

Programming Boot Camp

Adaloでのアプリ開発と外部連携

東京工業大学 2023/11/3

Ryo Imahashi

目次

- Adaloとは
- Adaloに登録しよう
- テンプレートアプリを操作してみよう
- Adaloでのアプリ開発の概要説明
- アプリを作ってみよう
- サンプルアプリを改善しよう
- 外部連携
- まとめ

Adaloとは

- Adaloは、アメリカ発のノーコードツールです。プログラミング不要でアプリを開発することができます。
- 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＆ドロップしていくことで、アプリを作ることが可能です。
- ブラウザで表示するWebアプリだけでなく、AndroidやiOS向けのスマートフォンアプリも開発できます。開発したアプリのGooglePlayやAppStoreでの公開も可能です。
- シンプルで分かりやすいので、ノーコード開発を初めて学ぶためのツールに適しています。

注意点

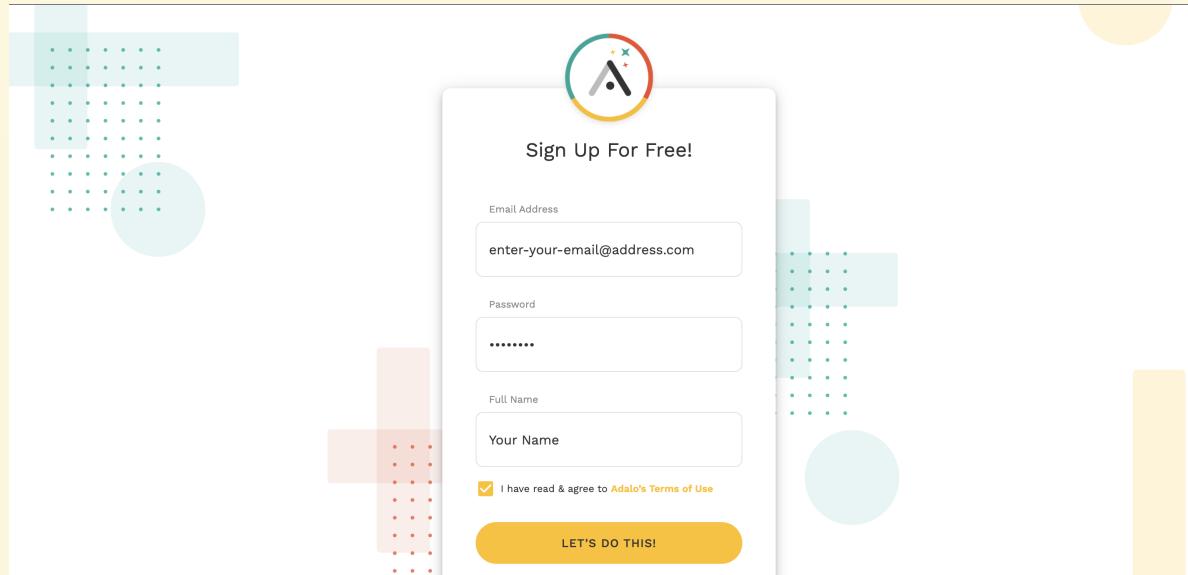
- 一つのアプリを複数人で同時編集するのが難しく、Development Phaseでのチーム開発に利用するのはあまりおすすめできません
 - ただし、画面の開発を担当するのが1人だけで、それ以外のメンバーは全員APIの開発を担当するという分担であれば、問題なく使えます

Adaloで作られたアプリの例

- Union: <https://union-jp.site/>
 - 大学生が開発した、大学生・大学院生・大学教員・大学職員限定のSNS
 - 2021年に1,000万円の資金調達を実施している
 - <https://prtimes.jp/main/html/rd/p/000000001.00007669.html>
- その他にも、 #MadeInAdalo でたくさん紹介されています
 - <https://www.adalo.com/made-in-adalo>

Adaloに登録しよう

- AdaloのSignUp画面にアクセスしてください
 - <https://app.adalo.com/signup>



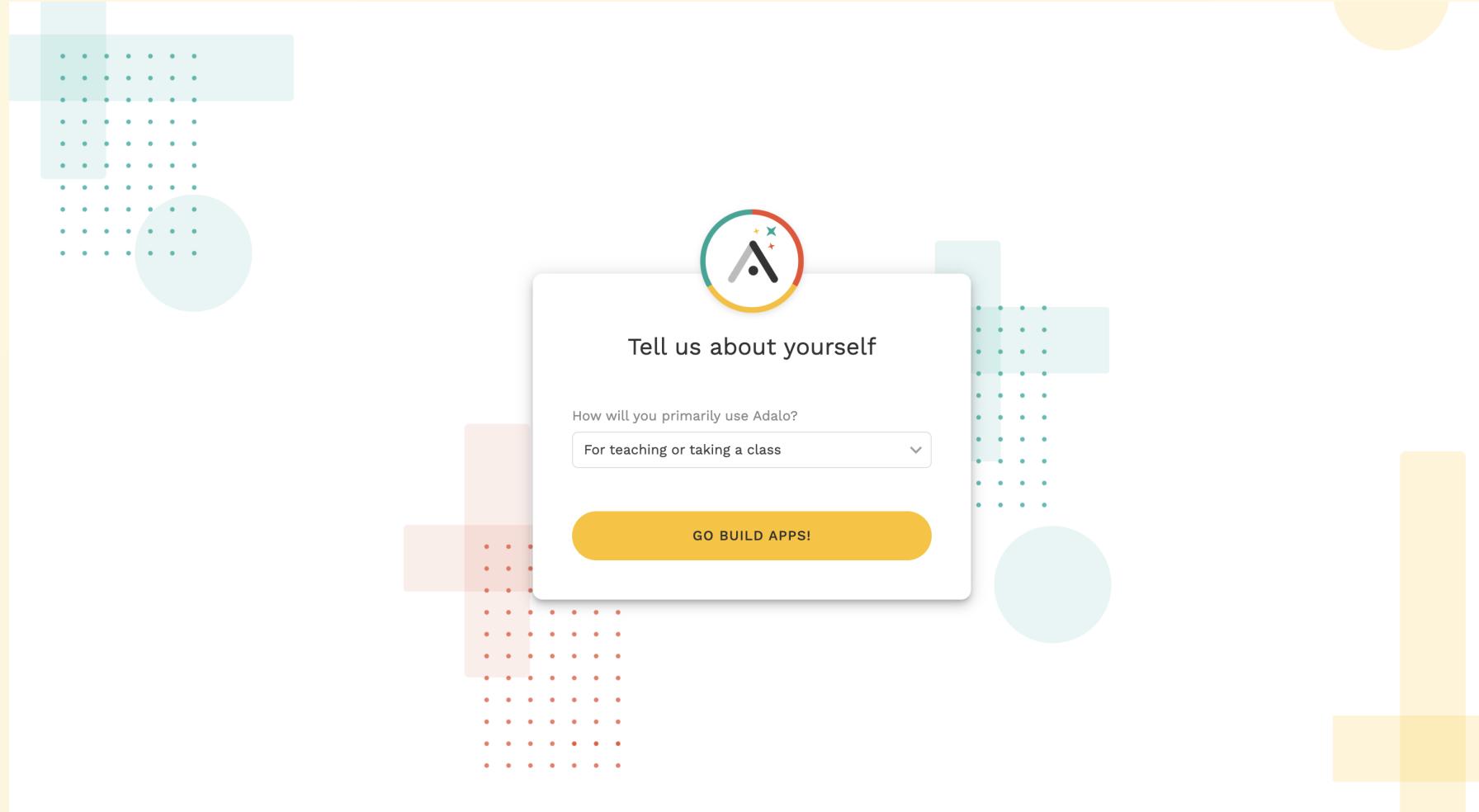
- 使用するブラウザは、Google Chromeを推奨します
 - インストール: https://www.google.com/intl/ja_jp/chrome/

- 無料で登録できます
- ご自身のEメールアドレス、パスワード、フルネームを入力してください
- 利用規約への同意のチェックを入れてください
- LET'S DO THIS! ボタンを押して、次の画面に進んでください

参考: 無料プランの制限

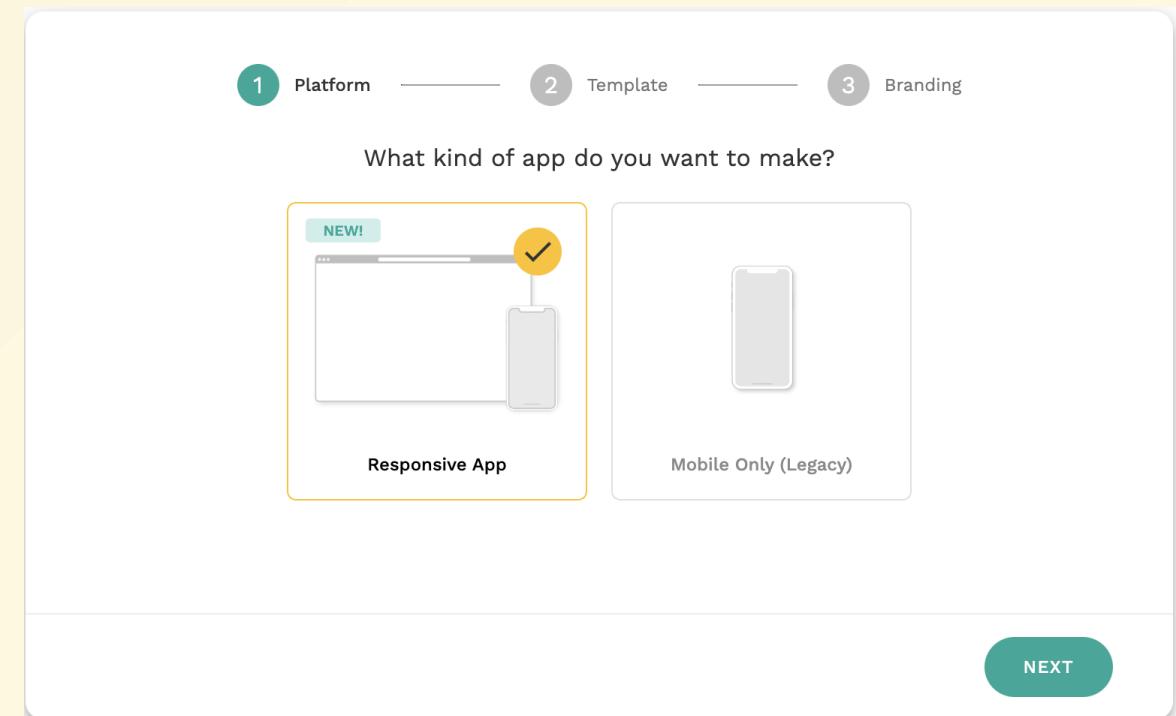
- 外部アプリケーションとの連携ができない
 - ただし、14日間はトライアル利用が可能
 - 1アプリで登録できるデータベースのレコード数の上限が200件
 - 月間アクション(ボタンクリック時等に行われる処理)回数の上限が1000回
-  実際にサービスを運用していく際は、有料プランへの移行を検討してください

- 利用目的を聞かれるので、For teaching or taking a class を選択して、Go Build Apps! ボタンを押してください



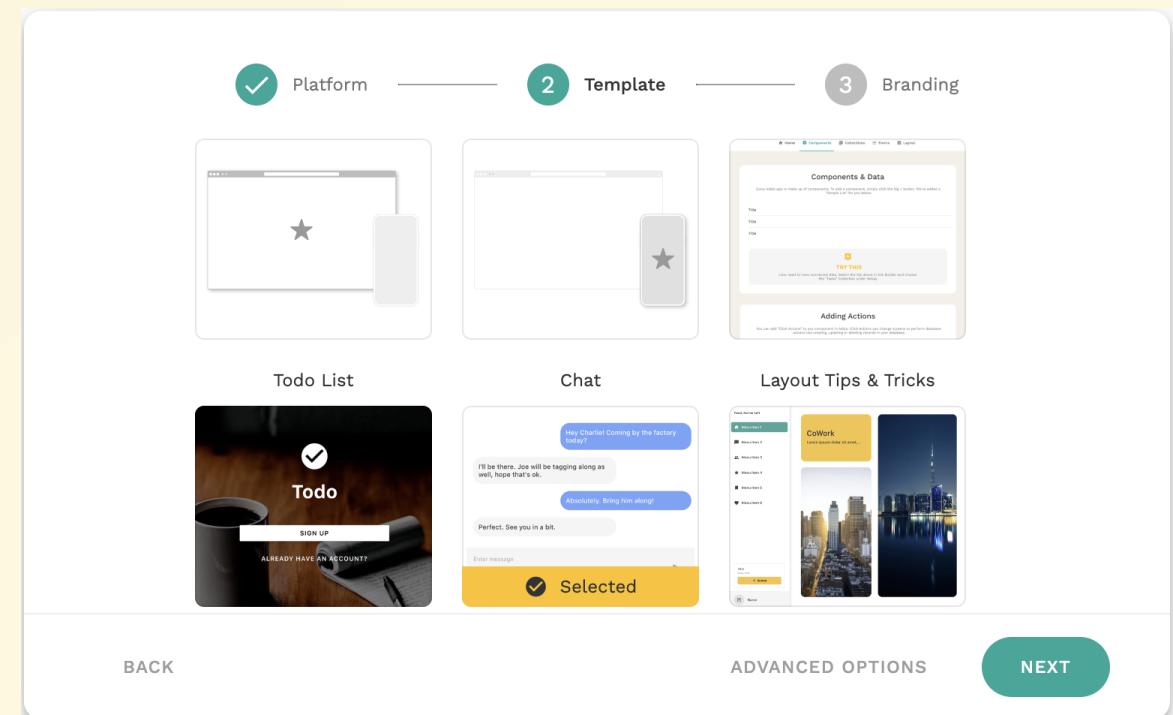
プラットフォームの選択

- Responsive AppとMobile Onlyが選べます
- PCとMobileの両方に対応できる、Responsive Appを選択しましょう



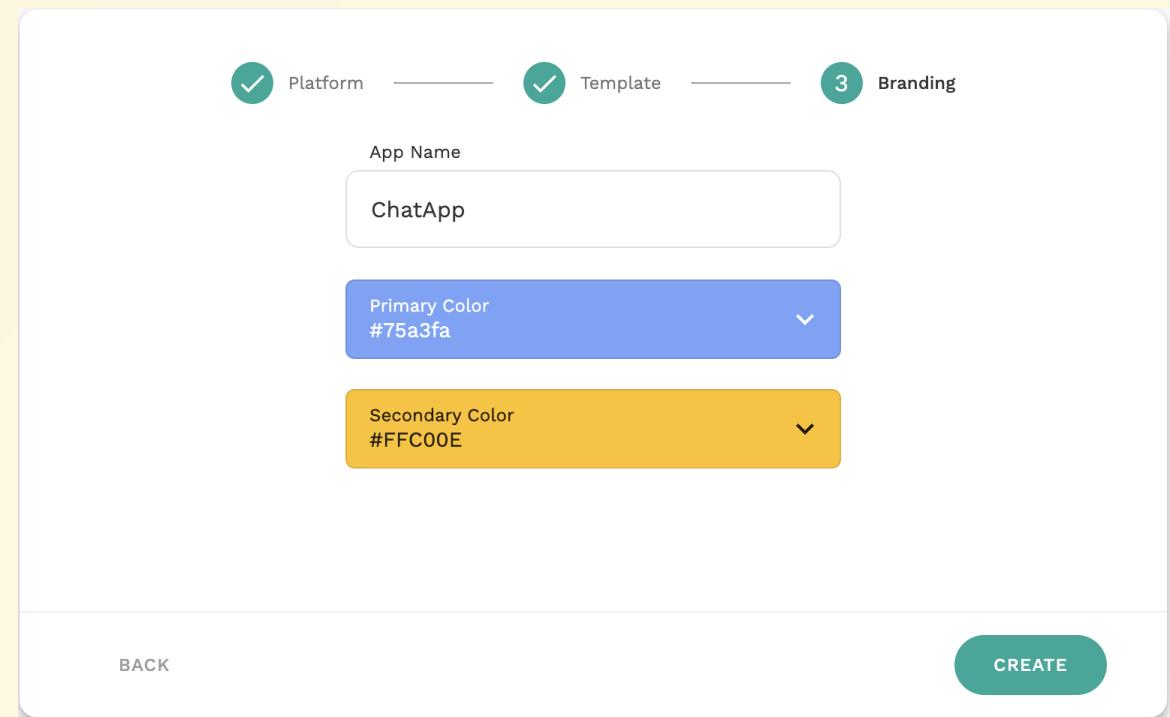
テンプレートの選択

- 完成されたアプリがテンプレートとして提供されています
- 今回は、Chatを選択しよう

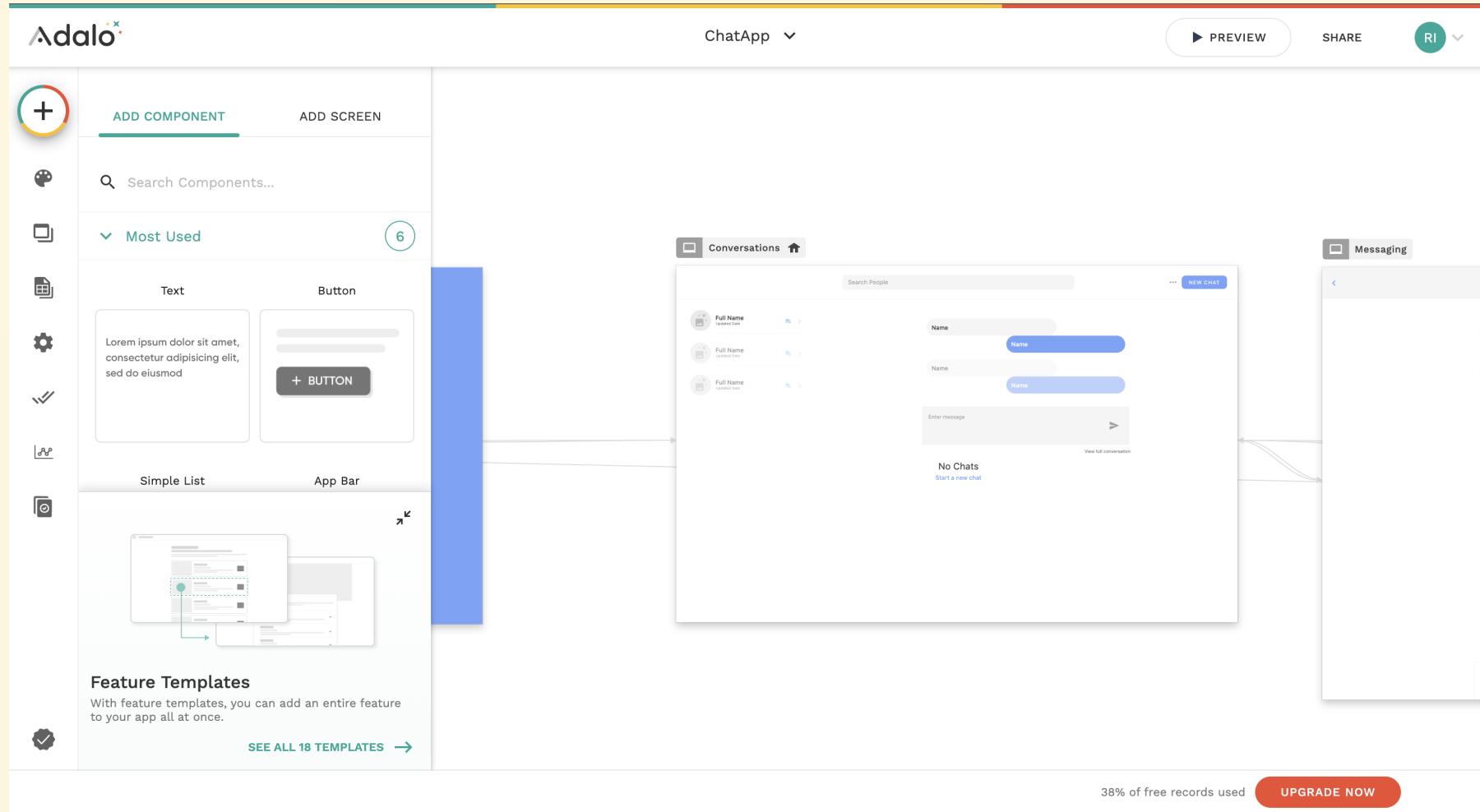


ブランディング

- App Name、Primary Color、Secondary Colorを自由に入力してください
 - Primary Colorは、アプリで最も使われるベースになる色です
 - Secondary Colorは、目立たせたい時に使う色です。重要なボタンなどに使われます。

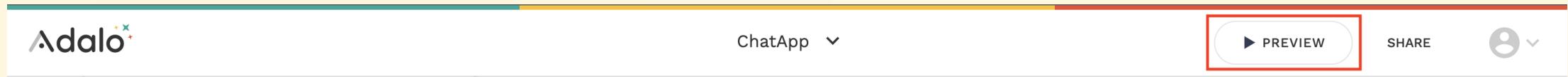


- このようなAdaloの管理画面が表示されたらOKです
- 今後はこの管理画面を使って、アプリを開発していきます

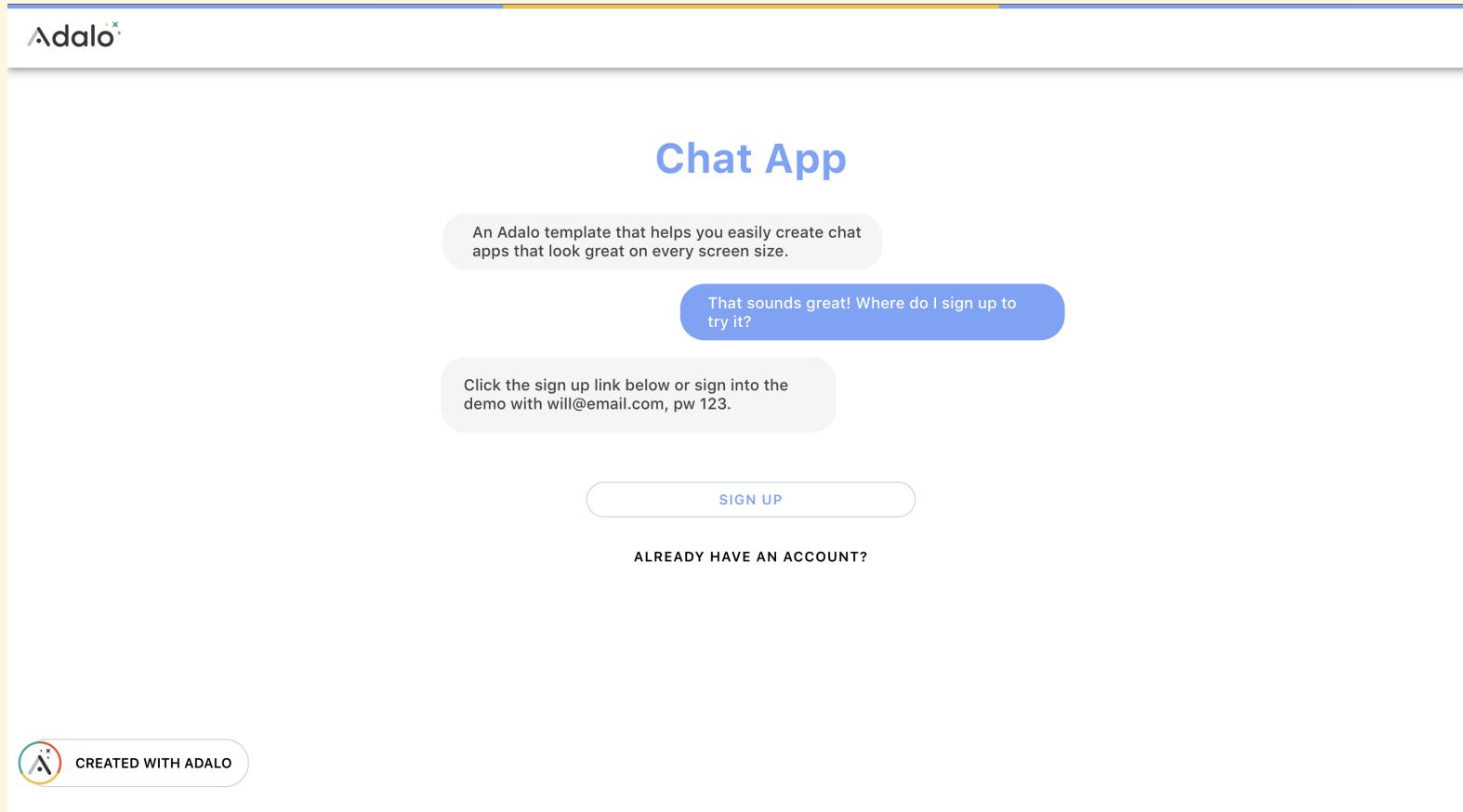


テンプレートアプリを操作してみよう

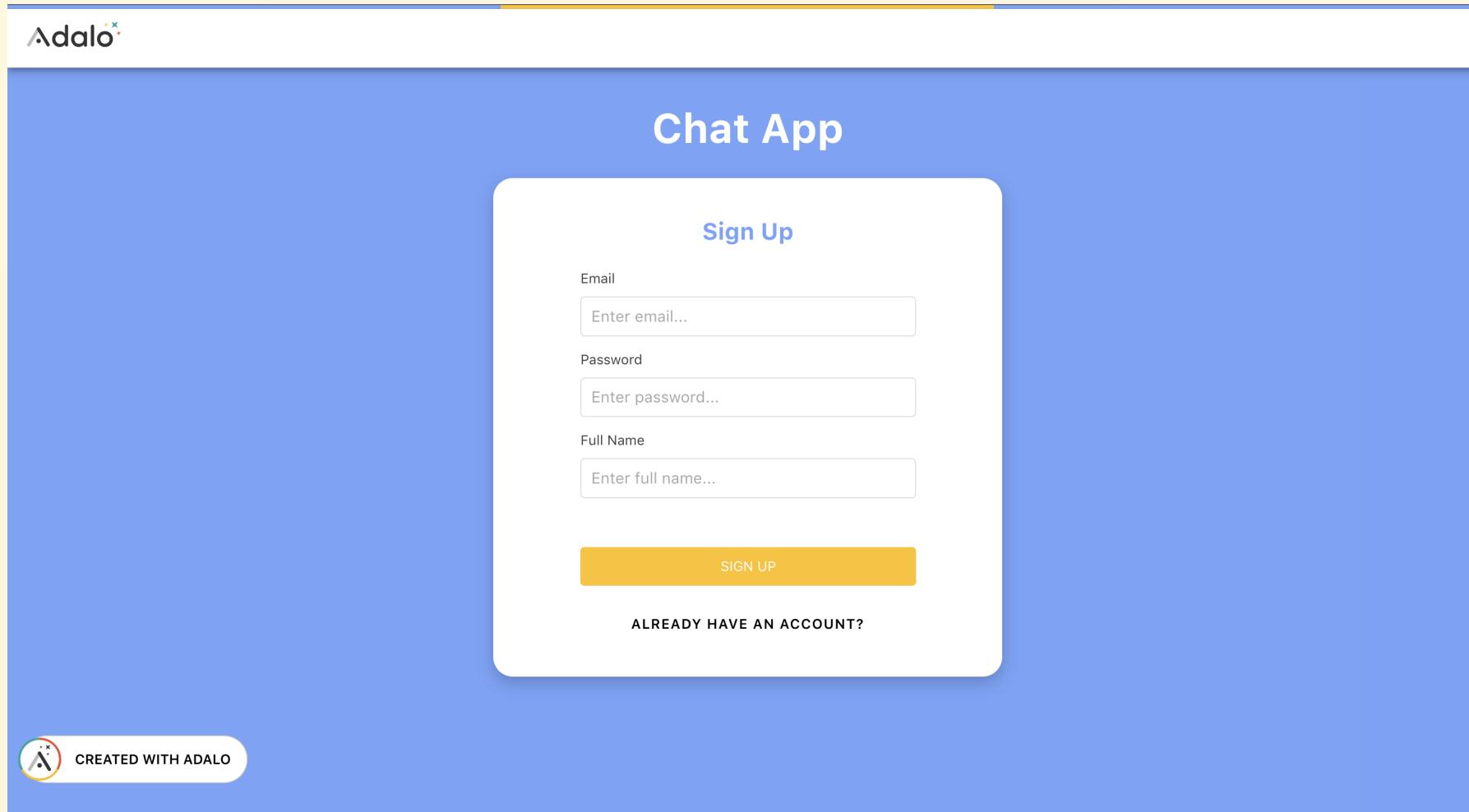
- まずはAdaloで作ったアプリがどのように動作するかを確認するために、先ほど選択したChatアプリを操作してみましょう
- 画面右上のPreviewボタンをクリックしてください



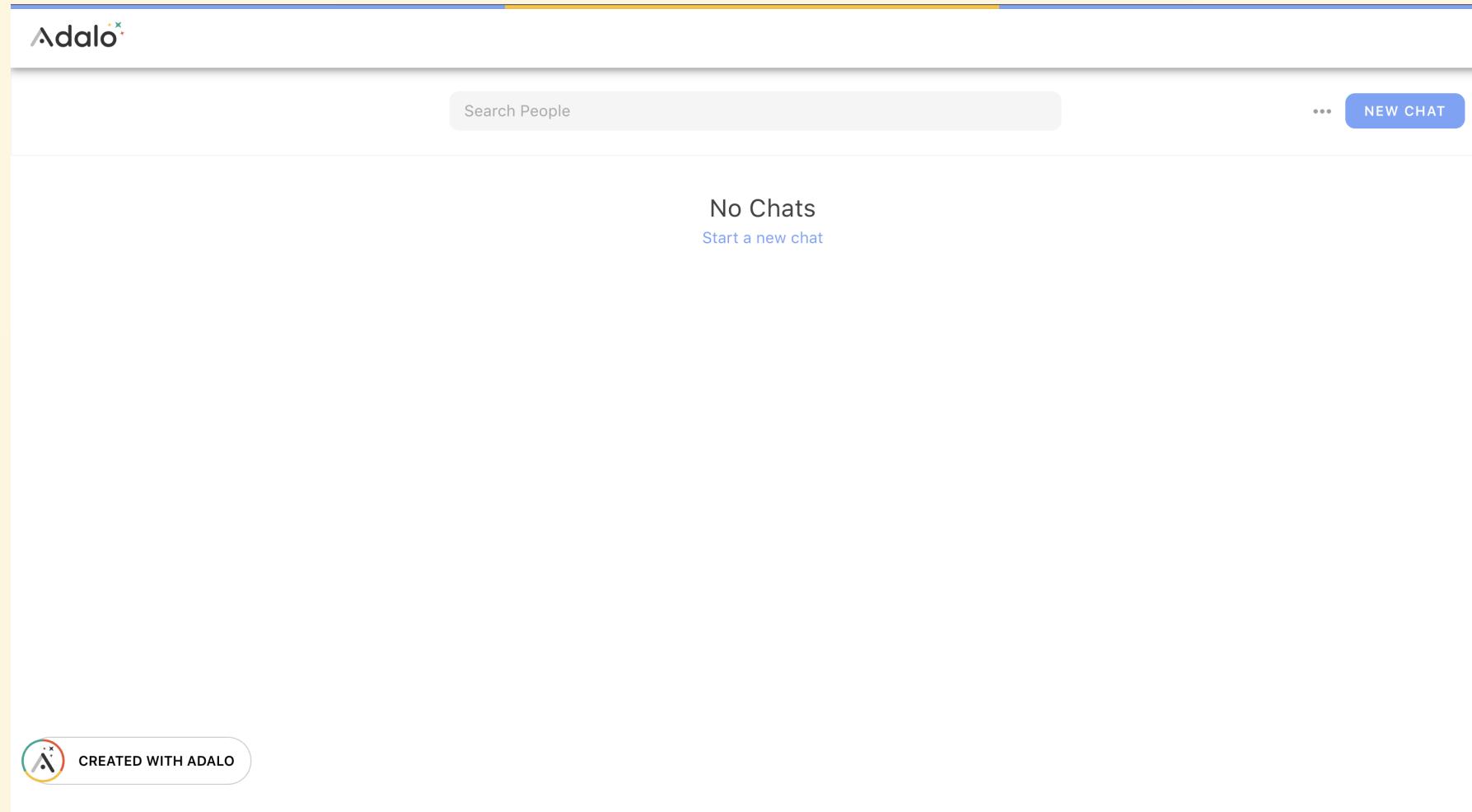
- プレビュー画面が起動します
- 一緒にChatアプリを操作してみましょう



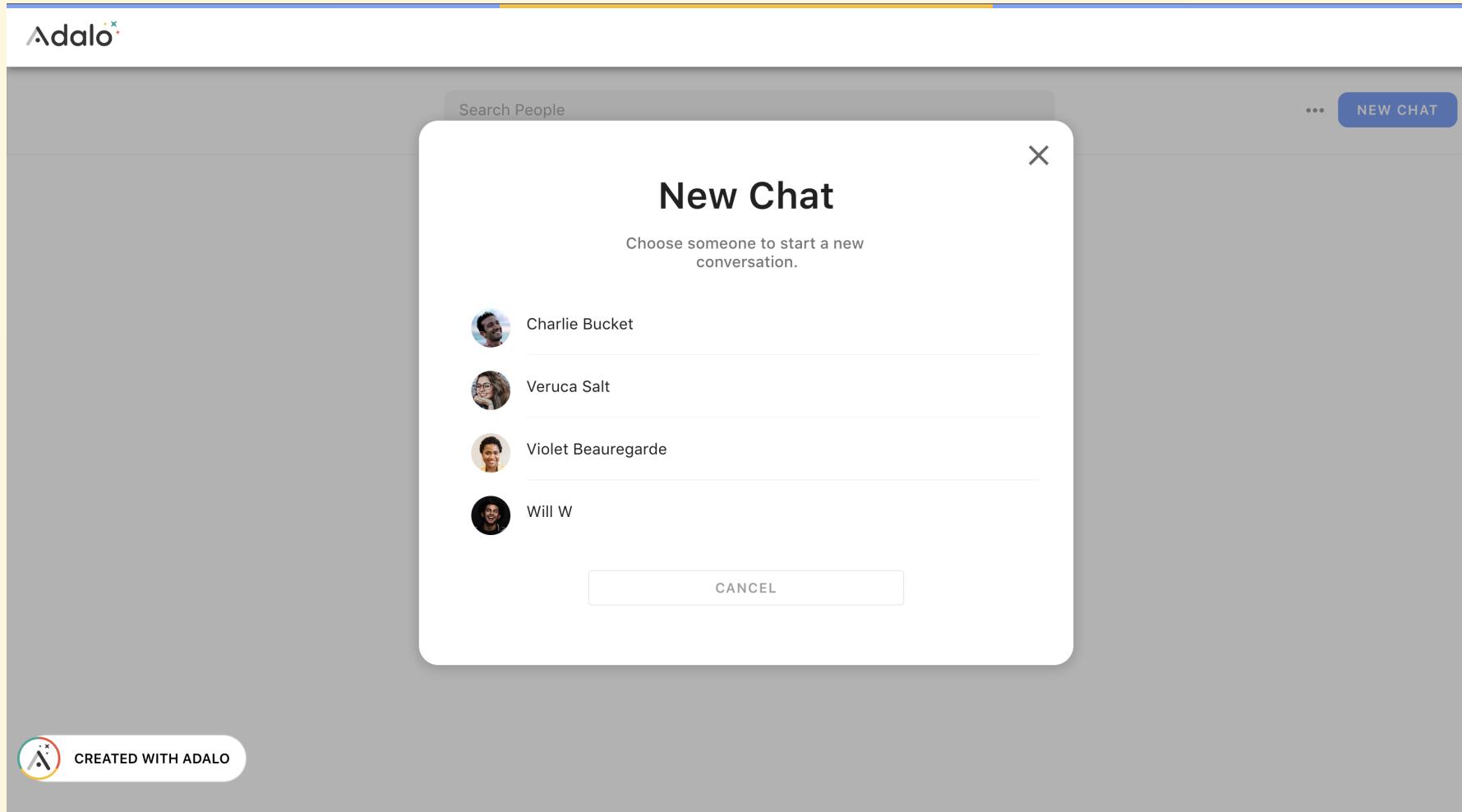
- Signupしましょう
 - EmailとPasswordはメモしておいてください(後で使います)



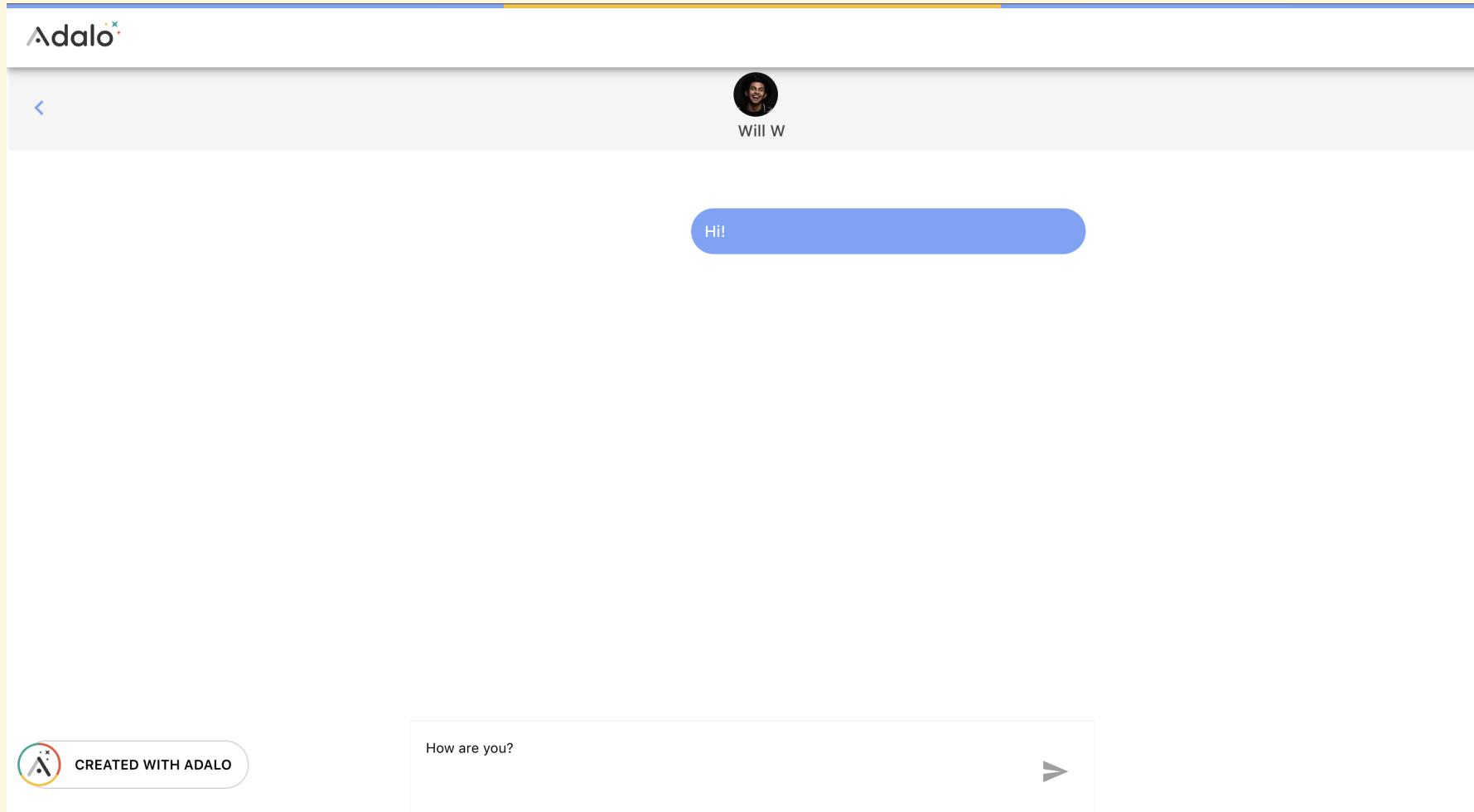
- まだ会話が行われていないため、リストが空の状態です
- 画面右上の NEW CHAT ボタンを押してみましょう



- サンプルユーザーのリストが表示されます
- Will Wを選んでクリックしてください



- メッセージを送信できます



他にも以下のような機能がありますので、試してみましょう。

- プロフィール写真の更新
- パスワードの変更
- ログアウト
- ログイン
 - 以下のように入力すると、サンプルユーザーの Will W としてログインができます
 - Email: will@email.com
 - Password: 123

- 今回試したChat以外にも、いくつかのテンプレートが提供されています。
- 自分たちが作りたいアプリに近いものがあれば、そのテンプレートを流用して開発スピードをアップさせられるかもしれません。時間がある時に、他のテンプレートも試してみましょう。

Adaloでのアプリ開発の概要説明

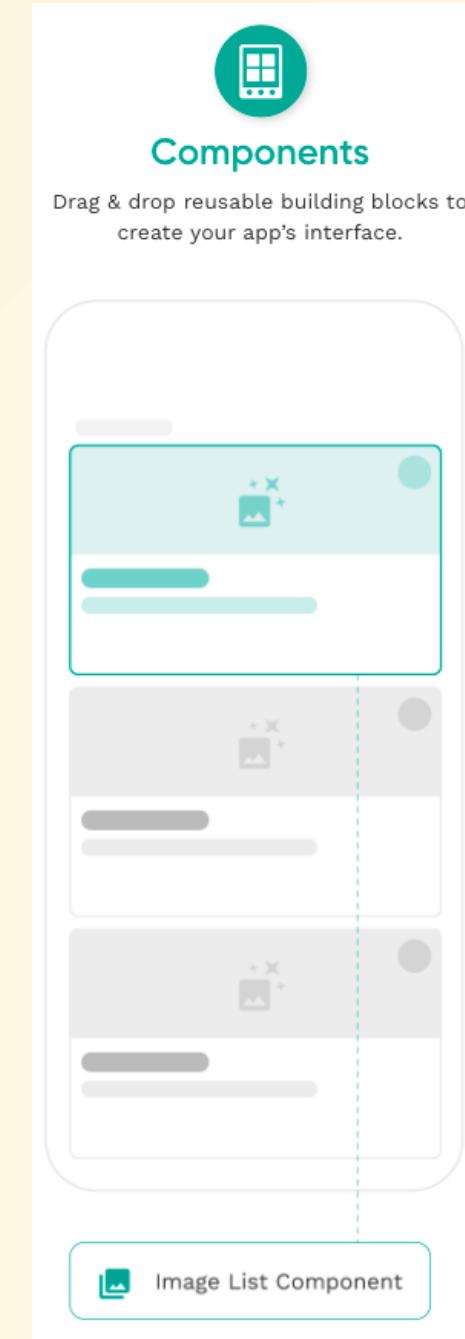
次に、Adaloでのアプリ開発の概要を説明していきます。

3つの基本コンセプト

- 基本となる以下3つのコンセプトについて紹介します
 - Components
 - Database
 - Actions

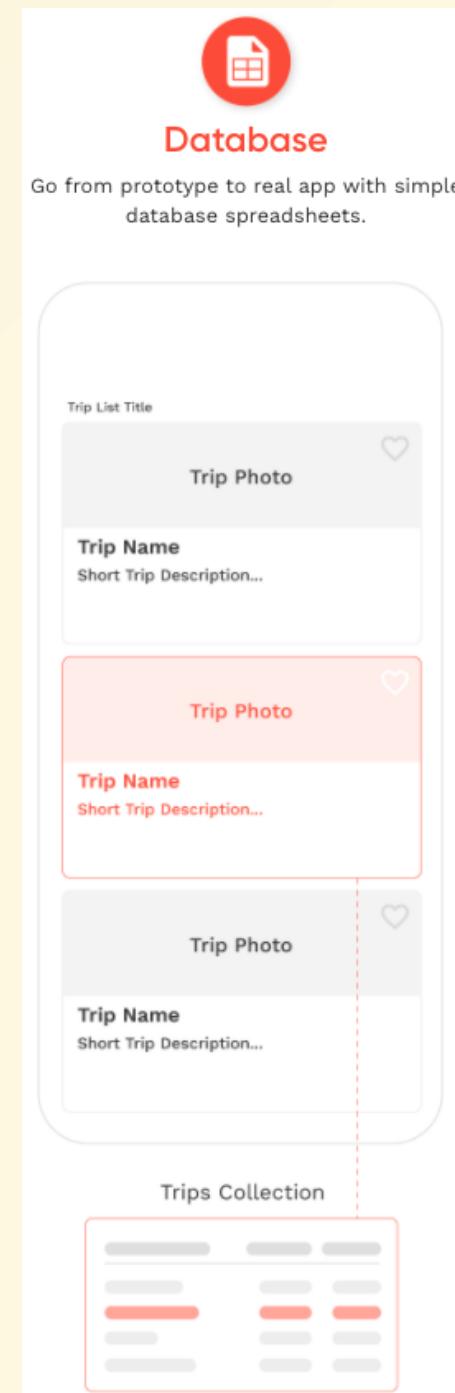
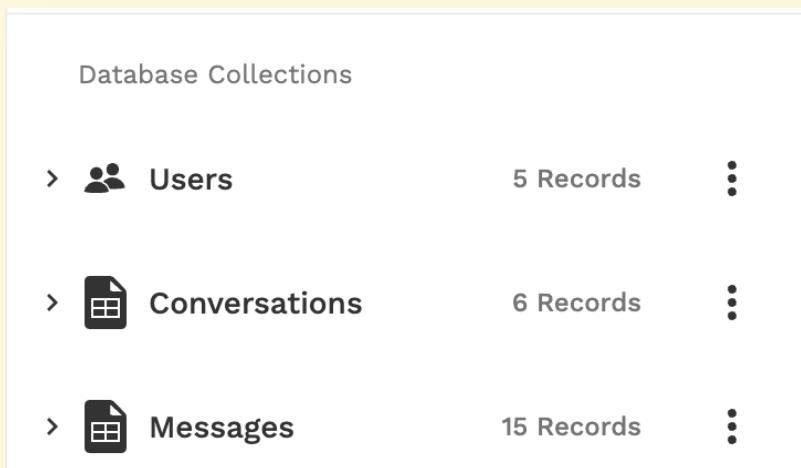
Components

- ユーザーインターフェースを作るために画面上に配置される要素。
- 例:
 - リスト
 - ボタン
 - テキスト
 - 画像



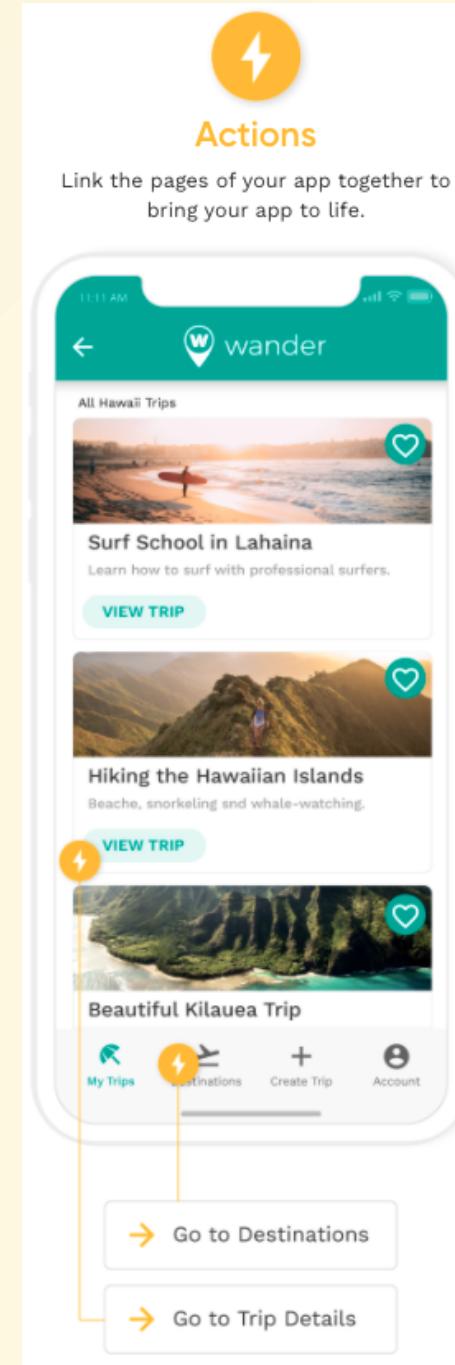
Database

- 整理されたデータの集合。
- データの登録、読み込み(表示)、更新、削除が行われる。
- 例: Chatアプリの場合



Actions

- 特定のコンポーネントをクリックした時に何を行うかを指定するために使われる。
- 例:
 - 別の画面に遷移させる
 - データベースのデータを登録、更新、削除する



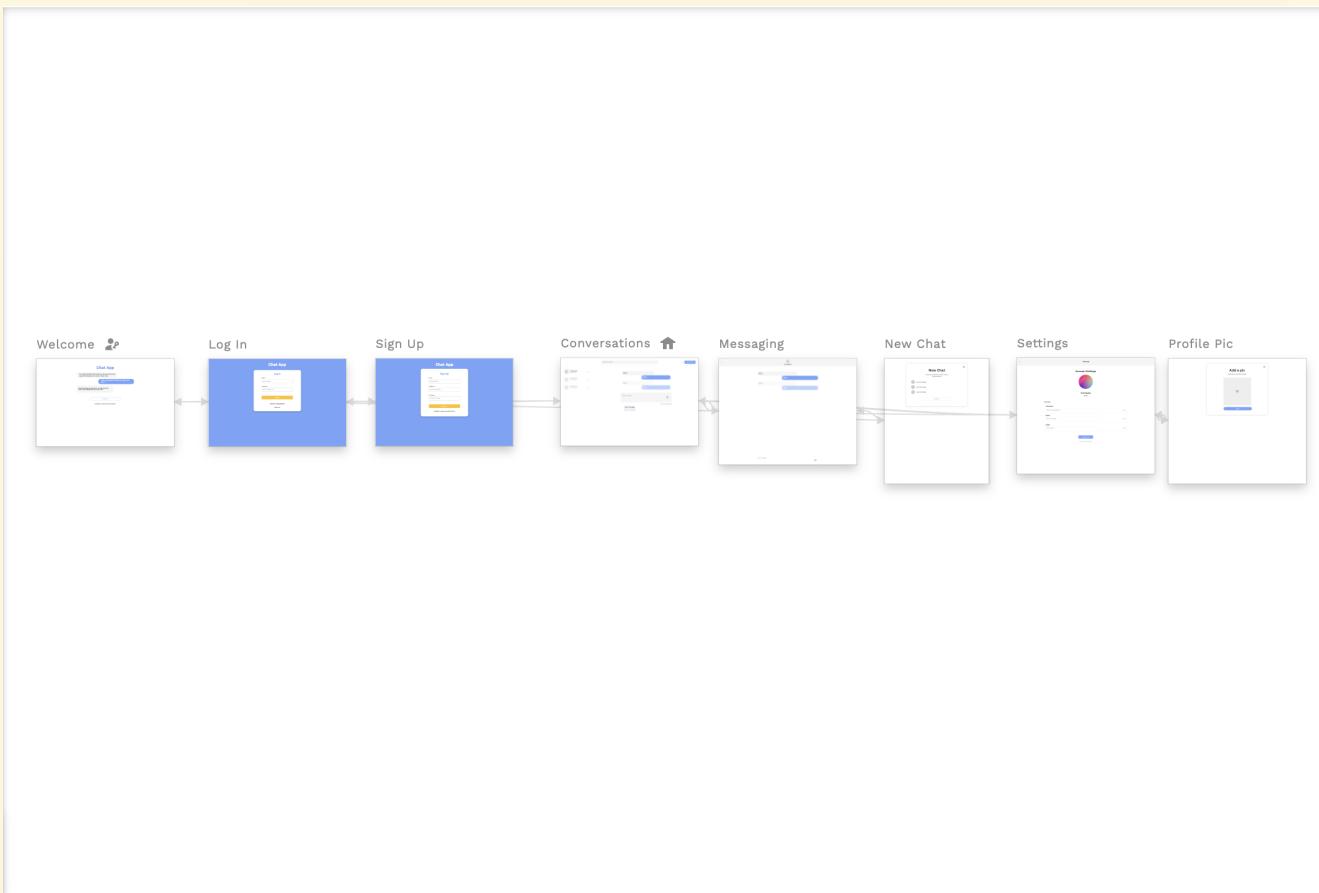
Adaloの機能説明

次に、Adaloの管理画面で利用できる機能を紹介していきます

The screenshot shows the Adalo app builder interface. On the left, there is a sidebar titled "ADD COMPONENT" containing various UI element templates like Text, Button, Simple List, App Bar, Image, and Form. A search bar at the top of the sidebar says "Search Components...". On the right, the main workspace displays a flowchart titled "ChatApp" with several screens connected by arrows: Welcome, Log In, Sign Up, Conversations, Messaging, New Chat, Settings, and Profile Pic. The "Log In" screen is highlighted with a blue background. At the bottom of the workspace, there is a progress bar stating "52% of free records used" and a red "UPGRADE NOW" button.

