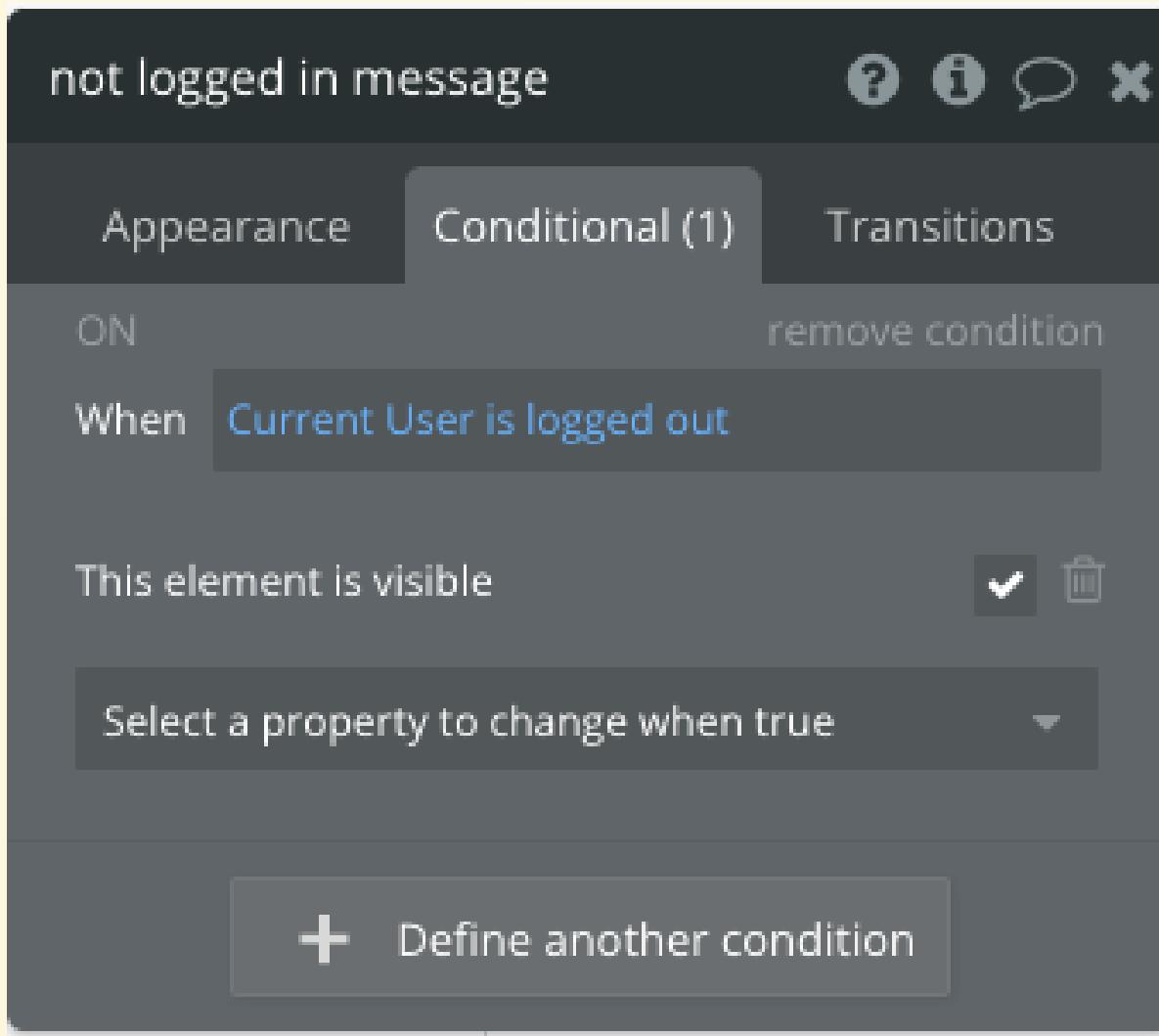
- ・ ペット一覧はログインしていないと表示されません
- ・ さらに、ログインしていてもペットが1件も登録されていないと何も表示されません
- ・ しかし、ユーザからすると、なぜペット一覧が表示されていないのかが判断つきません