Canvas

- 画面を作る作業領域
- 要素を選択したり、ドラッグアンドドロップで動かしたりできる



Left Toolbar

左側のツールバーの各機能を紹介します。





Add Panel

- コンポーネントや画面を選択してアプリに追加できる
- 機能のテンプレートを追加することも可能

The screenshot shows the 'Add Panel' interface with two main sections:

- ADD COMPONENT**:
 - Search bar: Search Components...
 - Category: Most Used (6 items)
 - Components:
 - Text: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
 - Button: + BUTTON
 - Simple List
 - App Bar
 - Image
 - Form
 - Bottom button: ADD FEATURE TEMPLATE
- ADD SCREEN**:
 - Search bar: Search Screens...
 - Category: Most Used (2 items)
 - Screens:
 - Blank Desktop First
 - Blank Mobile First
 - Layout (4 items)
 - Lists (16 items)
 - Detail (5 items)
 - Create (3 items)
 - Account Settings (1 item)
 - Bottom button: ADD FEATURE TEMPLATE



Branding

- 色やフォントを変えられる

The screenshot displays a branding configuration interface with two main sections: Colors and Fonts.

Colors:

- Primary Color: #75a3fa (blue box)
- Secondary Color: #FFC00E (yellow box)
- Default Background Color: #FFFFFF
- Default Text Color: #424242

Fonts:

- Headings: Libre Baskerville
- Body: Standard System Font

At the bottom right, there is a callout: "Need more cool fonts? We got you! Try out our custom fonts to see which one will work best for your app" with a "START FREE TRIAL" button.



Screens

- 画面の一覧や、その画面の構成を表示できる

The screenshot shows the Figma application interface. On the left, the 'Screens' panel lists various screen components:

- Conversations
- Log In
- Messaging
- New Chat
- Profile Pic
- Settings
- Sign Up
- Welcome

At the bottom of this panel is a '+ ADD SCREEN' button. To the right of the 'Screens' panel is the 'Components' panel, which displays a list of components under the 'Normal Screen Log In' section:

- Text Chat App
- Shape Card
- Text Log In
- Button Button
- Button Button
- Form Form

Below these sections are 'Actions' and 'Available Data' sections, each with a '+' ADD COMPONENT button. A vertical toolbar on the far left contains icons for creating screens, components, shapes, text, buttons, cards, and forms.



Database

- データベースの構成や保存されているデータを表示できる
- Collection: 同じ属性(プロパティ)を持ったデータの集まり

The screenshot shows a database application interface. On the left is a sidebar with various collection names and their record counts:

- Database Collections (5 Records)
- Users (5 Records)
- Email (1 Record)
- Password (1 Record)
- Username (1 Record)
- Full Name (1 Record)
- Messages (1 Record)
- Profile Image (1 Record)
- Messages (1 Record)
- Conversations (6 Records)
- Friends (1 Record)
- Active Conversation (7 Records)
- + ADD PROPERTY
- Conversations (6 Records)
- Messages (15 Records)
- + ADD TO DATABASE

The main area is titled "Users" and contains a table with the following data:

	A Email	>Password	Username	Full Name	Messages
<input type="checkbox"/>	imahashi@example.com	[hidden]		Ryo Imahashi	2 messages
<input type="checkbox"/>	veruca@email.com	[hidden]	veruca	Veruca Salt	2 messages
<input type="checkbox"/>	charlie@email.com	[hidden]	charlie	Charlie Bucket	2 messages
<input type="checkbox"/>	violet@email.com	[hidden]	sarah	Violet Beauregarde	1 message
<input type="checkbox"/>	will@email.com	[hidden]	will	Will W	8 messages

At the top right of the main area, there are several icons: + ADD USER, shield, up arrow, down arrow, double arrows, and a magnifying glass.



Settings

- アプリの名前やアイコンを設定できる
- キャンバスに関する表示設定ができる
- アプリへのアクセス権限の設定ができる
- アプリの複製や削除ができる
- 位置情報機能を使うためのAPIキーの設定ができる

The screenshot shows the 'Settings' screen. On the left is a vertical sidebar with eight icons: a plus sign in a circle, a paint palette, a square with a border, a document, a brush, a gear (selected), a checkmark, and a circular arrow. To the right of the sidebar is a list of settings options:

- > App Settings
- > Display Settings
- > App Access
- > Copy / Delete
- > API Keys



Publish

- 作ったアプリを公開できる
(有料プランのみ)

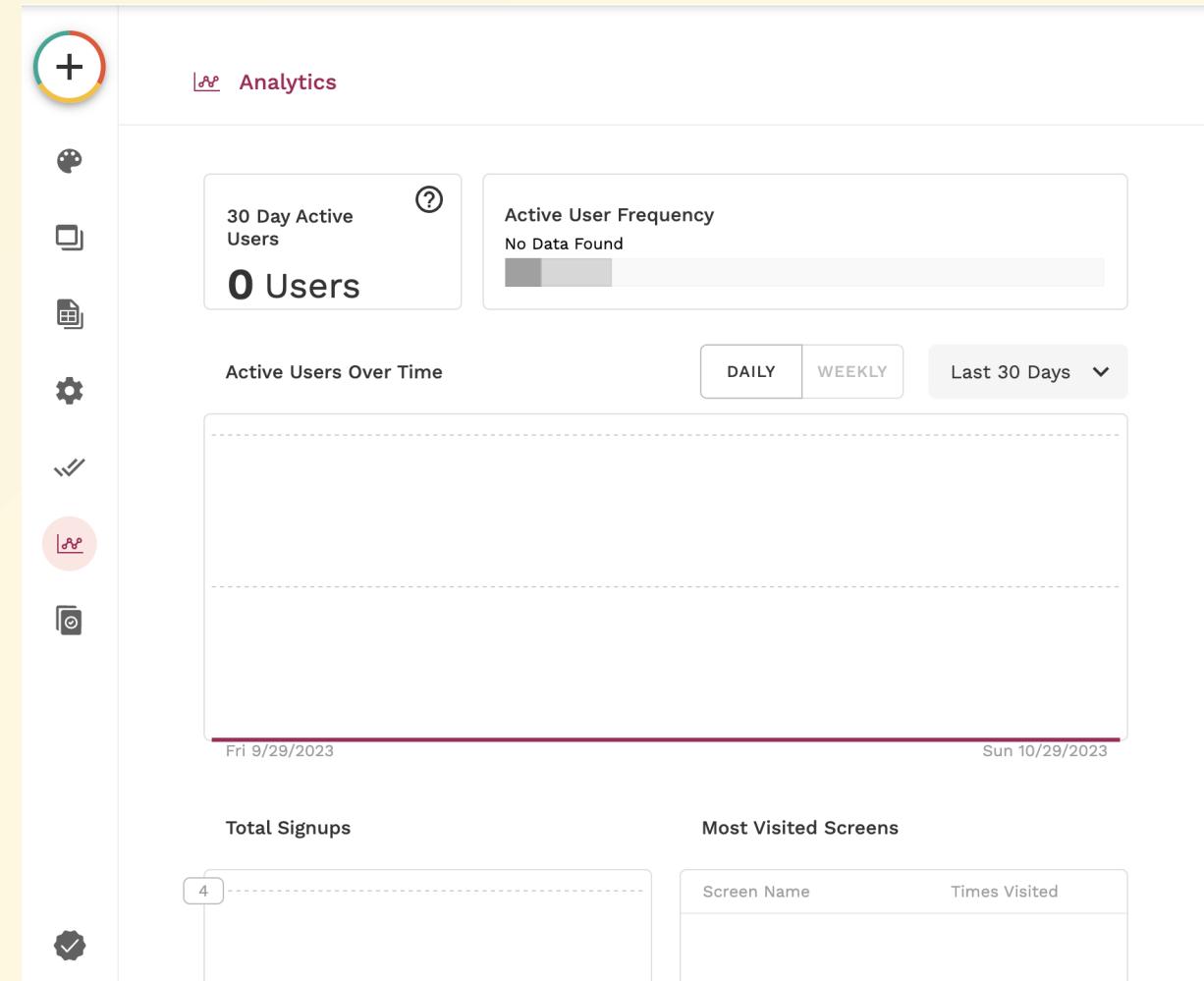
The screenshot shows the 'Publish' section of the Appy Pie platform. On the left, there's a vertical stack of icons: a green circle with a checkmark, a plus sign inside a red circle, a paint palette, a square with rounded corners, a document with a grid, a gear, a green circle with a checkmark, a speech bubble with a double arrow, and a square with a circular arrow. To the right of these icons is a white box containing a green checkmark and the word 'Publish'. Below this is a call-to-action box with the text 'Upgrade to Publish your Apps!' and 'Switch to a paid plan to start sharing your apps with the world.' A large green 'UPGRADE' button is centered in this box. To the right of the upgrade box is a list of three items, each preceded by a grey chevron: 'Web App', 'Android App', and 'iOS App'.

- > Web App
- > Android App
- > iOS App



Analytics

- 利用状況を分析したレポートを見ることができる





Version History

- バージョン履歴を作成したり、アプリをバージョン履歴作成時と同じ状態に戻すことができる(有料プランのみ)

Version History

+

?

▼ Design Versions 2/10

Free Trial | 14 Days Left

Create and restore versions of your app's design without affecting your database, user data, or existing native builds.

UPGRADE NOW

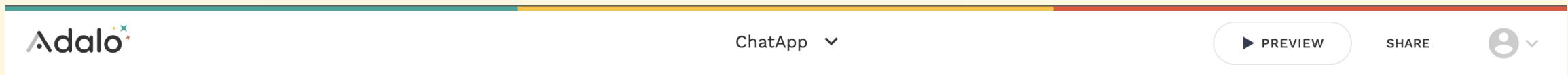
+ CREATE DESIGN VERSION

Second Version
Created by Ryo on Nov 4 11:29 pm
something changed.

Initial Version
Created by Ryo on Nov 4 11:28 pm
Nothing changed.

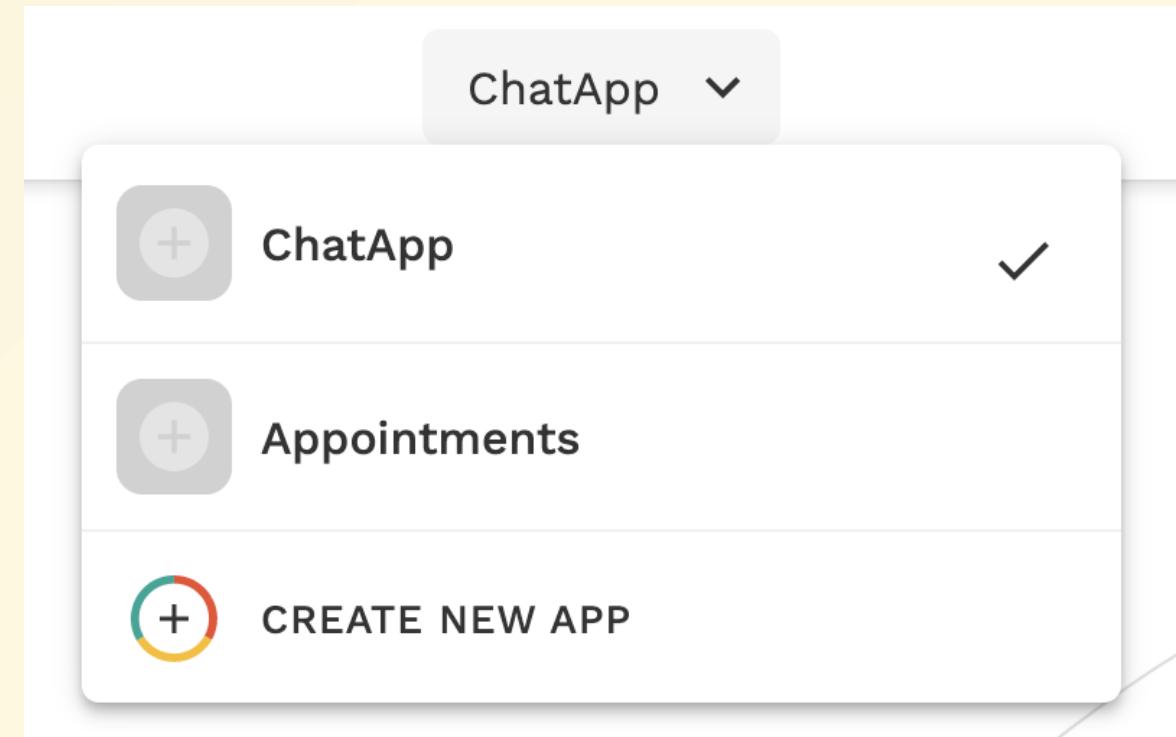
Top Bar

上部のツールバーの各機能を紹介します。



App Switcher

- 開いているアプリの名前が表示される
- アプリを切り替えられる
- 新しいアプリを追加できる



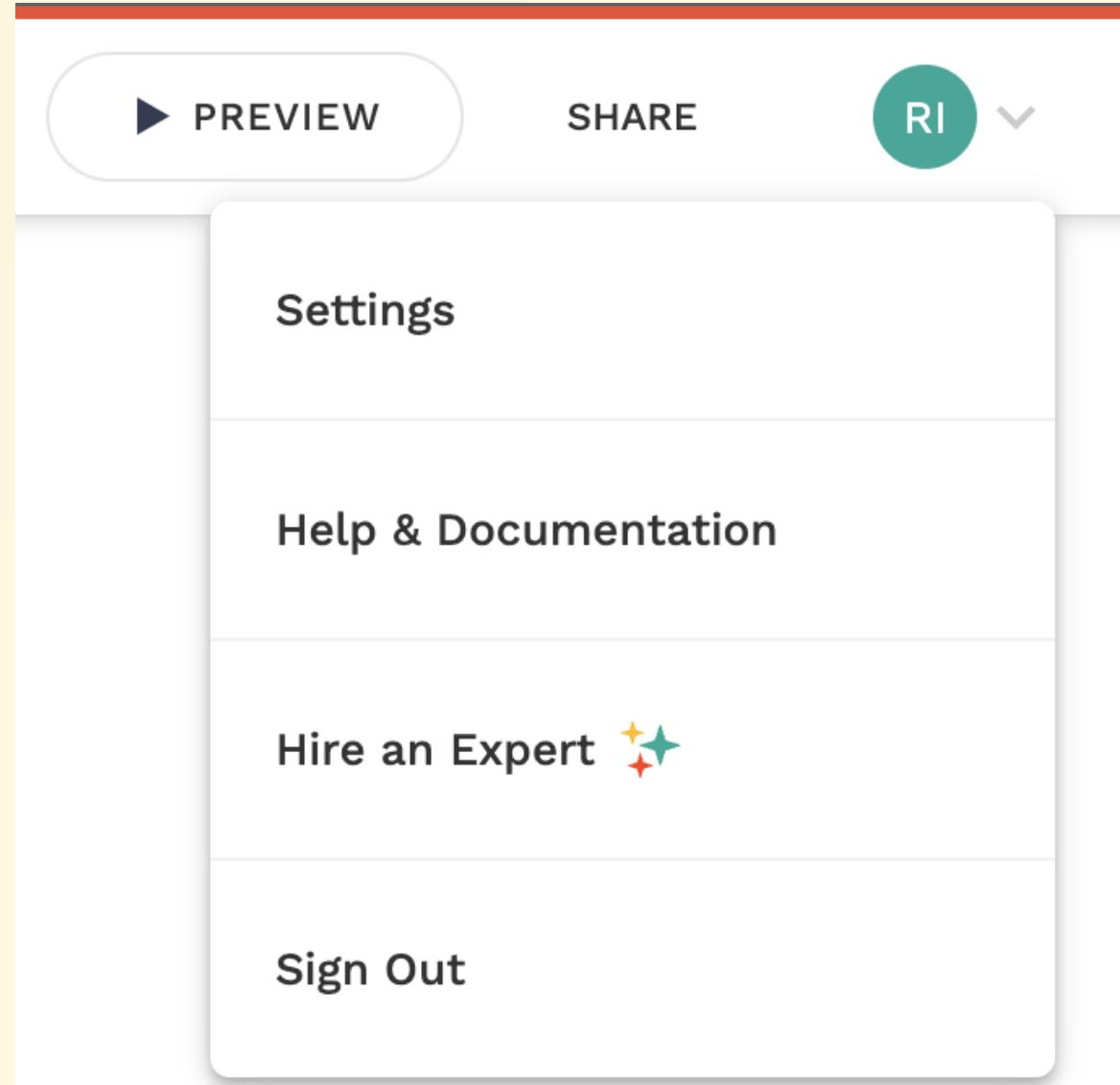
Preview, Share

- アプリを実行して試すことができる

The screenshot shows a web-based application preview for a "Chat App" template. At the top left is the Adalo logo. The main title "Chat App" is centered above a subtitle: "An Adalo template that helps you easily create chat apps that look great on every screen size." Below this is a blue button with white text that says "That sounds great! Where do I sign up to try it?". A gray callout box below the button contains the text "Click the sign up link below or sign into the demo with will@email.com, pw 123.". At the bottom center is a blue "SIGN UP" button. To its right, the text "ALREADY HAVE AN ACCOUNT?" is displayed. At the very bottom left is a small circular icon with a stylized letter 'A' and the text "CREATED WITH ADALO".

Account Menu

- 各種設定ができる
- ヘルプやドキュメントを開ける
- Adaloのエキスパートを探せる
- ログアウトできる



開発に役立つTipsを覚えておこう

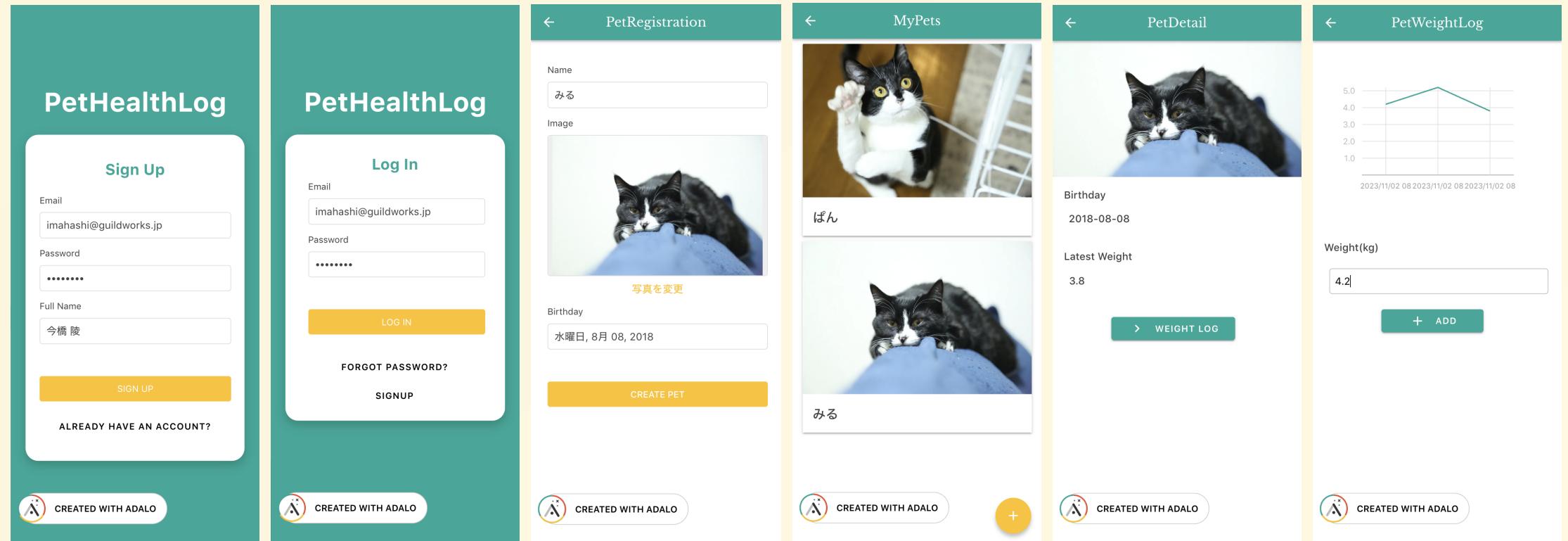
- 間違って編集してしまった時は、Windowsなら `Ctrl + Z`、Macなら `Command + Z` で元に戻せます
- Adaloの開発ツールでは、日本語の直接入力がうまくいかないことがあります。日本語のテキスト入力はコピー＆ペーストで行ってください。

アプリを作ってみよう

次に、新しくアプリを作ってみましょう。

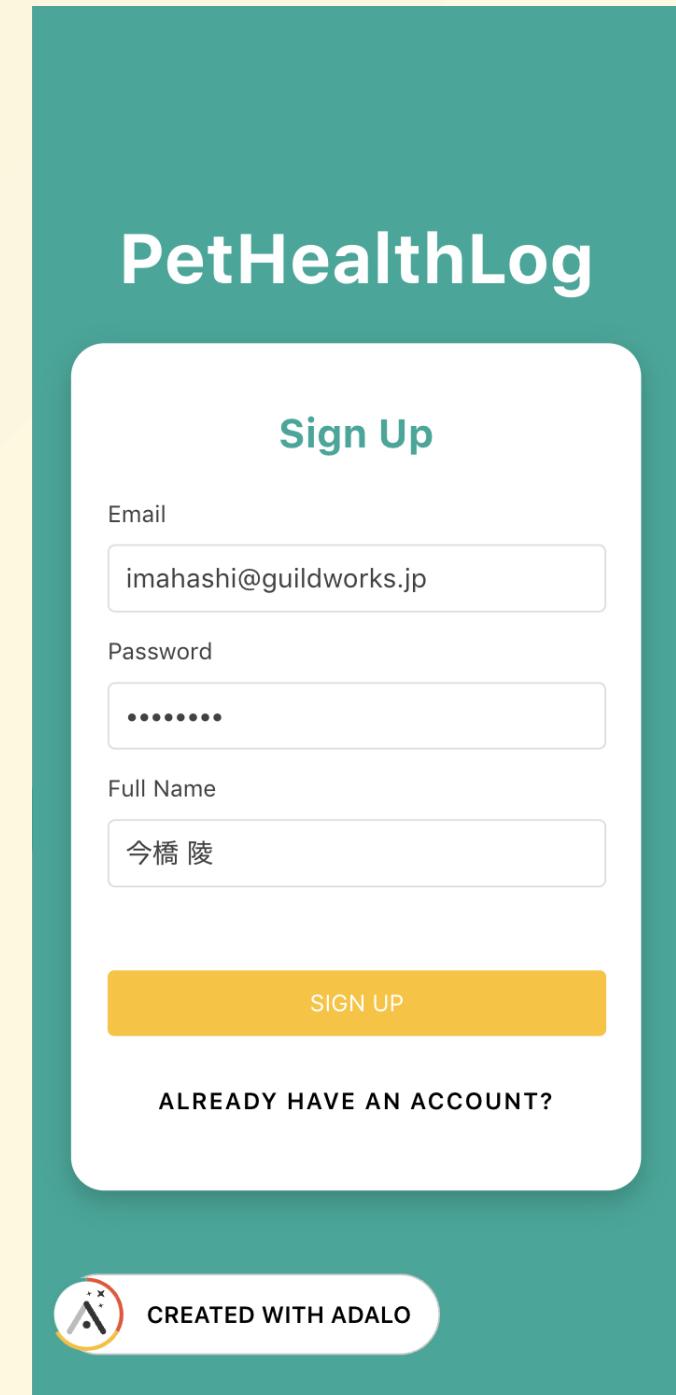
作りたいアプリのUI

ペットの健康管理アプリを作ってみましょう。
まずはUIを確認していきます。



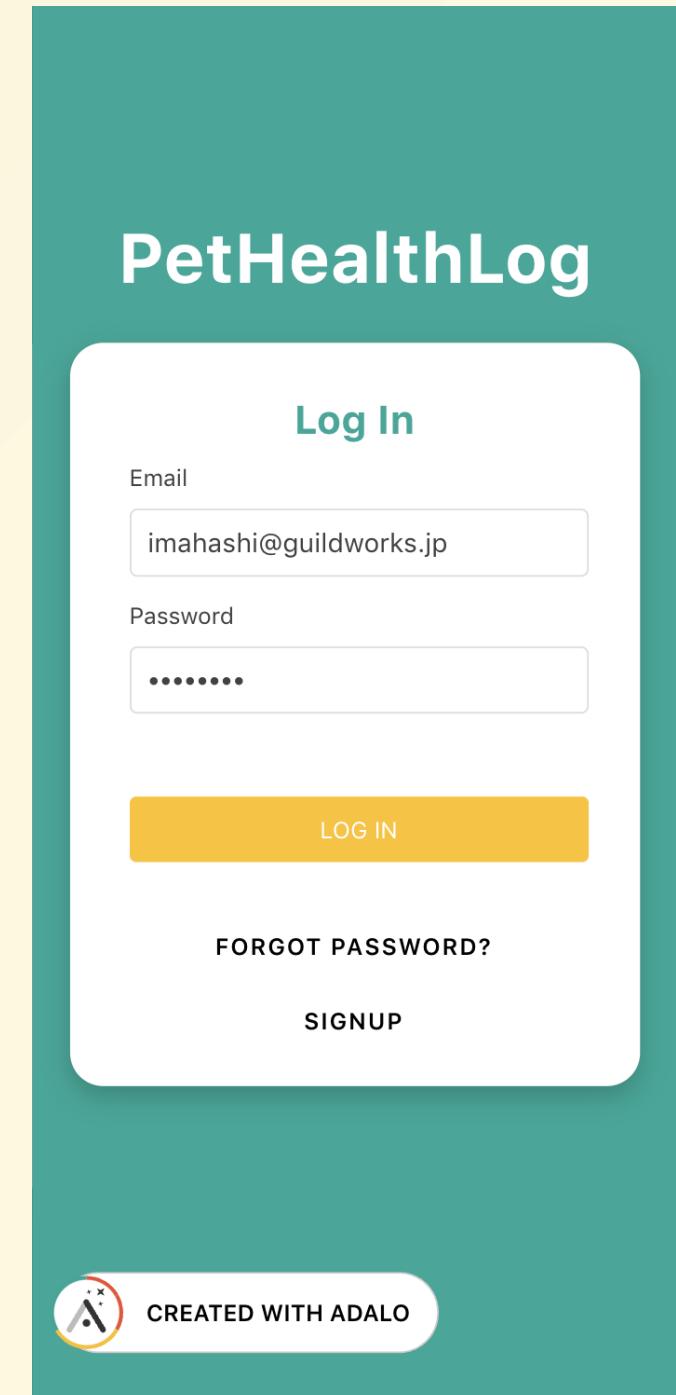
会員登録画面

- 以下の項目を入力して会員登録できる
 - Email
 - Password
 - Full Name
- 会員登録済の人向けに、ログイン画面へのリンクがある



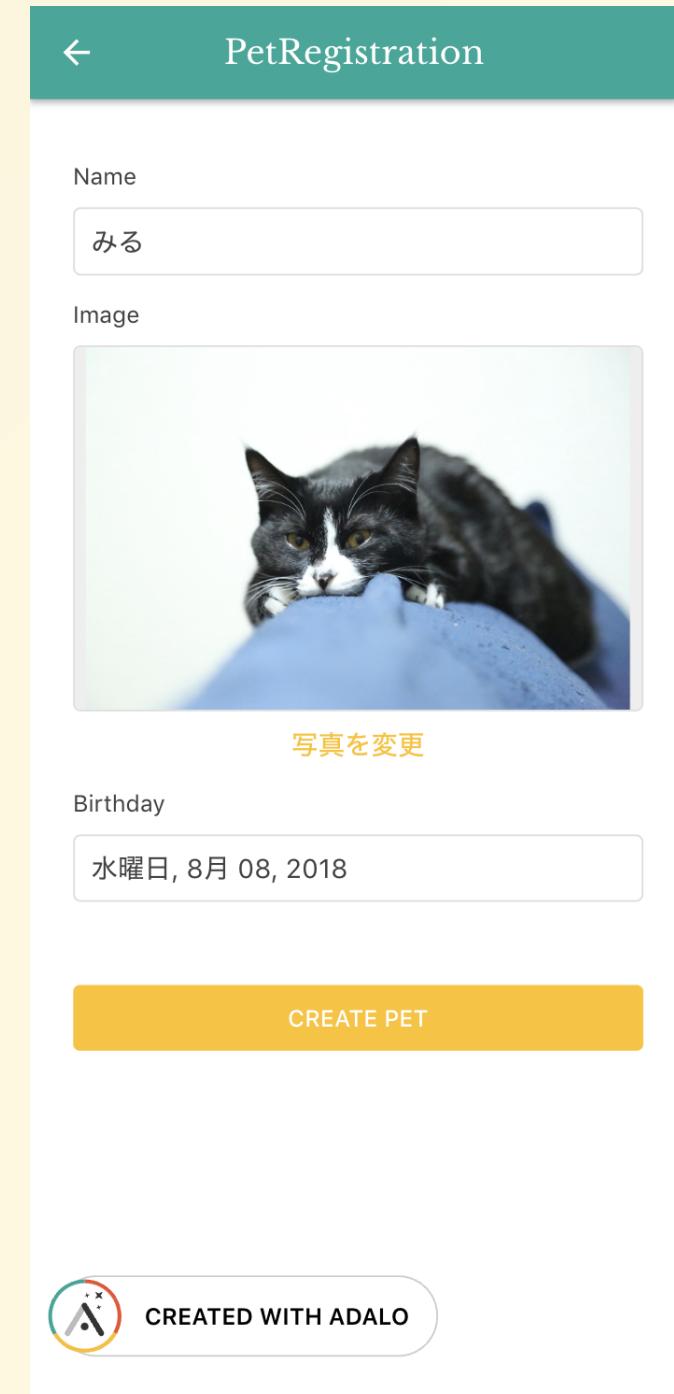
ログイン画面

- 以下の項目を入力してログインできる
 - Email
 - Password
- パスワードを忘れた人向けのリンクがある
- 会員登録画面へのリンクがある



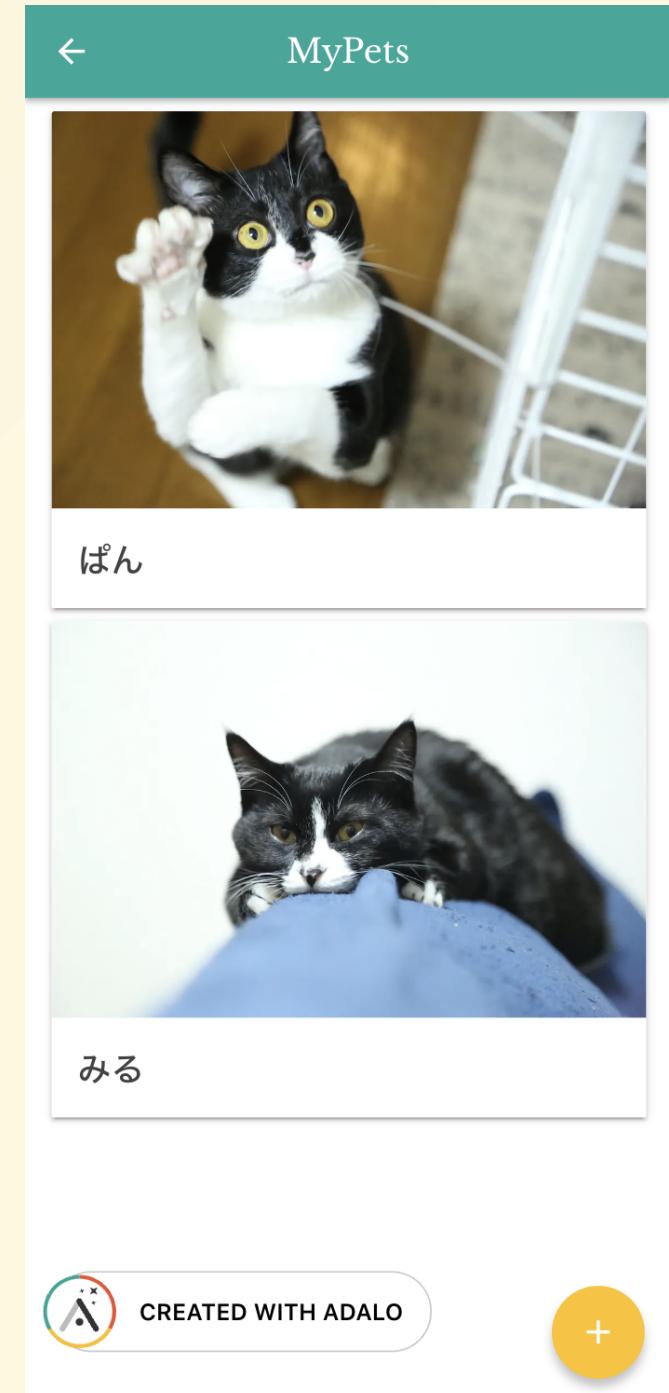
ペット登録画面

- 名前を入力できる
- 写真を選択できる
- 誕生日を入力できる
- 登録ボタンで確定し、ペツト一覧画面に遷移できる



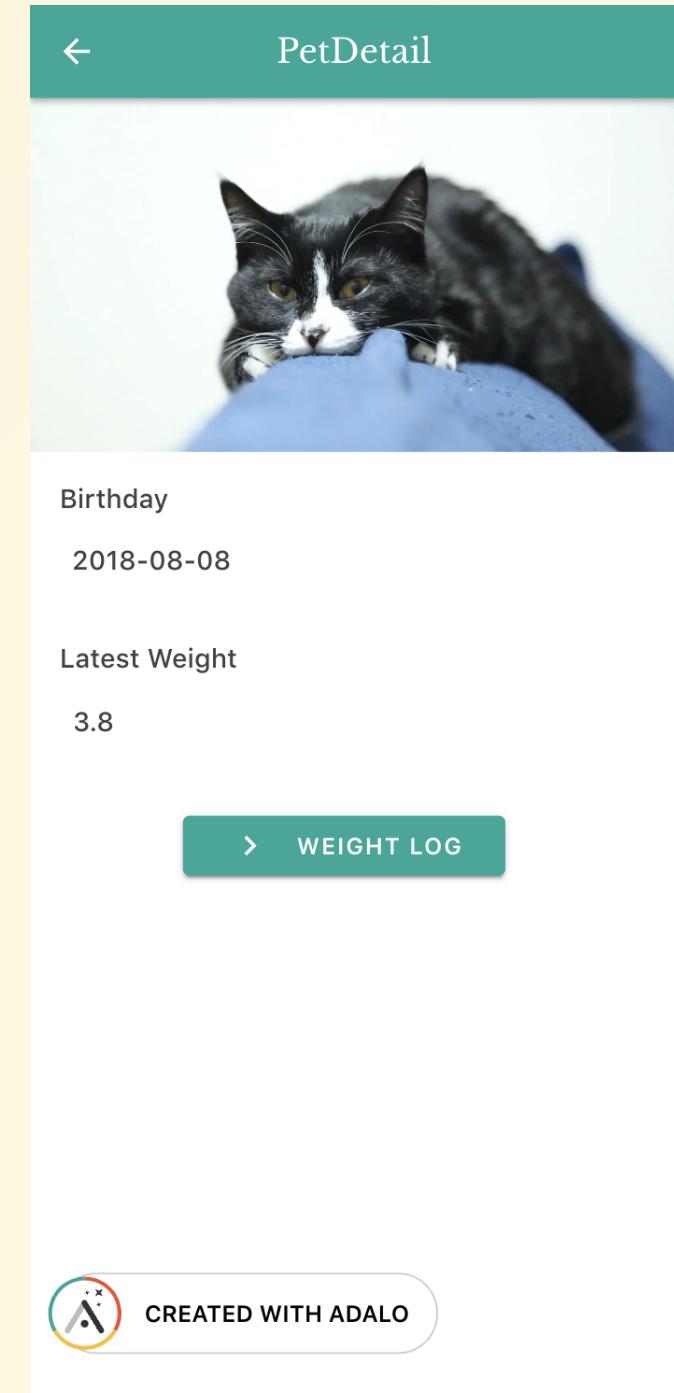
ペット一覧画面

- 登録したペットが一覧で表示できる
- ペットをクリックすると、そのペットのペット詳細画面に遷移できる
- 右下のアイコンを押すと、ペット登録画面に遷移できる



ペット詳細画面

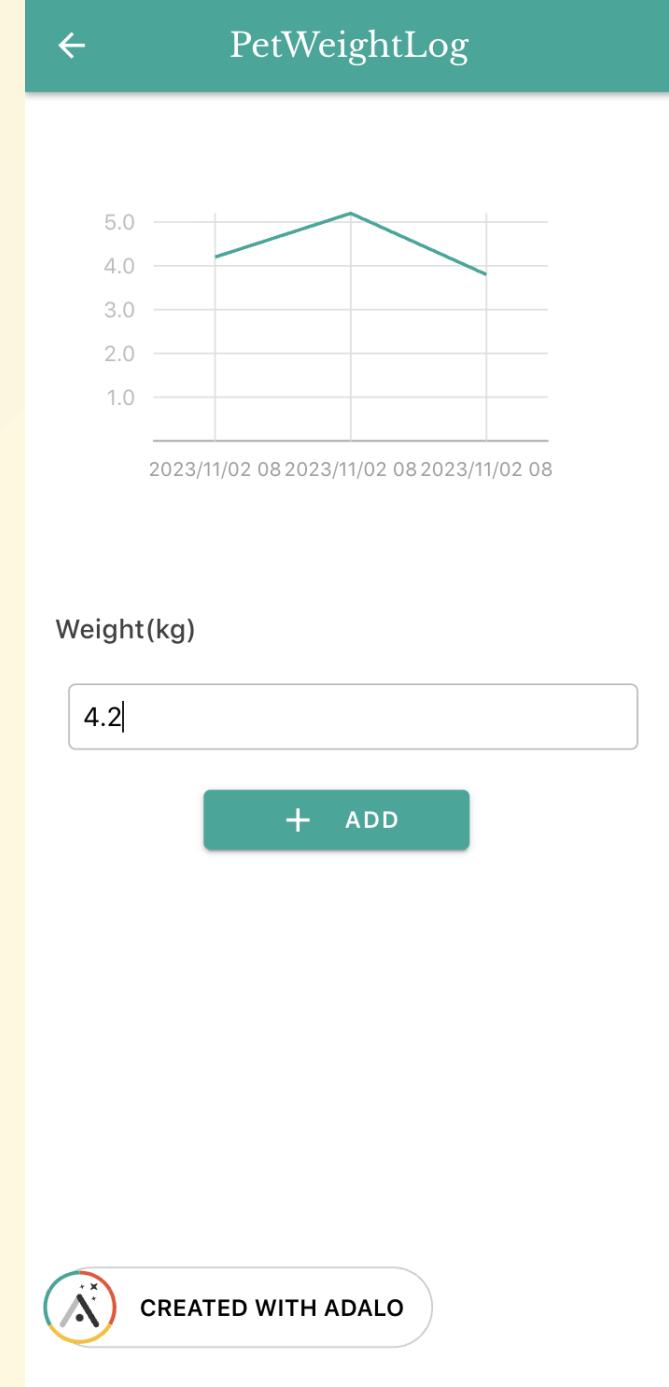
- 誕生日が表示される
- 最新の体重が表示される
- 体重記録画面へのリンクがある





体重記録画面

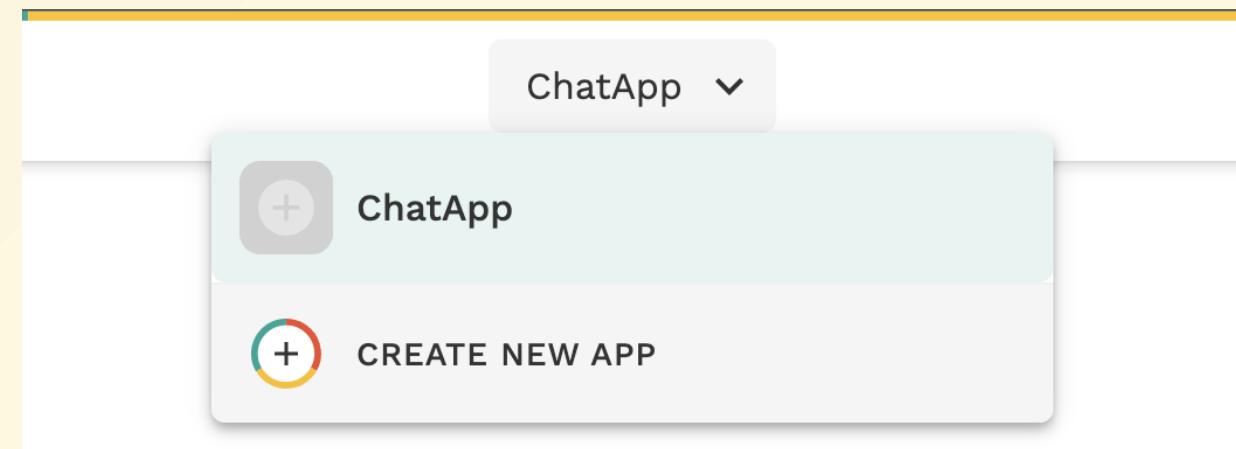
- 体重の遷移を示すグラフが表示される
- 現在の体重を入力できる
- ボタンを押して体重を追加できる



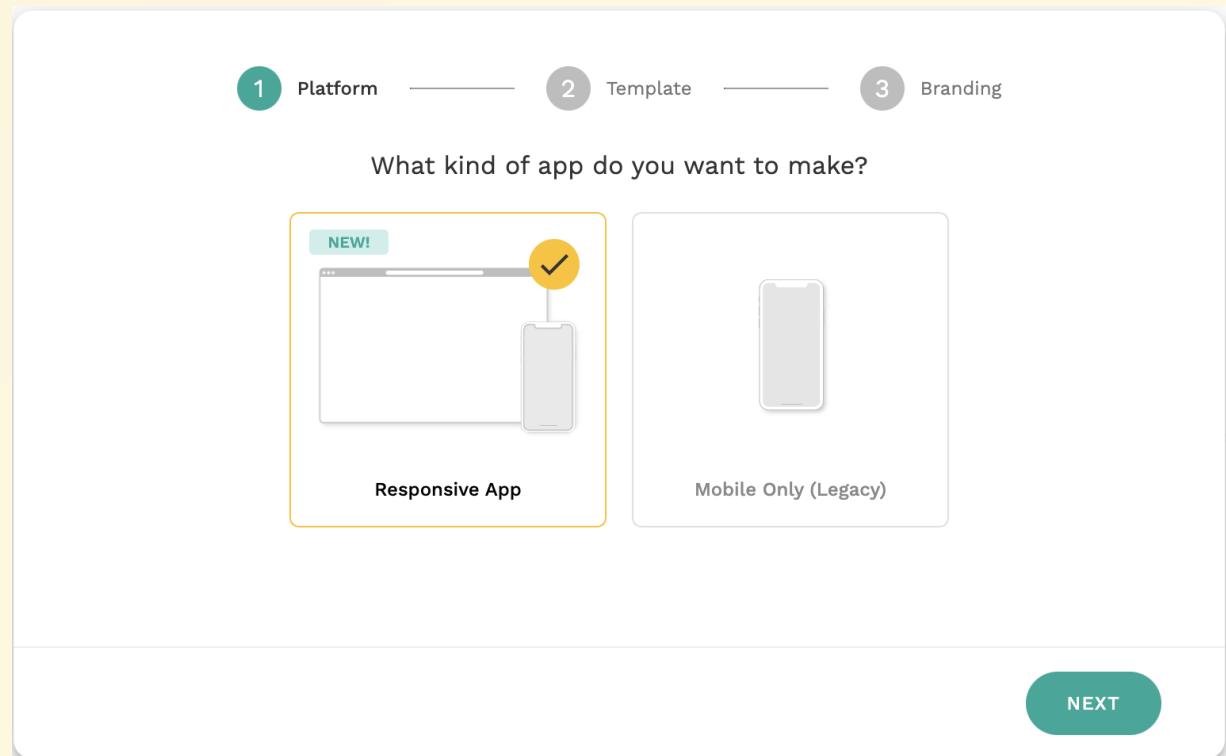
アプリ作成

それでは、アプリを作りまし
ょう。

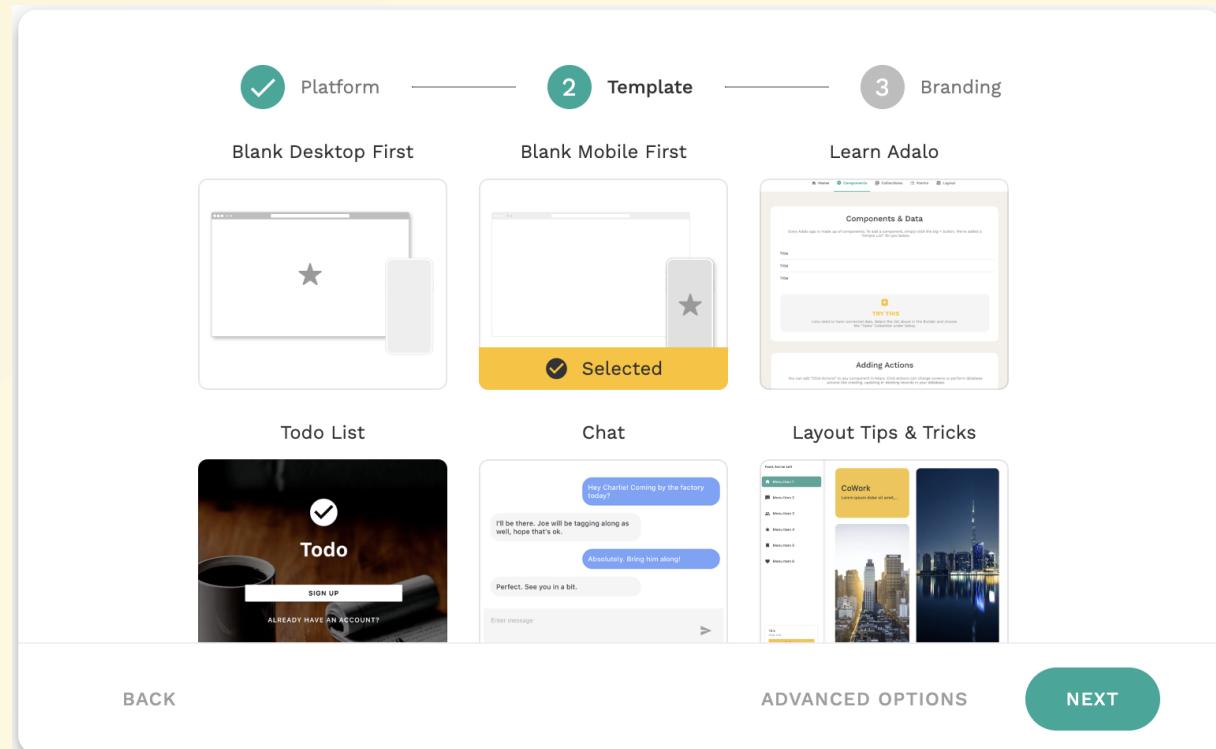
- CREATE NEW APPを選択し
てください



- Responsive Appを選択してください



- テンプレート: Blank Mobile First を選択してください



- App Name、Users of this app、Colorを自由に決めてください

The screenshot shows a branding configuration interface. At the top, there are three circular status indicators: 'Platform' (green checkmark), 'Template' (green checkmark), and 'Branding' (green circle with the number '3'). Below these are four input fields:

- App Name:** PetHealthLog
- Who are the users of this app?**: Mostly me, I'm just testing
- Primary Color:** #00A898
- Secondary Color:** #FFC00E

At the bottom left is a 'BACK' button, and at the bottom right is a green 'CREATE' button.

● アプリができました

The screenshot shows the Adalo app builder interface with the project name "PetHealthLog" at the top. On the left, there's a sidebar with icons for adding components, screens, and features, along with a search bar and a "Most Used" section containing components like Text, Button, Simple List, App Bar, Image, and Form. A button labeled "ADD FEATURE TEMPLATE" is also present. The main workspace displays four screens: "Welcome", "Log In", "Sign Up", and "Home". The "Log In" screen is currently active, showing fields for Email and Password, and buttons for LOG IN, FORGOT PASSWORD?, and SIGNUP. Arrows indicate transitions between the Welcome, Log In, and Sign Up screens. The "Home" screen is shown as a blank white page. At the bottom right, there's a circular icon with a user profile and the number "24", and a banner at the bottom right says "0% of free records used" with a "UPGRADE NOW" button.

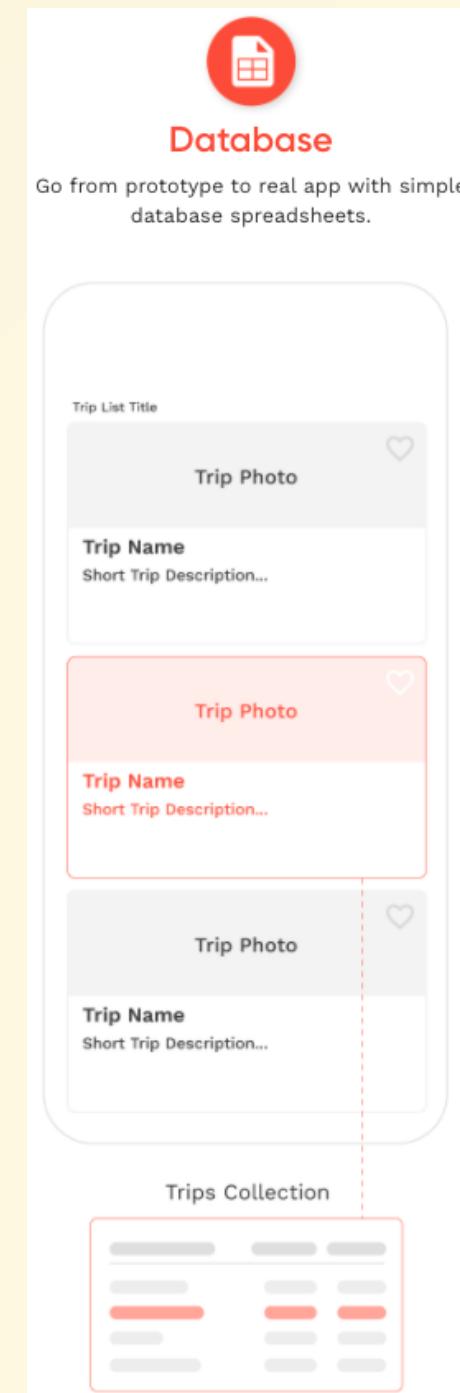
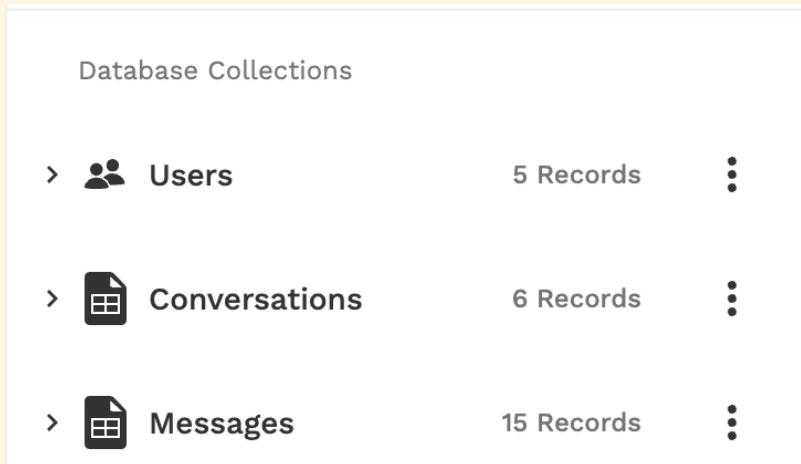
データベースを学ぼう

アプリ開発の最初にそのアプリで扱うデータを定義しておくことで、その後の開発をスムーズに進めることができます。

まずは、データベースがどのようなものかを確認します。

Database

- 整理されたデータの集合。
- データの登録、読み込み(表示)、更新、削除が行われる。
- 例: Chatアプリの場合



- データベースはよく「表計算ソフトのようなもの」と例えられます。
- データベースを使ってデータを作成(CREATE)、読み取り(READ)、更新(UPDATE)、削除(DELETE)することができます。これらの操作を総称してCRUDと呼びます。

Users

<input type="checkbox"/>	Email	 Password	Username	Full Name	 Profile Image
<input type="checkbox"/>	imahashi@example.com	[hidden]		Ryo Imahashi	
<input type="checkbox"/>	veruca@email.com	[hidden]	veruca	Veruca Salt	
<input type="checkbox"/>	charlie@email.com	[hidden]	charlie	Charlie Bucket	
<input type="checkbox"/>	violet@email.com	[hidden]	sarah	Violet Beauregarde	
<input type="checkbox"/>	will@email.com	[hidden]	will	Will W	

[DONE](#)

Adaloのデータベースの基本



このアイコンからAdaloのデータベースにアクセスできます。
Adaloのデータベースの構成要素は、以下の3つです。

- Collection
- Property
- Record

Collectionとは

同じ属性(Property)を持ったデータの集まり

The screenshot shows a database interface with a sidebar on the left containing icons for various collections: Database Collections, Users (selected), Email, Password, Username, Full Name, Messages, Profile Image, Conversations, Friends, and Active Conversation. The main area is titled 'Users' and displays a table with 5 records. The columns are: Email, Password, Username, Full Name, and Profile Image. The data is as follows:

	Email	Password	Username	Full Name	Profile Image
<input type="checkbox"/>	imahashi@example.com	[hidden]		Ryo Imahashi	
<input type="checkbox"/>	veruca@email.com	[hidden]	veruca	Veruca Salt	
<input type="checkbox"/>	charlie@email.com	[hidden]	charlie	Charlie Bucket	
<input type="checkbox"/>	violet@email.com	[hidden]	sarah	Violet Beauregarde	
<input type="checkbox"/>	will@email.com	[hidden]	will	Will W	

At the top right of the main area are buttons for '+ ADD USER', shield, upload, download, search, and refresh. At the bottom right is a red 'DONE' button.

- Collectionは、データベースで扱う様々なデータをデータの種類ごとに分割し、整理するためのものです。(類似の言葉として、テーブルがあります)
- ユーザーが1度の操作で登録や更新、削除といった操作をするデータのまとめがCollectionになることが多いです。
- デフォルトでは、UsersがCollectionとして用意されており、それ以外は開発するアプリに合わせて追加していきます。

※ どのようなCollectionを追加するかを決めるのはとても難しいです。練習しながら慣れていきましょう。(悩んだ時は、メンター陣に相談するのもおすすめです)

Recordとは

- Recordは、Collection内へ情報を保存する際の単位です。
 - 画像の1行が1つのRecordとなります。
- Users Collectionの例では、1人のユーザーが持つ情報をまとめて1Recordとして登録します。

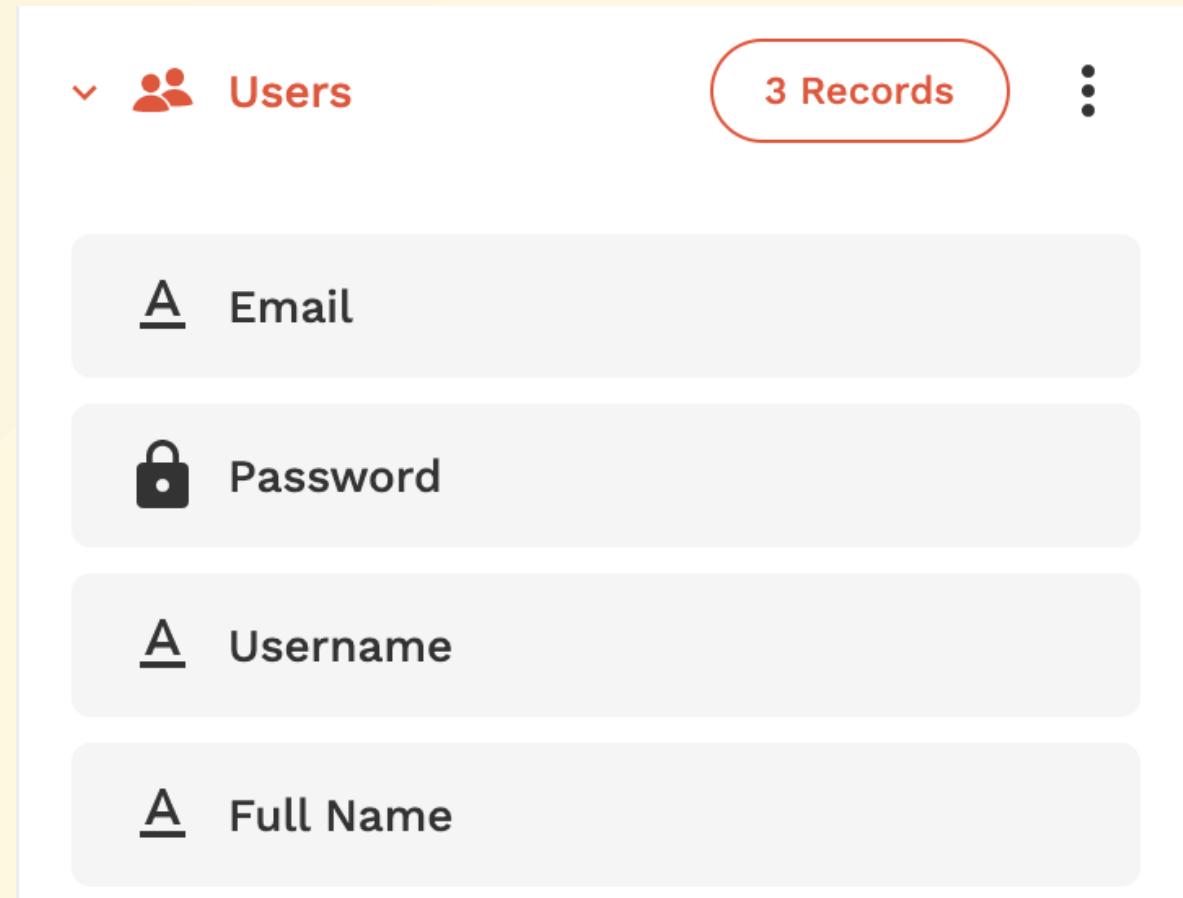
Users						
	<input type="checkbox"/>	Email	Password	Username	Full Name	Profile Image
	<input type="checkbox"/>	imahashi@example.com	[hidden]		Ryo Imahashi	
	<input type="checkbox"/>	veruca@email.com	[hidden]	veruca	Veruca Salt	
	<input type="checkbox"/>	charlie@email.com	[hidden]	charlie	Charlie Bucket	
	<input type="checkbox"/>	violet@email.com	[hidden]	sarah	Violet Beauregarde	
	<input type="checkbox"/>	will@email.com	[hidden]	will	Will W	

- Recordは基本的にアプリの画面上のフォームから登録できるようになりますが、Recordの表示中に右上の「+Add xxxx」ボタンを押して、右の画像のようなフォームから登録することも可能です。
- Collection内のRecordの検索や、CSVファイルのアップロード(インポート)・ダウンロードも可能です。

The image shows a user registration form titled "New User". The form includes fields for Email, Password, Username, Full Name, and Profile Image. The Profile Image field is a placeholder with the text "Choose image". At the bottom right, there are "CANCEL" and "SAVE" buttons.

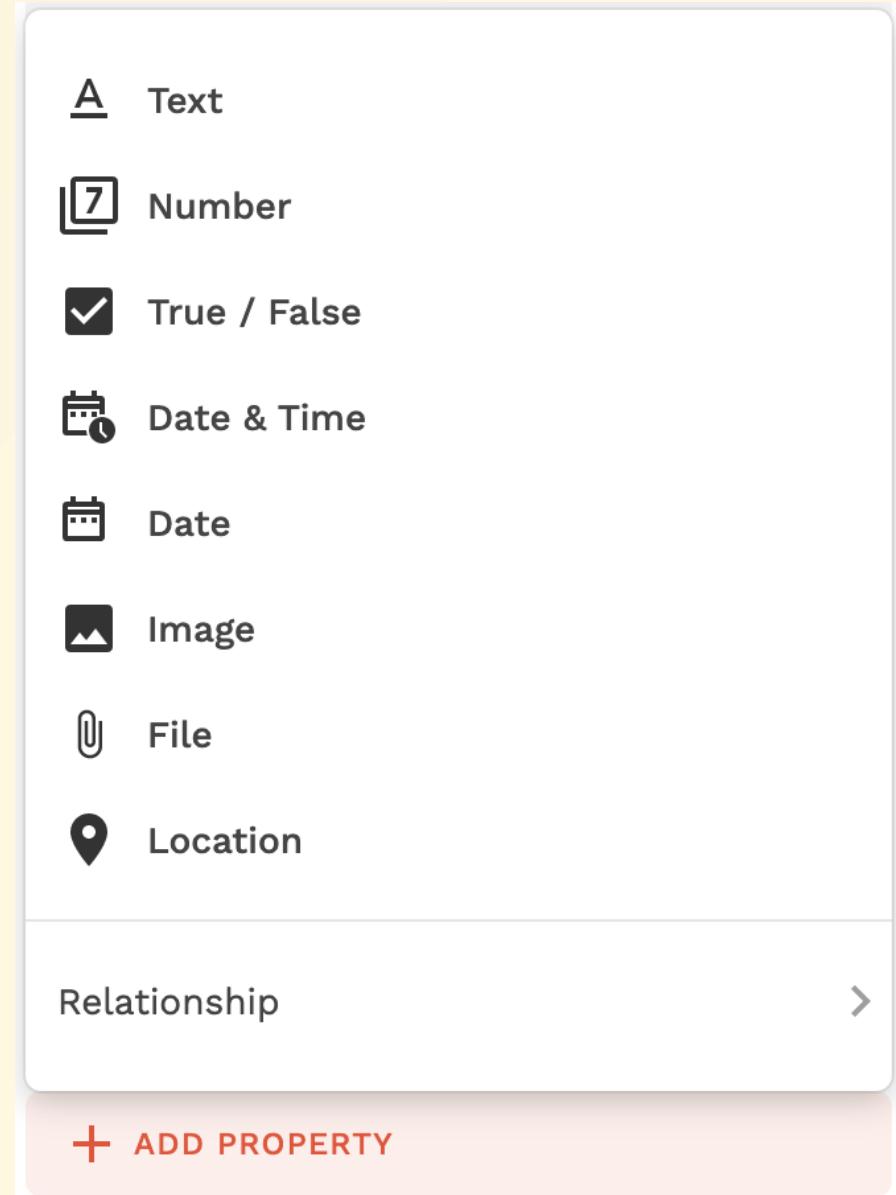
Propertyとは

- Propertyは、Recordを構成する一つ一つの項目です。
- Users Collectionは、Email、Password、用户名、氏名といったPropertyで構成されます。
- Propertyの値は空で登録される場合もあります。



Propertyがどのようなデータかを定義するため、Property追加時はTypeを選択します。

- Text
- Number
- True/False
- Date/Time
- Date
- Image
- File
- Location



Relationshipとは

- 1つのRecordに対して多数のPropertyを保存する代わりに、Relationshipと呼ばれる複数のCollectionを関連づけるための特別なPropertyを設定します。これにより、Collectionを扱いやすい形に分割することができます。

- 例えば、Chatアプリでユーザーが送信したメッセージはUsers Collectionとは別のMessages Collectionに保存され、これら2つのCollectionはRelationshipで関連づけられます。
 - Users側にはMessagesというRelationshipが、Messages側にはSenderという(Usersとの)Relationshipが登録されています。

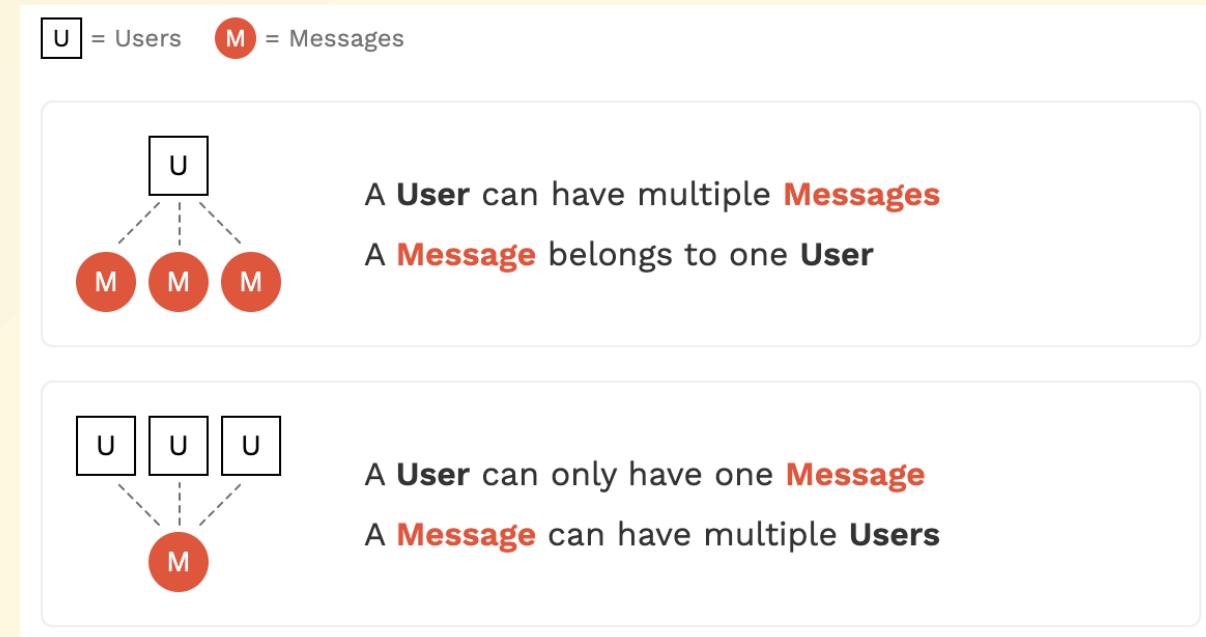
The screenshot displays two database collections: 'Users' and 'Messages'.
The 'Users' collection (left) has fields: Email, Password, Username, Full Name, and Profile Image. It shows 5 Records.
The 'Messages' collection (right) has fields: Text, Sender, and a note indicating it belongs to a relationship. It shows 15 Records.
A modal window is open between them, titled 'Messages'. It shows a 'Has Many Relationship' entry: 'One User, many Messages'. It also shows a 'Name' field set to 'Messages'. At the bottom are 'CANCEL' and 'SAVE' buttons.

Relationshipの種類

- AdaloのRelationshipでは、Collection間で紐付けられるRecordの数に応じて、1対多と多対多という2つの種類のいずれかを選択します。

1対多のRelationship

- 1つのRecordが、別のコレクションにある複数のRecordと関係を持つことを意味します。
- Relationshipを設定しようとしているCollectionを1と多のどちらにするかで、2種類の選択肢が現れます。



1対多のRelationshipの例

- Chatアプリでは1人のユーザーが複数のメッセージを送信しますが、メッセージの送信者は1人のユーザーなので、Users CollectionとMessages CollectionのRelationshipは1対多になります。

Users

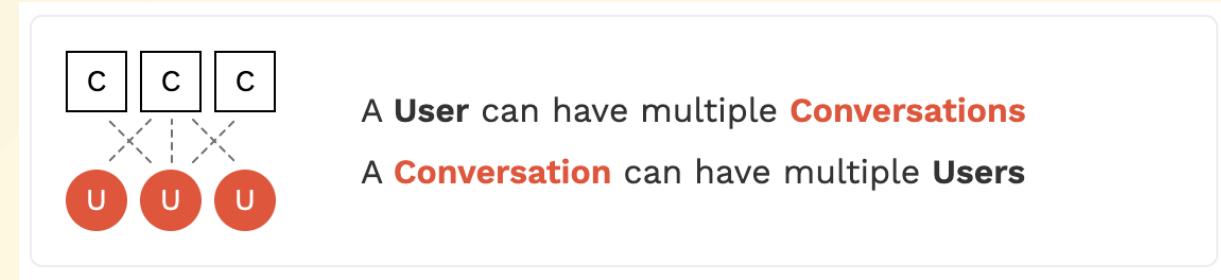
	Email	Password	Username	Full Name	Messages
<input type="checkbox"/>	imahashi@example.com	[hidden]		Ryo Imahashi	2 messages
<input type="checkbox"/>	veruca@email.com	[hidden]	veruca	Veruca Salt	2 messages
<input type="checkbox"/>	charlie@email.com	[hidden]	charlie	Charlie Bucket	2 messages
<input type="checkbox"/>	violet@email.com	[hidden]	sarah	Violet Beauregarde	1 message
<input type="checkbox"/>	will@email.com	[hidden]	will	Will W	8 messages

Messages

	Text	Sender	Conversation	Receiver	ID
<input type="checkbox"/>	Hey!	will@email.com	Will W and Ryo Imahashi		45
<input type="checkbox"/>	foo	imahashi@example.com	Violet Beauregarde and...		44
<input type="checkbox"/>	Hi!	imahashi@example.com	Will W and Ryo Imahashi	will@email.com	43
<input type="checkbox"/>	Sure thing Violet.	will@email.com	Violet Beauregarde and...		42
<input type="checkbox"/>	Hey Will! Before you st...	violet@email.com	Violet Beauregarde and...		41

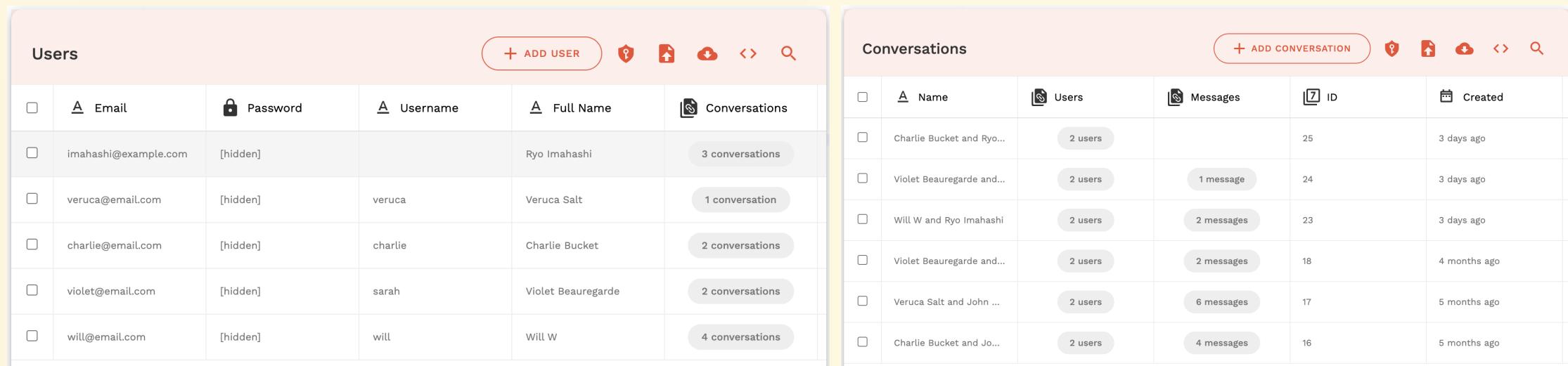
多対多のRelationship

- 両方のCollectionの1Recordが、もう一方のCollectionの複数のRecordに紐付けされることを意味します。



多対多のRelationshipの例

- Chatアプリでは1人のユーザーが複数の会話(誰とどんなメッセージをやりとりしたかを管理するもの)を持ち、1つの会話には複数のメンバー(ユーザー)が所属するので、Users CollectionとConversations CollectionのRelationshipは多対多になります。



Users					
	Email	Password	Username	Full Name	Conversations
<input type="checkbox"/>	imahashi@example.com	[hidden]		Ryo Imahashi	3 conversations
<input type="checkbox"/>	veruca@email.com	[hidden]	veruca	Veruca Salt	1 conversation
<input type="checkbox"/>	charlie@email.com	[hidden]	charlie	Charlie Bucket	2 conversations
<input type="checkbox"/>	violet@email.com	[hidden]	sarah	Violet Beauregarde	2 conversations
<input type="checkbox"/>	will@email.com	[hidden]	will	Will W	4 conversations

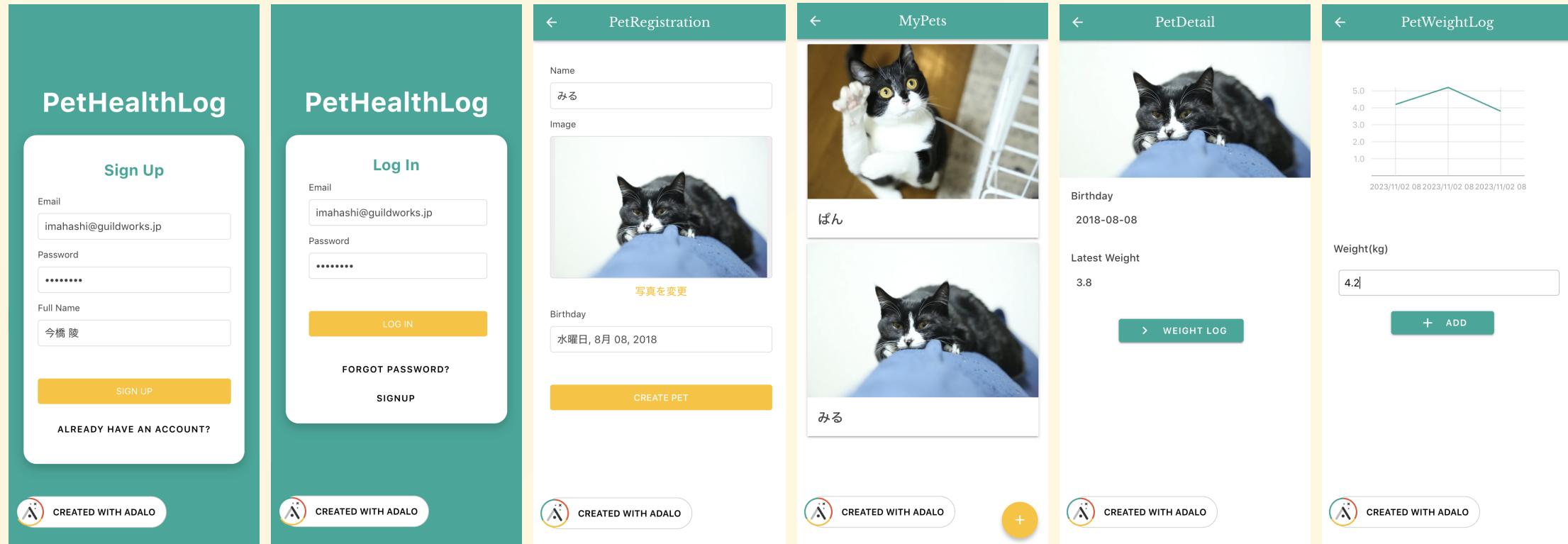
Conversations					
	Name	Users	Messages	ID	Created
<input type="checkbox"/>	Charlie Bucket and Ryo...	2 users		25	3 days ago
<input type="checkbox"/>	Violet Beauregarde and...	2 users	1 message	24	3 days ago
<input type="checkbox"/>	Will W and Ryo Imahashi	2 users	2 messages	23	3 days ago
<input type="checkbox"/>	Violet Beauregarde and...	2 users	2 messages	18	4 months ago
<input type="checkbox"/>	Veruca Salt and John ...	2 users	6 messages	17	5 months ago
<input type="checkbox"/>	Charlie Bucket and Jo...	2 users	4 messages	16	5 months ago

データベース設計

次に、実際にデータベースを設計しましょう。

データベースを設計してみよう

サンプルアプリのUIを見ながら、データベースを設計してみましょう。手順は次のページで紹介します。



データベース設計の手順

1. UIを見ながら、保存が必要になるデータをリストアップしましょう。テキストエディタ(メモ帳アプリ等)に書き起こしてください。
2. リストアップしたデータがどのようなCollectionに分類できるかを考えて、Adaloのデータベースに必要なCollectionを作成しましょう。
3. 1でリストアップしたデータを適切なCollectionにPropertyとして追加してください。その際、適切なTypeを選択してください。
4. 他のCollectionと関連を持つCollectionには、Relationship Propertyを設定しましょう。

※ 次のスライド以降に解説がありますが、答えを見る前に一度自分で手を動かして考えてみることをおすすめします。

※ 絶対的な正解はないです。悩んだら、直感に従って進めてみてください。

解説

UIを見ながら、保存が必要になるデータをリストアップすると、以下のようになりました。

- ユーザーのEmail
- ユーザーのパスワード
- ユーザーのFullName
- ペットの名前
- ペットの写真
- ペットの誕生日
- ペットの体重
- ペットの体重の登録日時

- その他のデータを挙げられた人がいれば、教えてください！

リストアップしたデータがどのようなCollectionに分類できるかを考えて、今回はこの3つに分類することにします。

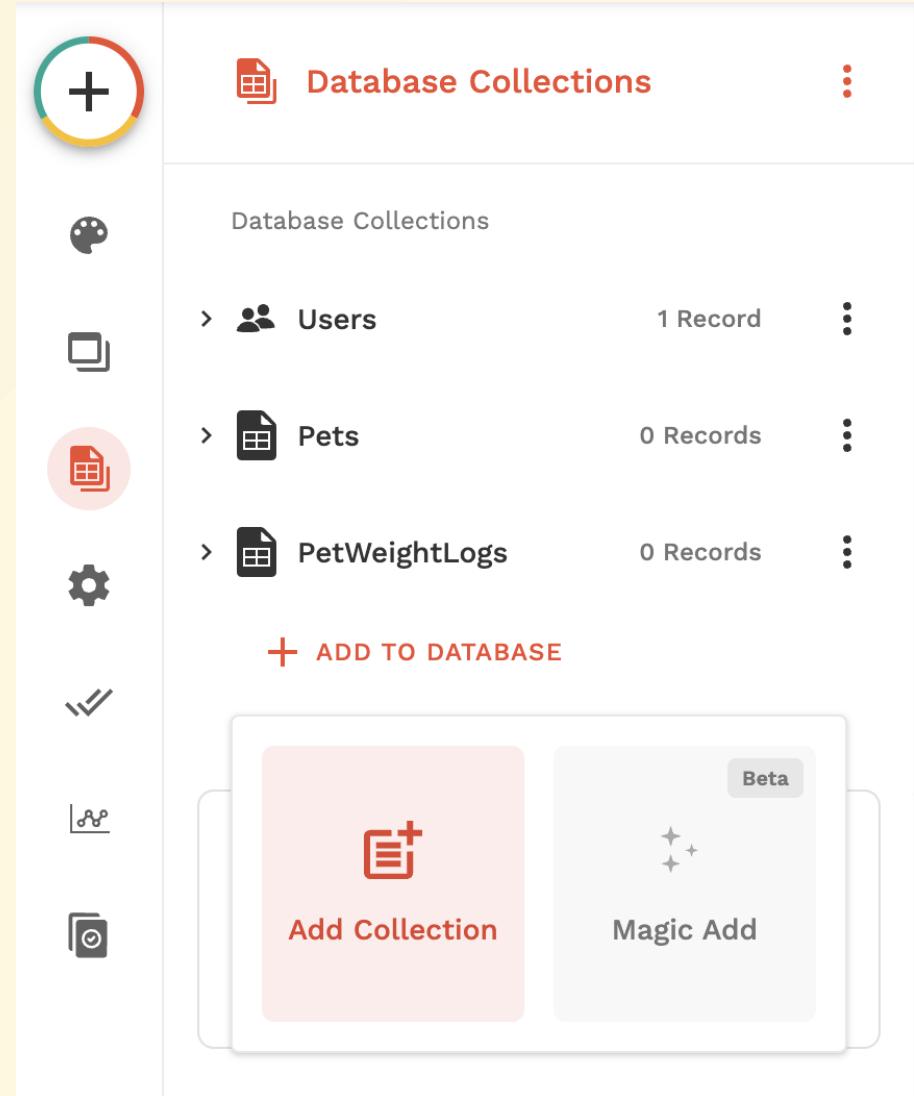
- Users
- Pets
- PetWeightLogs

- ユーザーのCollectionとペットのCollectionの2つを用意した人は多いのではないでしょうか？
- ペットの体重記録のCollectionは用意しなかった人もいるかもしれません。(ペットの体重とその登録日時をペットのCollectionに含める方法も間違いではありません。後ほど解説します。)
- その他のCollectionの分類をした人はいますか？

Collectionの分類に関する補足

- 「Aが決まればBが1つに決まる」という関係が成り立つ時、AをCollectionに、BをそのCollectionのPropertyにする場合が多いです。
 - ユーザーが決まれば、ユーザーのEmail、パスワード、FullNameがそれぞれ1つに決まります。
 - ペットが決まれば、ペットの名前、写真、誕生日がそれぞれ1つに決まります。
- 「Aに対してBが複数存在する」という関係が成り立つ時、AとBは別々のCollectionに分割することが多いです。
 - (1匹の)ペットに対して、ペットの体重とその登録日時は複数存在します。

- CollectionをAdaloのデータベースに登録しておきます。
 - ADD TO DATABASE > Add Collectionから手動で追加してください
 - Magic Add は追加したい機能をテキスト入力するとAIがデータベースの変更を提案してくれますが、今回は使いません
- Usersはデフォルトで作成されています。

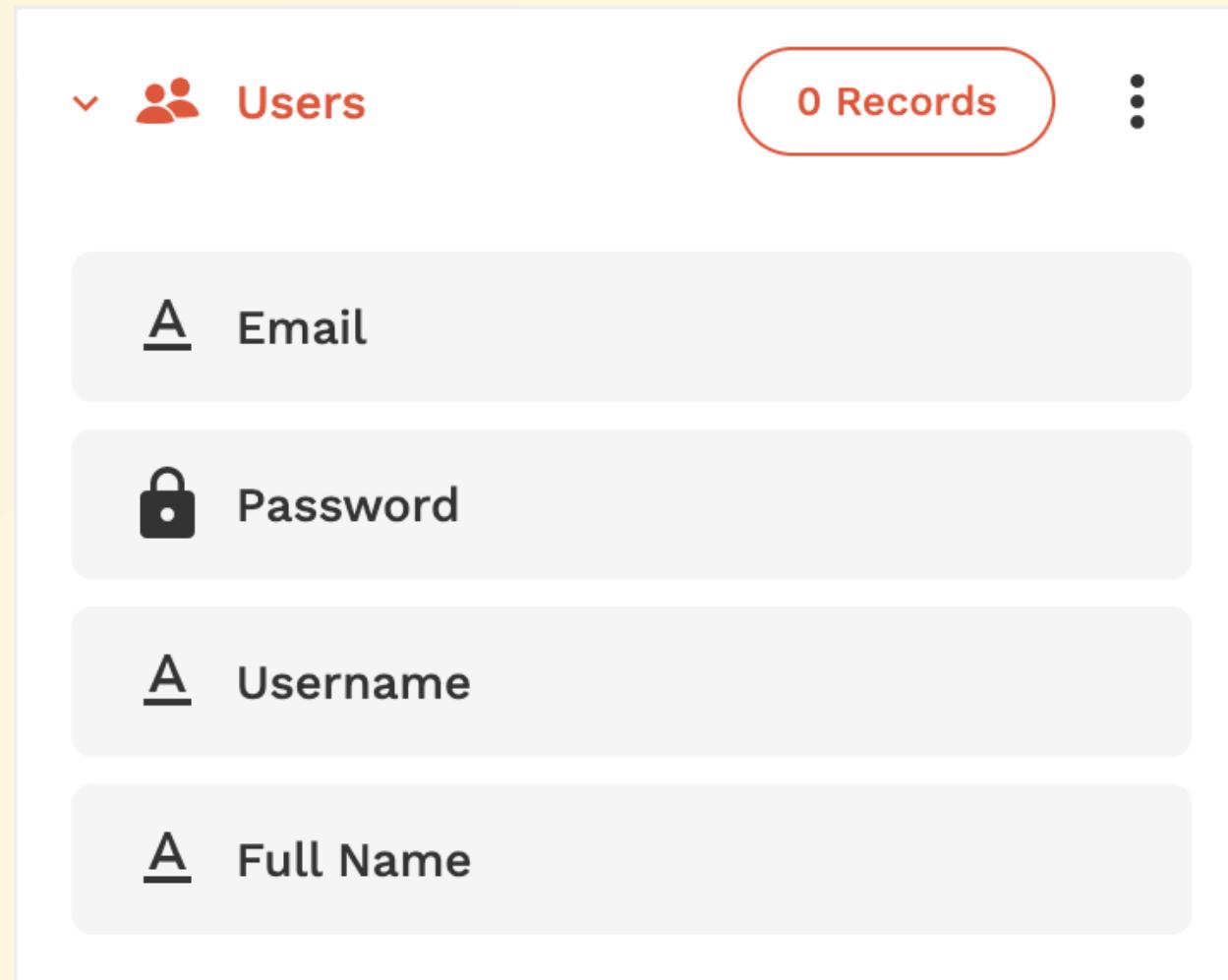


次に、ステップ1でリストアップしたデータを適切なCollectionの配下にPropertyとして追記すると、以下のようになりました。()の中は選択するTypeです。

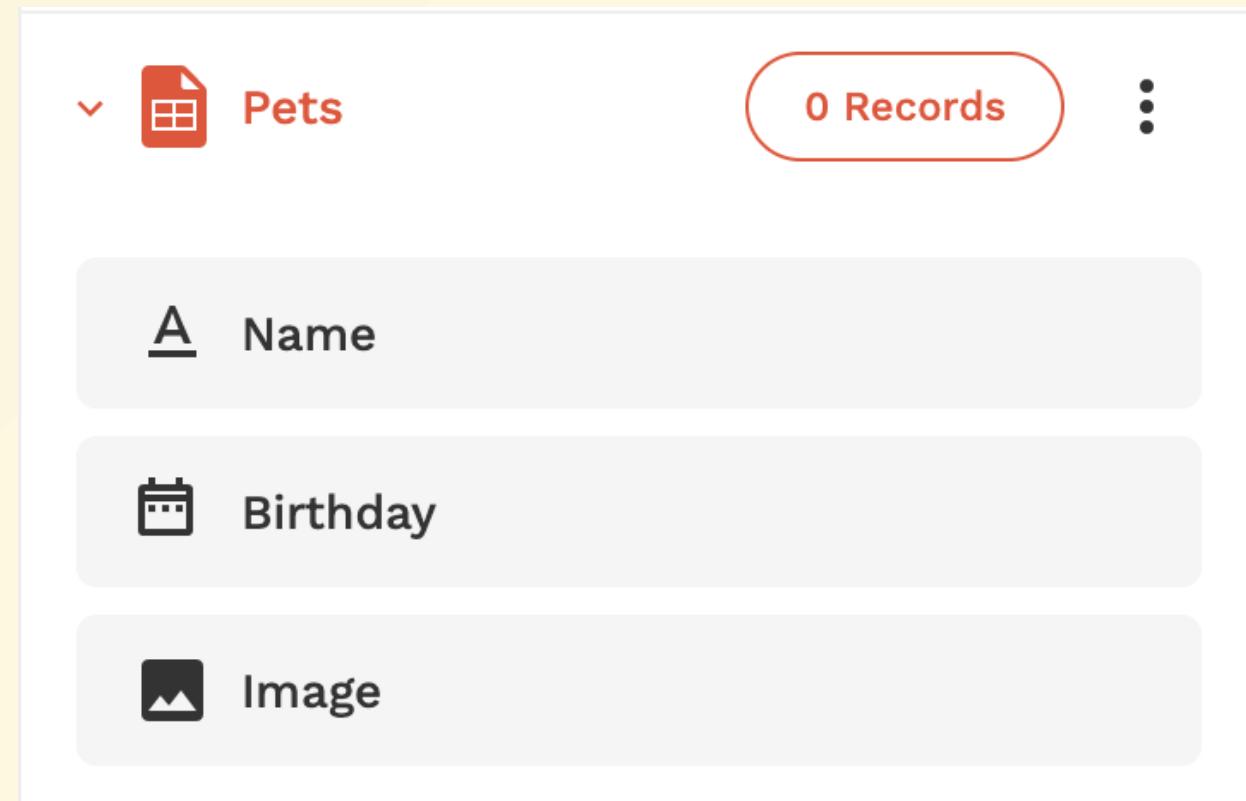
- Users
 - Email(Text)
 - Password(*Password)
 - FullName(Text)
- Pets
 - Name(Text)
 - Image(Image)
 - Birthday(Date)
- PetWeightLogs
 - WeightKg(Number)
 - RegisteredTime(Date&Time)

* Passwordはデフォルトで設定される特殊なTypeです。

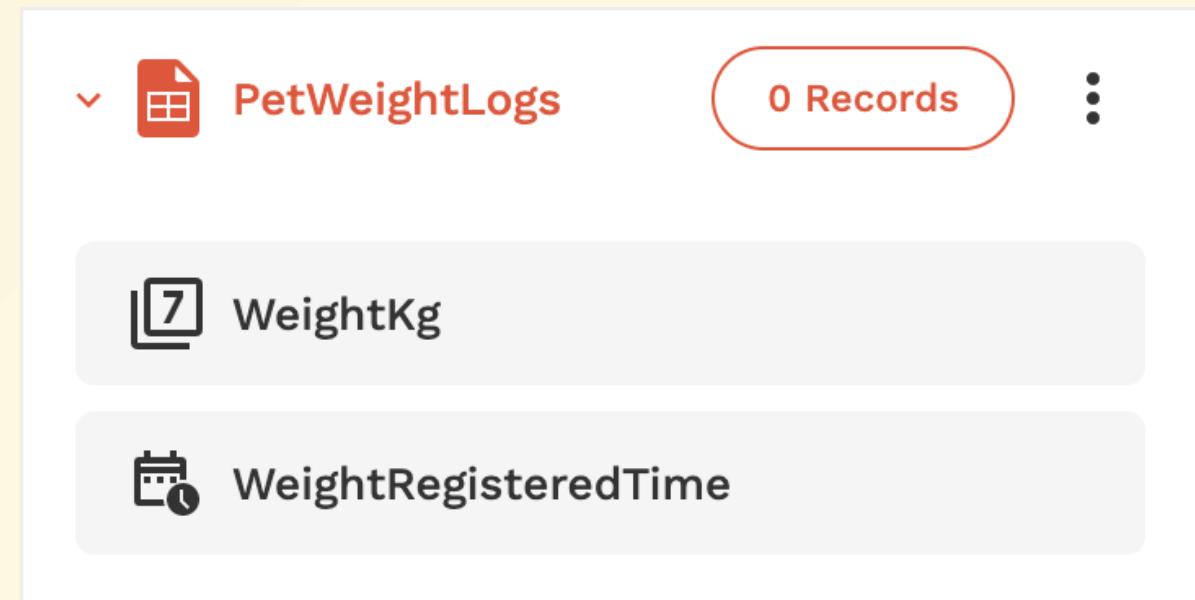
- Adaloで実際にPropertyを追加します。
- Users Collectionはデフォルトで設定済みで、必要な項目は含まれています。
- Usernameは不要ですが、削除できないのでそのままにしておきます。



- Pets CollectionのPropertyはこのようになります。

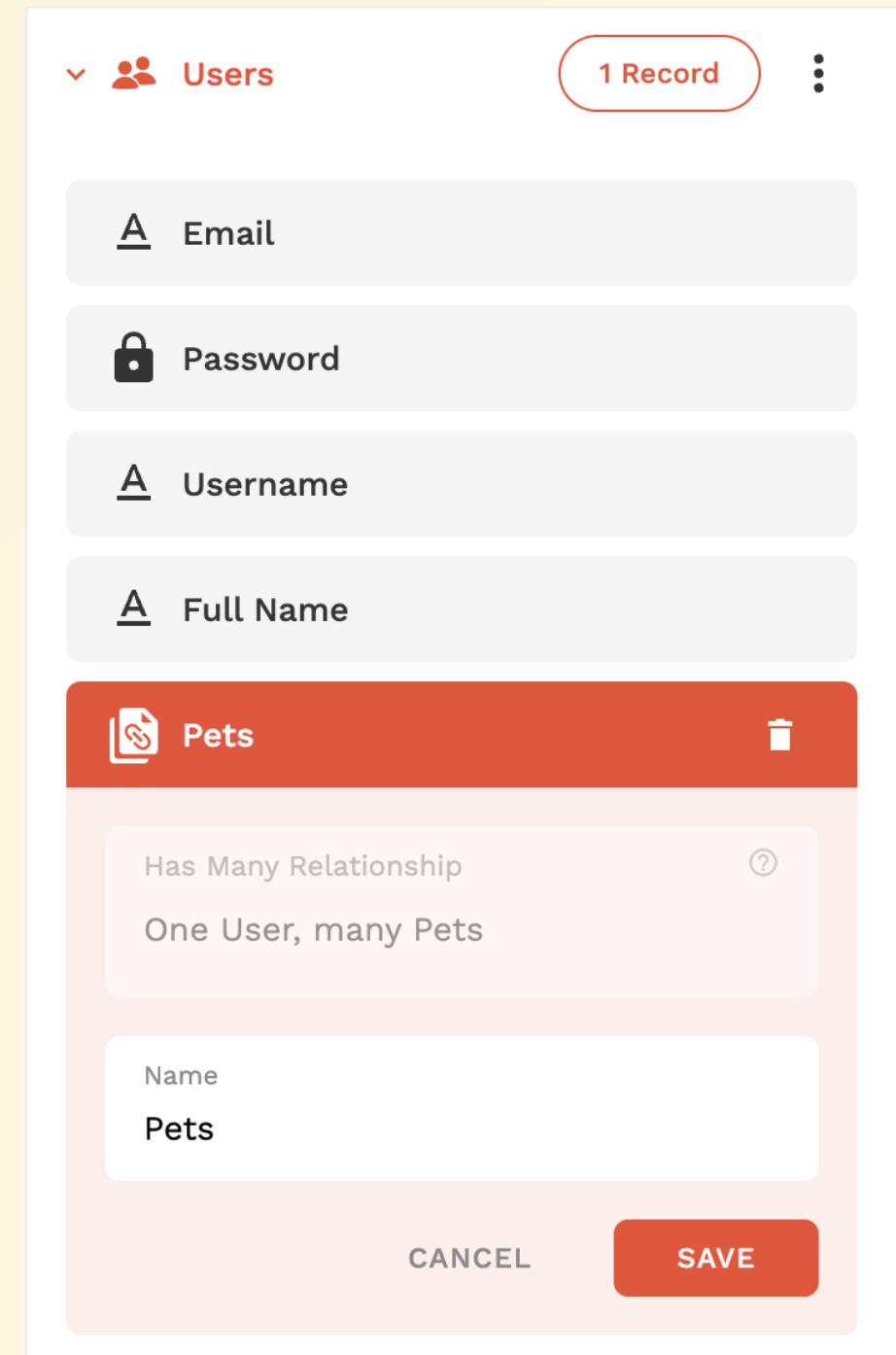
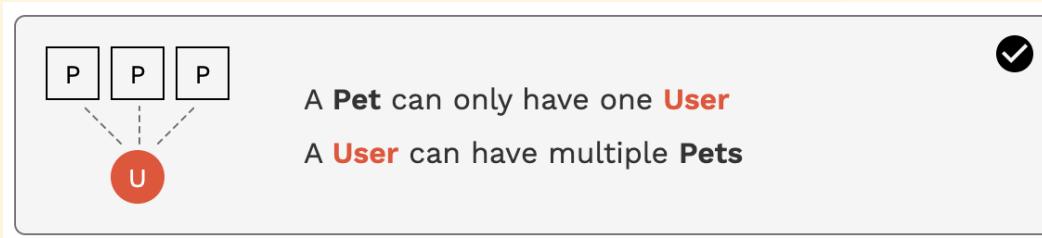