- そこで、それぞれの状態に応じてメッセージを出し分けて見ましょう

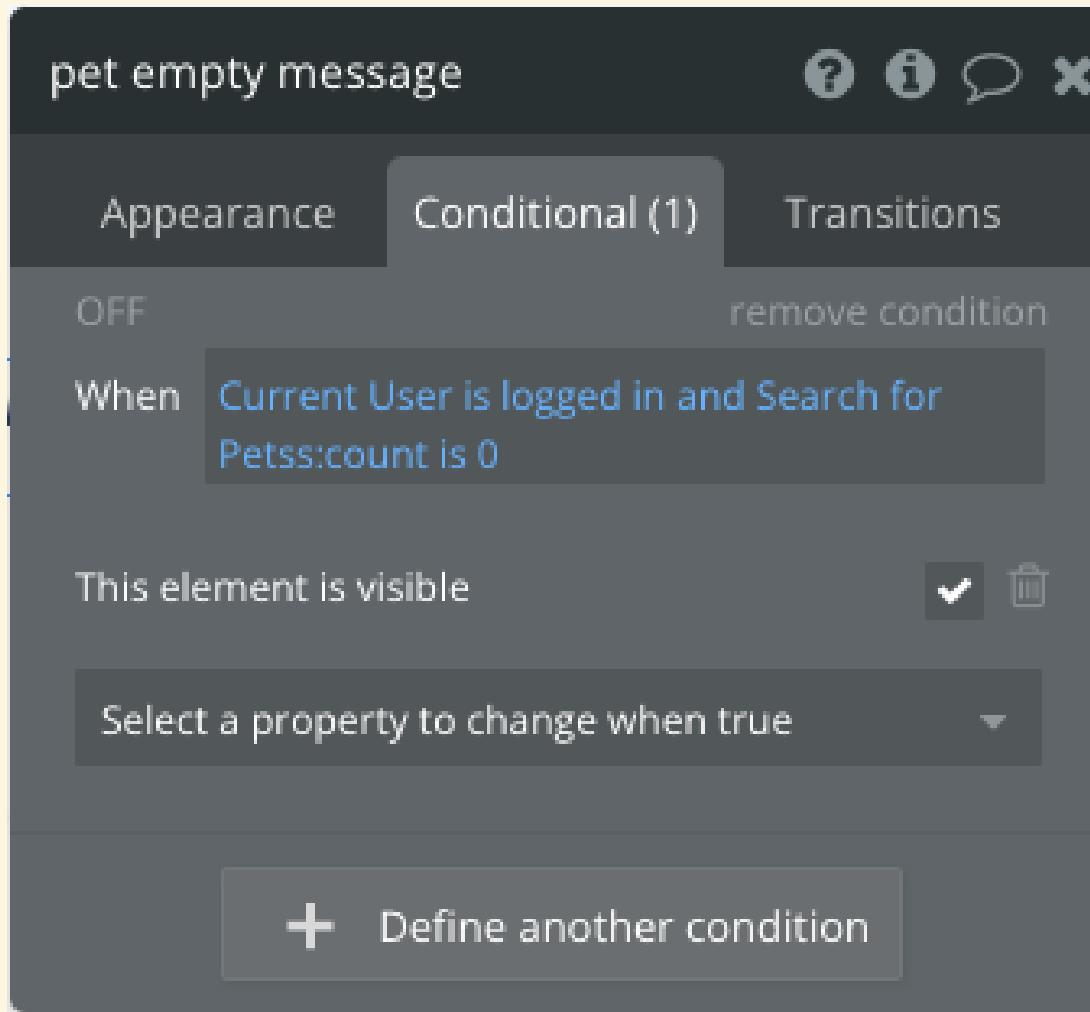


- 設定の考え方は先ほどのペットの体重が0件の時の出し分けと同じです
- それぞれのメッセージを用意しつつ、デフォルトでは非表示設定にしておきます
- その上で、それぞれの条件に合致する場合に、メッセージを表示させます
 - 現在のユーザがログイン済みかどうか？
 - 現在のユーザがログイン済みで、かつそのユーザが登録したペットが1件以上存在するか？
- 次のページに答え（設定内容）を書いています！

- 現在のユーザがログイン済みかどうか？



- 現在のユーザがログイン済みで、かつそのユーザが登録したペットが1件以上存在するか？



Appendix7: 詳細画面から更新画面への導線を設け、更新機能を作ってみよう

- ポイントとしては次のシートの通りです！ぜひ実践してみてください！

Name

Image



Birthday

Gender

[← Back to detail](#)

UPDATE

- 更新画面は登録画面とほぼ同じ画面要素を持つので `pet_register` を Clone して `pet_update` を作ってみましょう
- 登録画面では、各入力要素（名前や誕生日など）の中身は空っぽでしたが、更新画面では既に登録済みのデータの値を画面表示時点でセットしてあげる必要があります
- 各入力要素のダイアログからそれっぽい項目がありますので、そこに初期値となる値を指定します
 - Hint  Insert dynamic data
 - Hint  Initial content

- 更新ボタンを押したときの振る舞いについては、新規登録時には Data(Thing) の Create a new thing... でしたが、今回は更新ですね
 - 更新といえば change... お、それっぽいアクションがありそうですね 😊
- 最後に詳細画面から更新画面へ遷移するとき、詳細画面で表示しているペットの情報を渡してあげる必要があります。これは今日の講義で学んだ部分です！
 - Hint 💡 Data to send

まとめ

- ・ 今回のレクチャーでは Bubble の使い方から始まり、実際に画面をデザインするところから、データベースとの連携まで行いました
- ・ 基本的な流れは Adalo と似ている部分もありつつ、ワークフロー周りの考え方にはビジュアルプログラミングツールと呼ばれる所以を少しは感じていただけたかなと思います
- ・ 次回は Bubble のより深いところをレクチャーしていければと思いますので乞うご期待！

以上です！

お疲れさまでした！