- PetWeightLogs Collection のPropertyはこのようになります。
- Collection追加時にデフォルトで設定されるName Propertyは不要なので、削除します。
 - ドラッグアンドドロップで順番がCollection内の一一番上でなくなるように移動すれば、削除できます。

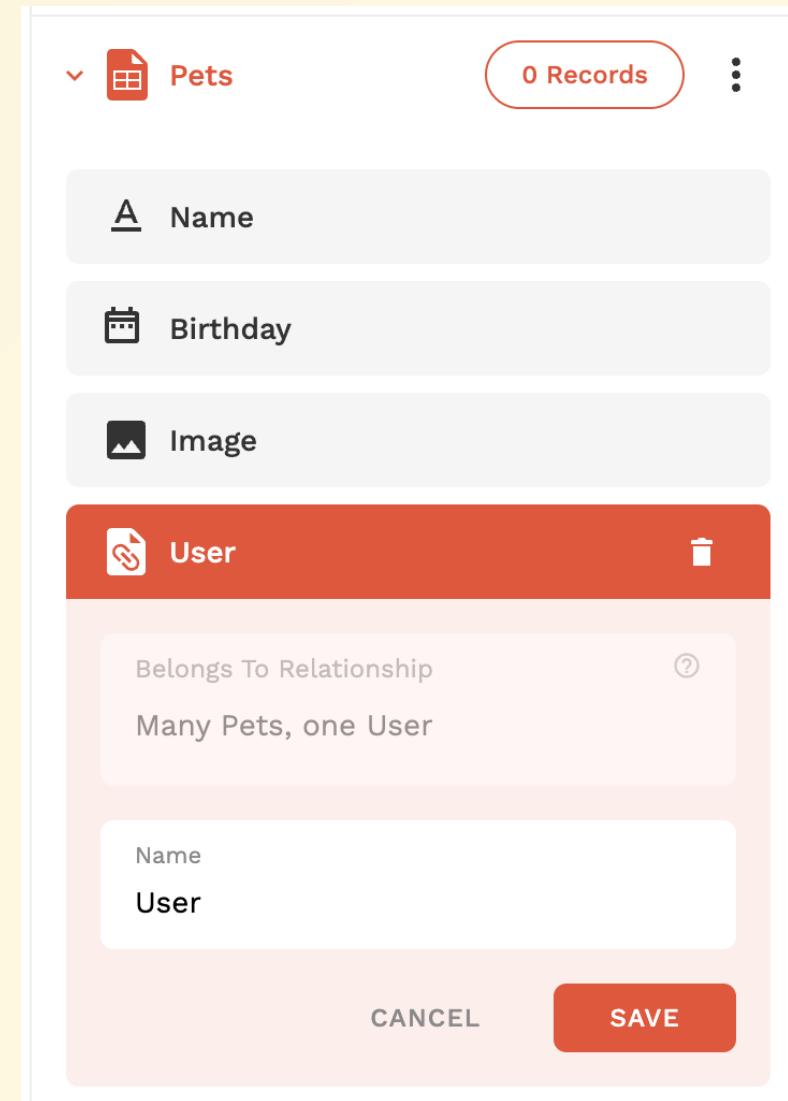


最後に、他のCollectionと関連を持つCollectionには、Relationship Propertyを設定します。

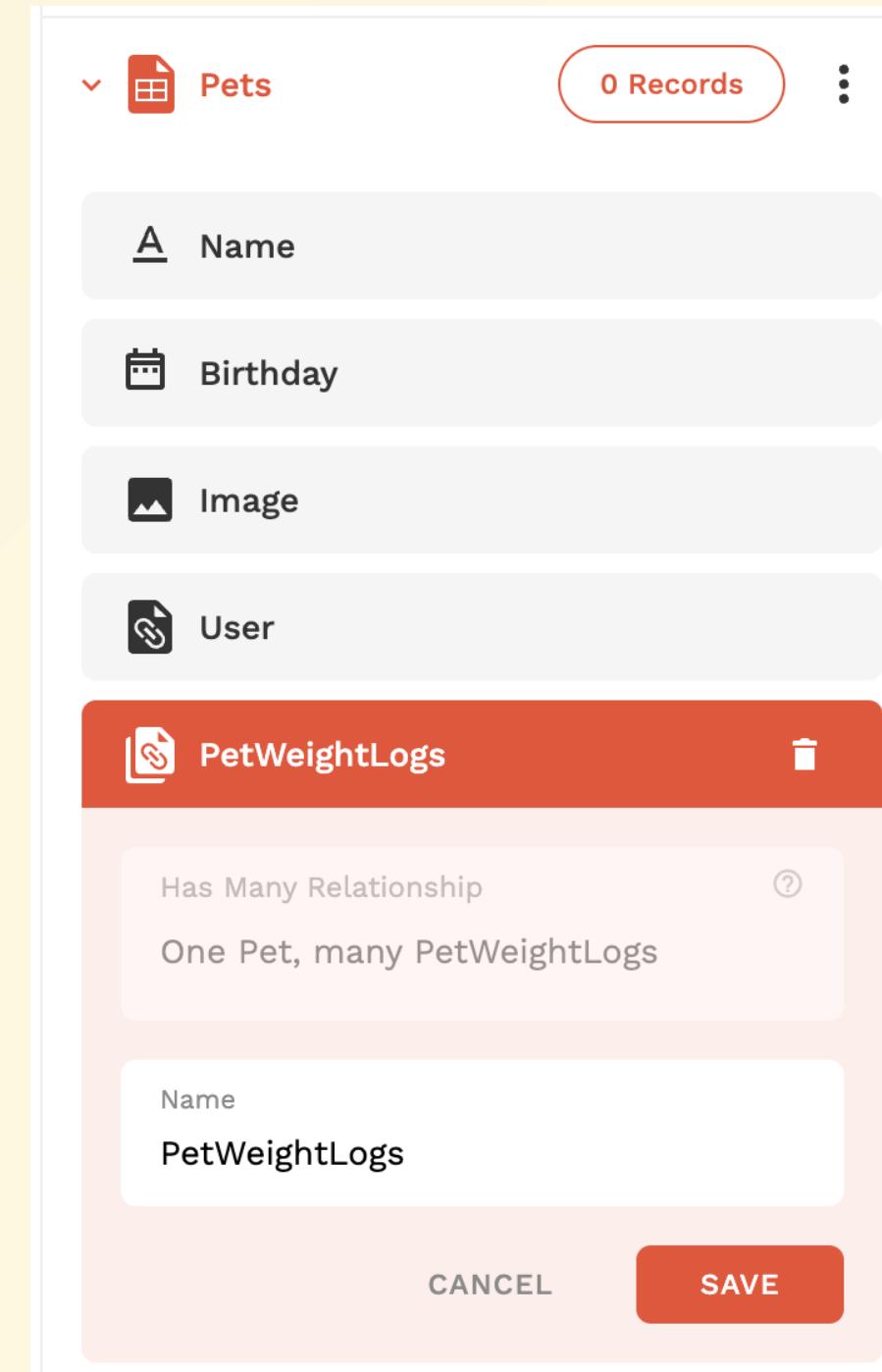
- Users Collectionを選択して、Pet Collectionとの1対多のRelationshipを追加します。



- Pets Collectionを確認すると、Users Collection側でRelationshipの設定をしたので、自動でUsers CollectionとのRelationshipが追加されています。
 - Users Collection側が1なので、末尾のsが省略されて、UserというProperty名になっています。

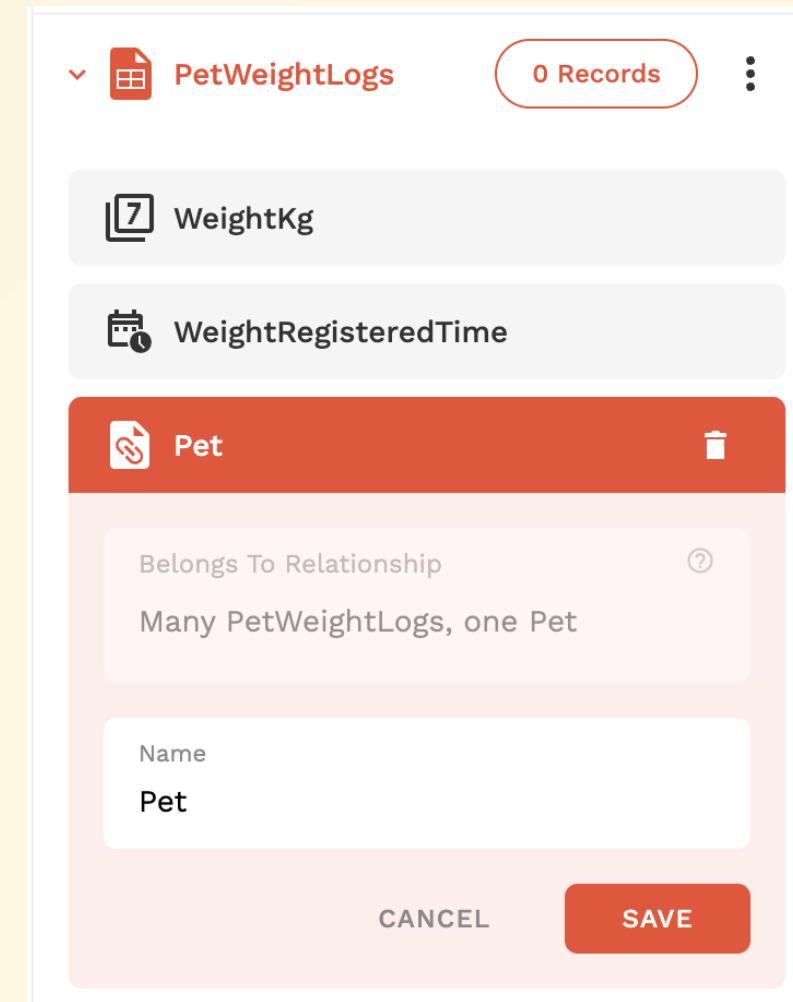


- Pets Collectionに、PetWeightLogs CollectionとのRelationshipを追加します。
 - Pets Collectionを選択して、PetWeightLogs Collectionとの1対多のRelationshipを追加します。



PetWeightLogs Collectionを確認すると、Pets Collection側でRelationshipの設定をしたので、自動でPets CollectionとのRelationshipが追加されています。

- Pets Collection側が1なので、末尾のsが省略されて、PetというProperty名になっています。



参考: Pets Collectionにペットの体重とその登録日時を含めた場合どうなるか

以下のようにレコードが登録されますが、この場合、少し困ることが出てきます。

Pets						
	Name	Image	Birthday	Weight(kg)	WeightRegisteredTime	Created
<input type="checkbox"/>	みる		8/8/2018	5.2	November 3, 2021 12:00 AM	a few seconds ago
<input type="checkbox"/>	みる		8/8/2018	5.1	November 2, 2021 12:00 AM	a minute ago
<input type="checkbox"/>	みる		8/8/2018	5	November 1, 2021 12:00 AM	a minute ago

困ること

- 1匹のペットに対して異なるペットの体重とその登録日時が結合されたRecordが複数登録されるため、ペットの情報(Name,Image,Birthday)が重複して登録されてしまう。
 - 1匹のペットの情報を変更するために、重複して登録されたRecordを全て更新しないといけなくなり、処理が複雑になる。
- Adaloには一つのCollectionを選んでそこにRecordを登録するためのフォームを自動生成する便利な機能があるが、データを登録する単位でCollectionが分かれていないので、それが使えなくなる。

データベース設計は以上です。

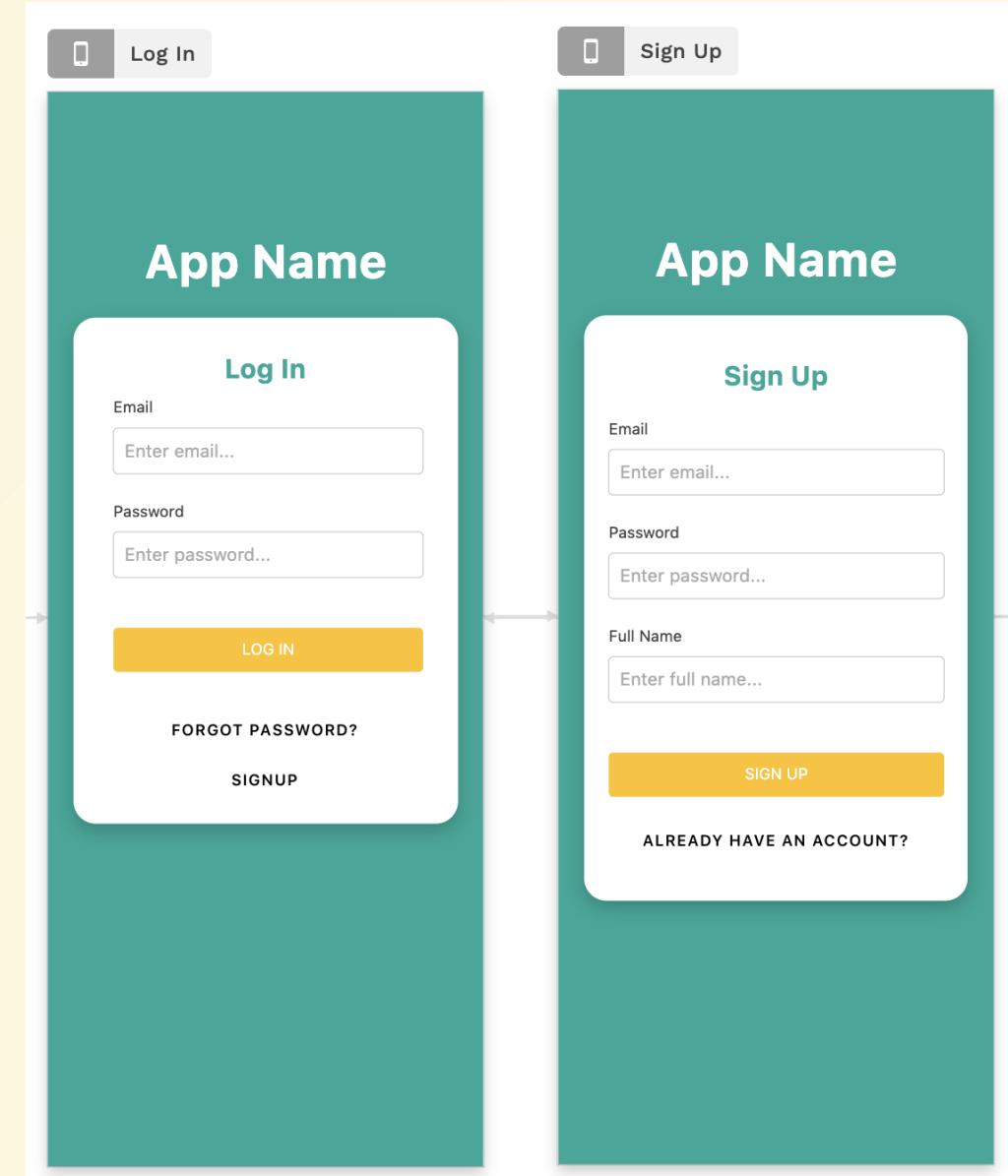
※ この後の作業で混乱しないために、Adaloのデータベースを資料と同じ状態に設定しておくことをおすすめします。

データベースを使ってアプリを開発しよう

設計したデータベースを使って、サンプルアプリを開発していきましょう。

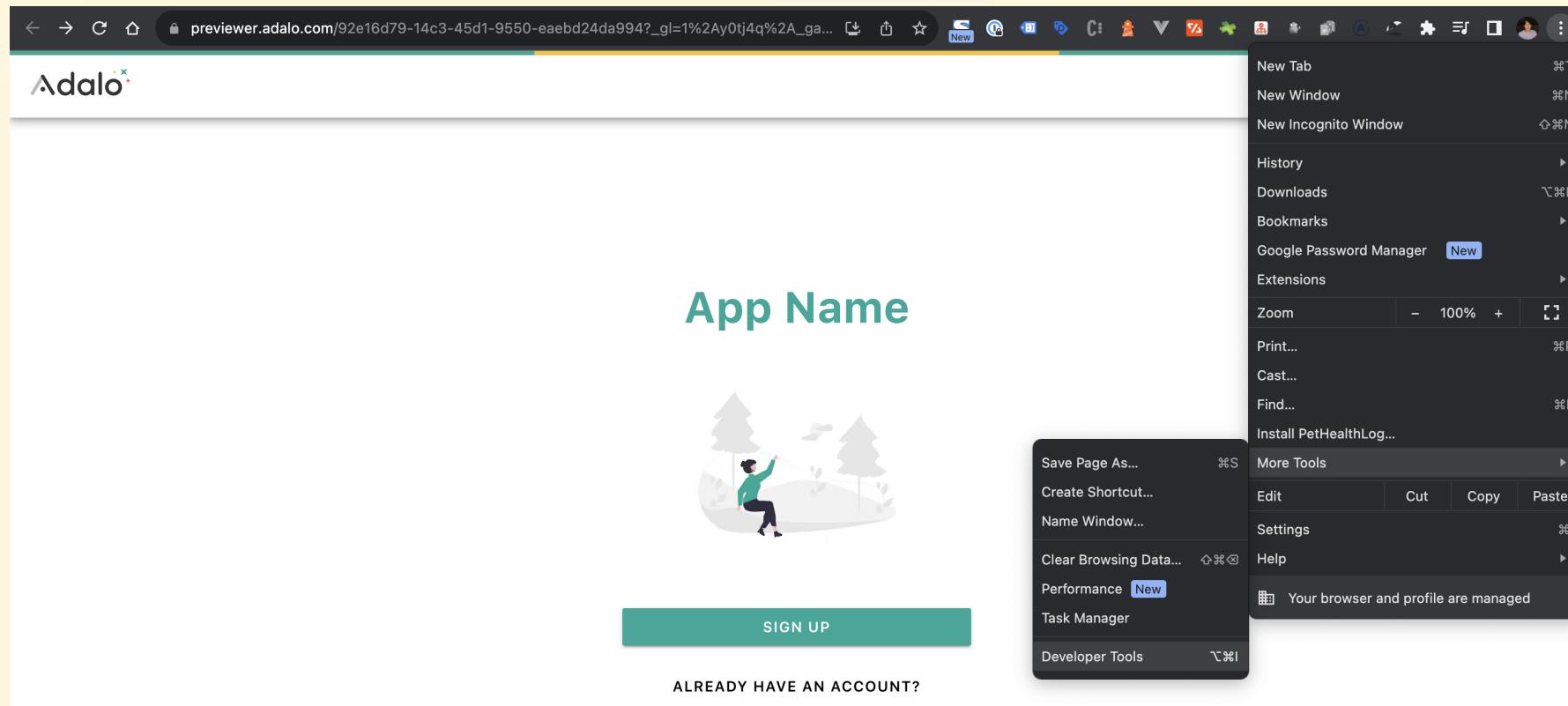
会員登録画面、ログイン画面

- ✓ 会員登録画面、ログイン画面はデフォルトで生成されるようになっています。
- 画面右上のPREVIEWボタンをクリックして、プレビュー機能で動作を確認してみましょう。



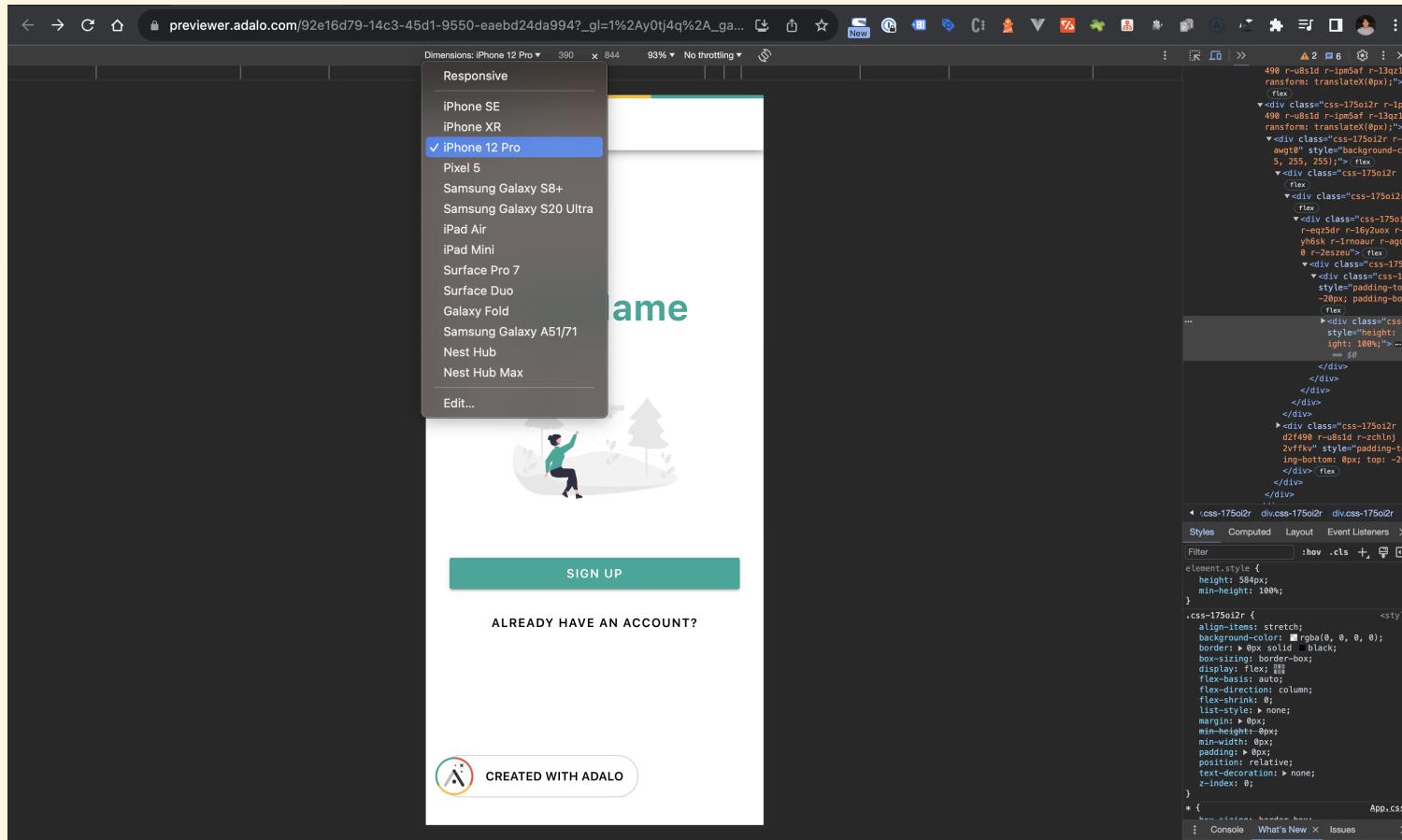
Mobile Firstでアプリを作っていくため、ここからはブラウザの開発者ツールを使い、Mobileサイズでプレビューをします。

- Chromeの場合、以下の手順で開発者ツールを開きます
 - 右上の三点リーダ > More Tools > Developer Tools



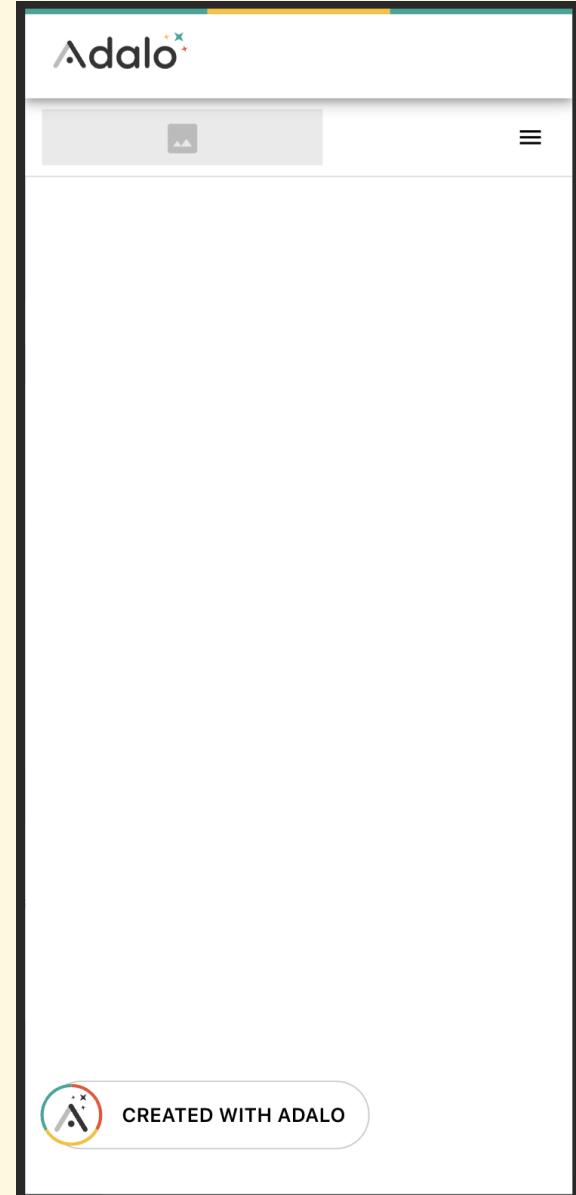
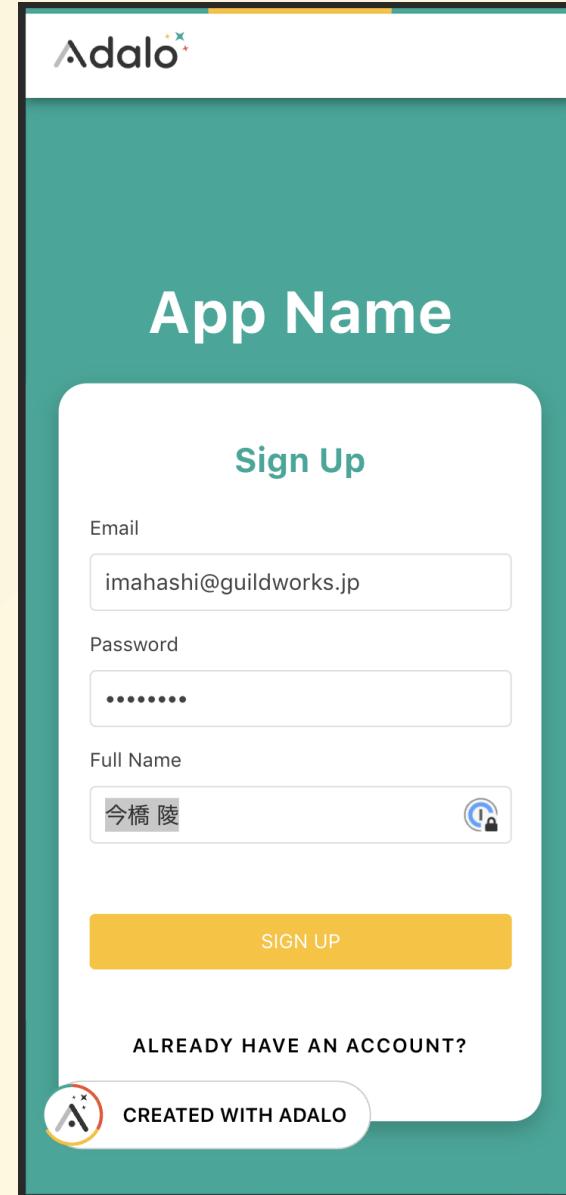
Chromeの開発者ツールでは、様々なデバイスの画面サイズで表示を確認することができます。

- iPhone 12 Proを選択してプレビューをします

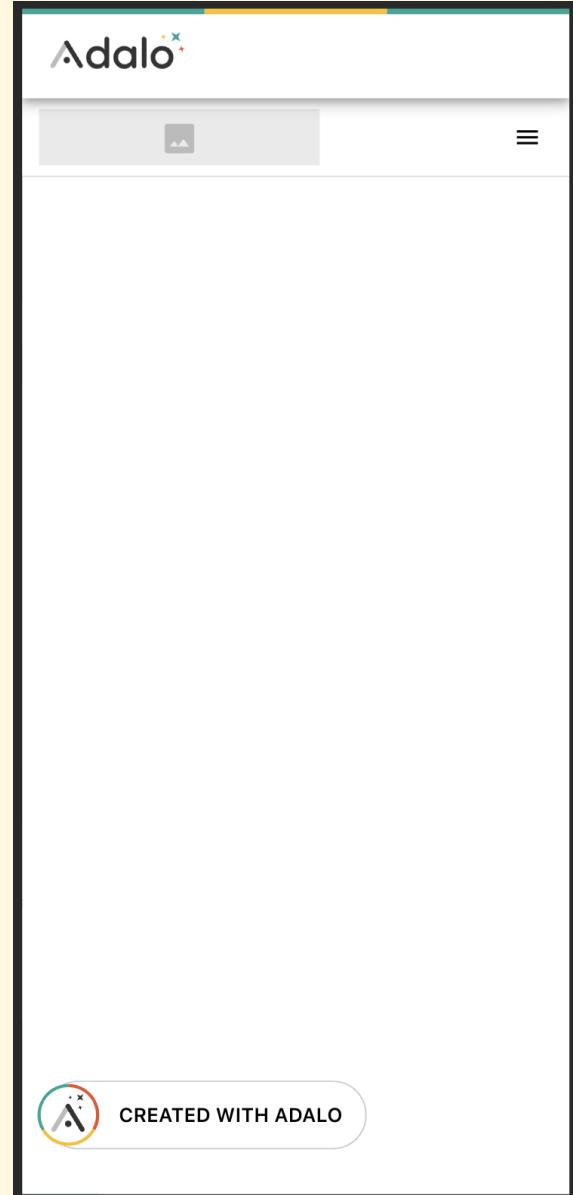
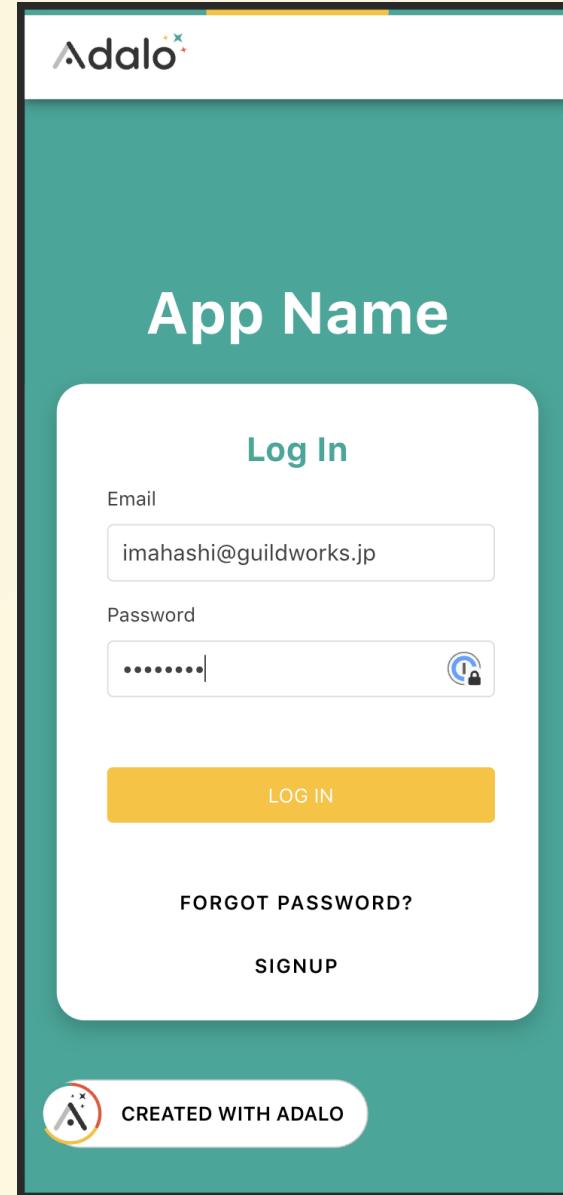


- 会員登録画面

- 会員登録をすると、Home画面に遷移します
- ブラウザの戻るボタン(←)で戻り、ALREADY HAVE AN ACCOUNT? をクリックしてログイン画面に遷移しましょう

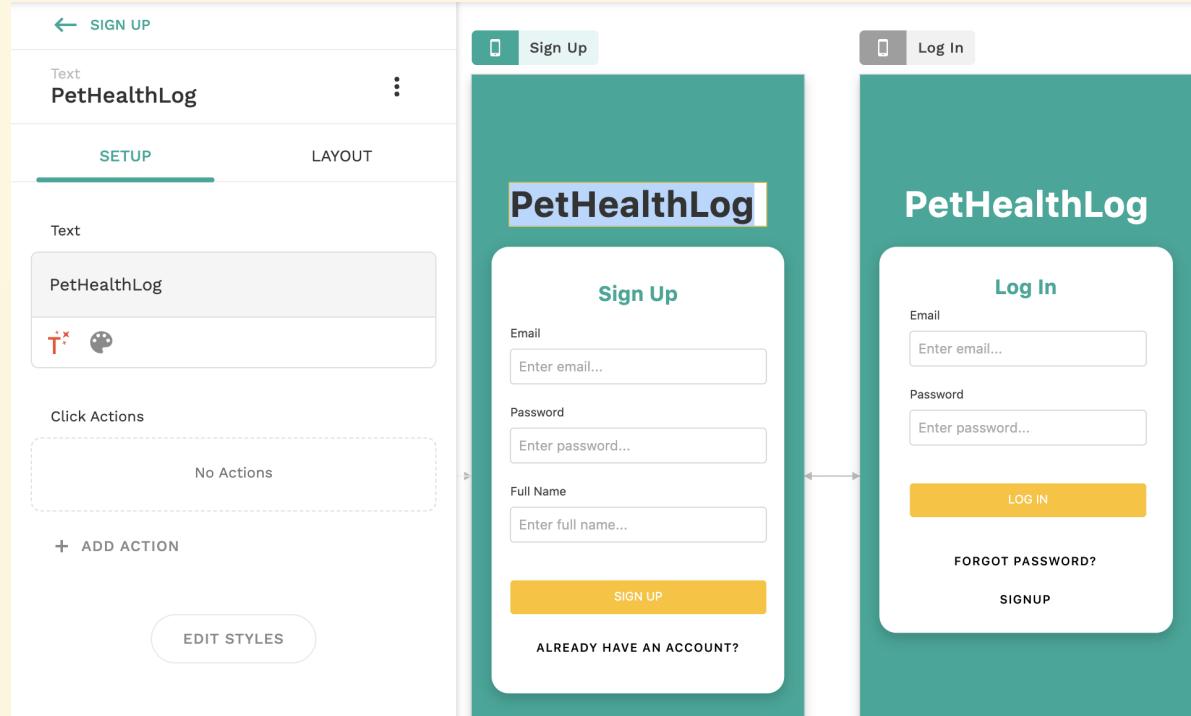


- ログイン画面
 - 先程会員登録したのと同じEmail、Passwordでログインすると、Home画面に遷移します



会員登録画面、ログイン画面はそのまま使えることがわかりました。

- App Nameと表示されている部分を、自分が決めたアプリ名に変更しておきましょう。

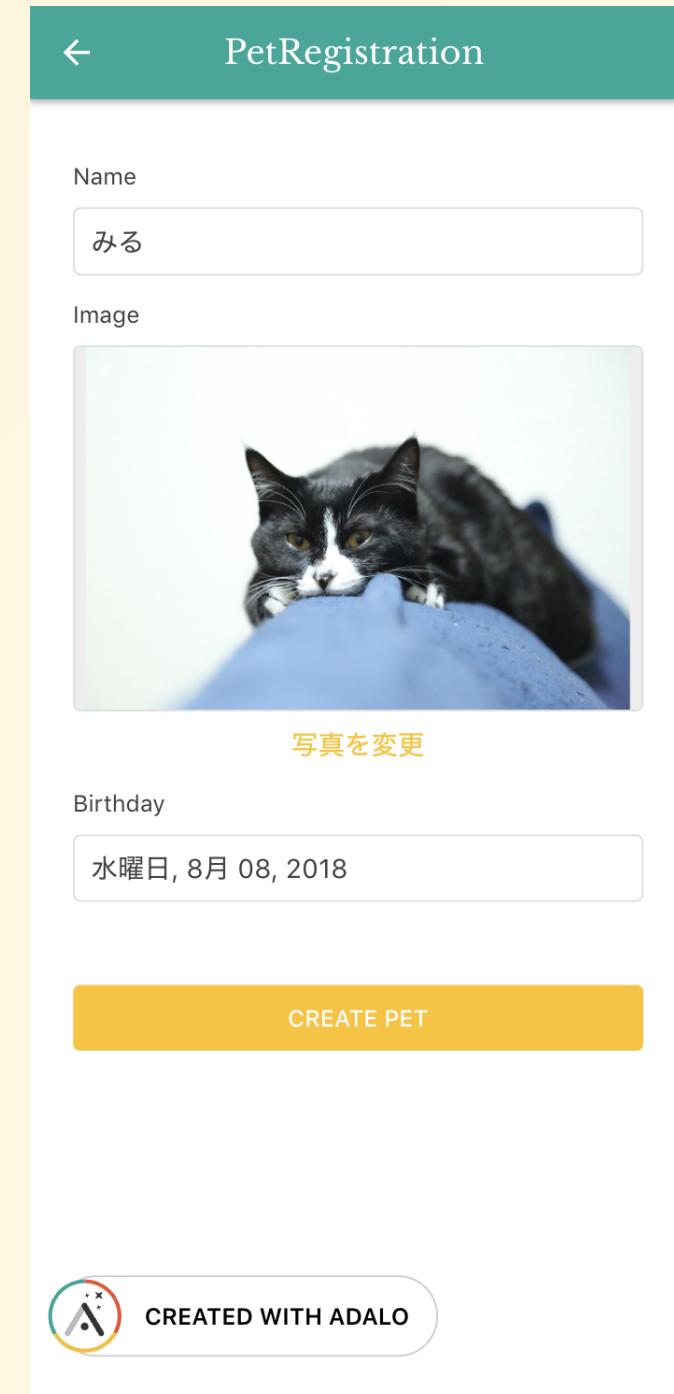


続いて、その他の4画面を作成していきましょう。

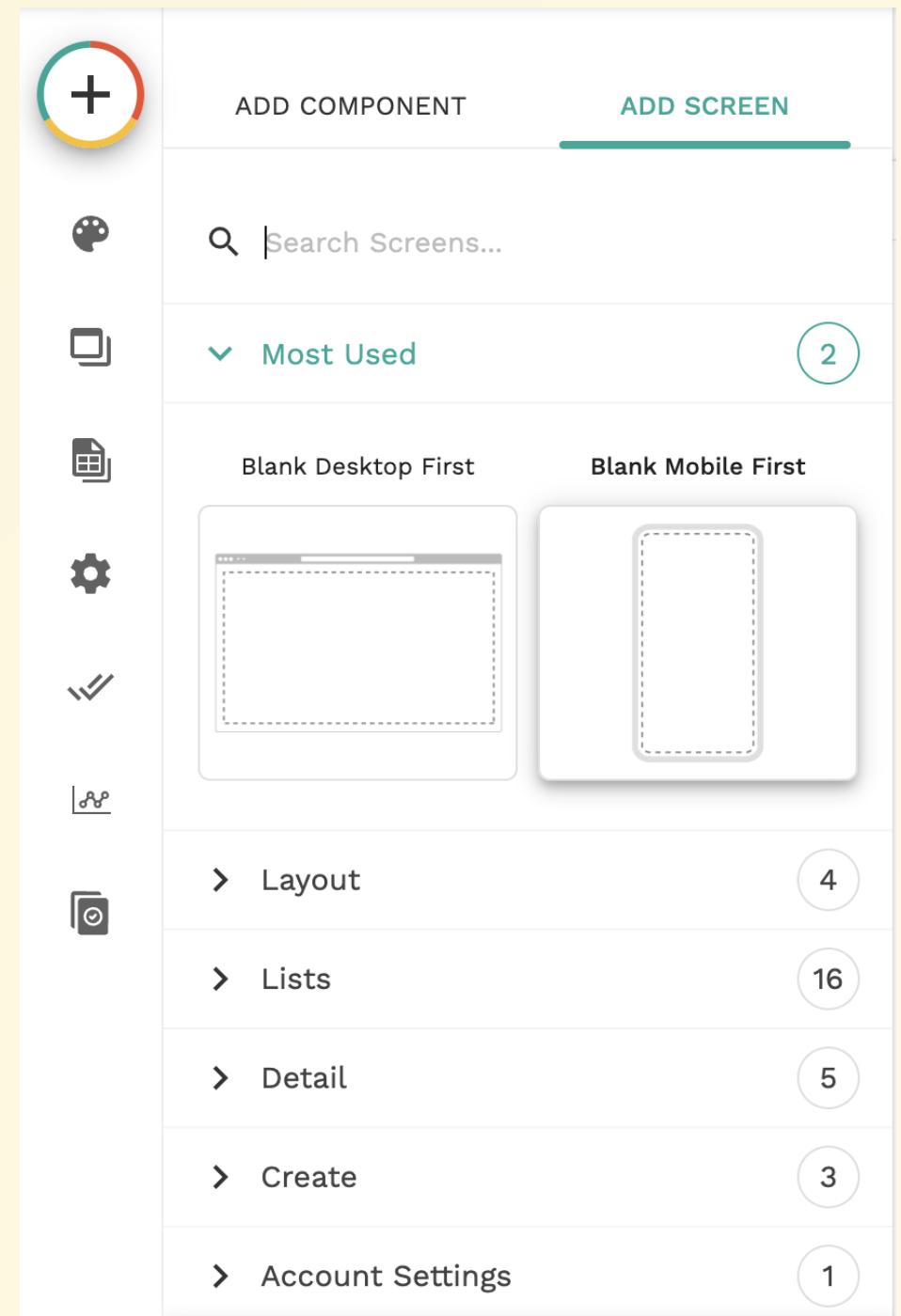
ペット登録画面

- 名前を入力できる
- 写真を選択できる
- 誕生日を入力できる
- 登録ボタンで確定し、ペツト一覧画面に遷移できる

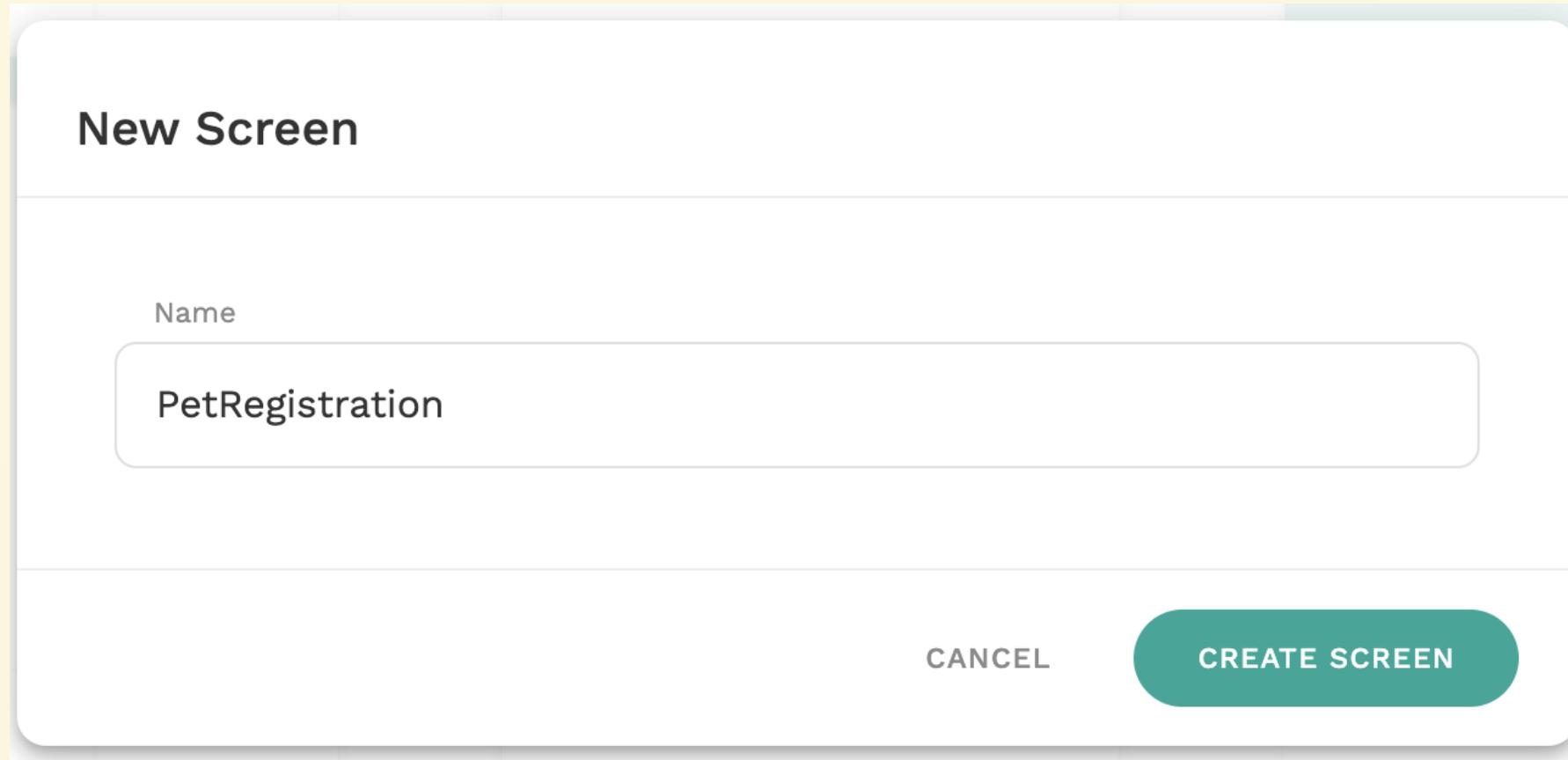
こちらの画面を作りましょう



- ADD SCREENからBlank Mobile Firstを選択します

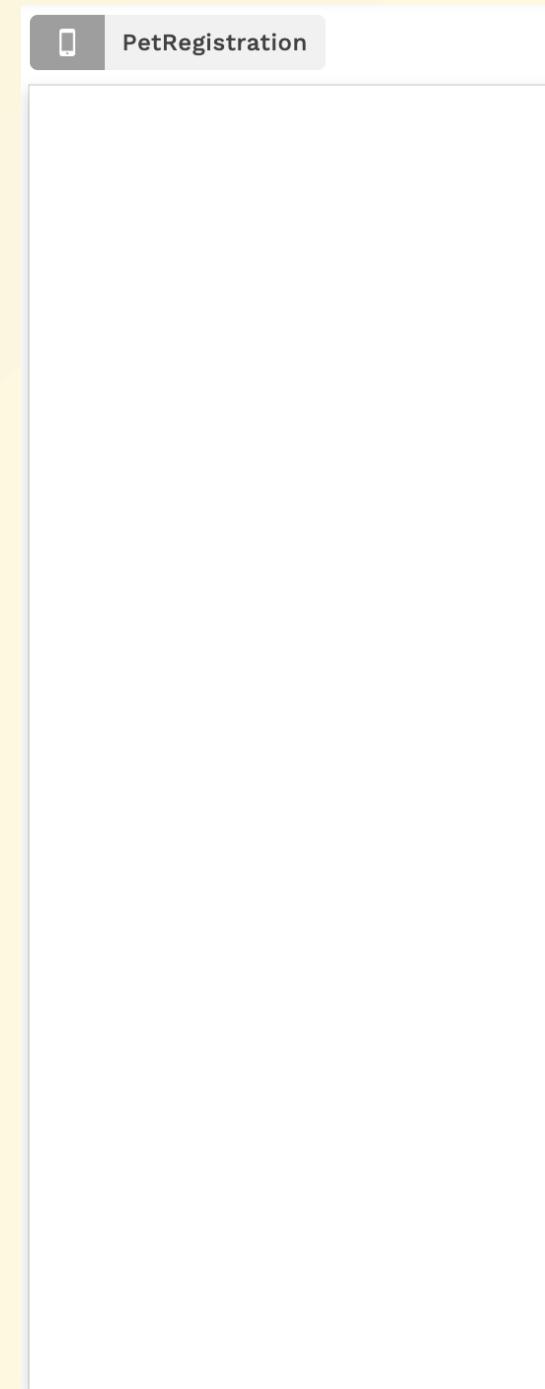


- Screen Nameを入力します

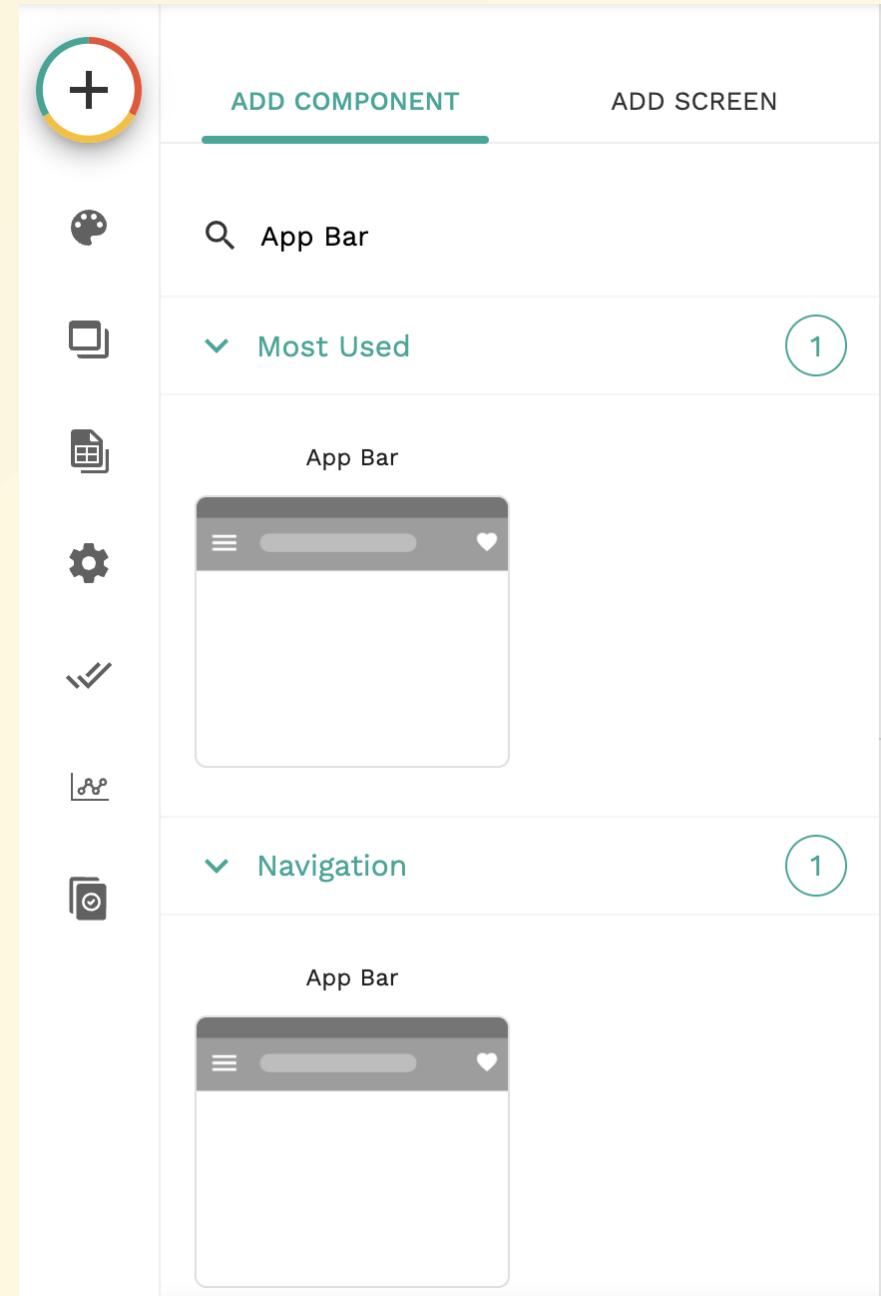


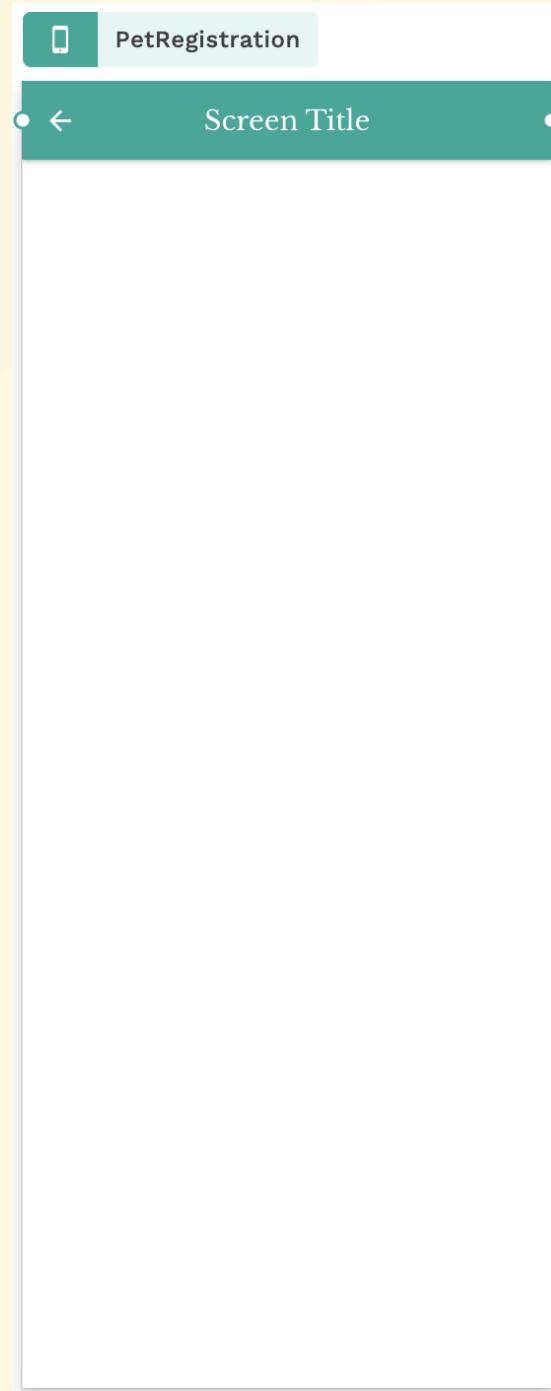
Screenが追加されました。

この上に必要なコンポーネントを追加していきます。



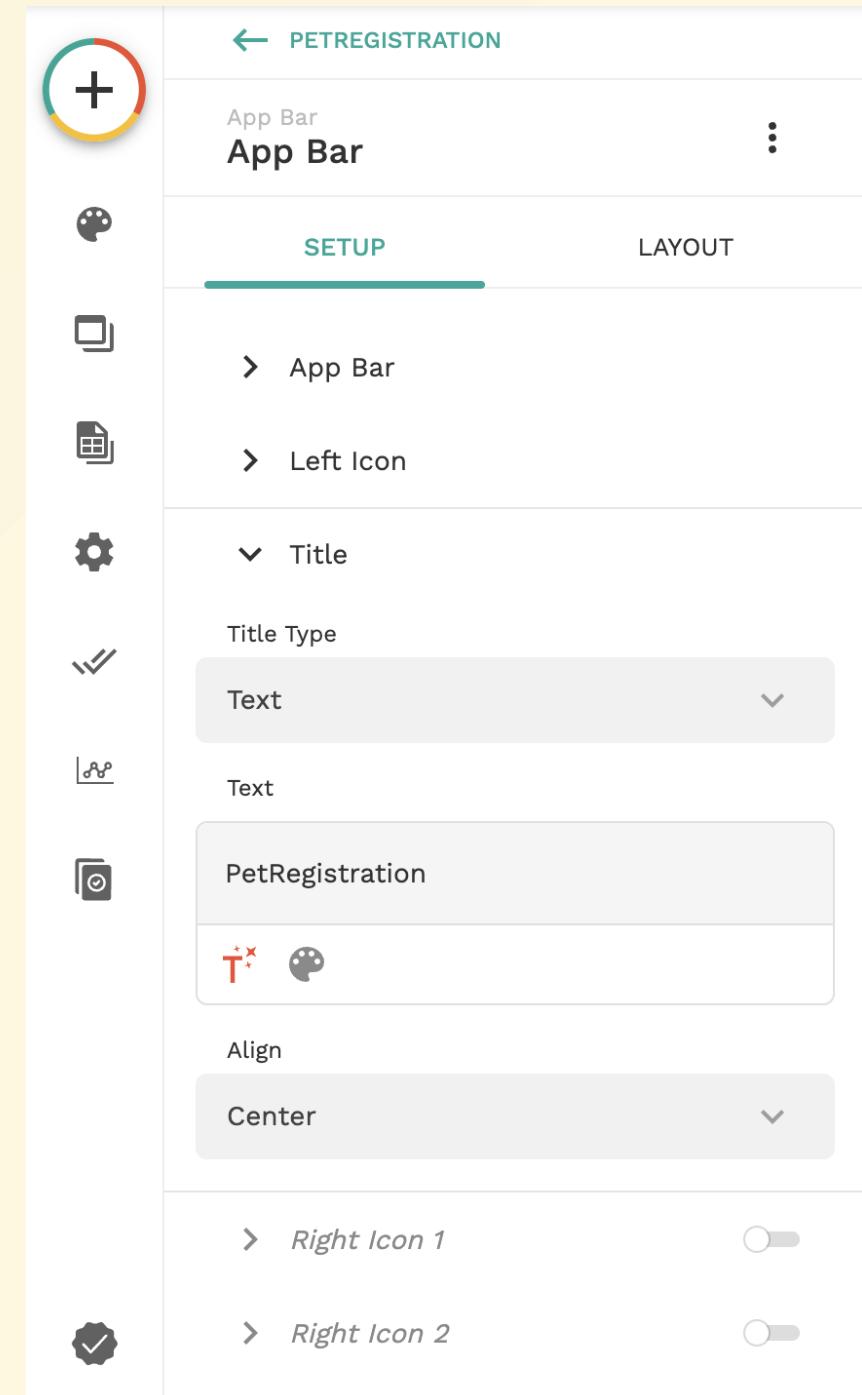
- ADD COMPONENTからApp Barを選択します



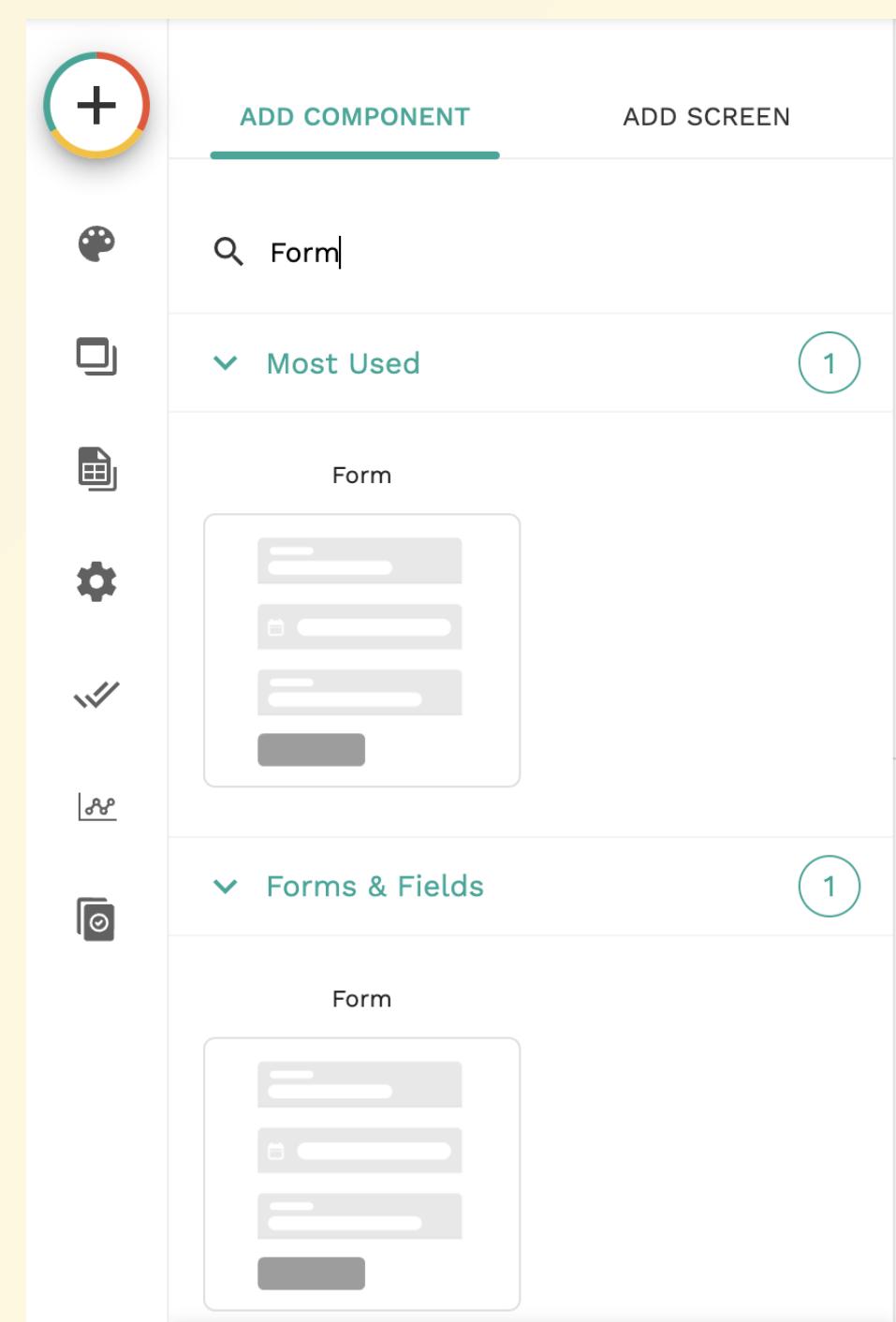


- 画面上に配置します

- Title > Textの値を画面名(PetRegistration)に変えます



- ADD COMPONENTからFormを選択します



- 画面上に配置します

A wireframe diagram of a mobile application screen titled "PetRegistration". The screen has a teal header bar with a back arrow and the title "PetRegistration". Below the header are three input fields labeled "Field 1", "Field 2", and "Field 3", each consisting of two rows. At the bottom is a large yellow "SAVE" button.

PetRegistration

← PetRegistration

Field 1

Field 2

Field 3

SAVE

- Which data collection?で Petsを選択します
 - 選択したCollectionに合わせたフォームが自動生成されます
- What do you want the form to do?でCreate New petを選択します

PETREGISTRATION

Form
Form

SETUP LAYOUT

Form

Which data collection?

Pets

What do you want the form to do?

Create New pet

Name
Enter name...

Birthday
Thursday, May 4th, 2019

Image

Choose Photo

CREATE PET

- FieldsのBirthdayとImageの順番をドラッグ&ドロップで入れ替えます

The screenshot illustrates the configuration of a form titled "PetRegistration".

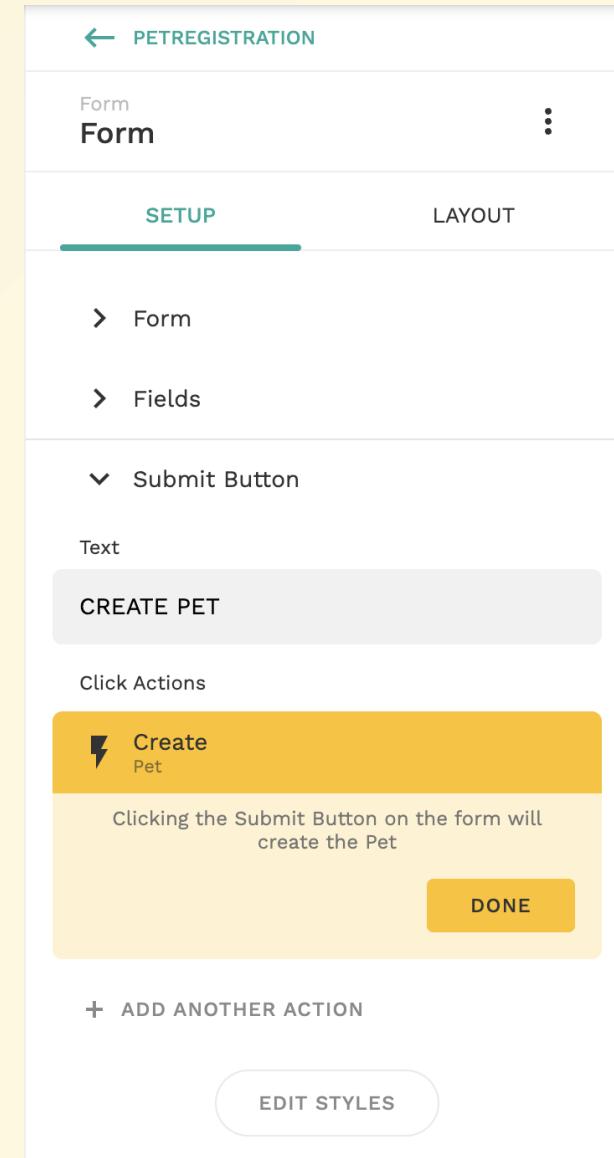
Left Panel (Setup):

- The title bar shows "PETREGISTRATION" and "Form".
- The "SETUP" tab is selected, while "LAYOUT" is visible.
- The "Fields" section is expanded, showing the current order of fields: "Name", "Image", and "Birthday".
- Below the fields, there are buttons for "ADD VISIBLE FIELD" and "ADD AUTOMATIC FIELD".
- The "User" field is set to "Logged In User".

Right Panel (Mobile View):

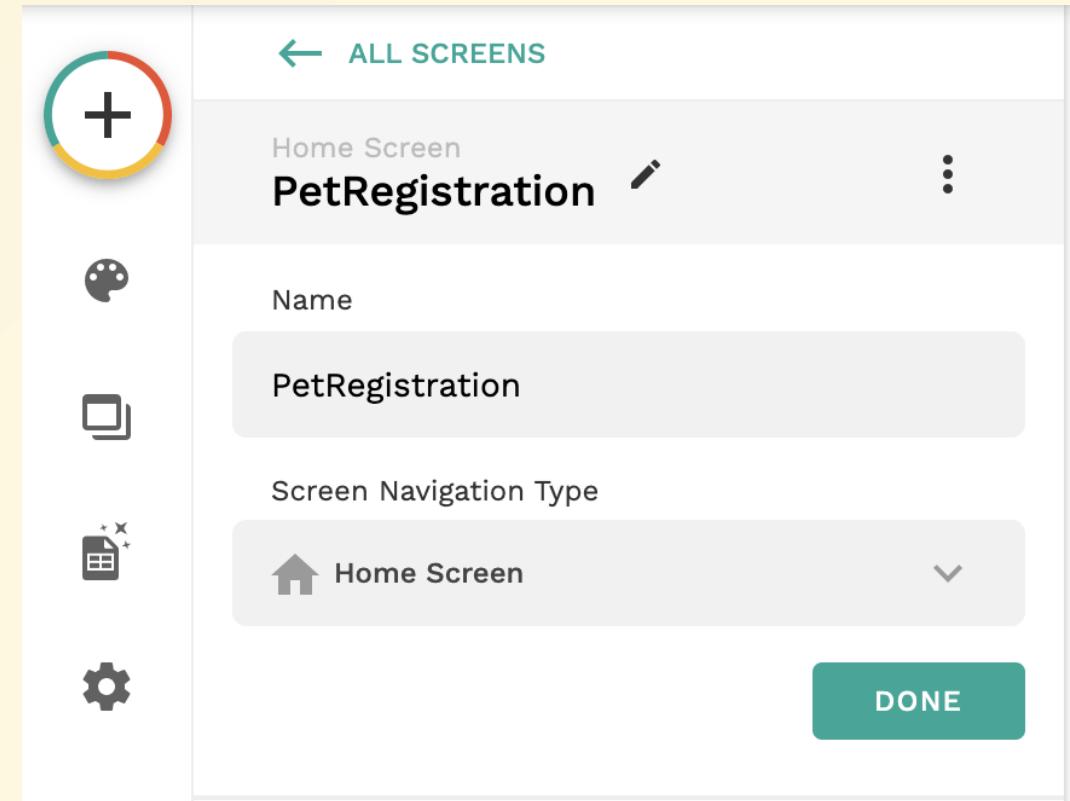
- The title bar shows "PetRegistration".
- The fields are displayed in the following order from top to bottom: "Name" (with placeholder "Enter name..."), "Image" (with a placeholder "Choose Photo" and a camera icon), and "Birthday" (with the value "Thursday, May 4th, 2019").
- A large yellow "CREATE PET" button is at the bottom.

- Submit ButtonのClickActionsにCreate Petが設定されていることを確認します

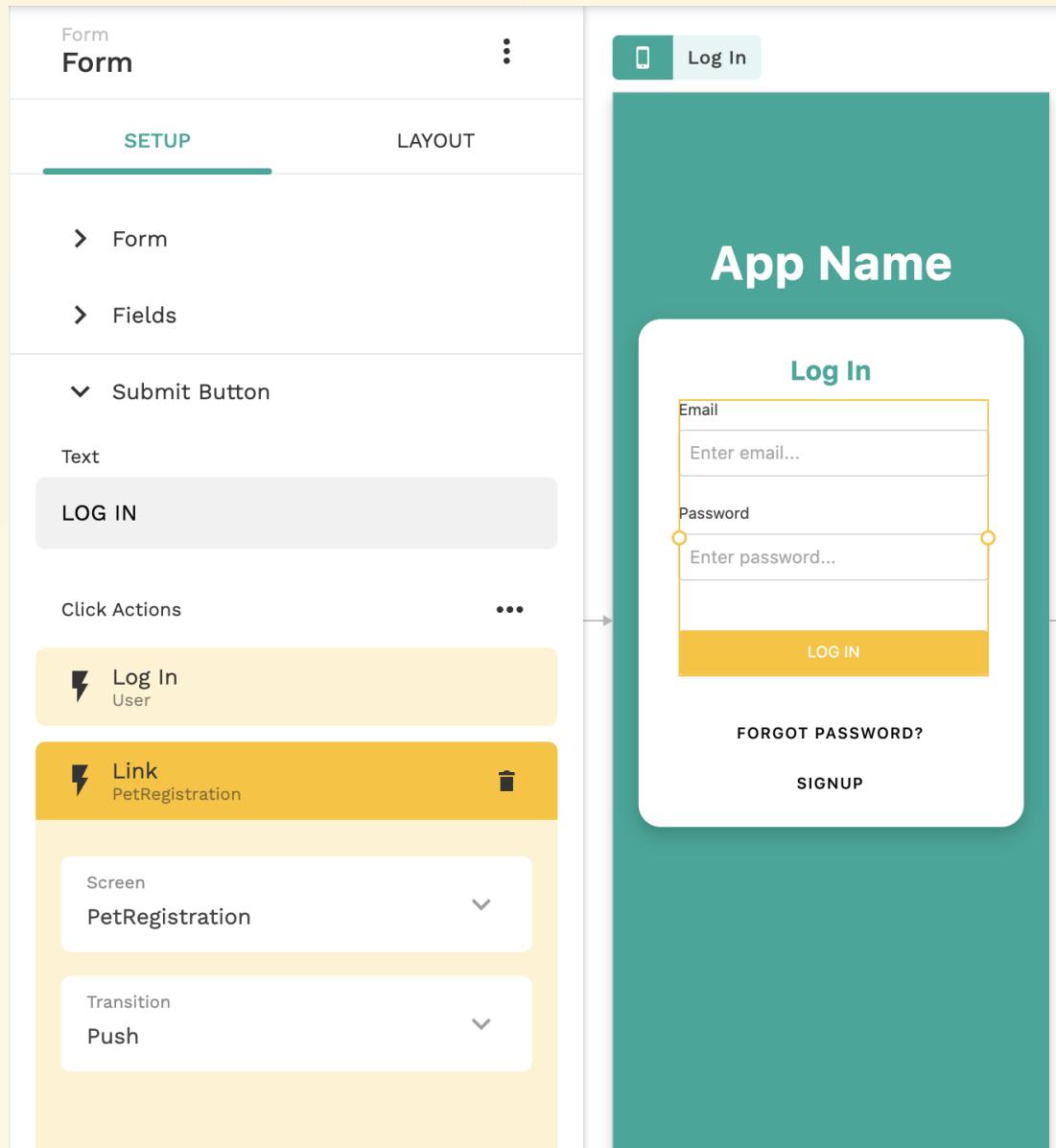


まだ導線がなく、表示することができないので、表示できるようにしましょう

- ログイン状態の時に最初に表示されるHome Screenに設定します
 - Screensからペット登録画面を選び、Screen Navigation TypeをHome Screenに変更します

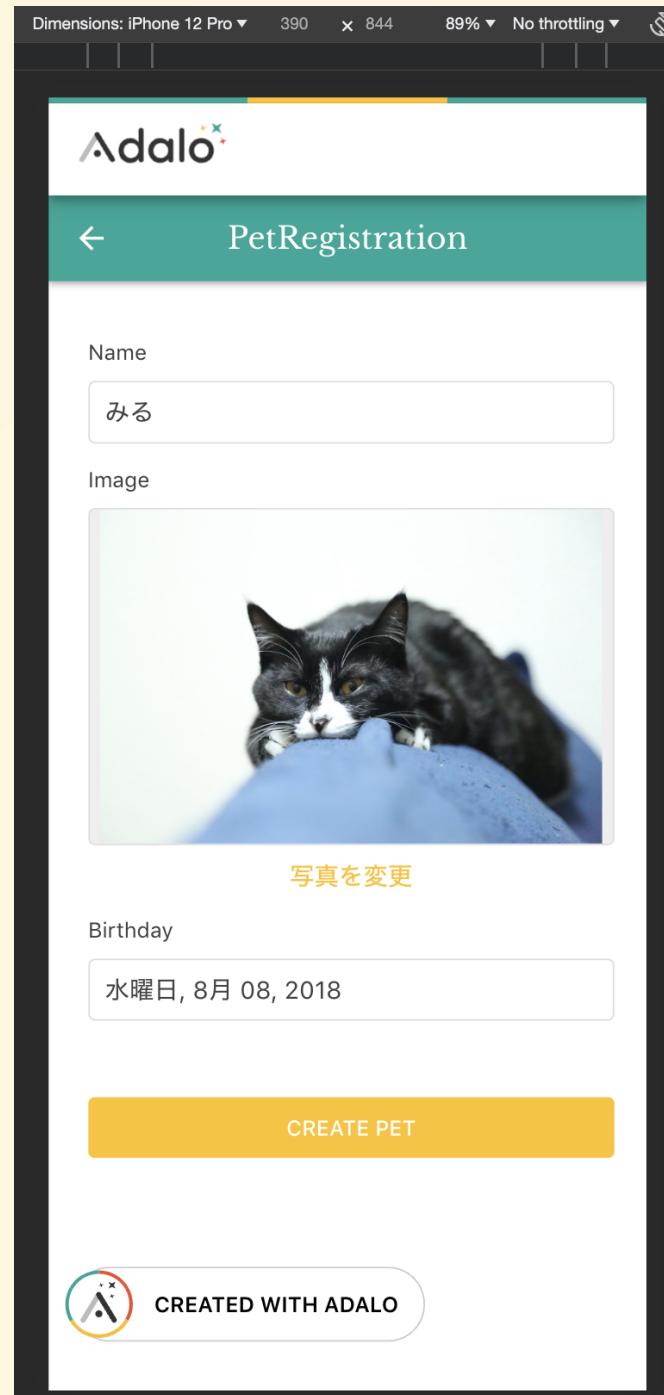


- ログイン画面の Form > Submit buttonを開き、ClickActionsに設定されているLinkのScreenをペット登録画面に変更します



ペット登録画面の動作を確認してみましょう。

- プレビュー機能でログインをすると、ペット登録画面が表示できます。
- 名前の入力、画像の選択、誕生日の選択ができます。
- CREATE PETボタンを押して、ペットを登録できます



ペットの登録後、データベースのPets CollectionにRecordが登録されたらOKです。

The screenshot shows the Airtable interface for the 'Pets' collection. On the left, there's a sidebar with icons for creating new records, editing, deleting, and more. The main area displays a table with the following data:

	Name	Birthday	Image	User	PetWeightLogs
<input type="checkbox"/>	みる	8/8/2018		imahashi@guildworks.jp	

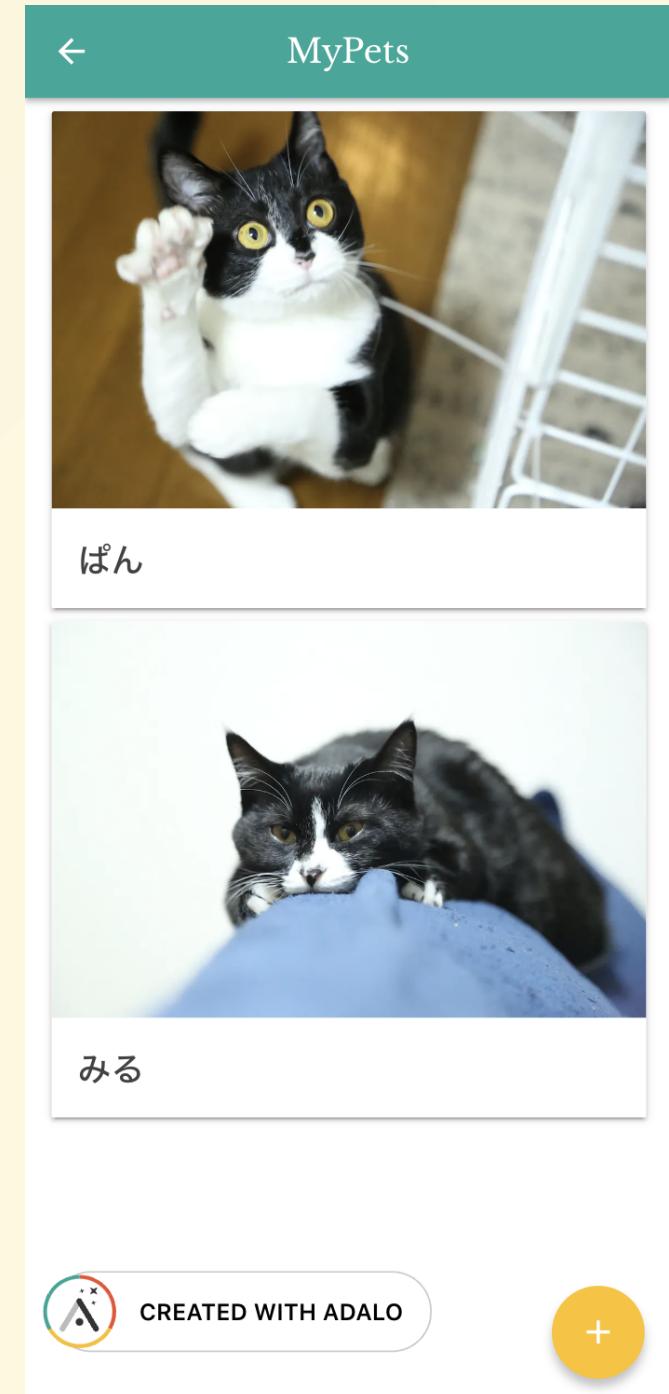
At the top right of the table, there are several action buttons: '+ ADD PET', a shield icon, an upward arrow, a downward arrow, a double-headed arrow, and a magnifying glass. A red 'DONE' button is located at the bottom right of the table area.

これで、ペット登録画面は完成です。

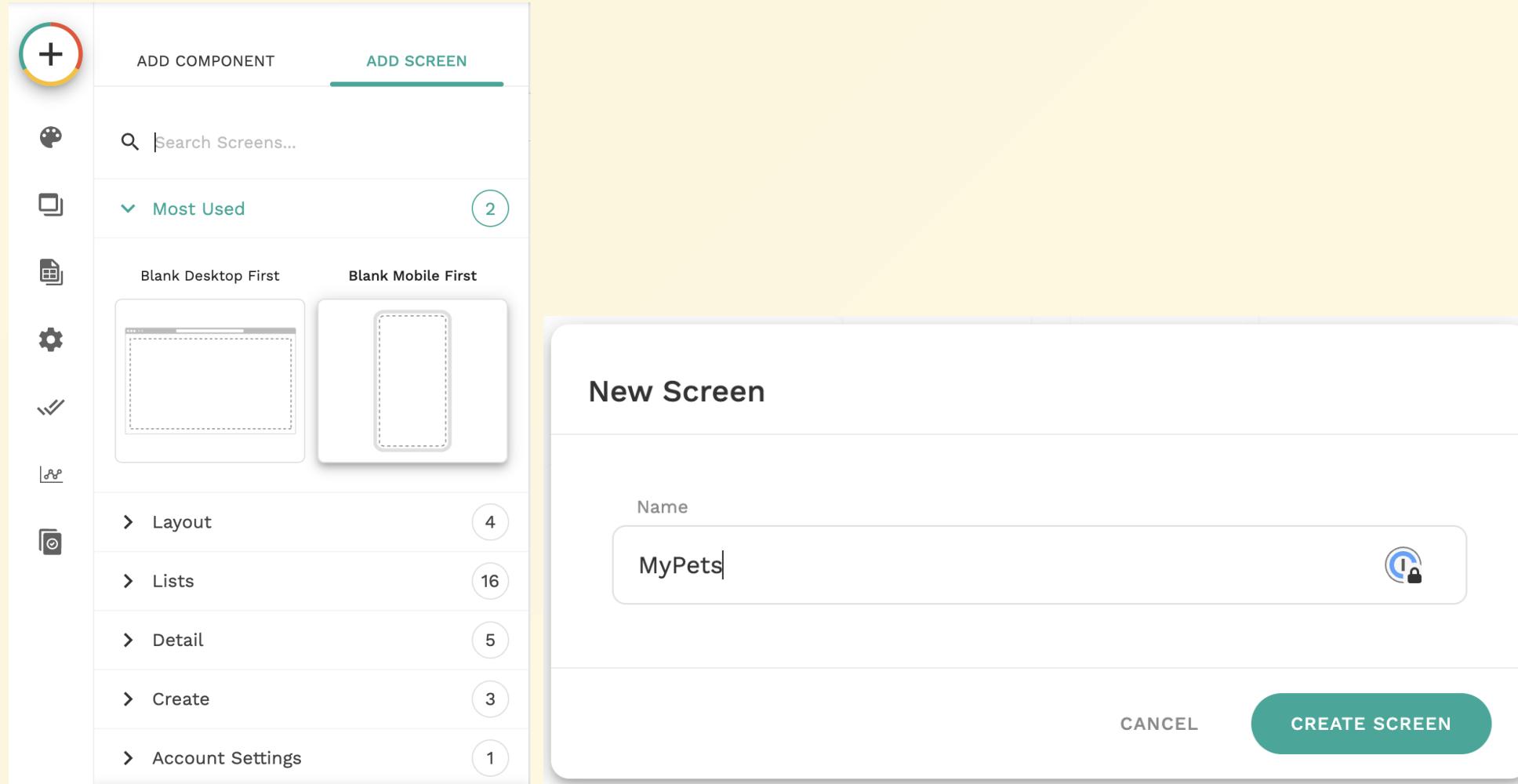
ペット一覧画面

- 登録したペットが一覧で表示できる
- ペットをクリックすると、そのペットのペット詳細画面に遷移できる
- 右下のアイコンを押すと、ペット登録画面に遷移できる

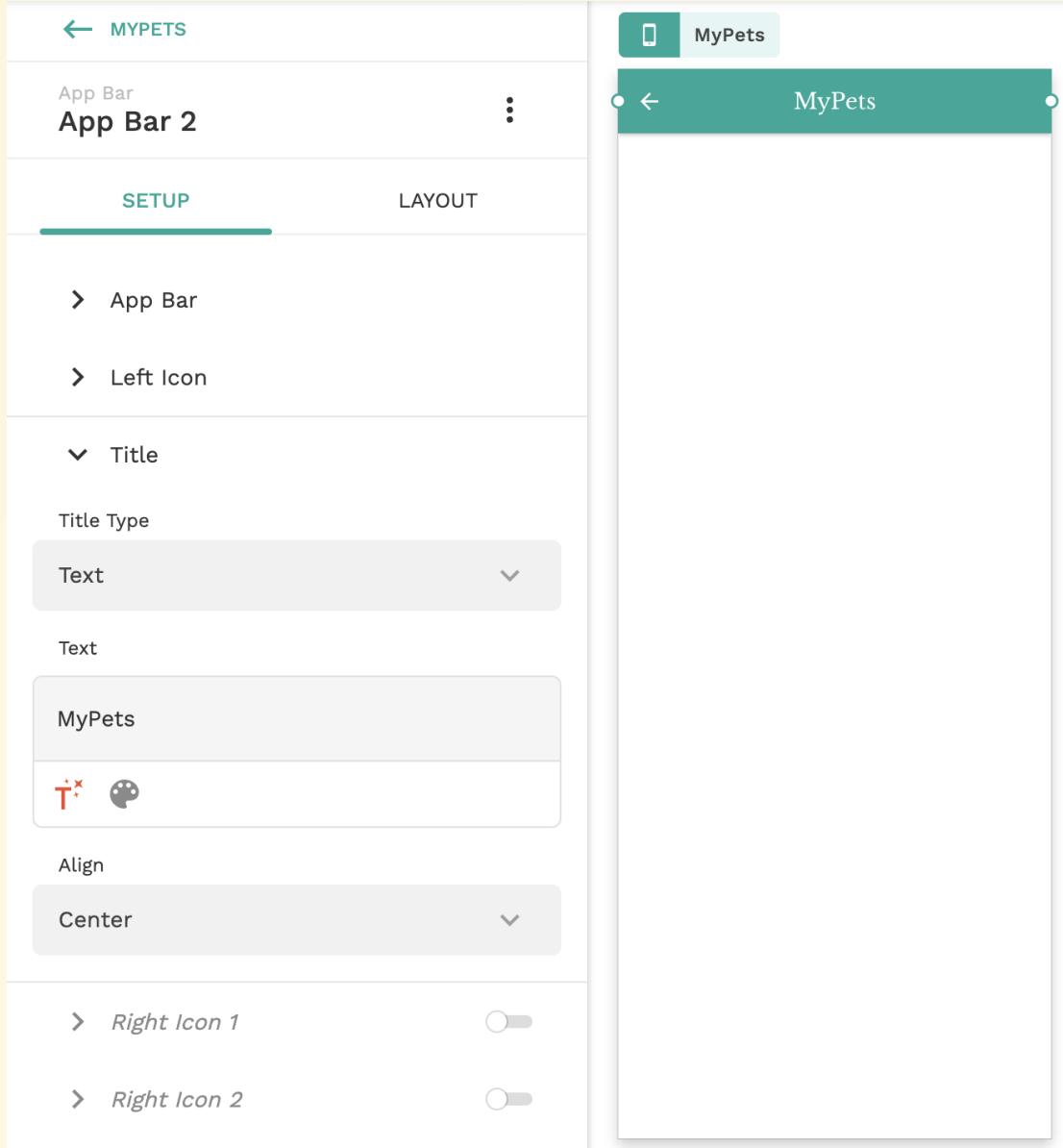
次は、こちらの画面を作りましょう



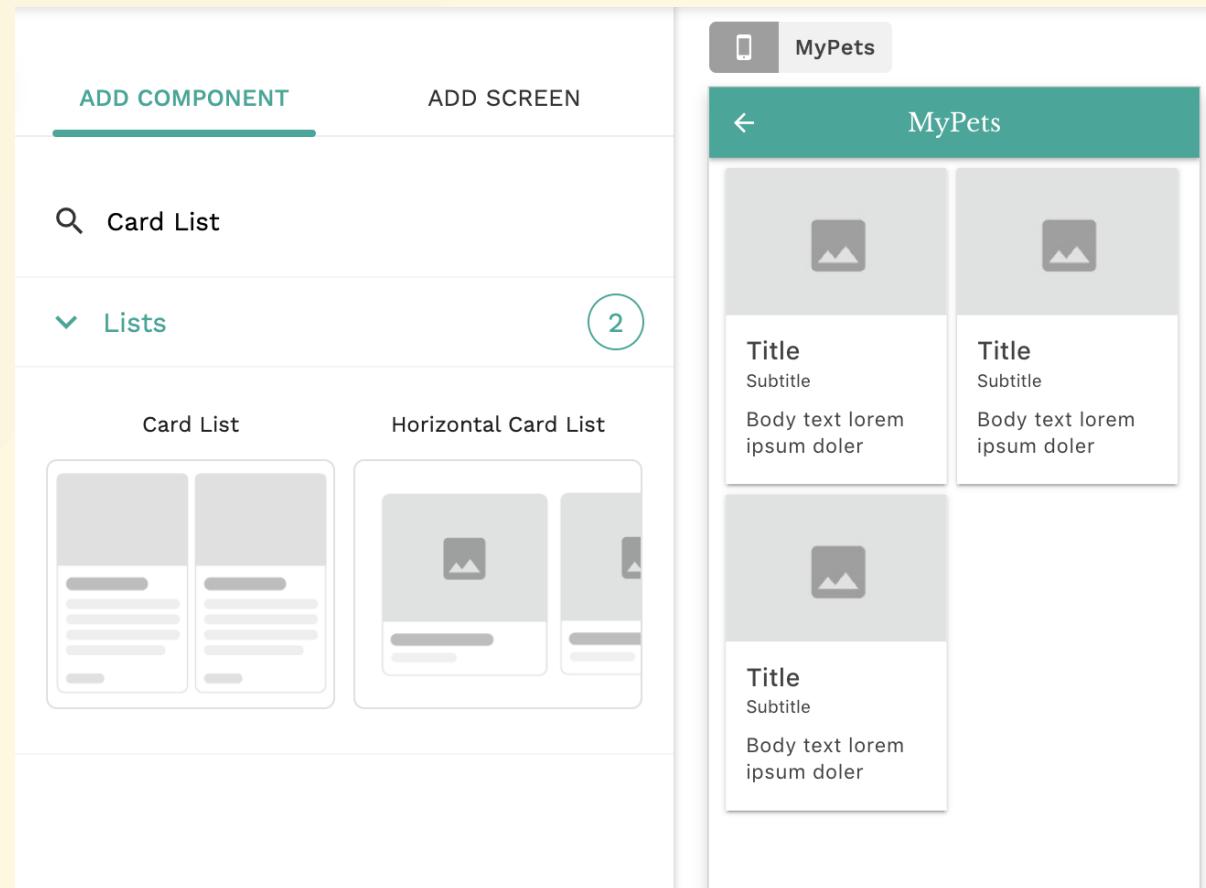
- ADD SCREENからBlank Mobile Firstを選択して、Screen Nameを入力します



- ADD COMPONENTからApp Barを選択して、画面上に配置します
- Title > Textの値を画面名(MyPets)に変えます



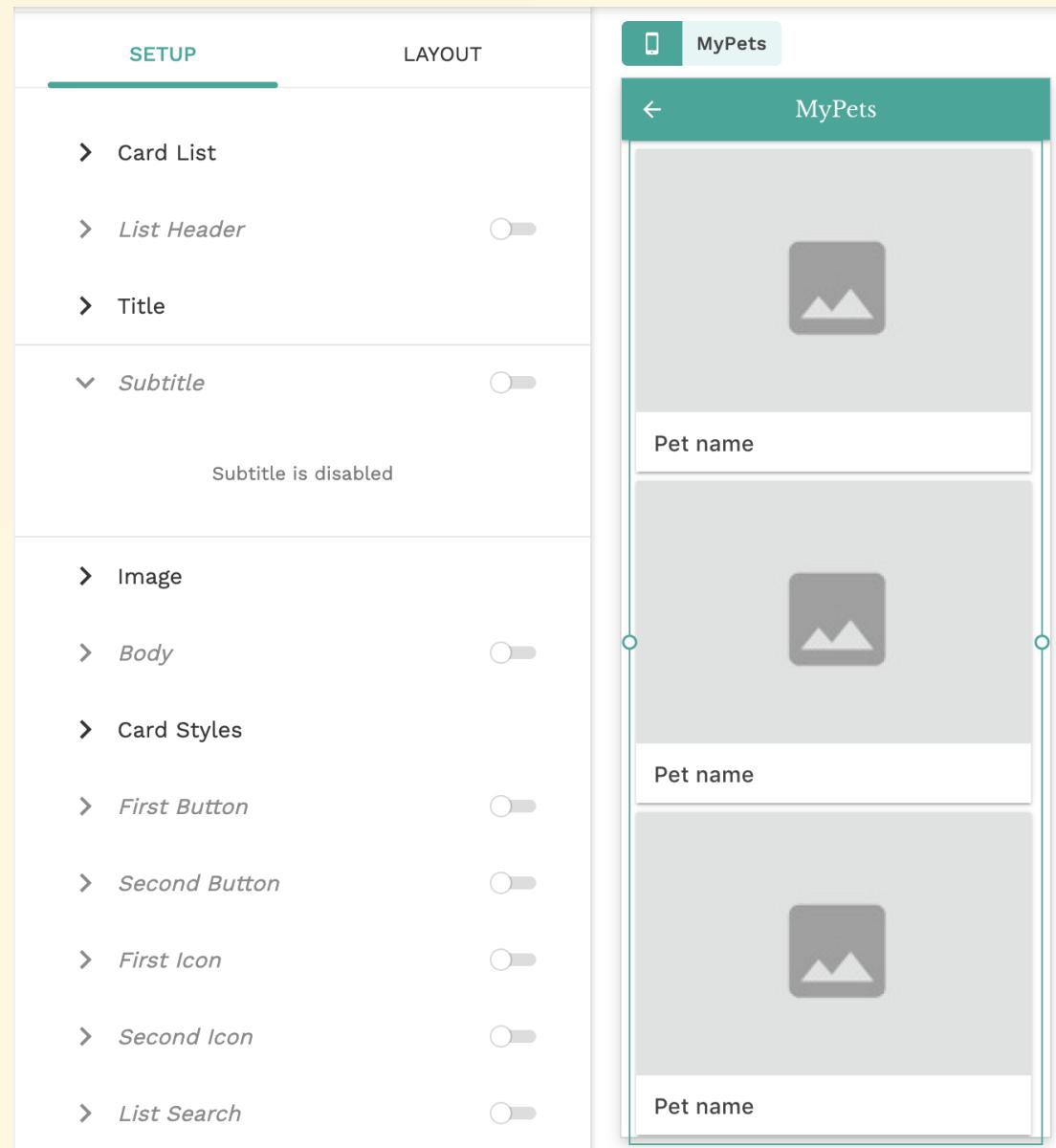
- ADD COMPONENTからCard Listを選択して、画面
上に配置します



- What is this a list of?でPetsを選択します
- FilterでLogged In User > Petsを選択します
- Columnsの値を1に変更します

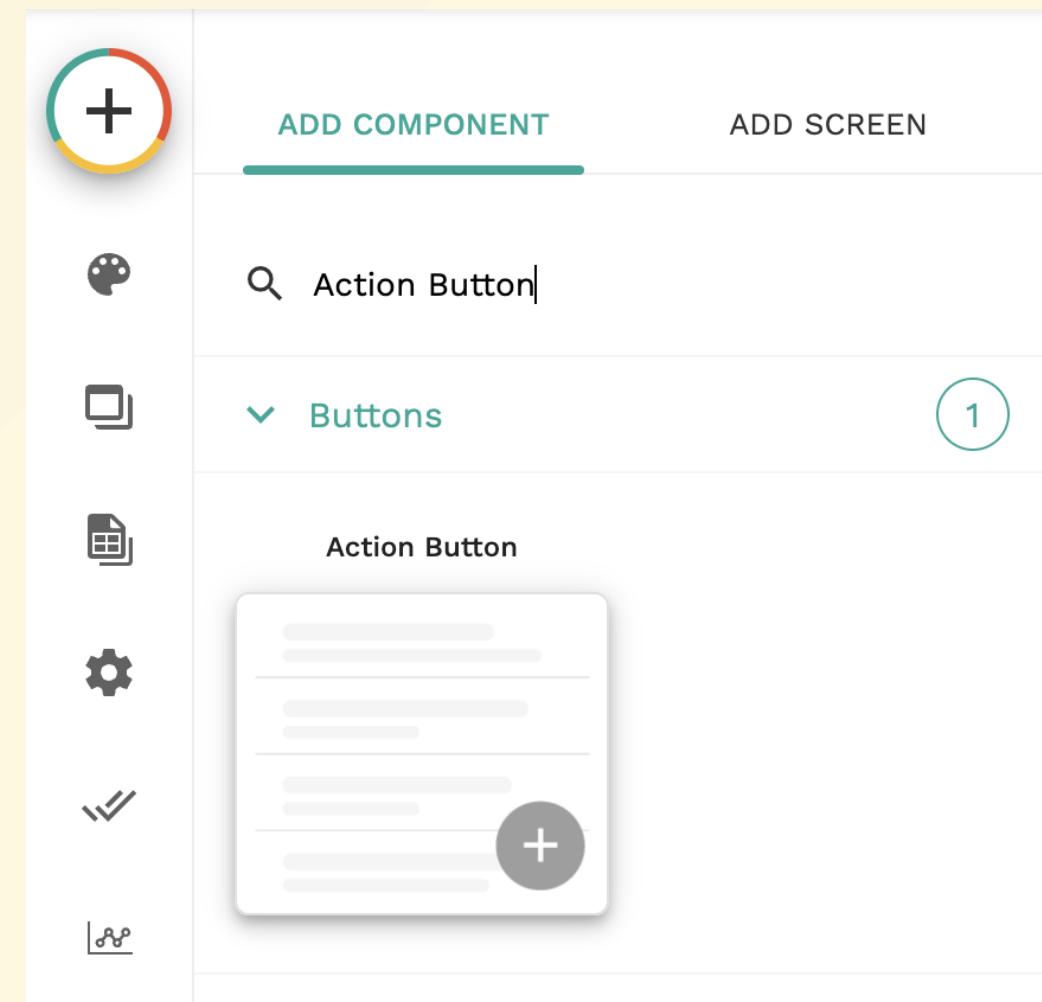
The screenshot displays the MyPets application interface. On the left, the 'Card List' setup screen is shown under the 'SETUP' tab. It includes fields for 'What is this a list of?' (set to 'Pets'), 'Filter' (set to 'Logged In User > Pets'), 'Sorting' (set to 'Add sorting...'), and 'Columns' (set to '1'). On the right, the 'MyPets' screen shows three cards, each featuring a placeholder image, a title 'Pet name', a subtitle 'Subtitle', and a body text 'Body text lorem ipsum dolor'.

- 不要なSubtitleとBodyのトグルをOFFにして、非表示にします

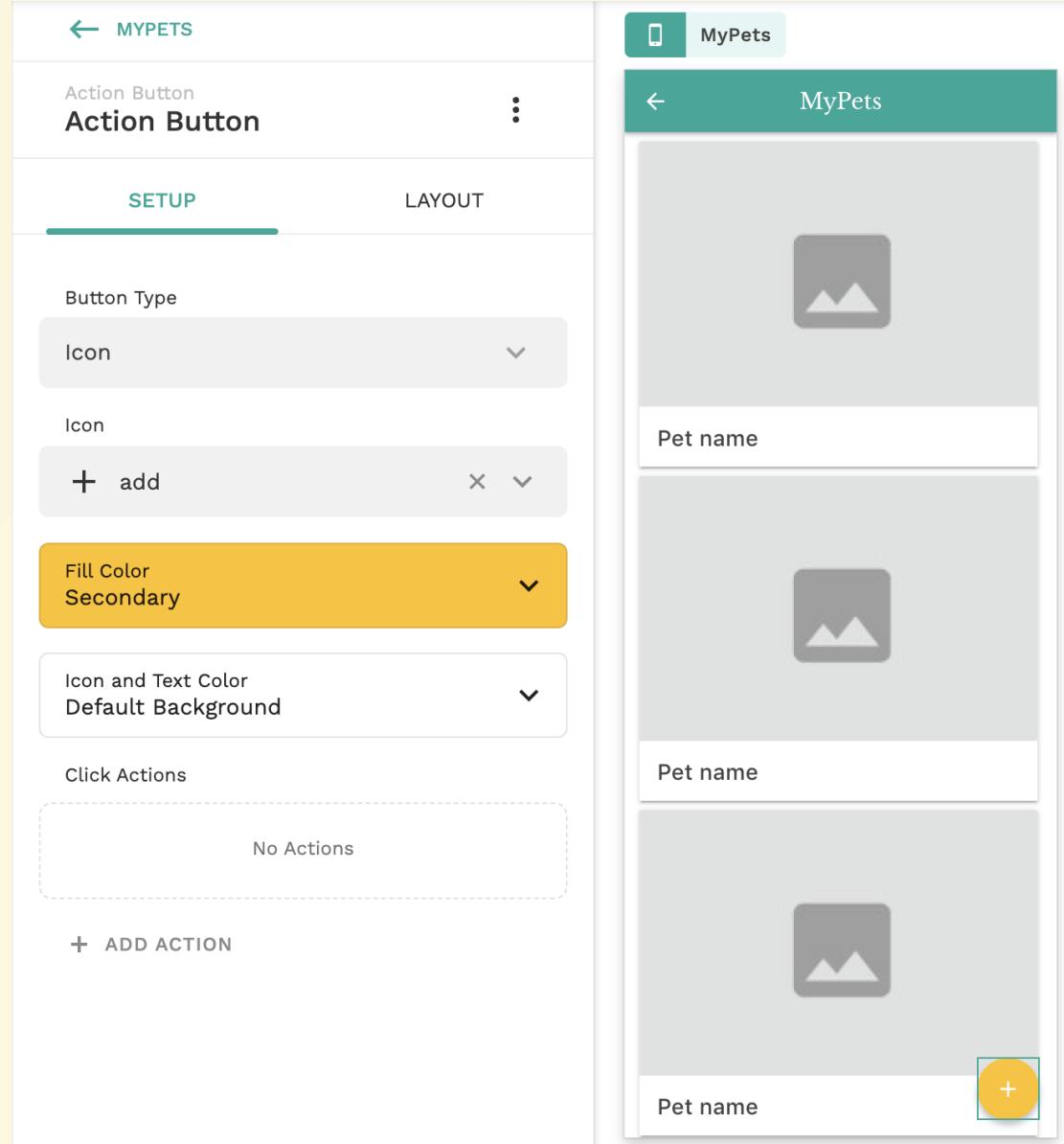


次に、ペット登録画面への導線を追加します

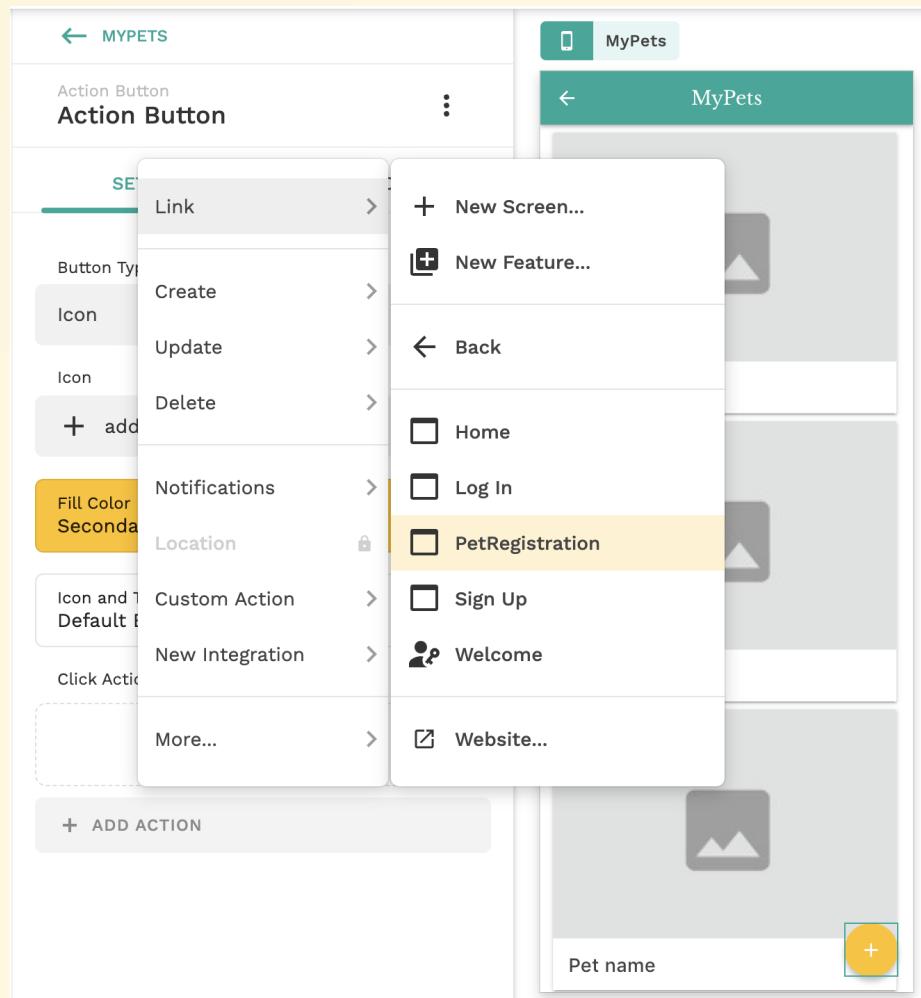
- ADD COMPONENTから Action Buttonを選択してください



- 画面右下に配置してください
- Icon and Text Colorを Default Background(White) に変更してください

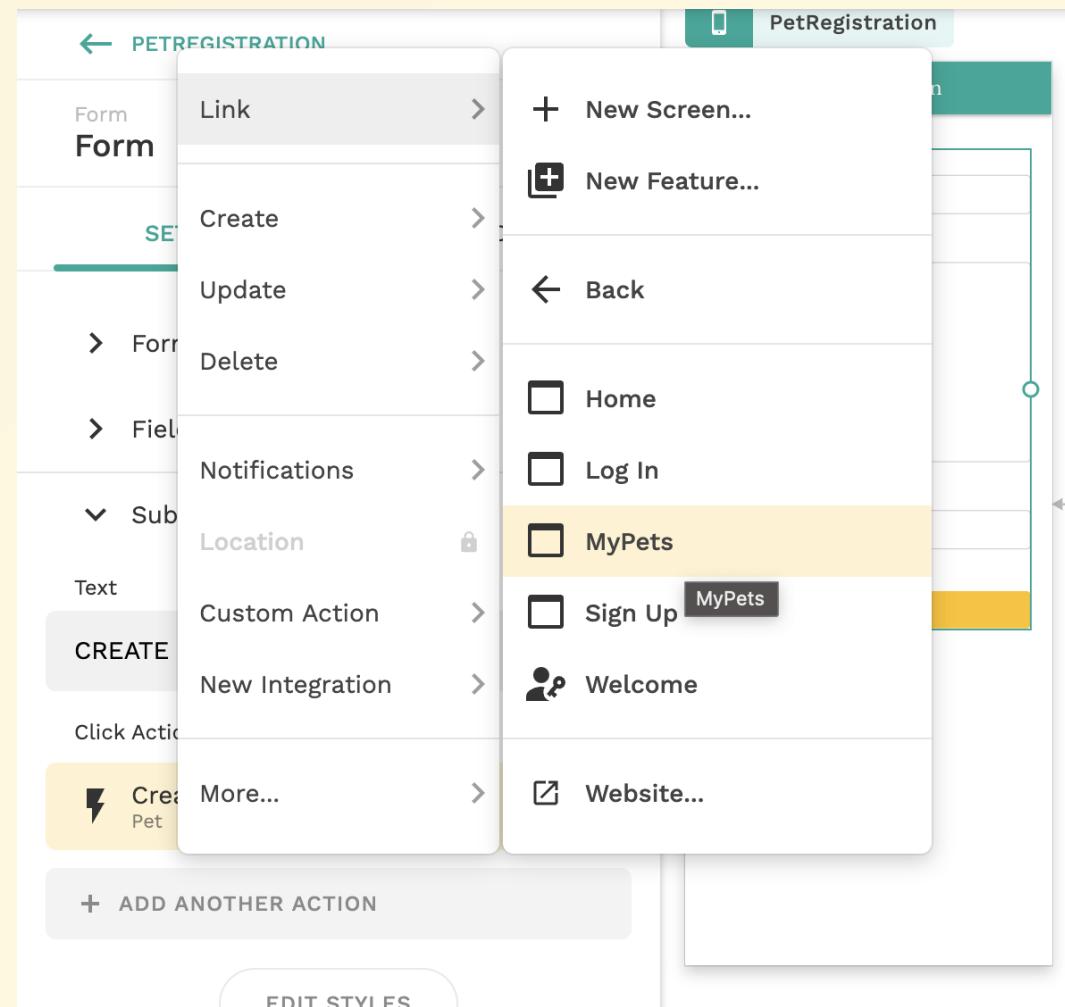


- ADD ACTION -> Link ->
PetRegistration を選択して
ください
- TransitionはNoneのままで
OKです

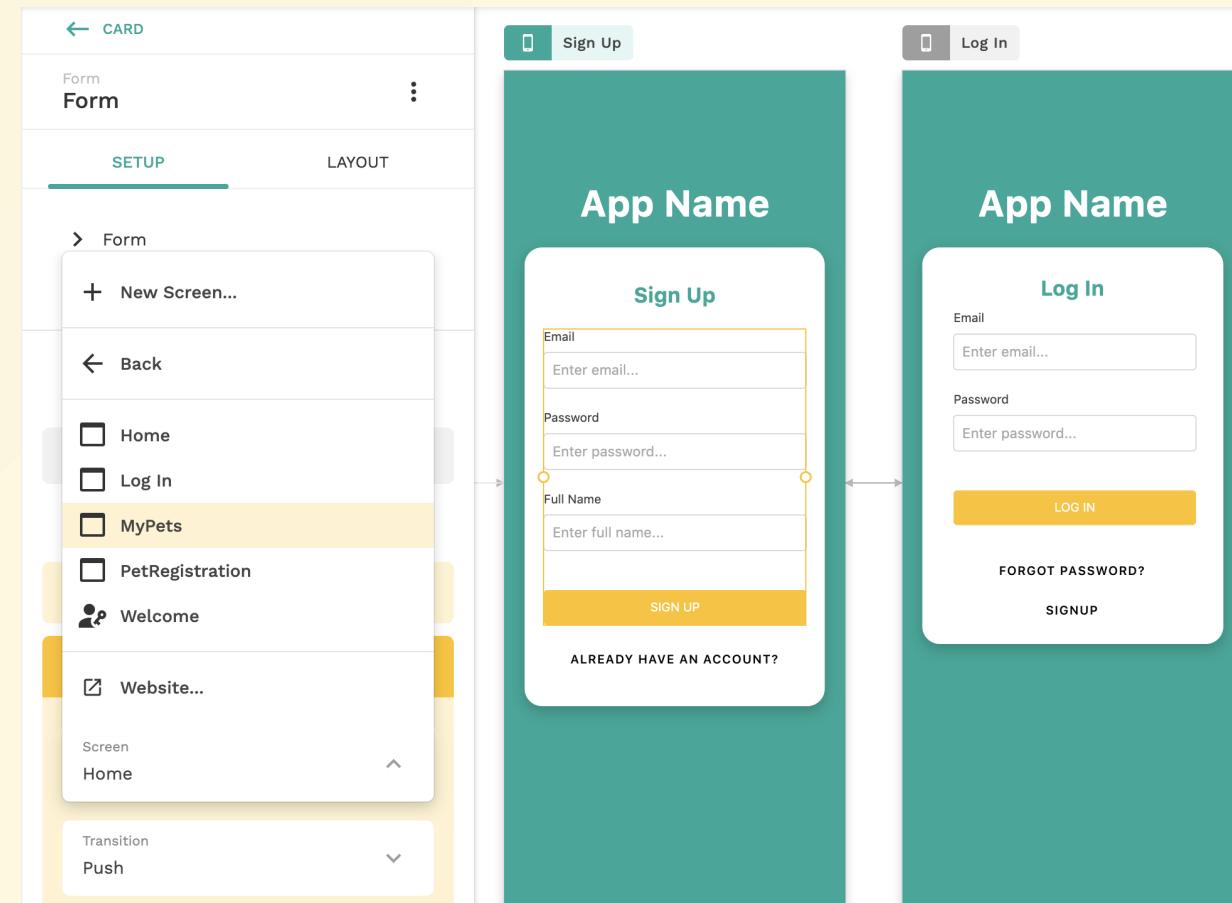


ペット登録画面からペット一覧画面への導線も追加しておきましょう

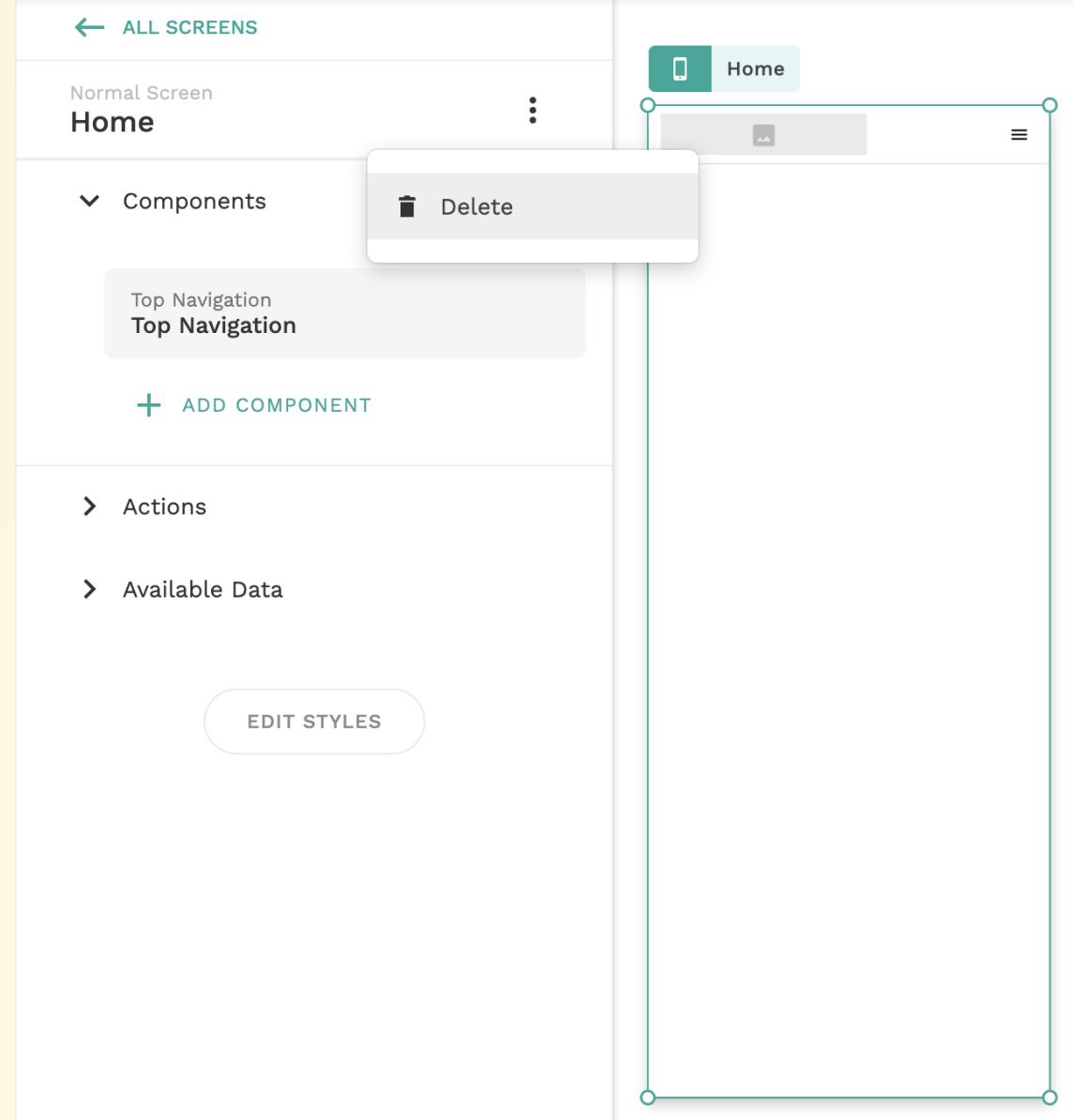
- ペット登録画面で Form > Submit Button を選択してください
- ADD ANOTHER ACTION -> Link -> MyPets を選択してください



- SignUp画面のSIGNUPボタンとLogin画面のLOGINボタンの遷移先も、HomeからMyPetsに変更しておきましょう

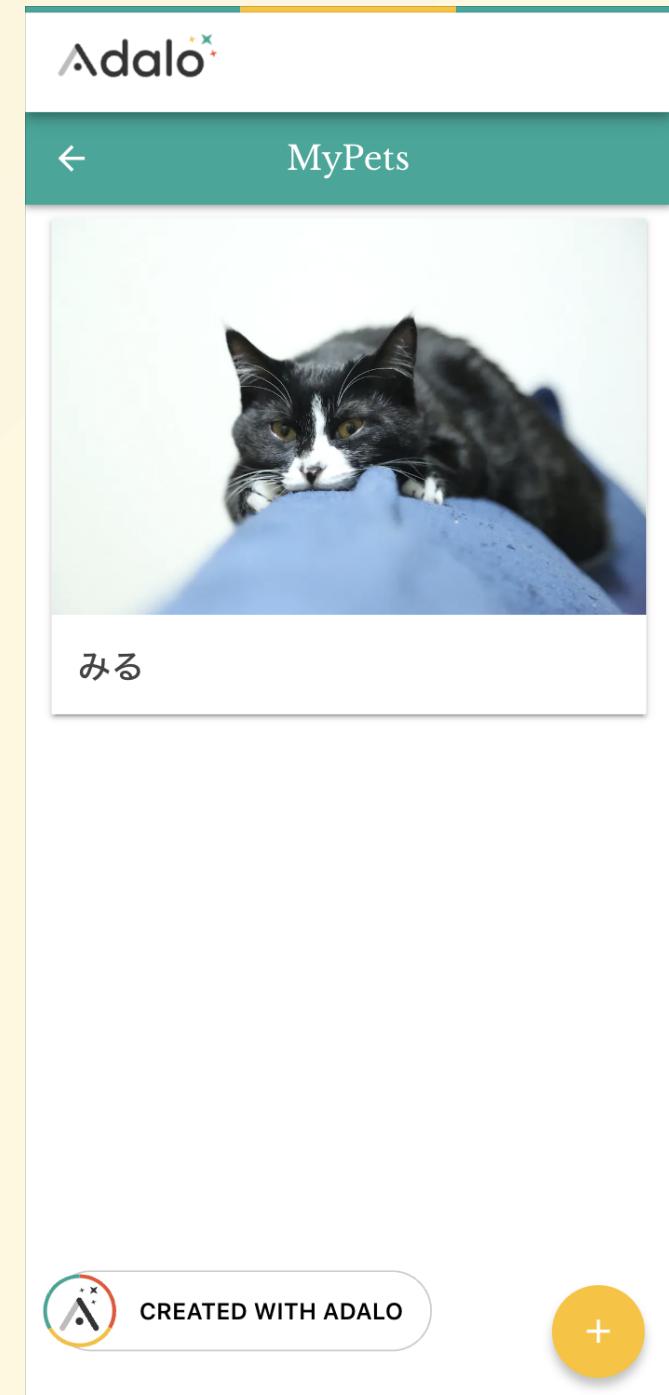


- ・デフォルトで作成されたHome画面は不要になってしまったので、削除しましょう



ペット一覧画面の動作を確認
しましょう。

- ・ペット登録画面のプレビューをした時と同様に、ペット一覧画面のScreen Navigation TypeをHome Screenに変更してください
- ・プレビュー機能でログインをすると、ペット一覧画面が表示できます。



- ペット登録画面でペットを追加すると、ペット一覧画面にも表示されます

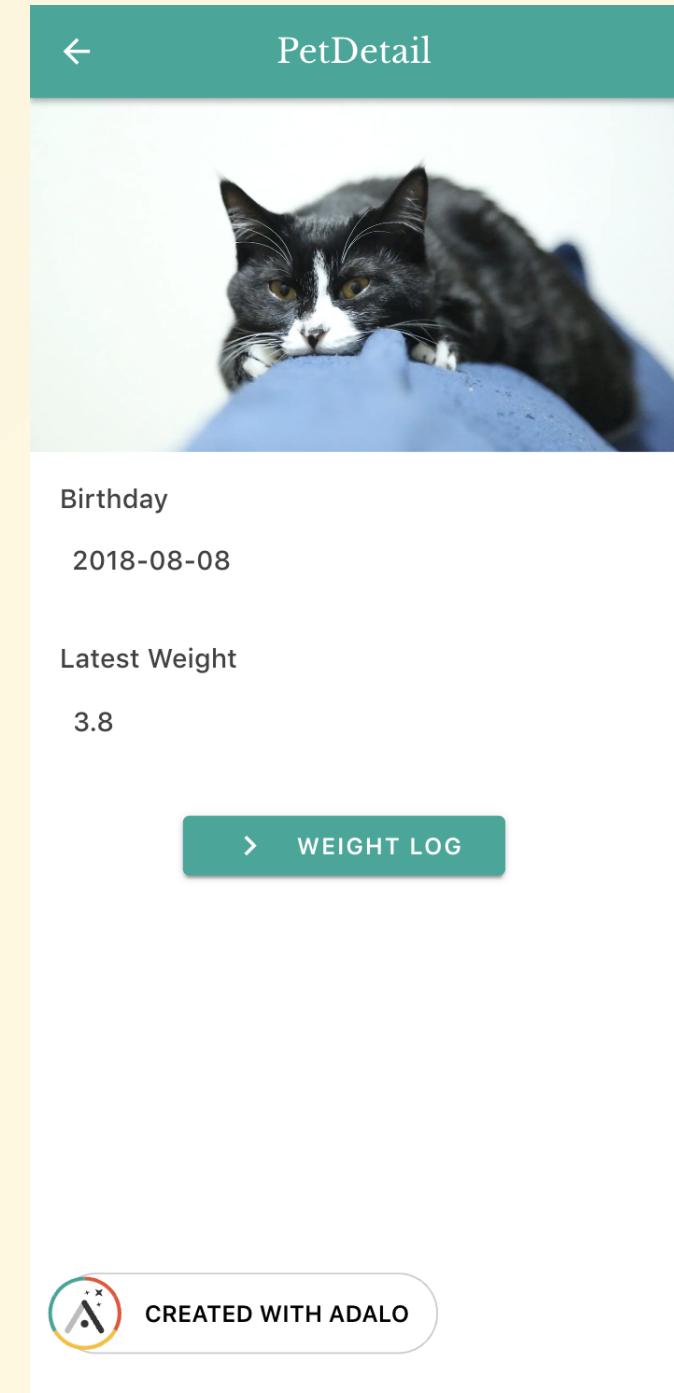
これで、ペット一覧画面は完成です。



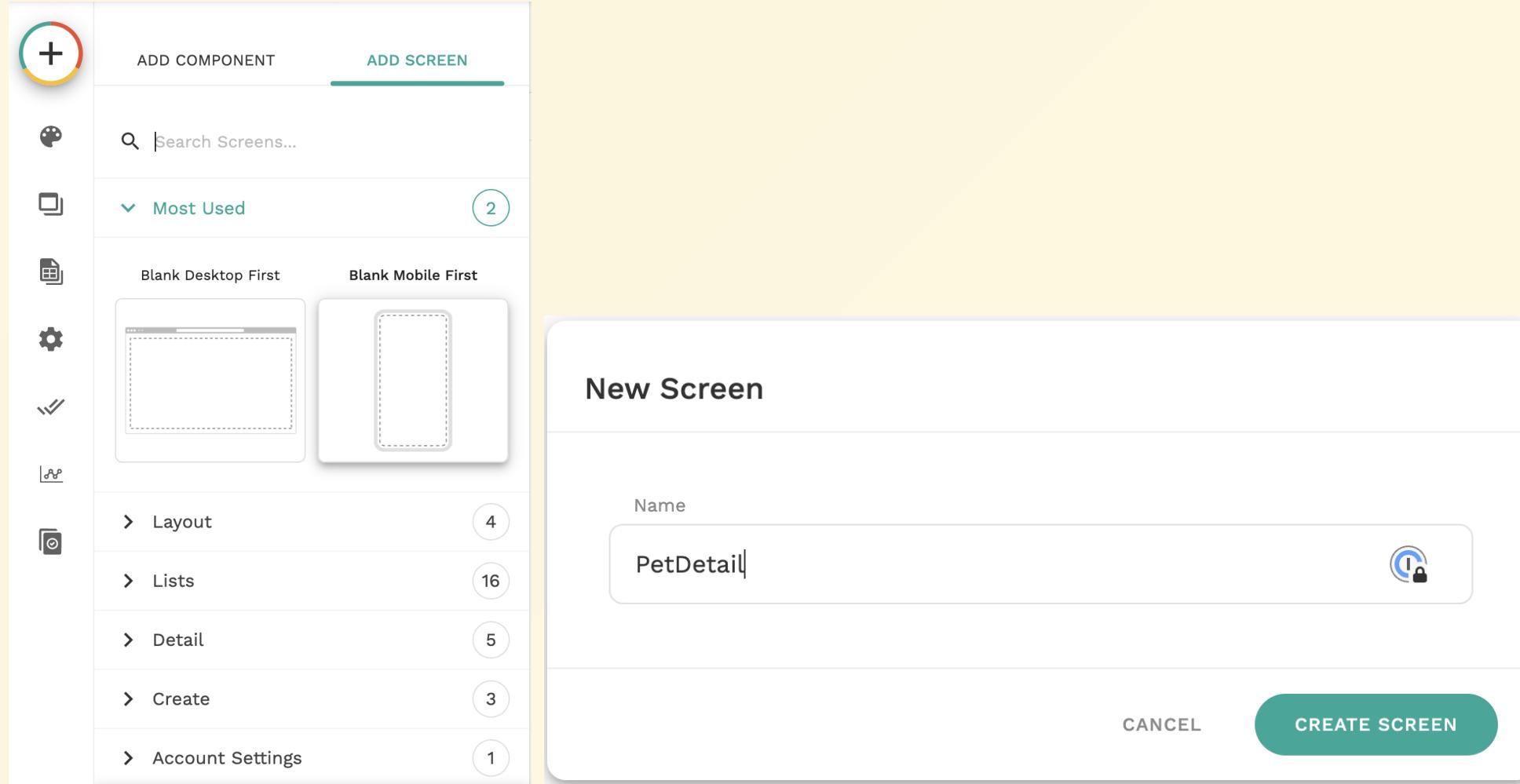
ペット詳細画面

- 誕生日が表示される
- 最新の体重が表示される
- 体重記録画面へのリンクがある

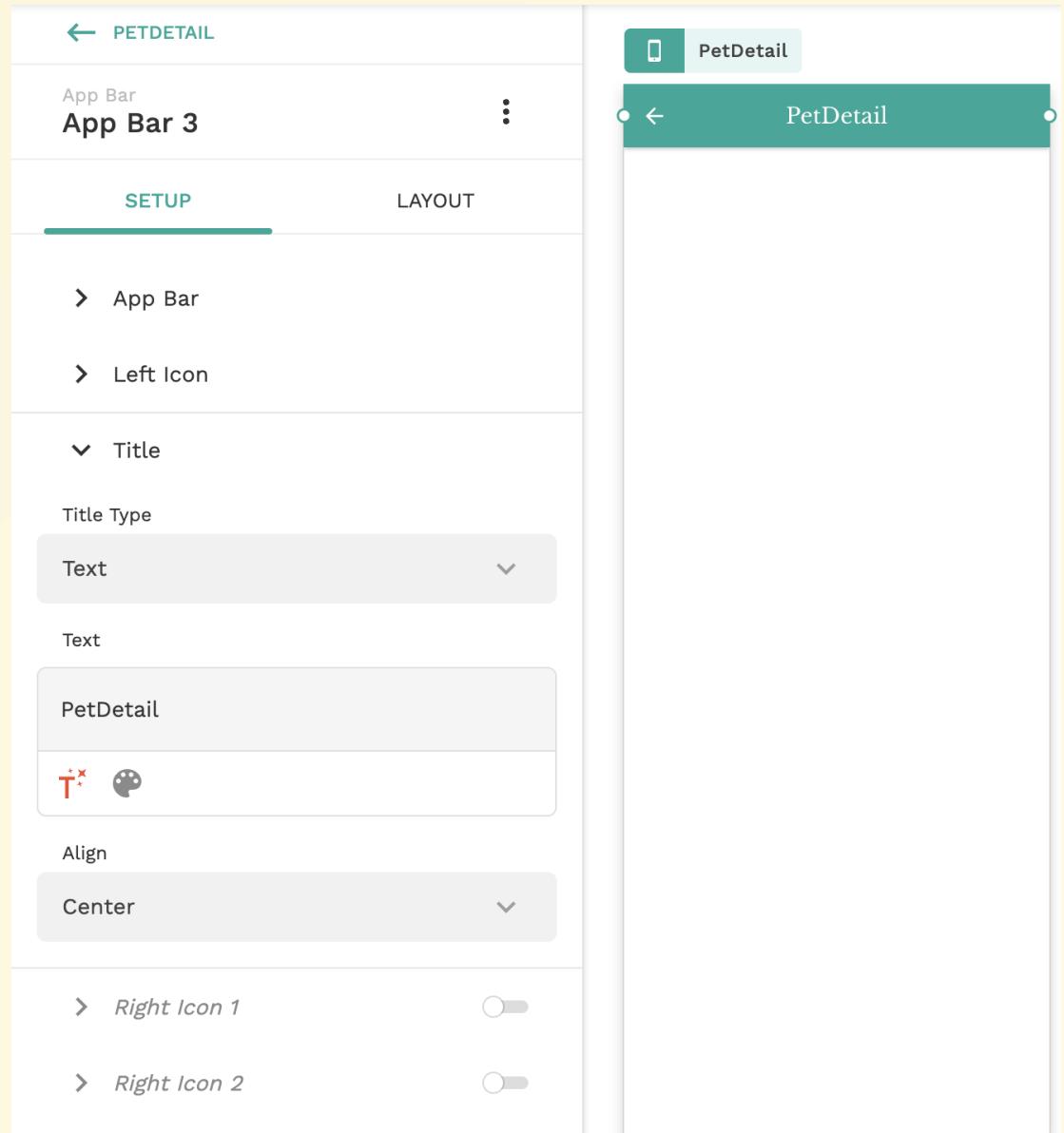
次は、こちらの画面を作りましょう



- ADD SCREENからBlank Mobile Firstを選択して、Screen Nameを入力します



- ADD COMPONENTからApp Barを選択して、画面上に配置します
- Title > Textの値を画面名(PetDetail)に変えます



- ADD COMPONENTからImageを選択して、画面上に配置します

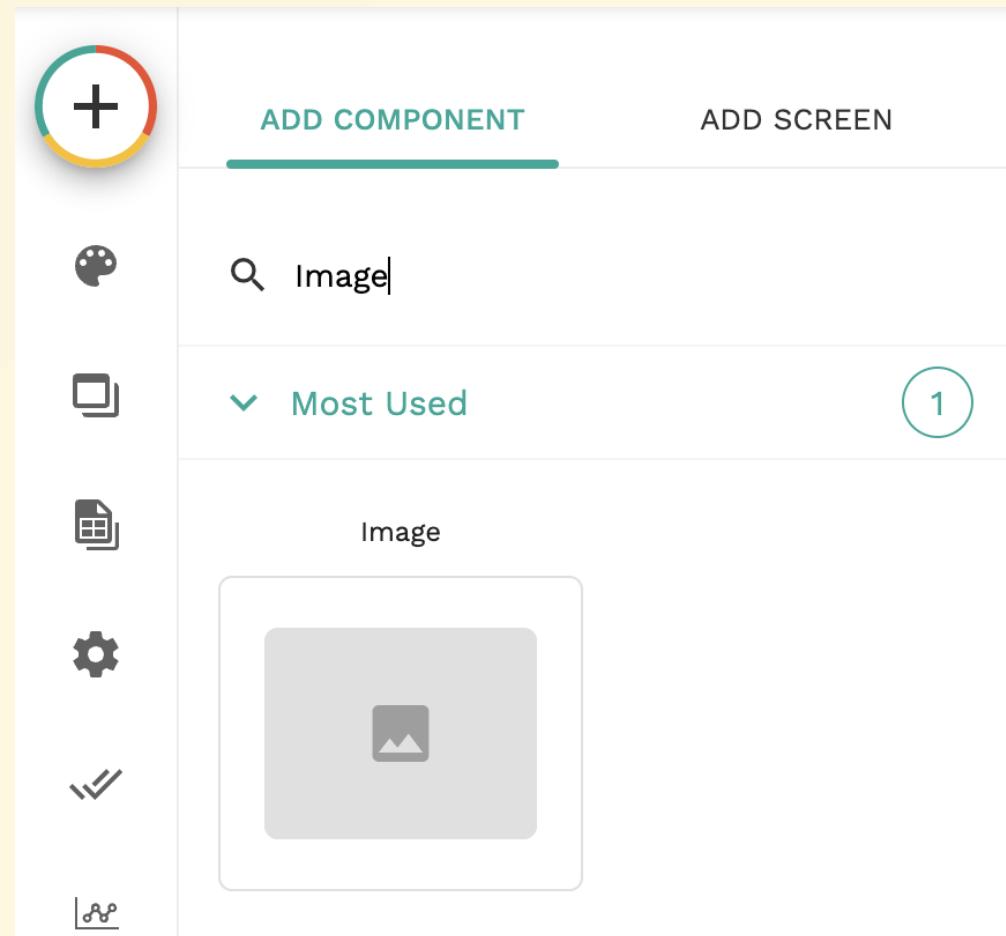
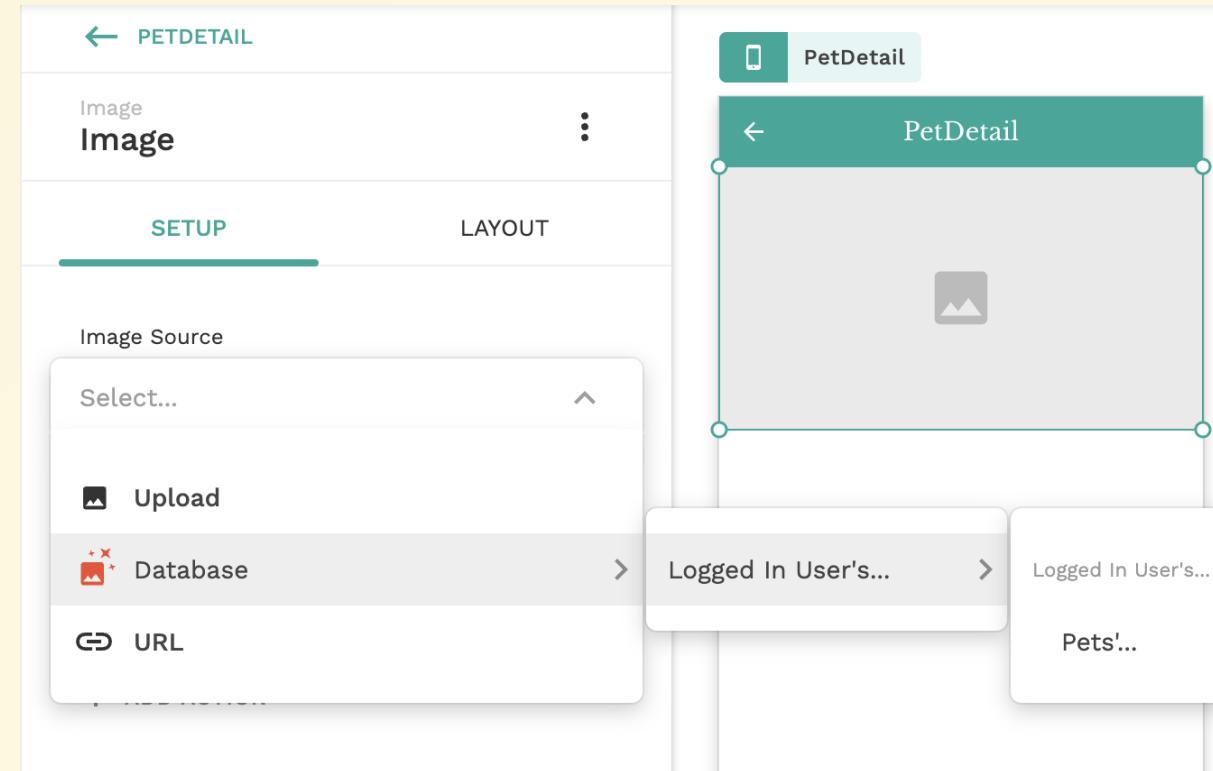


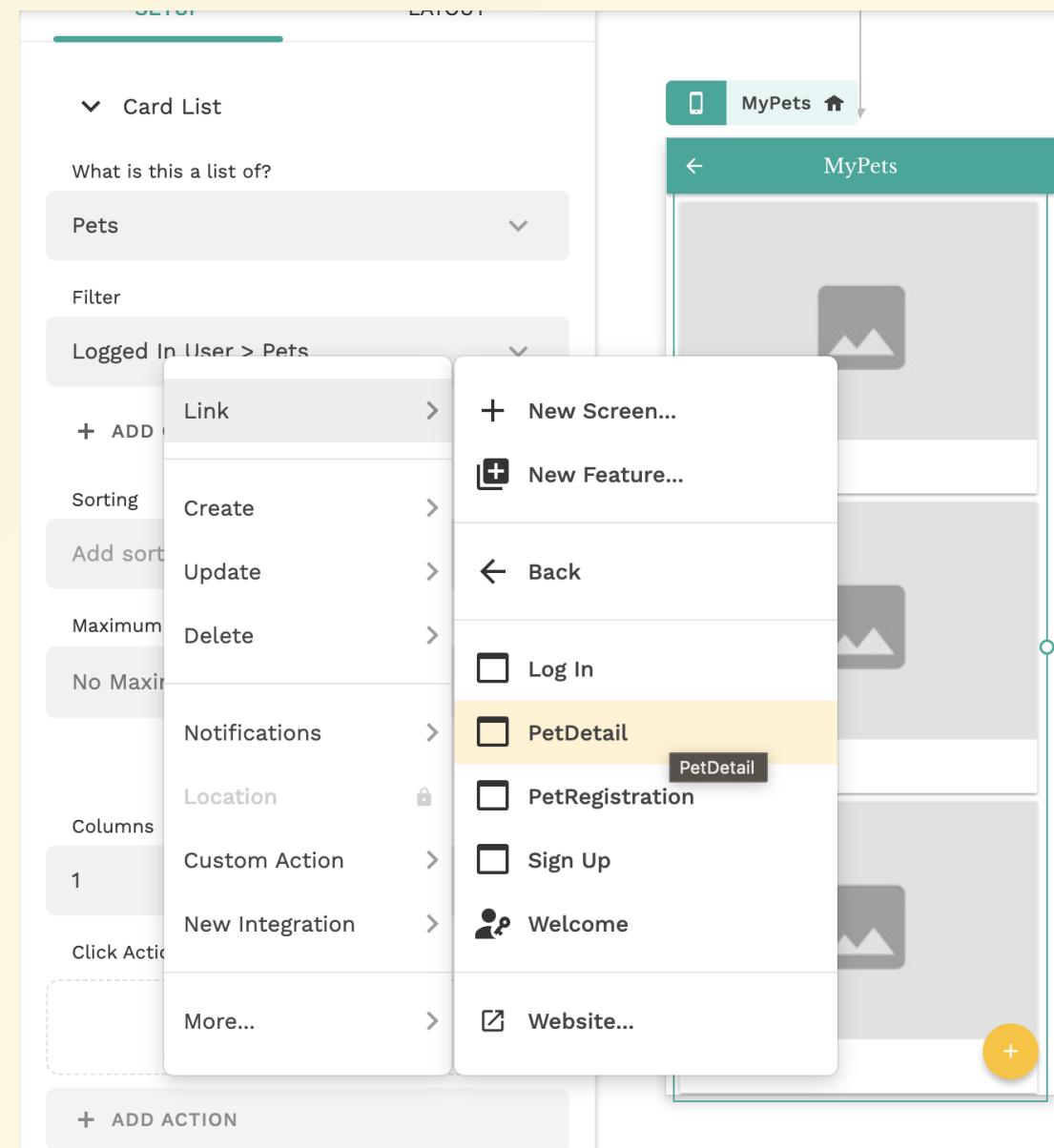
Image Sourceに表示したい特定のペットの画像を表示したいですが、そのままで選択肢にありません。

* Logged In User's > Pets > ...と選べそうに思えますが、ユーザーはペットを複数持つので、ペット1匹を特定できません

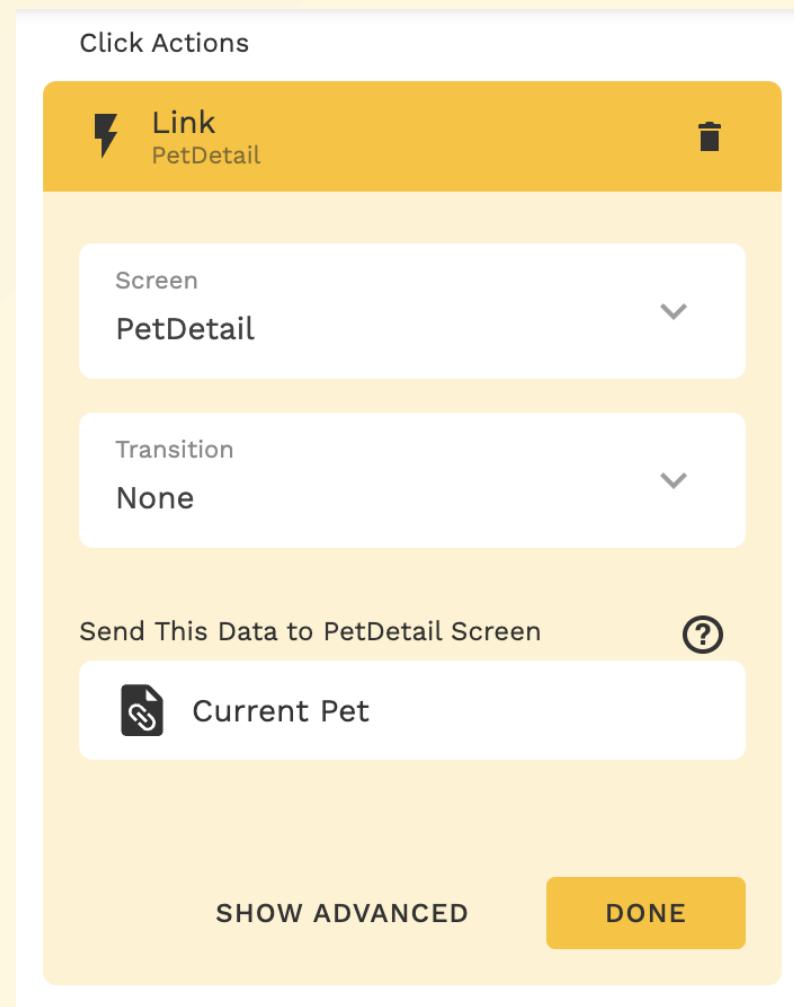


この画面への導線を設定することで、表示したい特定のペットが選択肢に追加されます

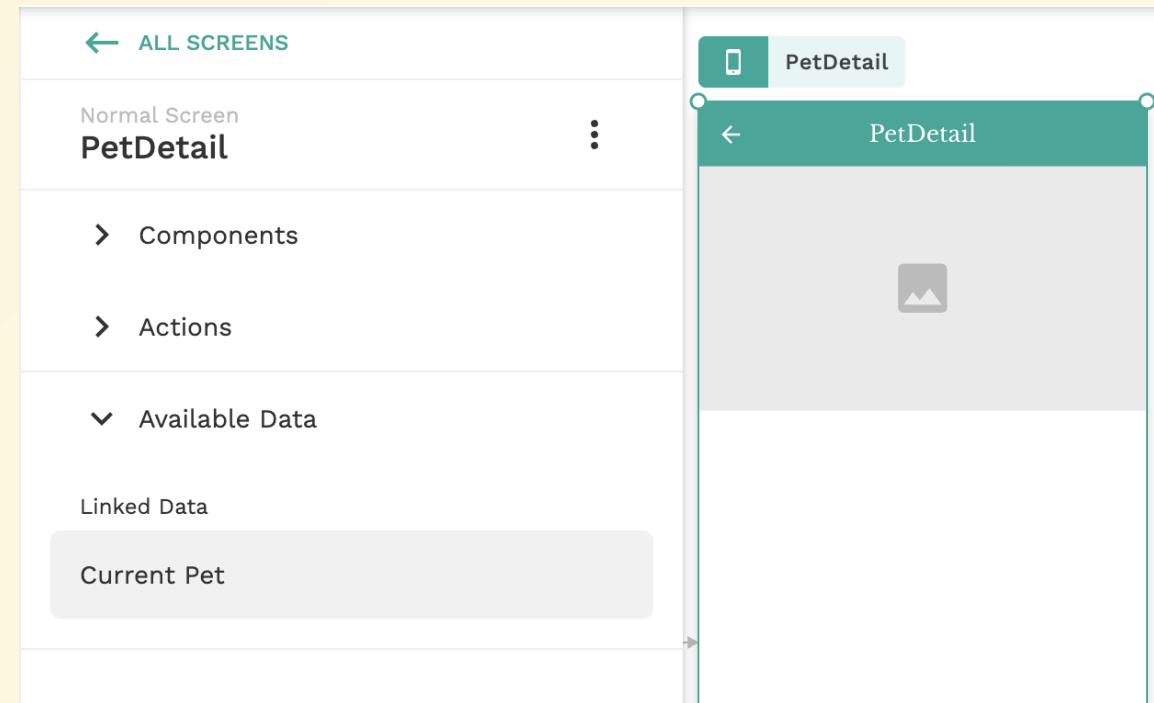
- ・ペット一覧画面のCardListでADD ACTIONをクリックし、PetDetailへのLinkを追加しましょう



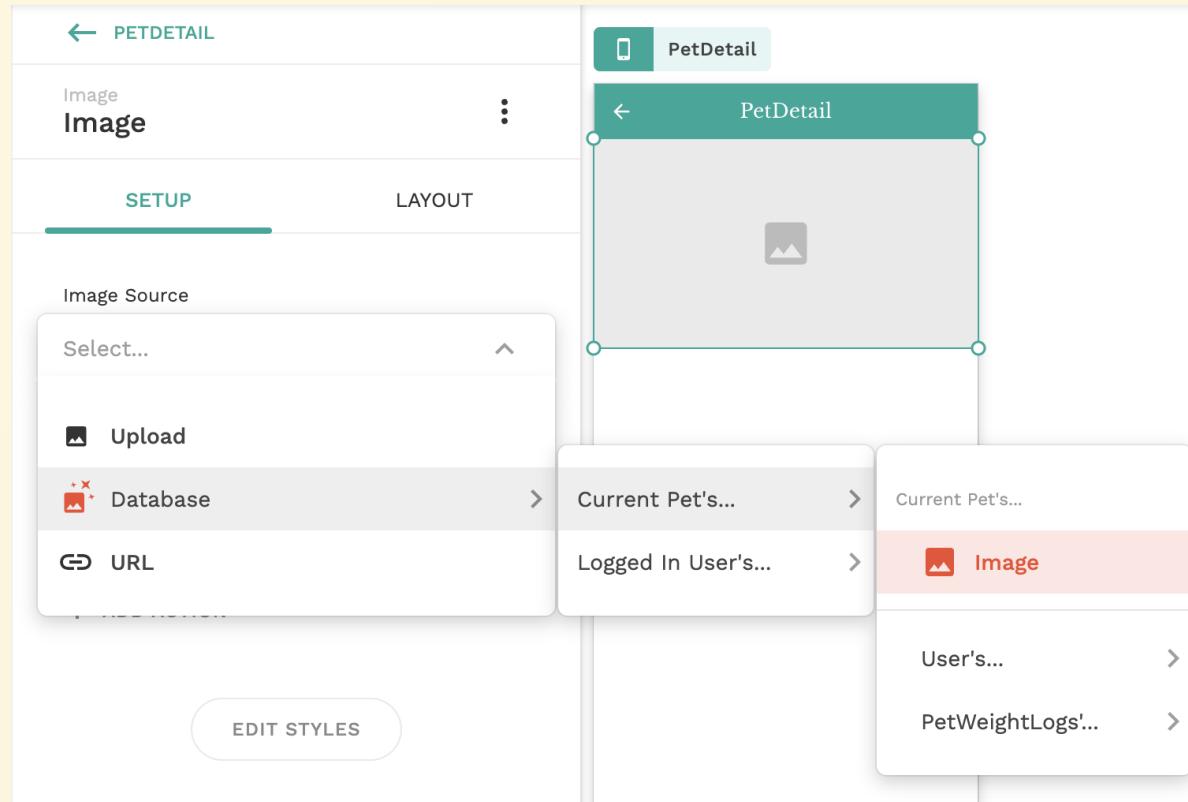
- 追加したLinkのSend This Data to PetDetail ScreenにCurrent Petが自動で設定されています。



- ・ペット一覧画面からのLinkのSend This Data to PetDetail ScreenにCurrent Petが設定されたため、ペット詳細画面のAvailable Data内にLinked DataとしてCurrent Petが設定されています。
 - これにより、ペット詳細画面でペット一覧画面で選択したペット(Current Pet)を扱えるようになります。



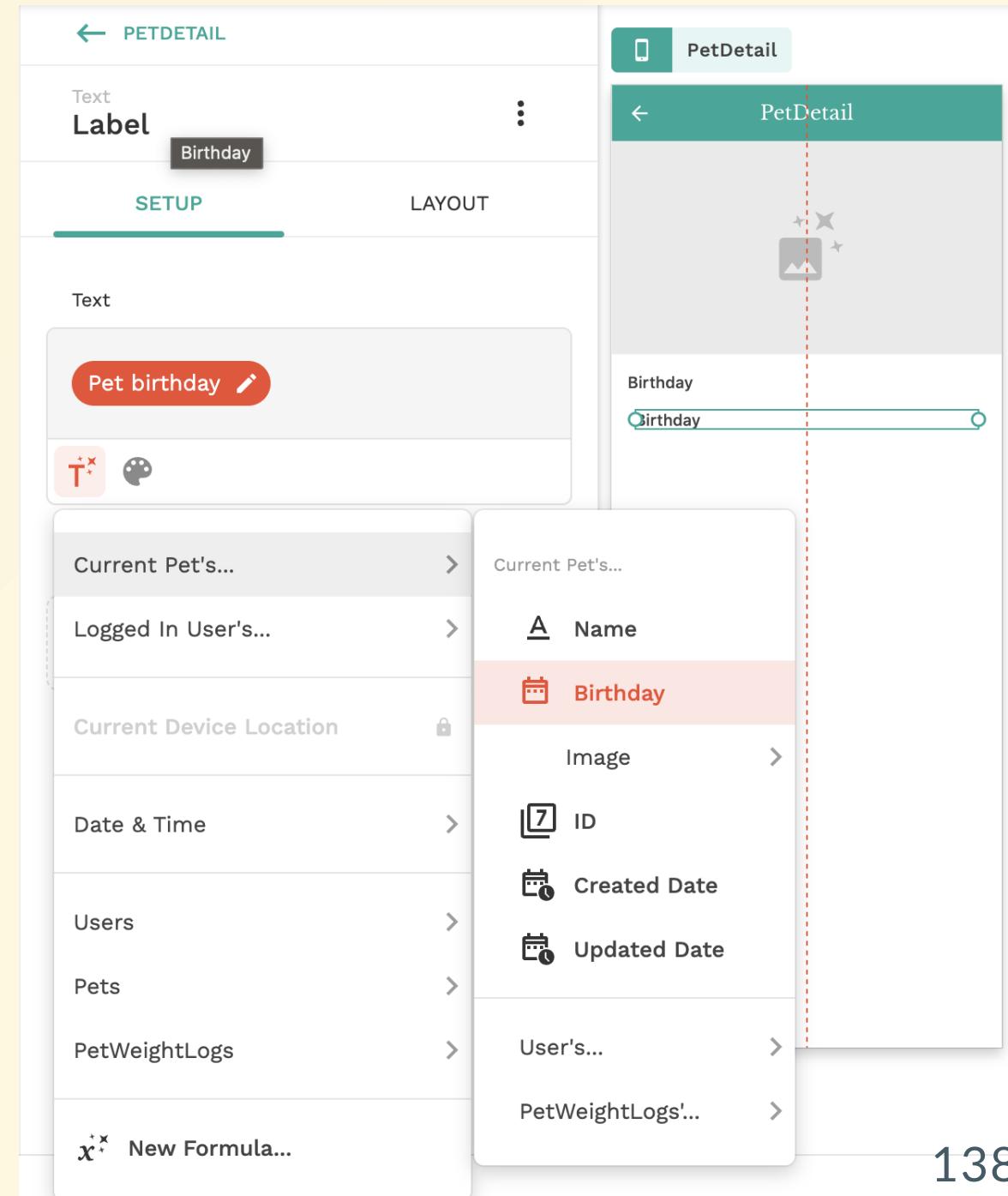
- ImageコンポーネントのImageSourceに Database > Current Pet > Image を設定します



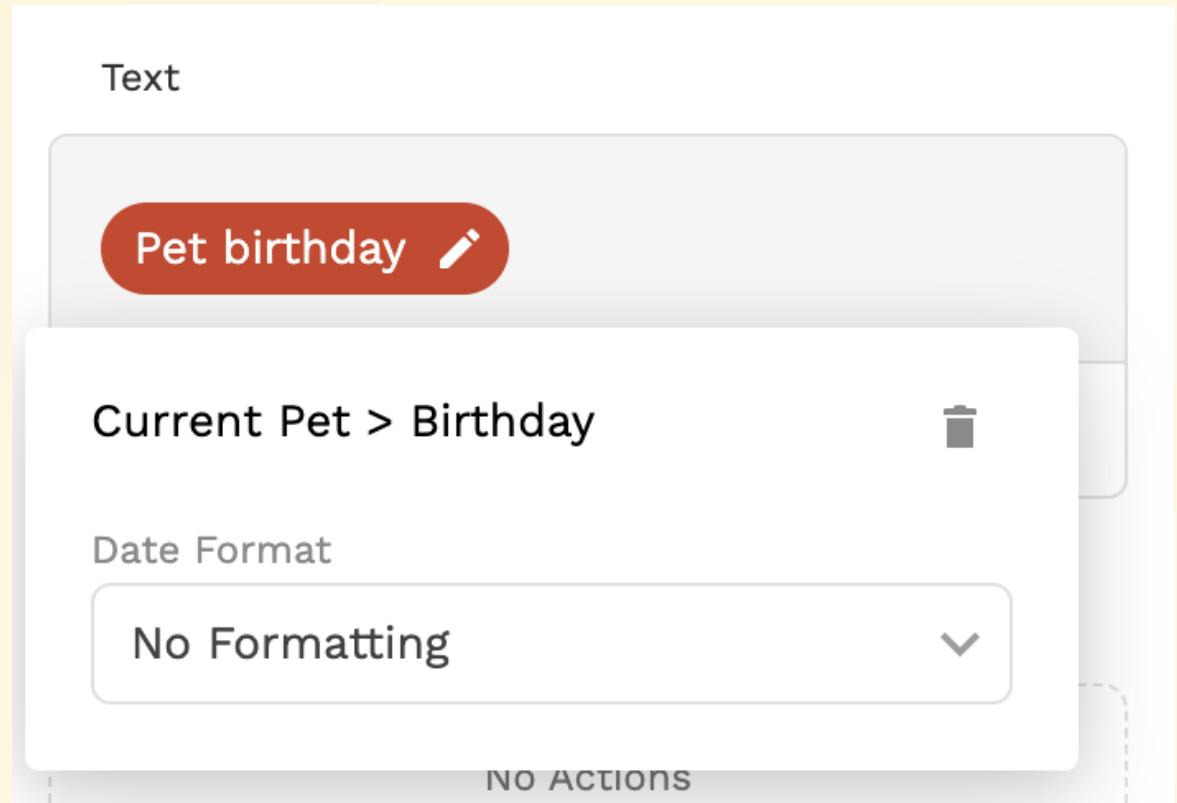
これで、特定のペットの画像が表示されるようになりました。

その他の項目も追加していきます。

- ADD COMPONENTからTextを追加し、Birthdayというラベルとその値を表示します
 - 値の表示は、Add Magic TextでCurrent Pet's > Birthdayを選択します



- Textのpet birthdayの右側の編集アイコンから、Data Formatを No Formattingに変更します



- ADD COMPONENTからTextを追加し、Latest Weightというラベルを表示します

The screenshot shows the Figma interface for a 'Text' component setup. At the top, there's a header bar with a back arrow and the text 'PETDETAIL'. Below the header, the component is labeled 'Text' and has a title 'Latest Weight'. There are two tabs at the top right: 'SETUP' (which is underlined in green) and 'LAYOUT'. The main area contains a 'Text' input field containing 'Latest Weight', with a red 'T' icon and a color palette icon below it. Below the input field is a section titled 'Click Actions' with a dashed box containing the text 'No Actions'. At the bottom right is a button labeled '+ ADD ACTION'. At the very bottom is a rounded rectangle button labeled 'EDIT STYLES'. To the right of the main interface, there's a preview window showing a mobile screen with the title 'PetDetail' and a placeholder image with three sparkles.

PETDETAIL

Text

Latest Weight

SETUP LAYOUT

Text

Latest Weight

T

Click Actions

No Actions

+ ADD ACTION

EDIT STYLES

PetDetail

Birthday

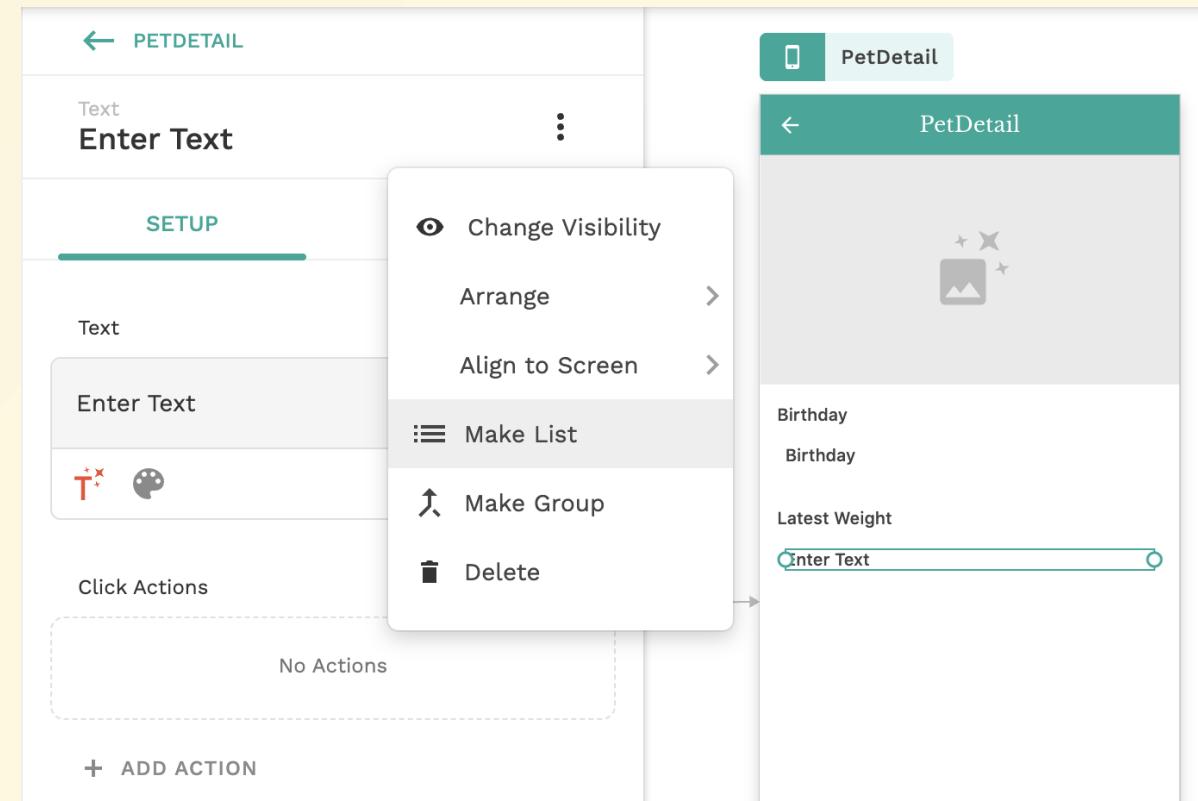
Birthday

Latest Weight

140

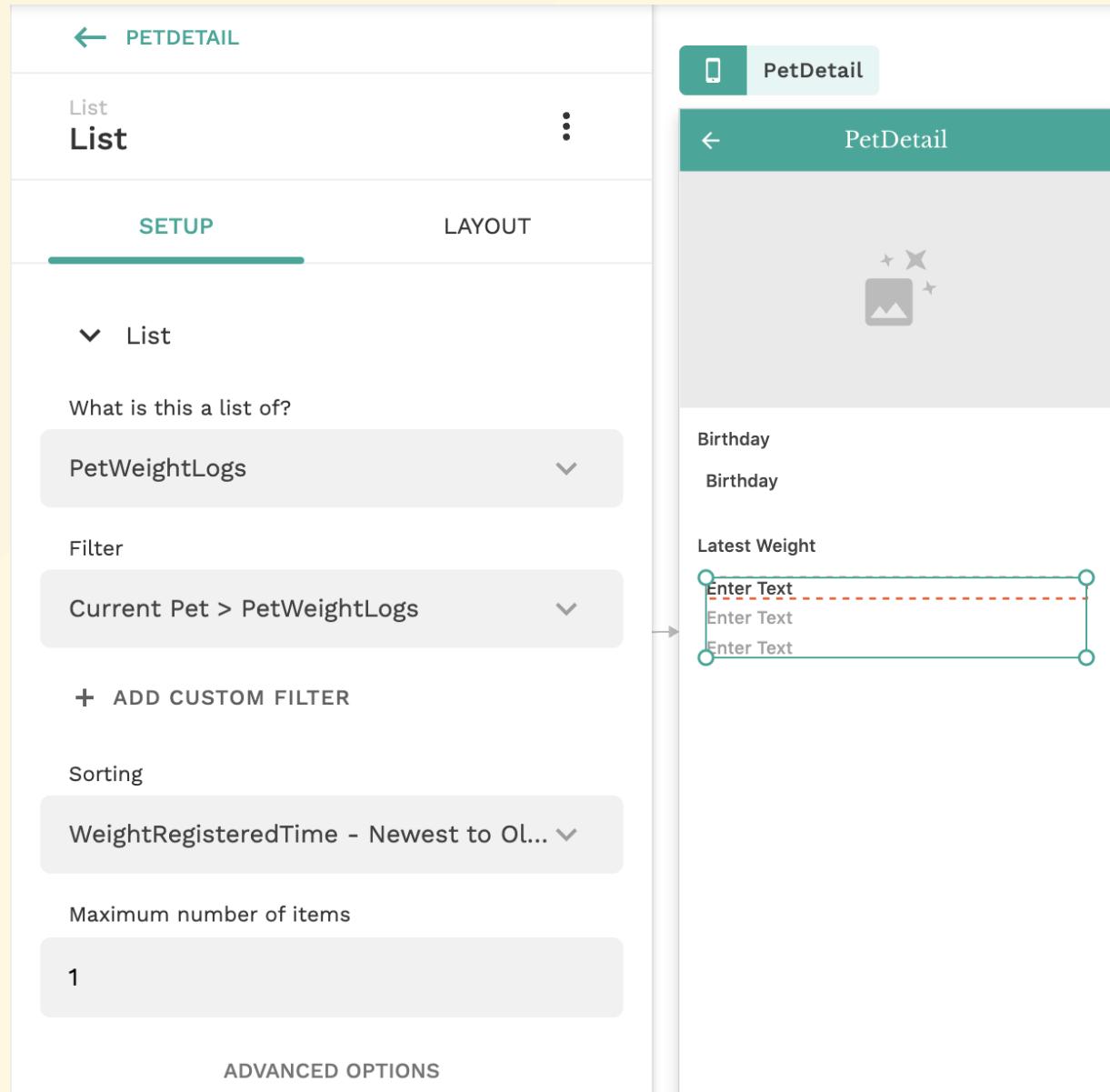
Latest Weightの値の表示は、少し工夫が必要です。

- ADD COMPONENTからTextを追加し、Make Listでリストにします。

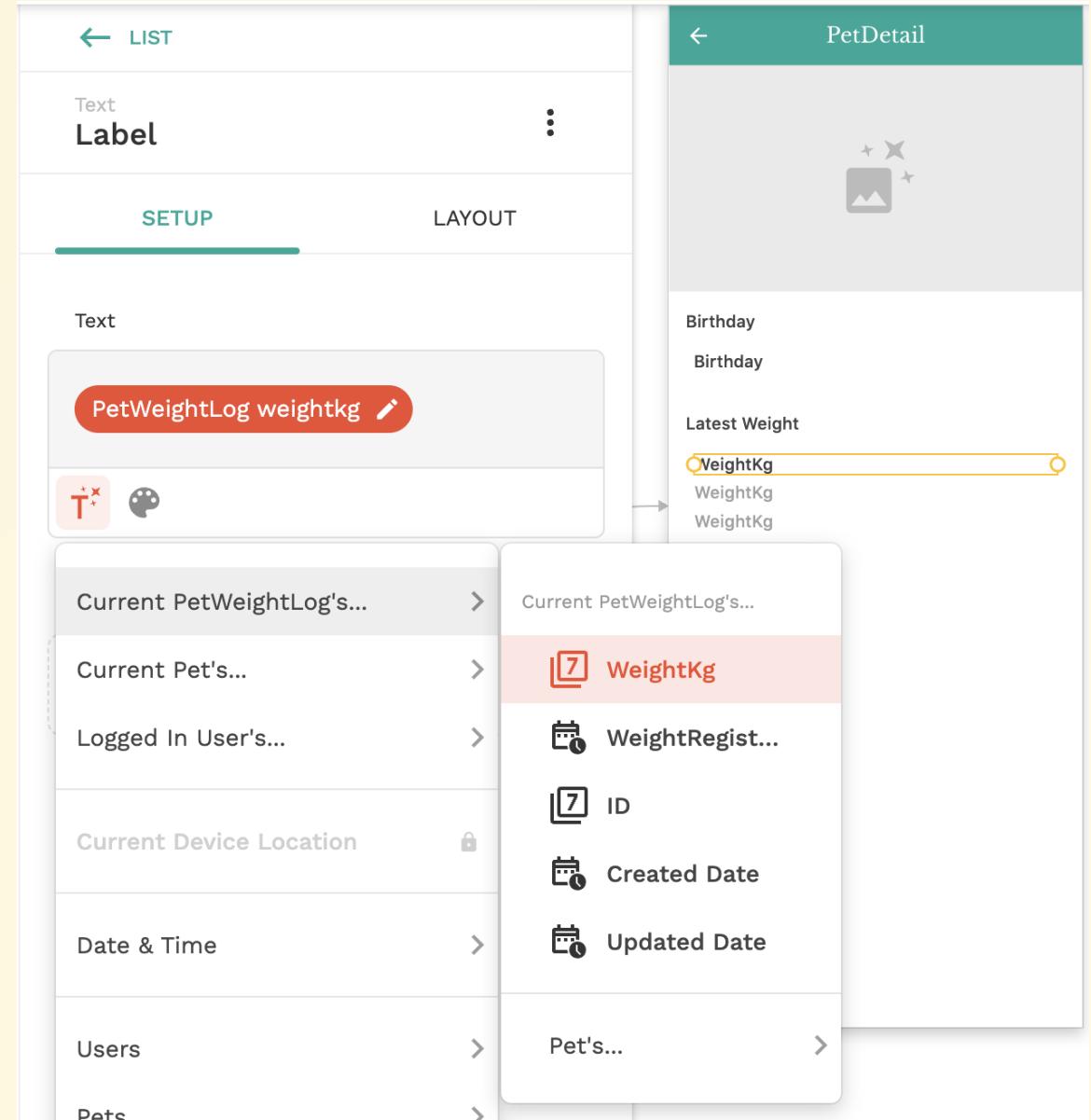


- What is this a list of?で PetWeightLogsを選択
- Filterで Current Pet > PetWeightLogsを選択
- Sortingで WeightRegisteredTime - Newest to Oldestを選択
- Maximum number of items に1を設定

これにより、最新の1件だけに絞り込まれます。

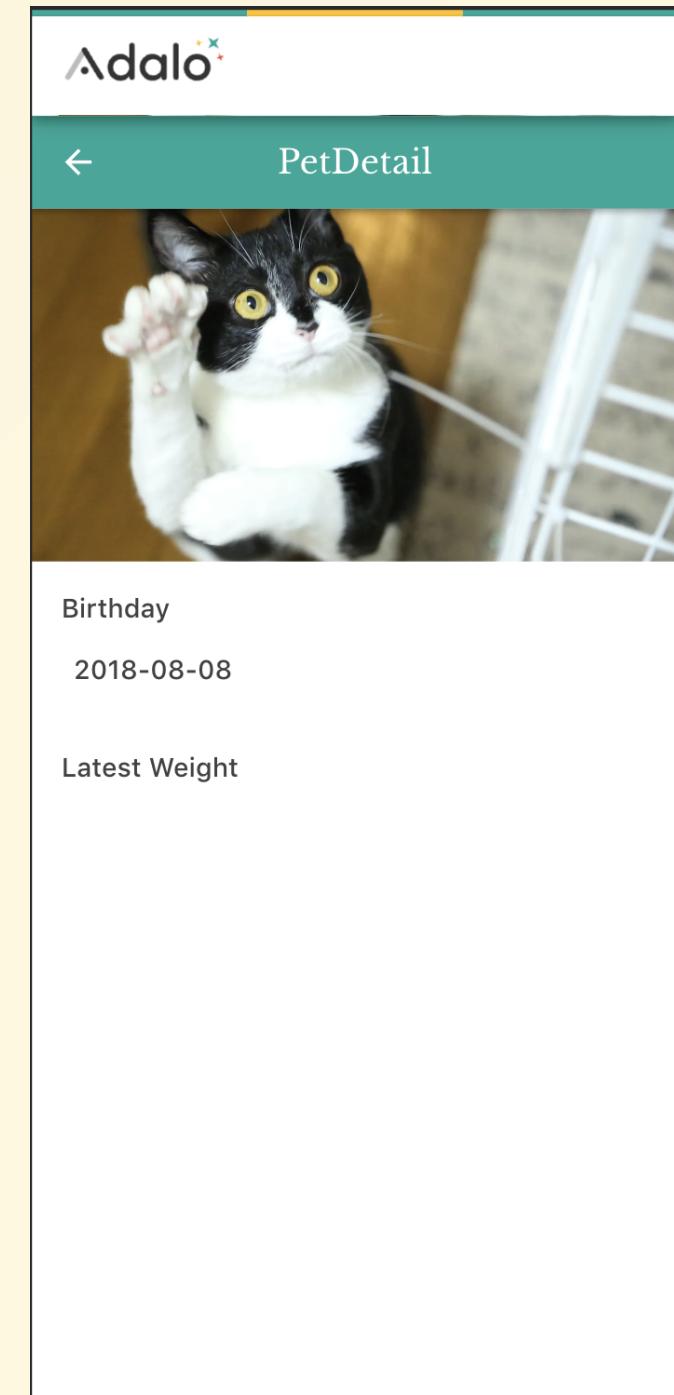


- List内のTextコンポーネントの値は、Current PetWeightLog's > WeightKg に設定します



- プレビュー画面で動作を確認しましょう

まだPetWeightLogのデータが登録されていないので、Latest Weightは表示されませんが、現時点ではこれでOKです。

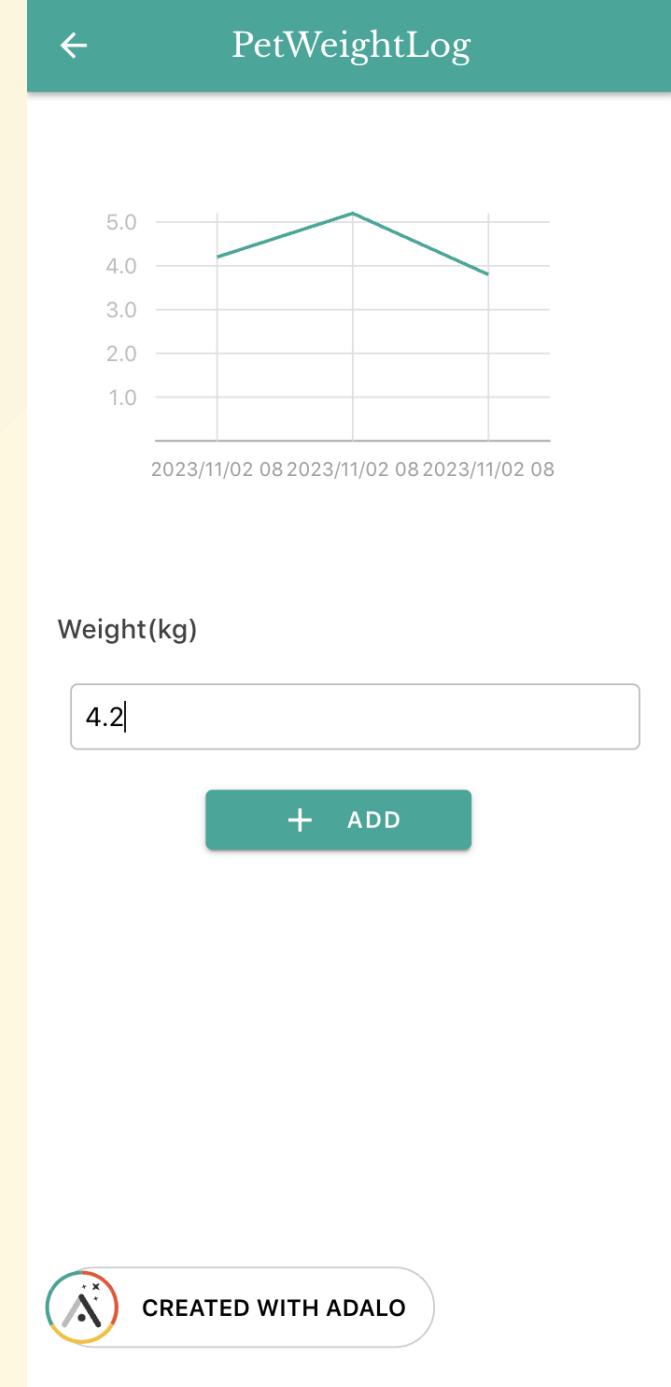




体重記録画面

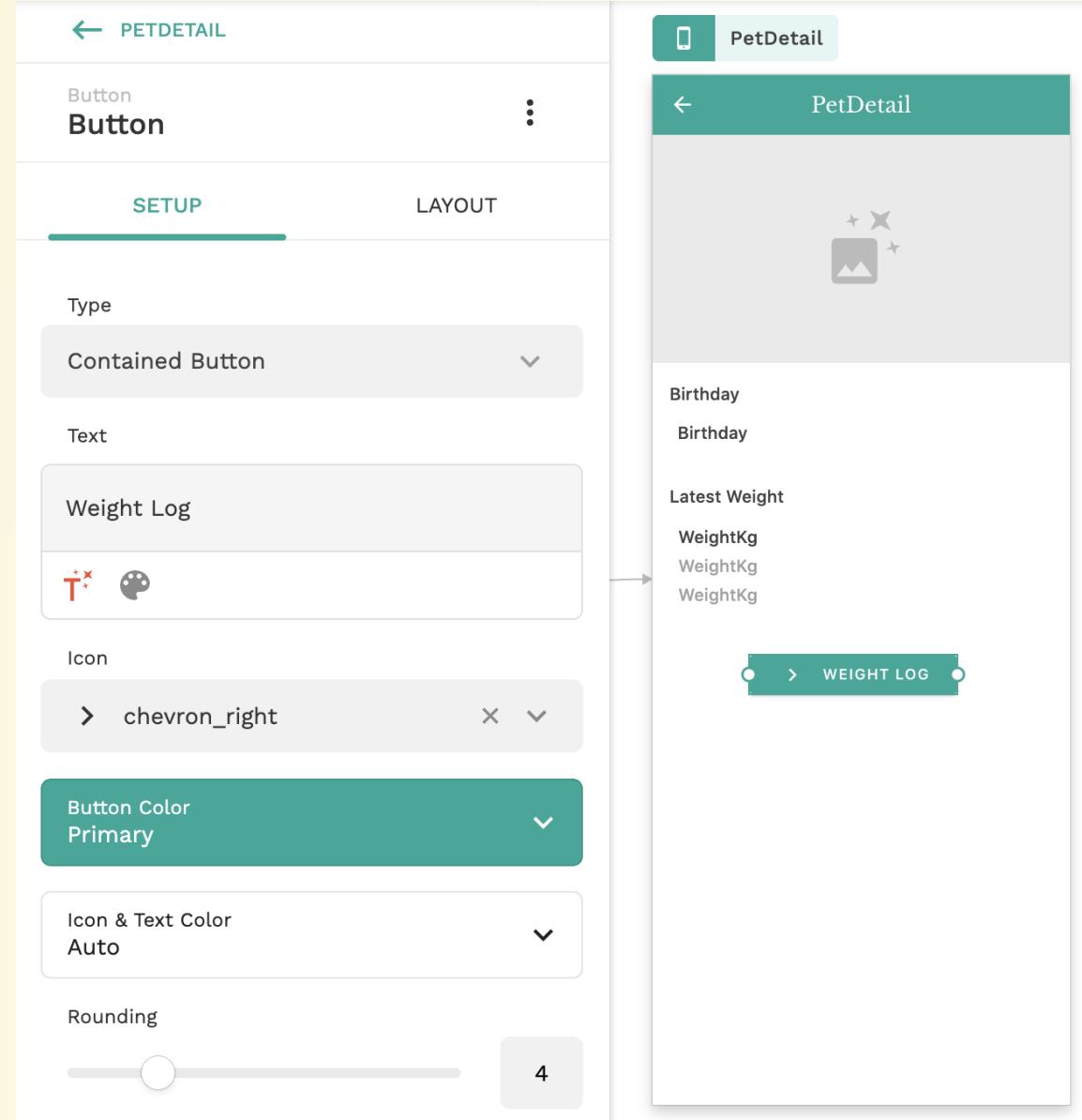
- 体重の遷移を示すグラフが表示される
- 現在の体重を入力できる
- ボタンを押して体重を追加できる

最後に、こちらの画面を作りましょう

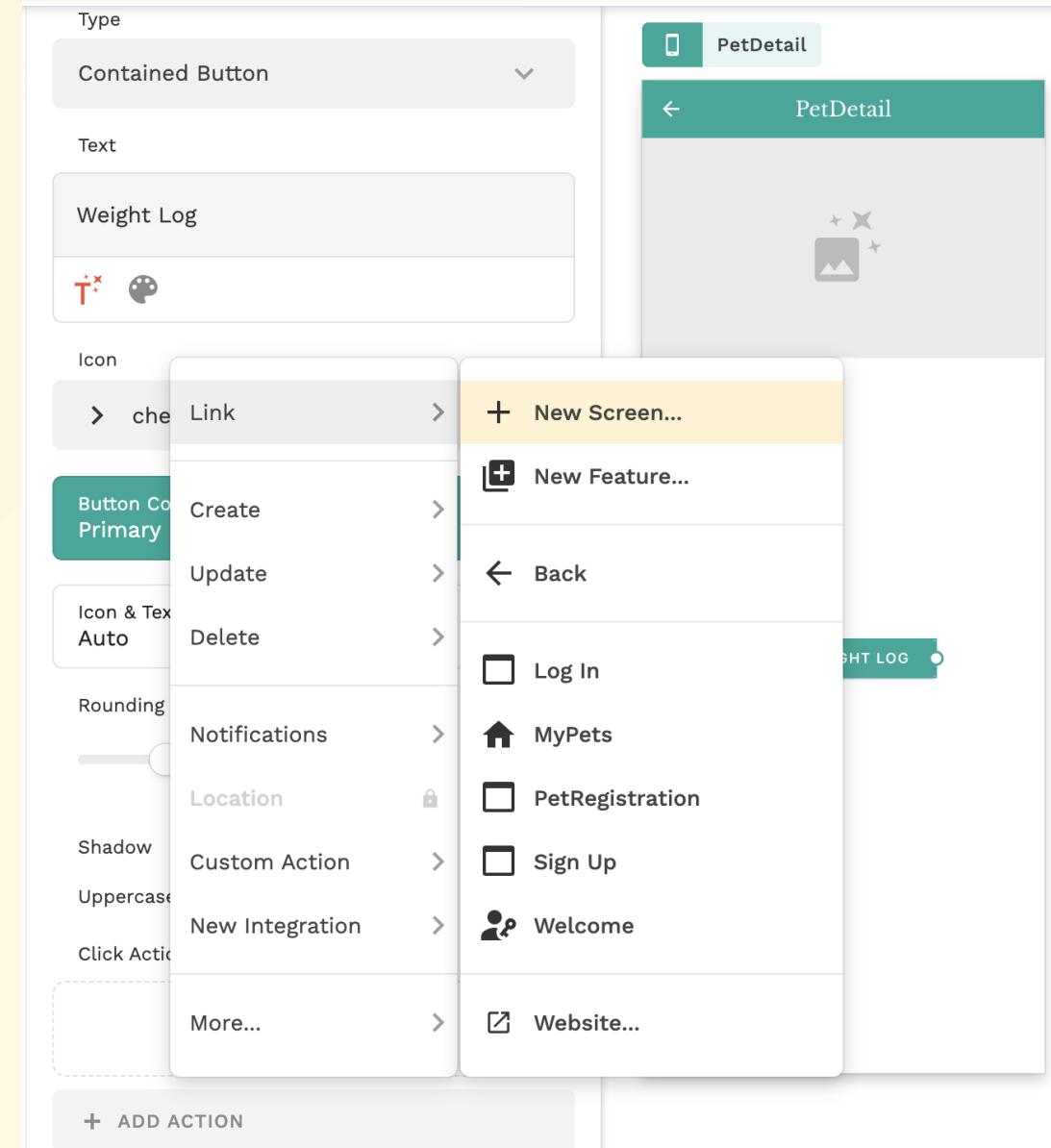


ペット詳細画面に、体重記録画面への導線となるボタンを追加します

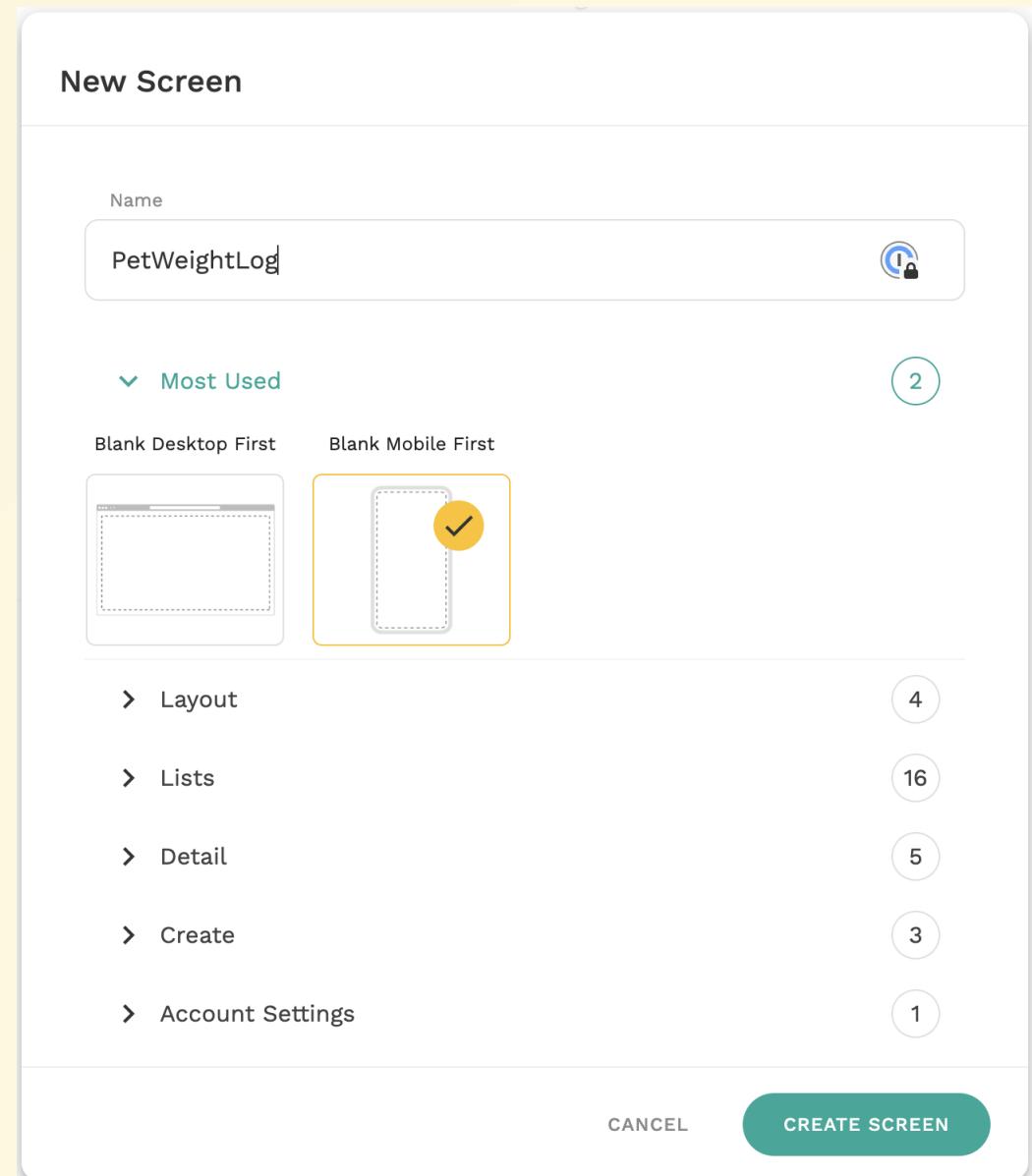
- ADD COMPONENTから Buttonを選択して、画面上に配置します
- TextをWeight Logに変更します
- Iconをchevron_rightに変更します



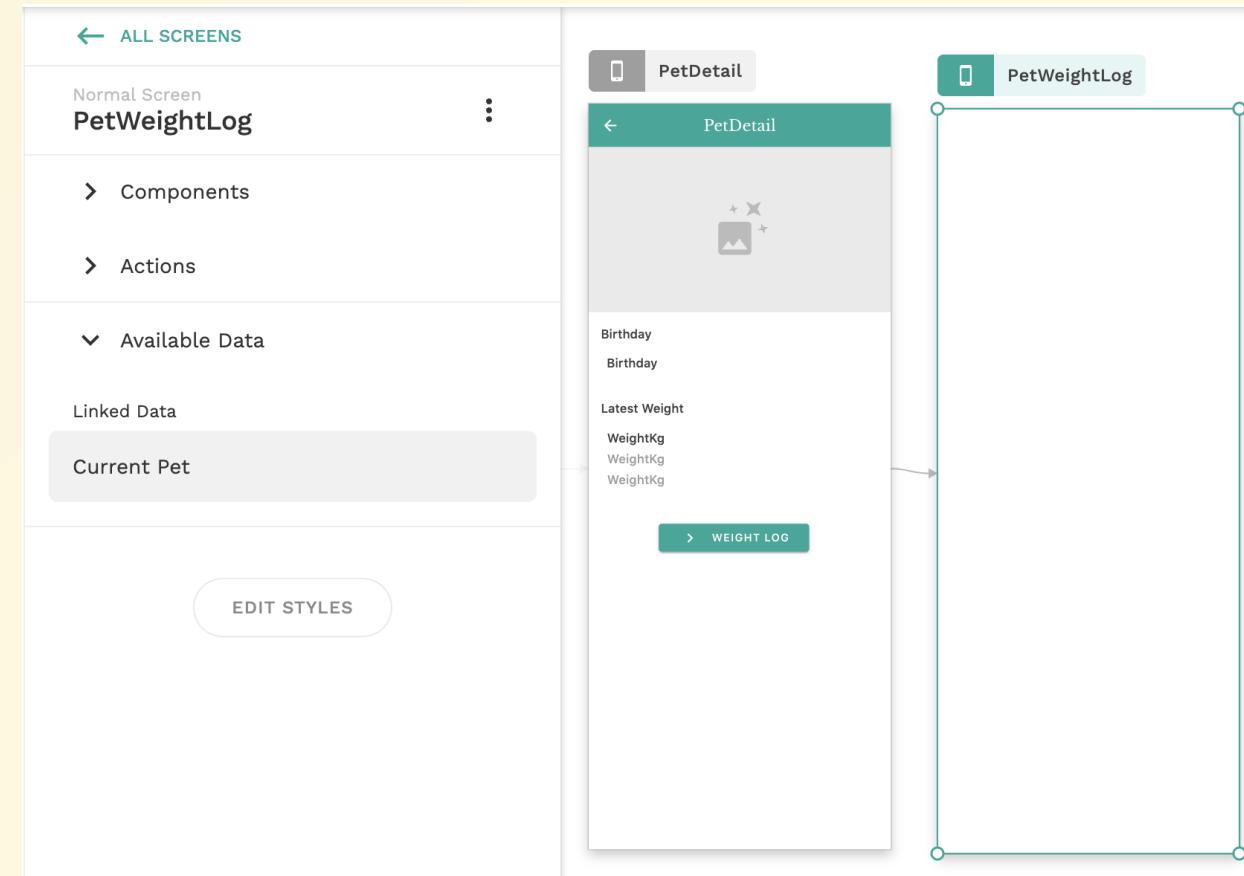
- ADD ACTION > Link > New Screenを選択します



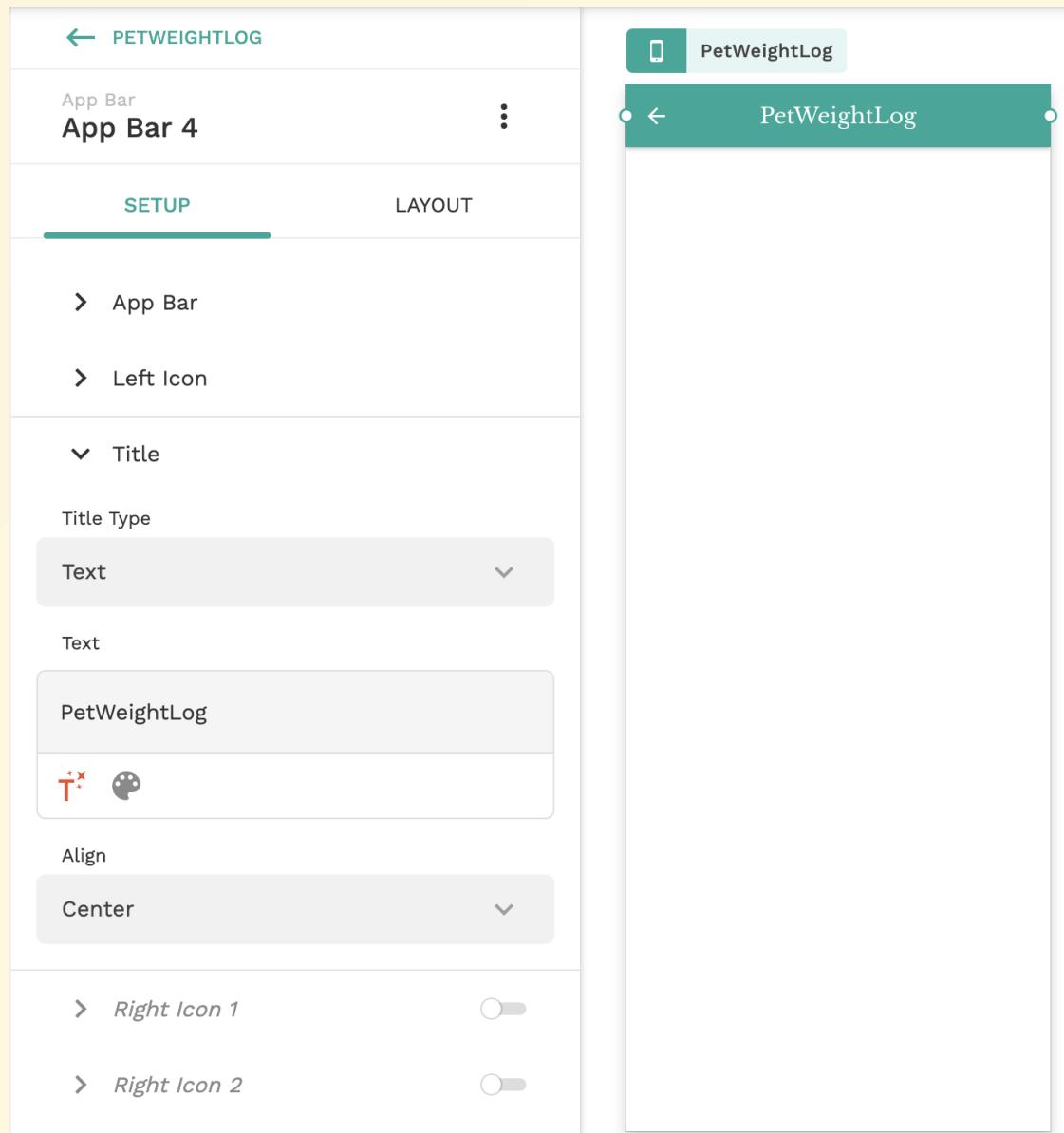
- Screen Nameを入力し、Blank Mobile Firstを選択して、画面を作成します



作成された画面では、ペット
詳細画面から受け渡された
Current Petのデータを扱える
ようになっています

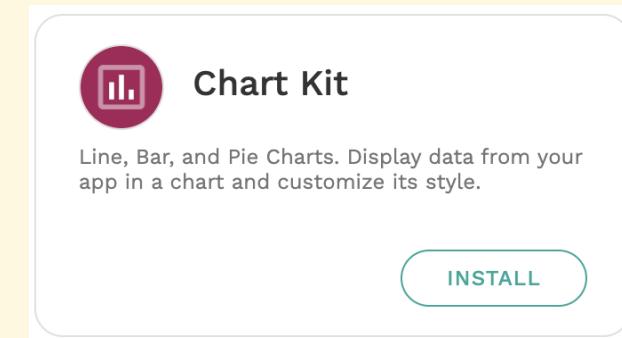
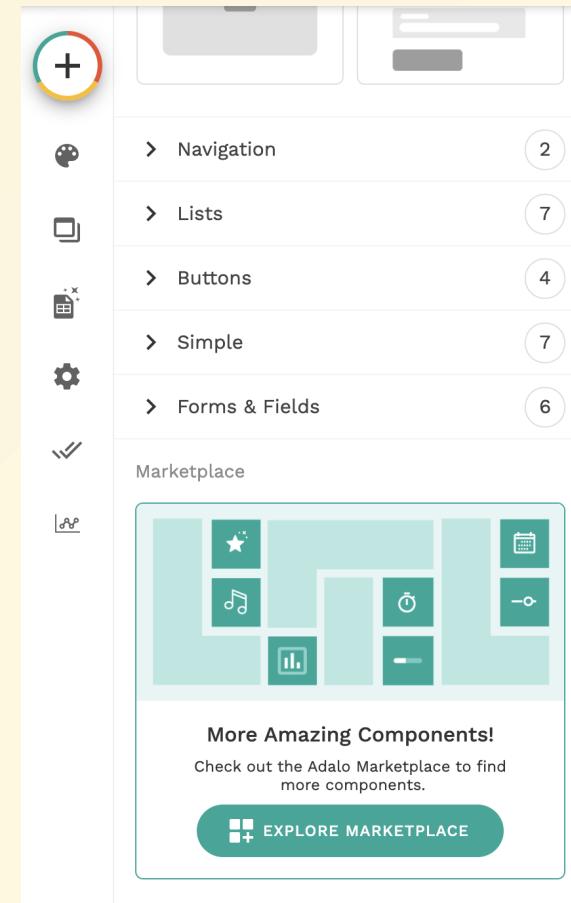


- ADD COMPONENTからApp Barを選択して、画面上に配置します
- Title > Textの値を画面名(PetWeightLog)に変えます

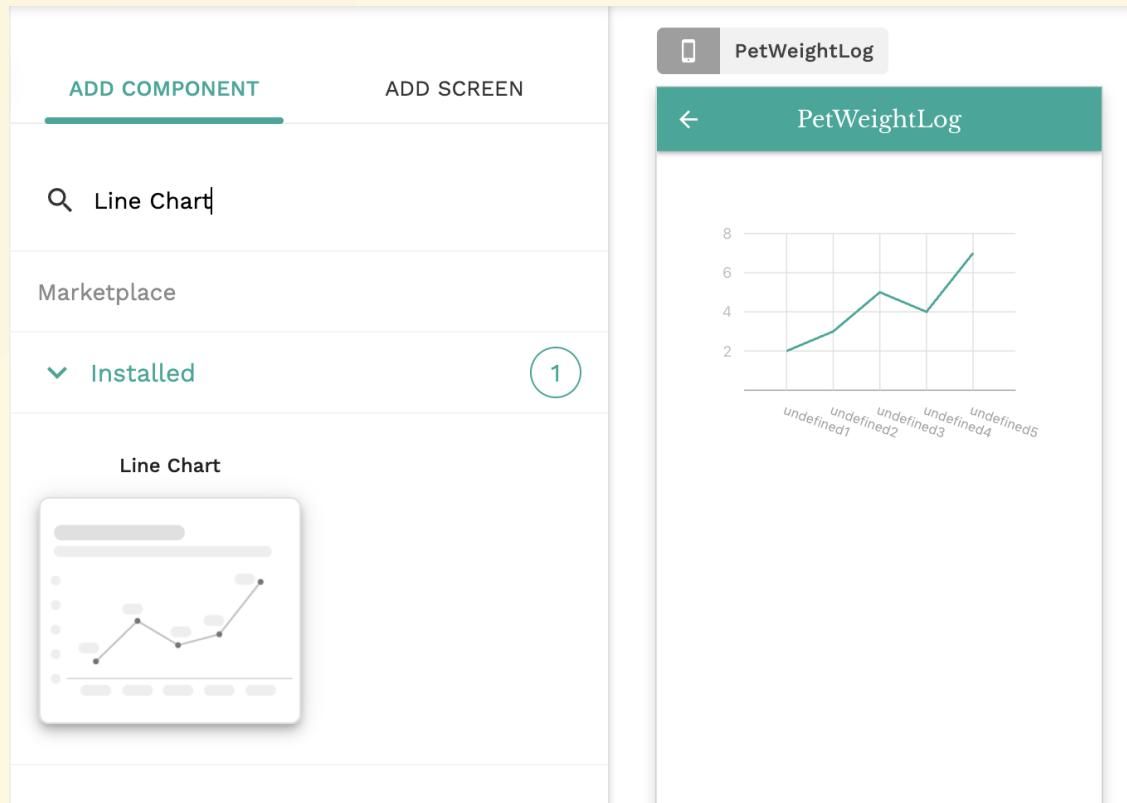


次に、過去に登録したペットの体重がグラフとして表示されるようにしていきます。

- ADD COMPONENTから EXPLORE MARKETPLACEを選択
- Chart KitをINSTALL



- Line Chartを画面に追加



Line Chartを設定します。

- What is this a chart of?で PetWeightLogsを選択
- FilterでCurrent Pet > PetWeightLogsを選択
- Custom Filterに WeightRegisteredTime Is after 30 days ago を設定し、表示期間を指定
- Sortingで WeightRegisteredTime - Oldest to Newestを選択

▼ Line Chart

What is this a chart of?

PetWeightLogs

Filter

Current Pet > PetWeightLogs

Custom Filter

WeightRegisteredTime

Is after

30 days ago ×

+ AND

+ ADD CUSTOM FILTER

Sorting

WeightRegisteredTime - Oldest to Ne...

Maximum number of items

No Maximum

ADVANCED OPTIONS

The screenshot shows a configuration interface for a line chart titled 'Line Chart'. It includes sections for 'What is this a chart of?' (set to 'PetWeightLogs'), 'Filter' (set to 'Current Pet > PetWeightLogs'), 'Custom Filter' (with a dropdown for 'WeightRegisteredTime' and a condition 'Is after' set to '30 days ago'), 'Sorting' (set to 'WeightRegisteredTime - Oldest to Ne...'), and 'Maximum number of items' (set to 'No Maximum'). There is also an 'ADVANCED OPTIONS' button at the bottom.

- X Axis Valueに
PetWeightLog >
WeightRegisteredTimeを設
定
 - Date FormatにDate /
Timeを設定
- Y Axis Valueに
PetWeightLog > WeightKg
を設定

The image shows two side-by-side panels from a software application's configuration interface.

Left Panel (X Axis Value):

- X Axis Value:** A red button labeled "PetWeightLog weightregisteredtime" with a pencil icon.
- Y Axis Value:** A red button labeled "PetWeightLog weightkg" with a pencil icon.
- Point Action:** A section containing a dashed box labeled "No Actions". Below it is a button "+ ADD ACTION".

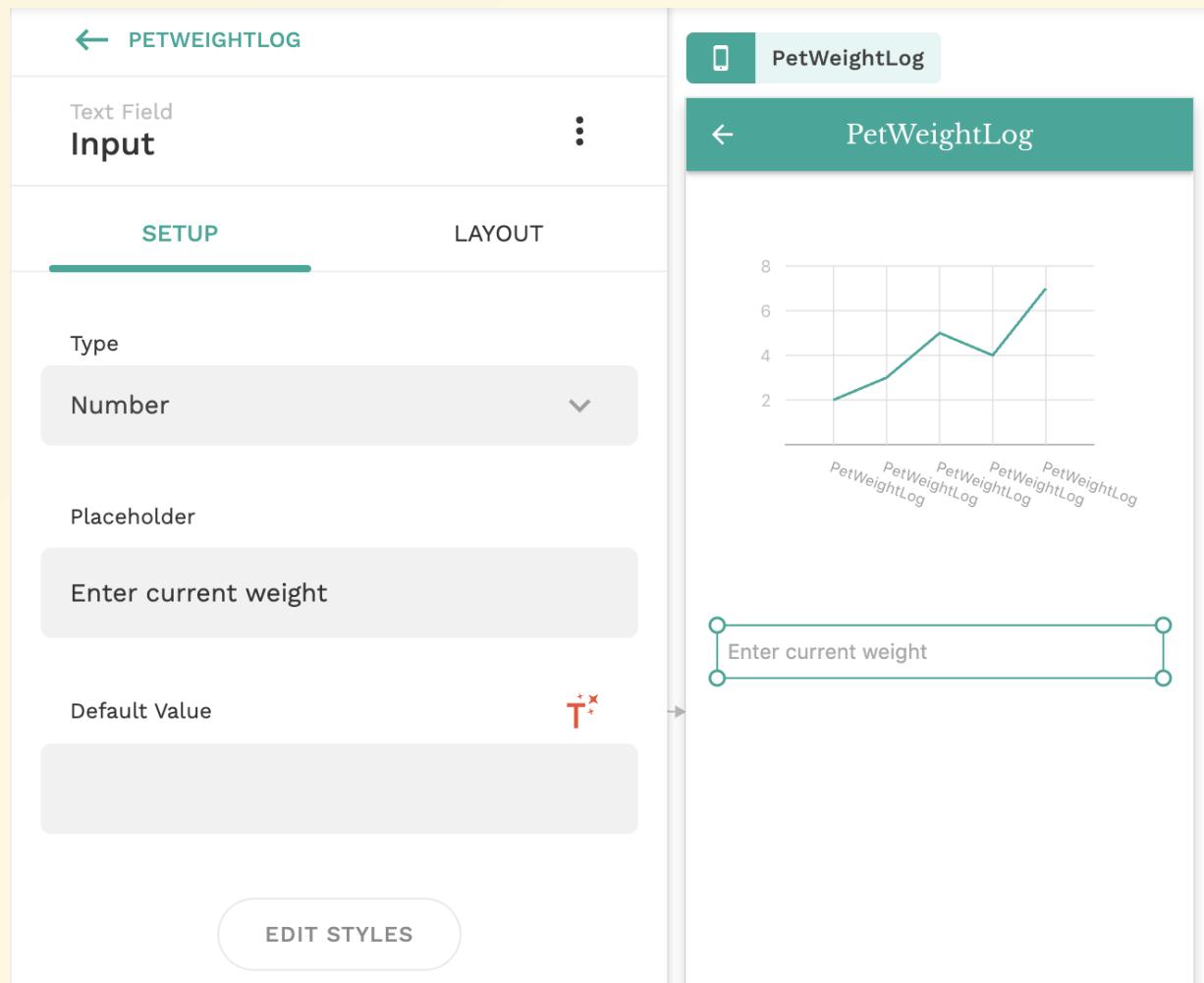
Right Panel (Current PetWeightLog > WeightRegisteredTime):

- Date Format:** A dropdown menu set to "Date / Time".

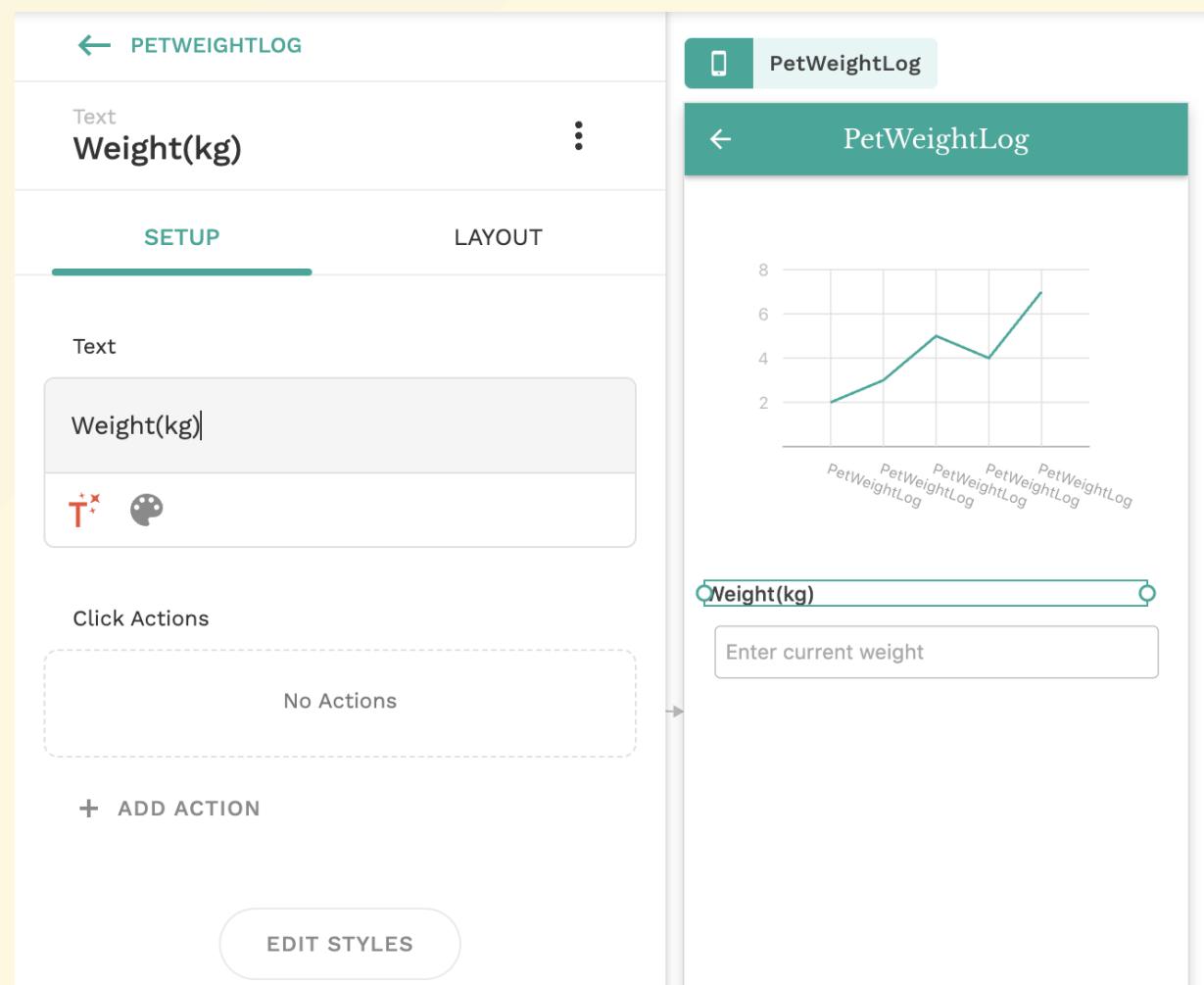
これで、グラフを表示するた
めの設定ができました。

次に、体重を記録できるよう
にしましょう。

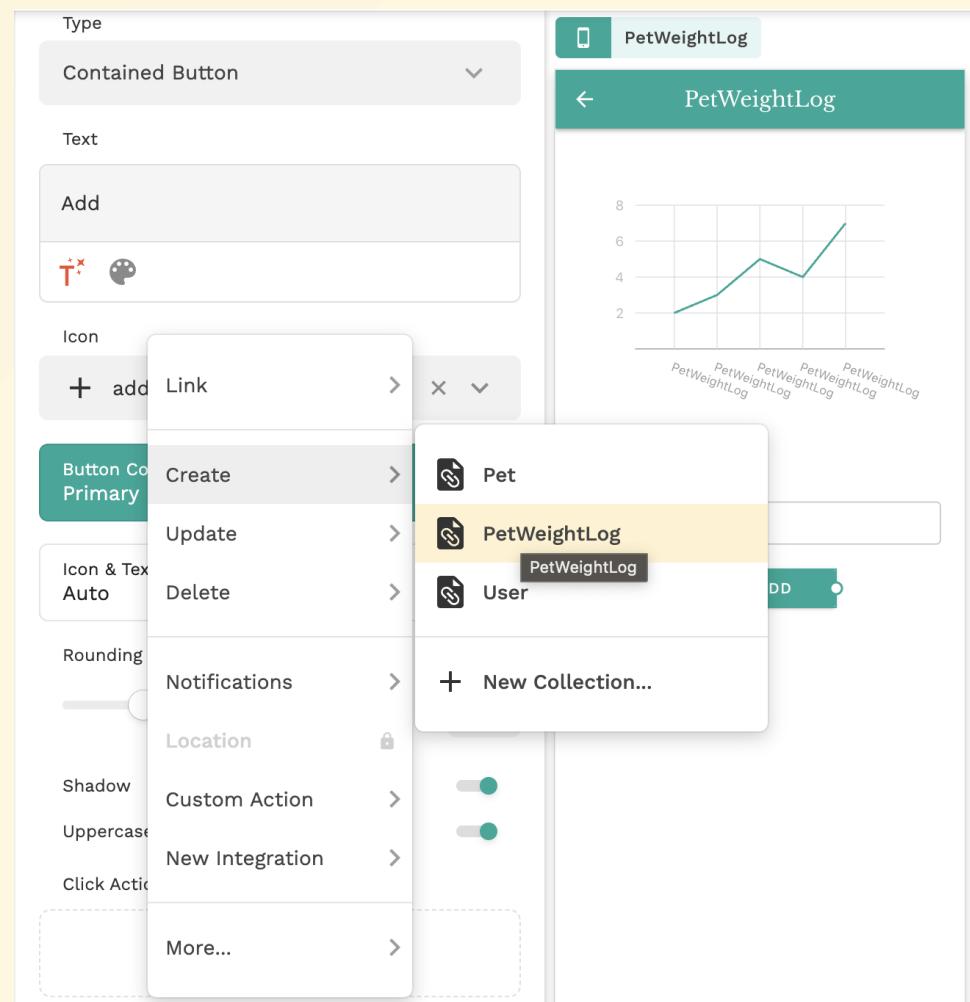
- ADD COMPONENTから、
Text Inputを追加してください
- TypeをNumberに変更して
ください
- PlaceholderをEnter current
weightに変更してください



- ADD COMPONENTからTextを追加してください
- 値をWeight(kg)に変更してください



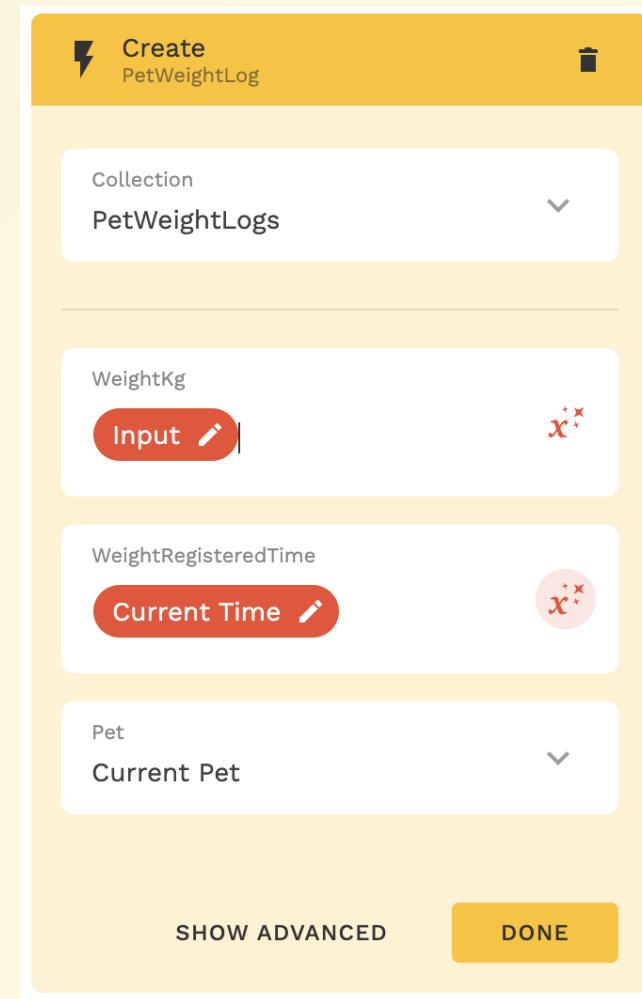
- ADD COMPONENTから Buttonを追加してください
- TextをAddに変更してください
- ADD ACTIONからCreate > PetWeightLogを選択してください



Actionに対して、以下のように設定します。

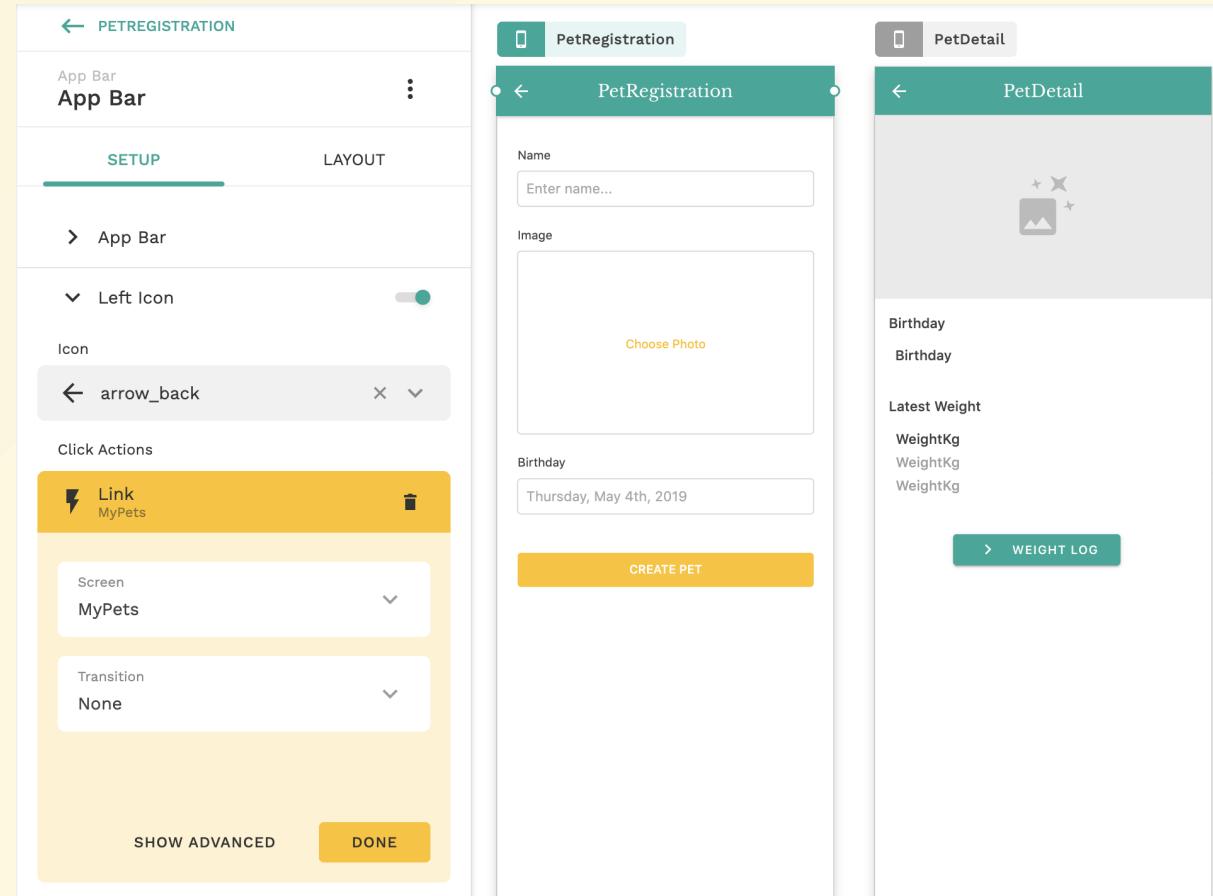
- WeightKg: Other Components > Input
- WeightRegisteredTime: Date & Time > Current Time
- Pet: Current Pet

これで体重記録画面が作成できました。

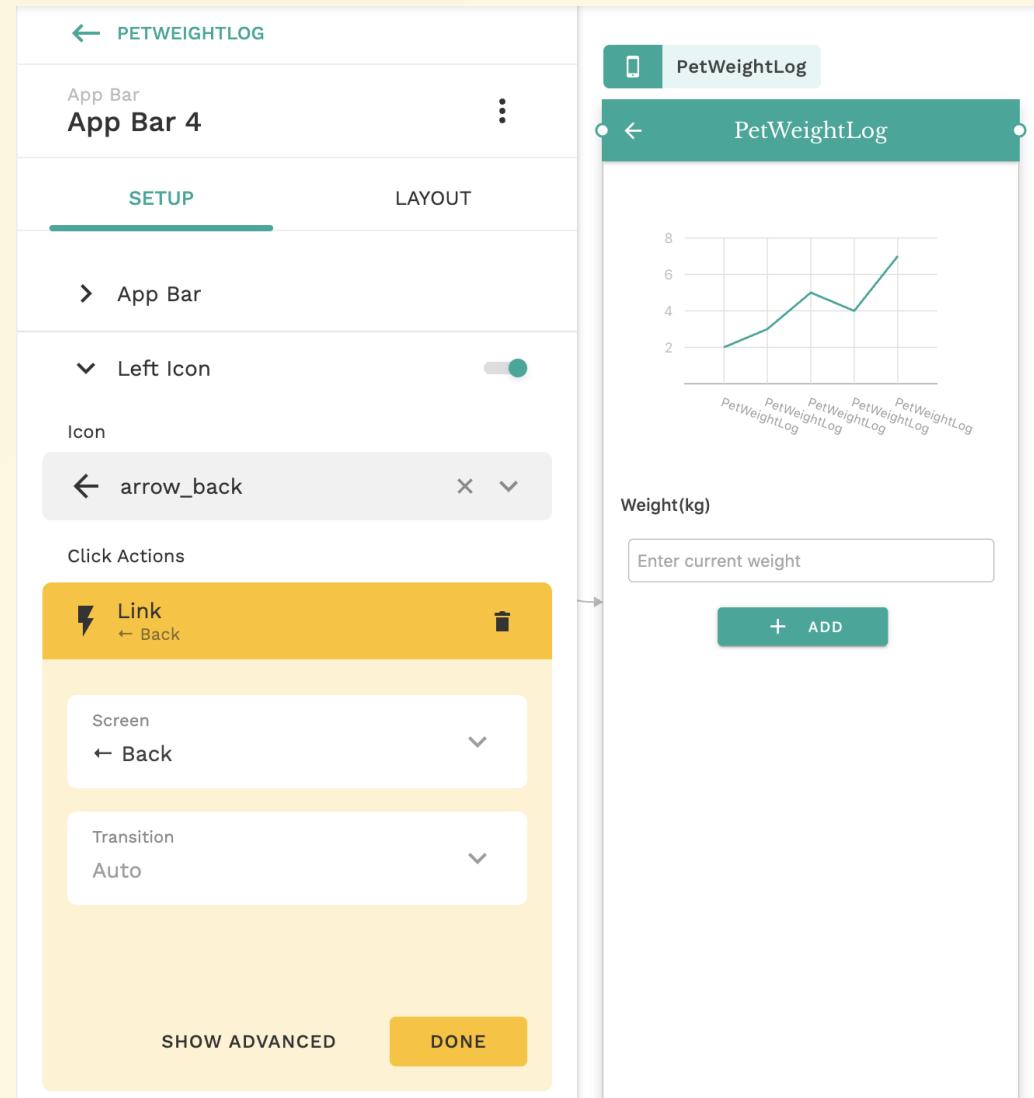


画面を行き来して動作が確認できるように、ヘッダーの戻るアイコンにリンクを設定しましょう。

- ペット登録画面、ペット詳細画面のそれぞれで、App BarのLeft IconにMyPetsへのリンクを追加します

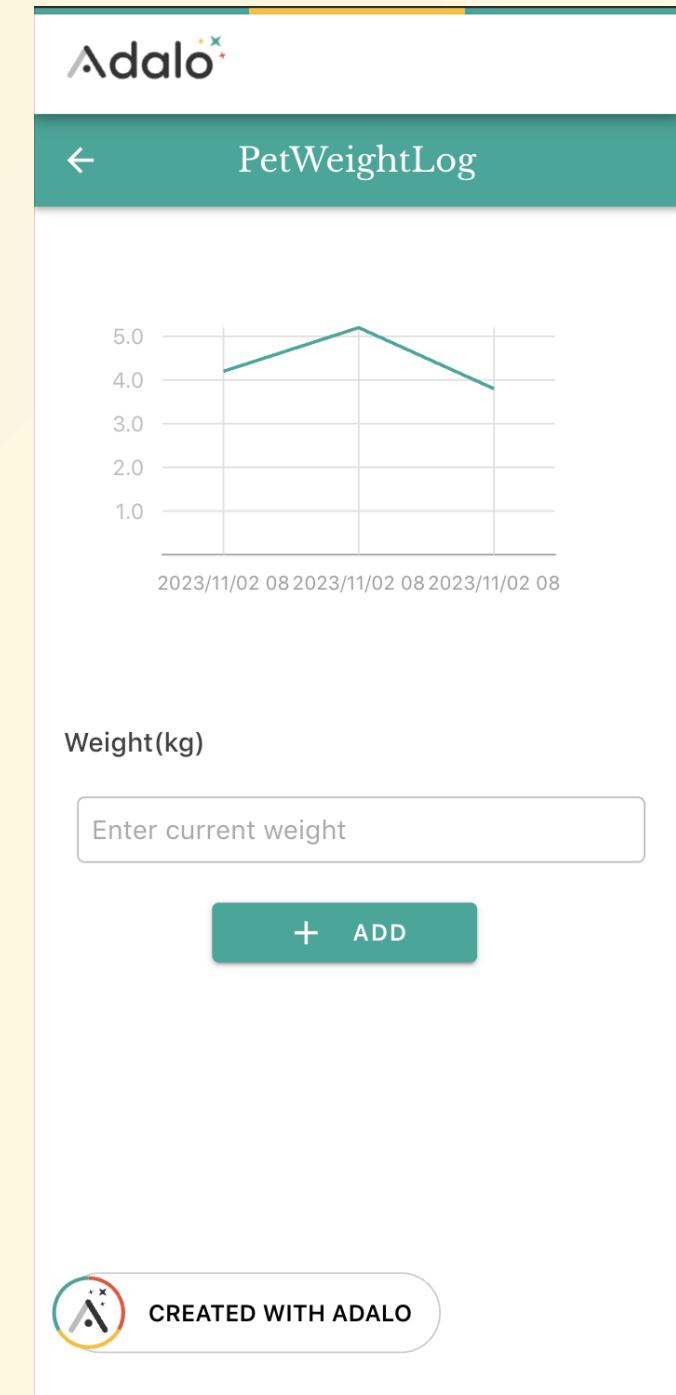


- 体重記録画面では、App Bar のLeft Iconのリンクで、Backを選択します

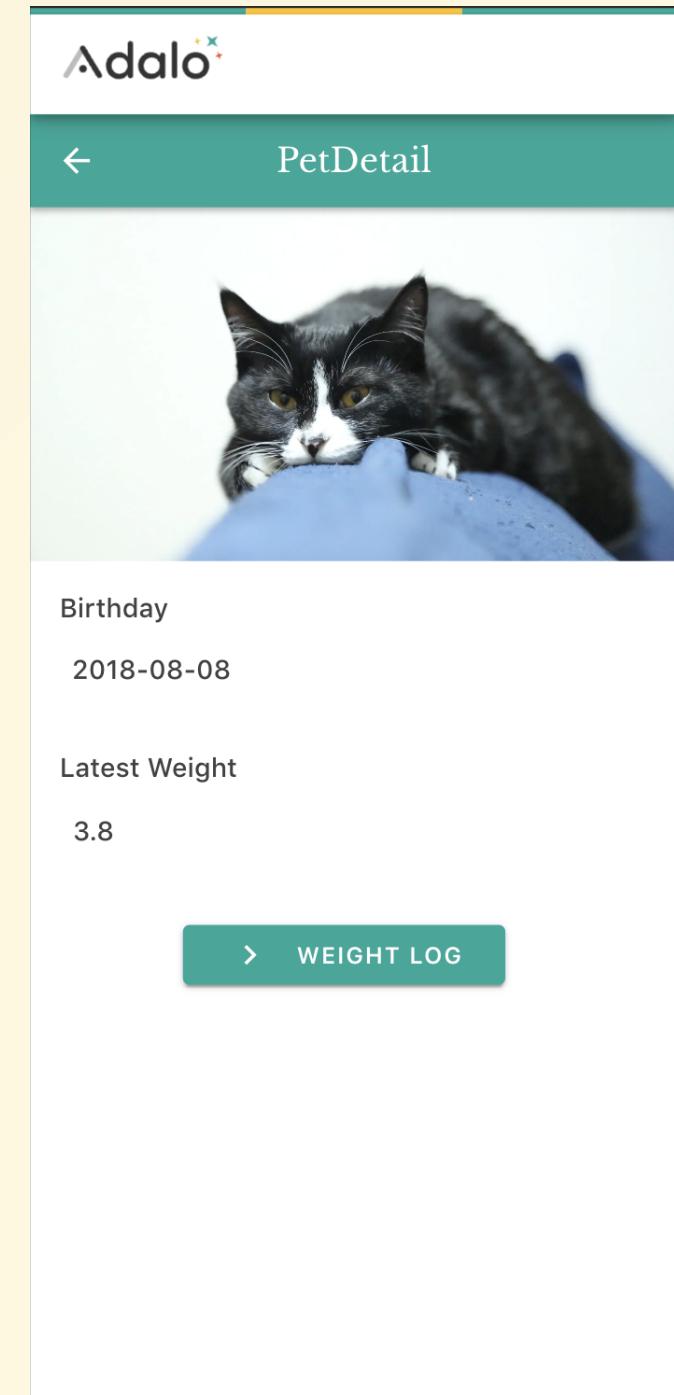


Preview機能でグラフの表示を確認しましょう。体重を複数追加すると、グラフが描画されます。

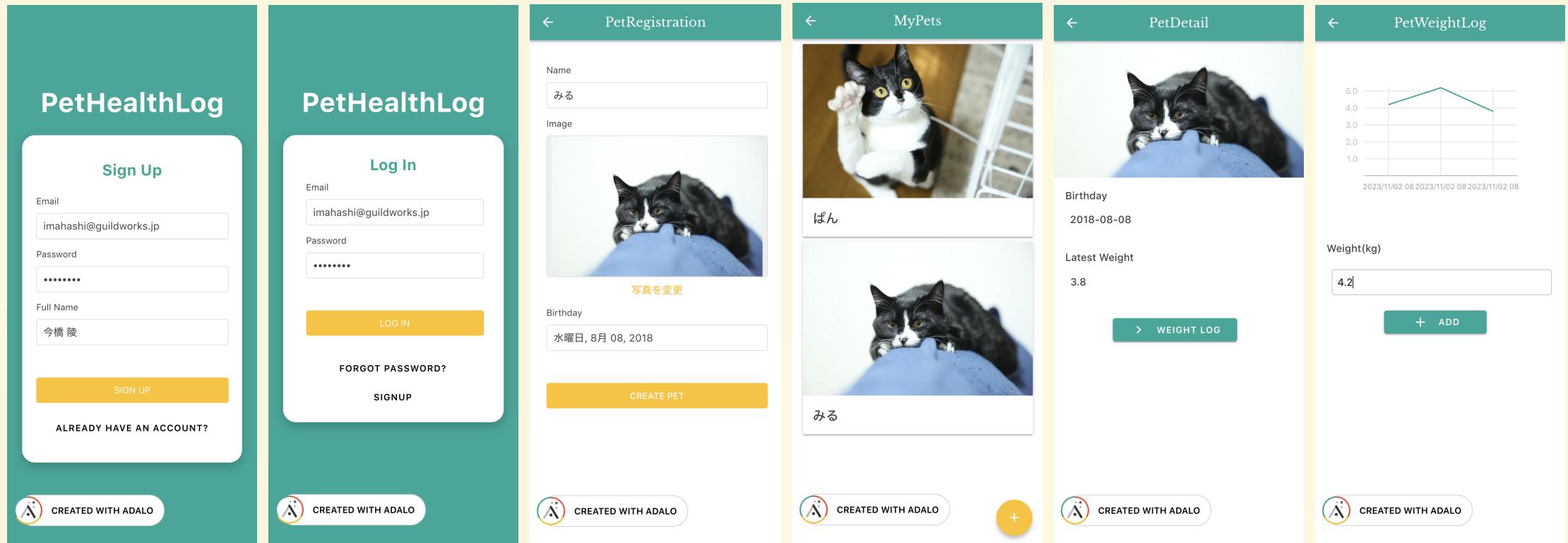
※ 体重の登録日時が長く、省略して表示されてしまいます。同じ日に複数の体重を登録してテストをしたかったのでDate&Time型にしていますが、本来はDate型にして、1日に1回しか登録できないよう制御する方が良さそうです。



ペット詳細画面でLatest Weightが表示されるようになったことも確認できます。



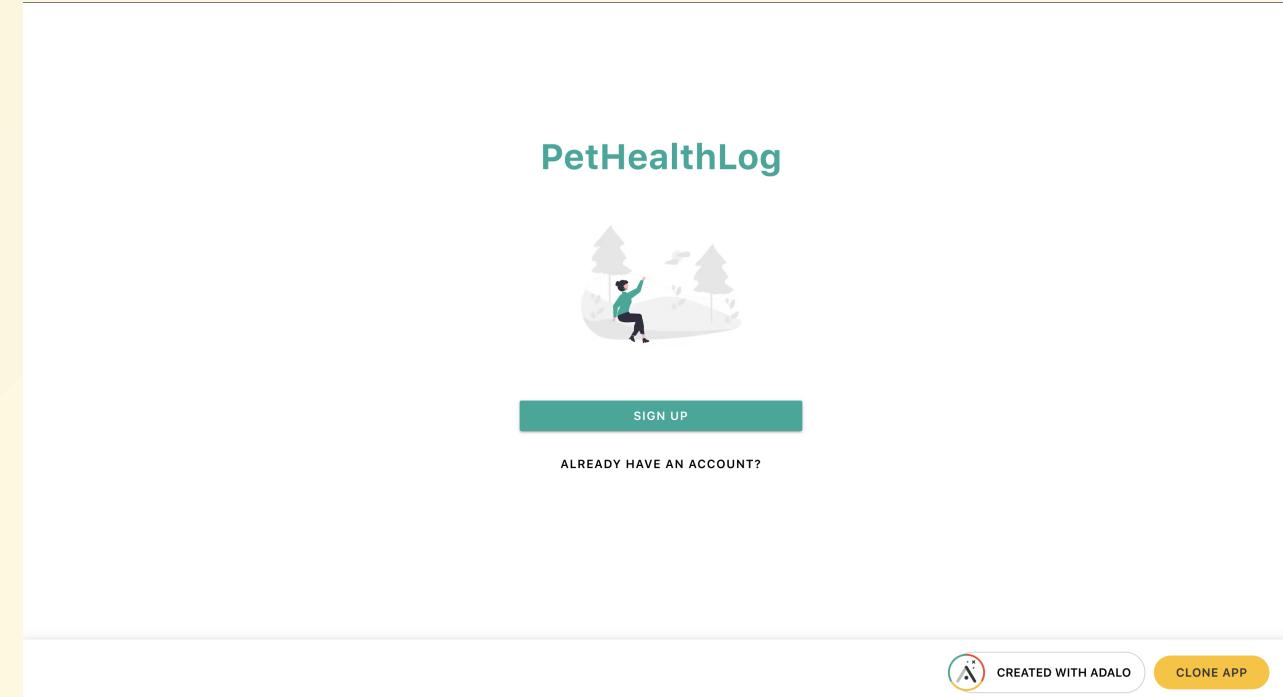
アプリの画面を一通り作成することができました 🎉



クローン用URL

- 以下のURLの右下のCLONE APPボタンでアプリをクローンできますので、答え合わせに使ってください

<https://ryo-imahashisteam-6.adalo.com/pethealthlog>

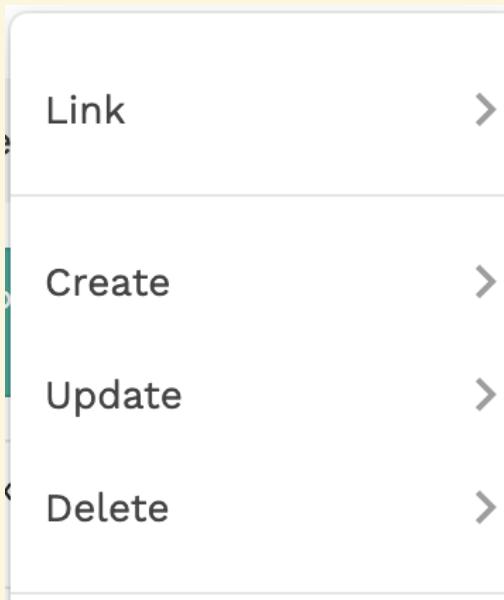


演習1

1. 登録済のペットの情報を編集できる、ペット編集画面を作成しましょう
2. 登録済のペットを削除できる、ペット削除機能を追加しましょう

ヒント

- ・ペット編集画面を作る時は、ペット登録画面の作り方が参考になります
- ・ペット削除機能のために新規の画面は不要なので、どこに削除ボタンを追加するのが良いかを考えてください
- ・ACTIONを追加する時に、UpdateやDeleteが選べます





演習1の答え合わせ

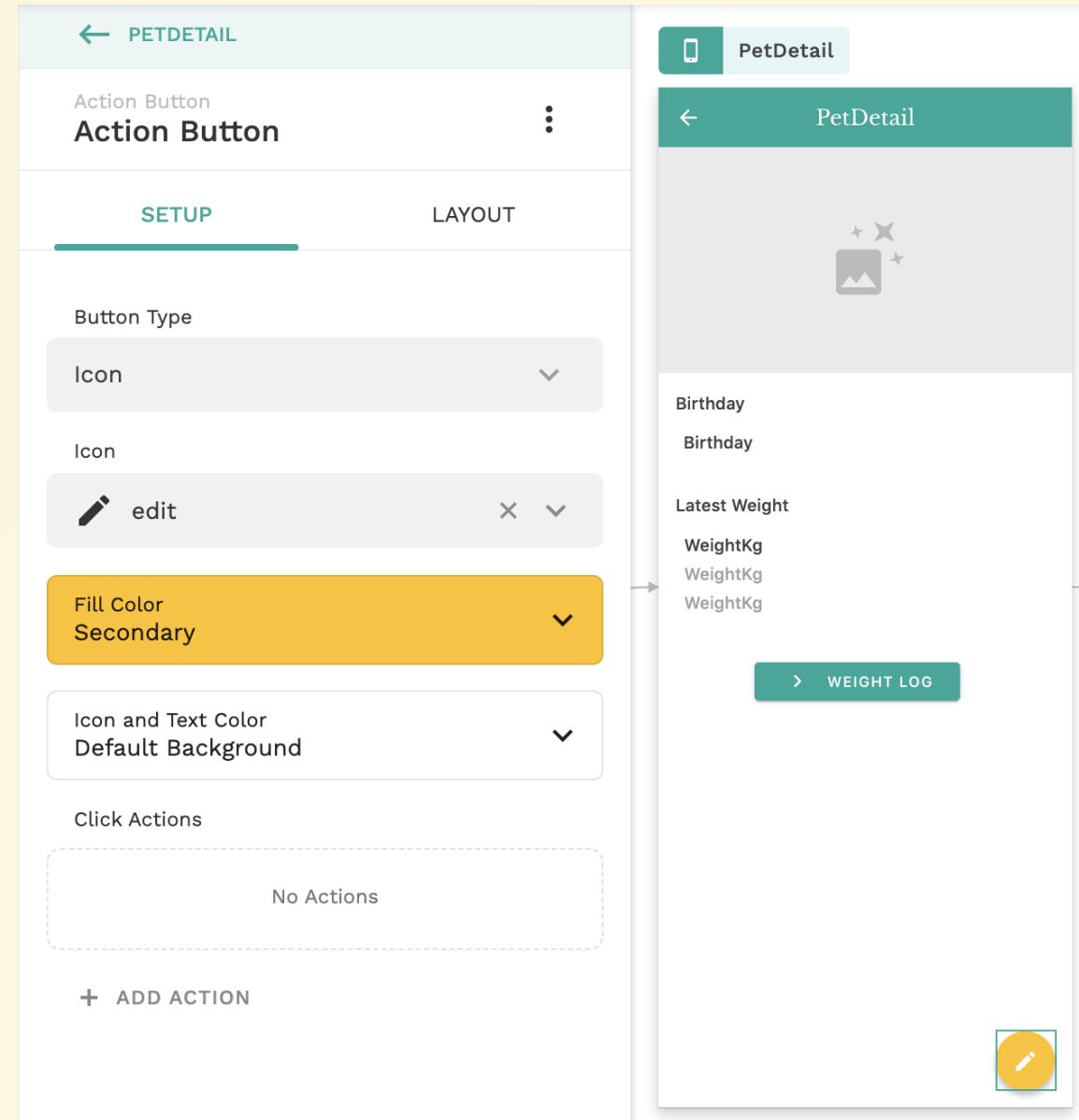
まずは自分でやってみて、その後でここから先を見てください 🙏

ペット編集画面

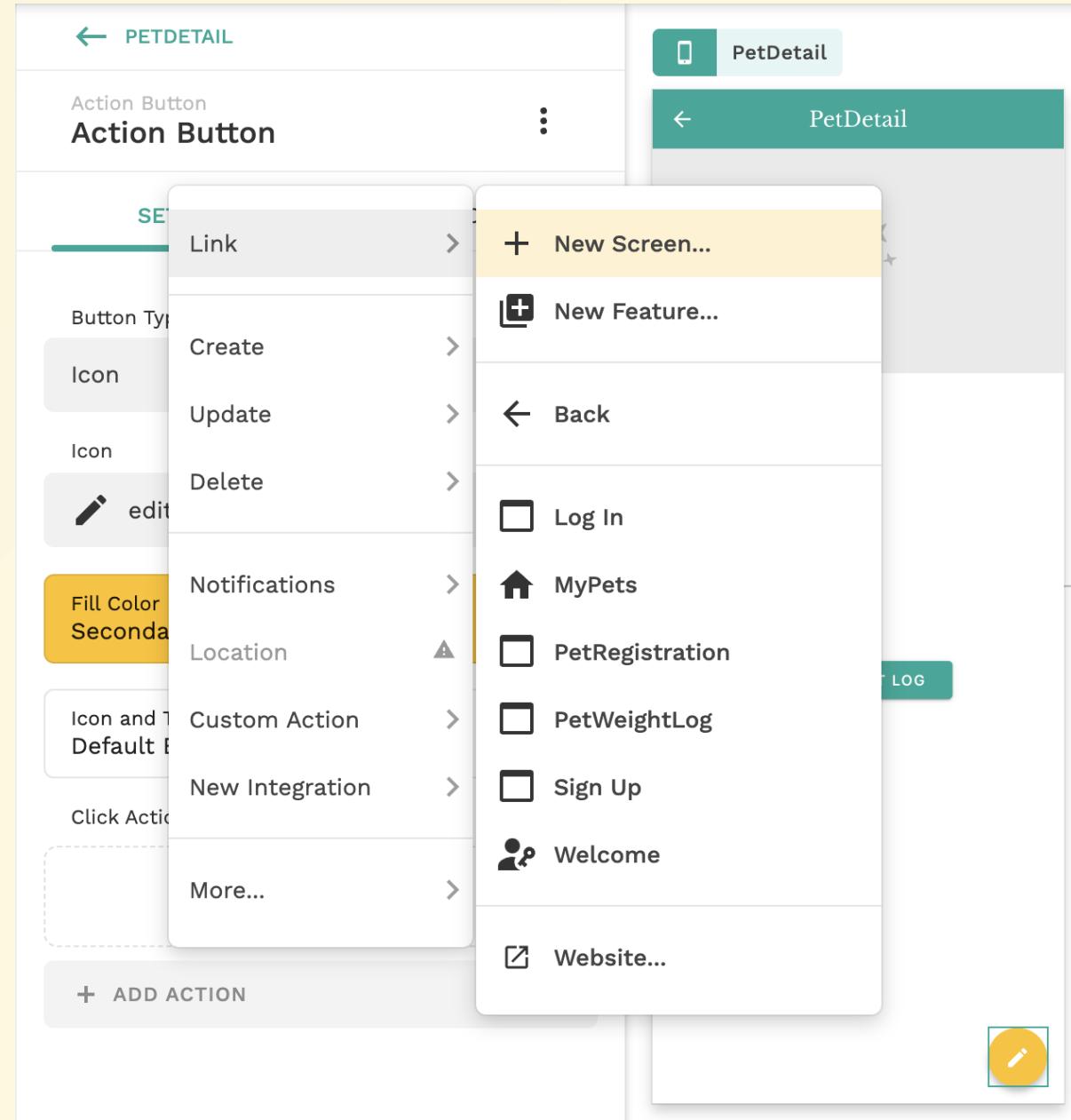
登録したペットの情報を更新できるペット編集画面を新たに作ります。

まず、ペット詳細画面にペット情報編集画面への導線を追加します。

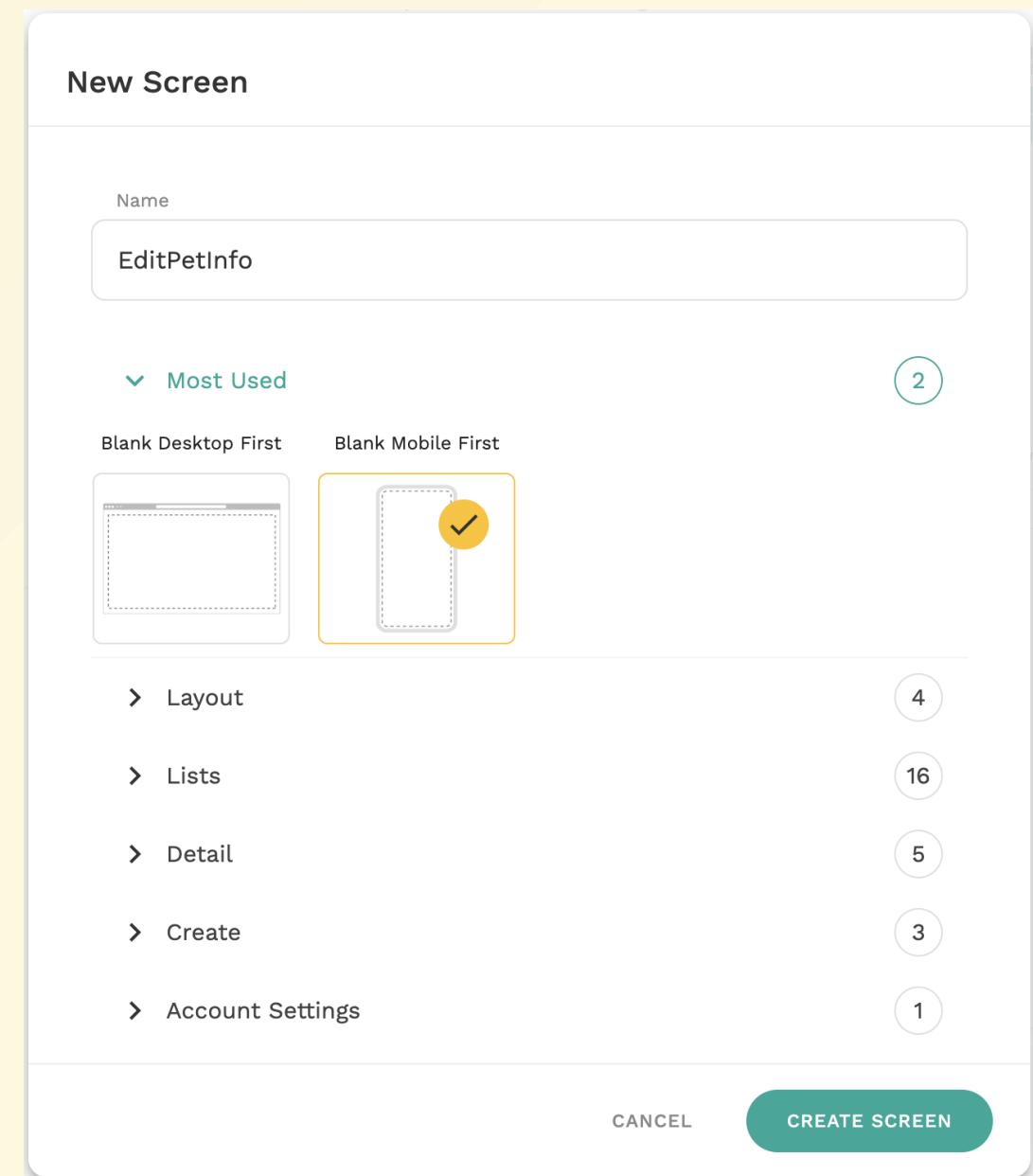
- ADD COMPONENTから Action Buttonを追加
- Iconをeditに変更
- Icon and Text Colorを Default Background(white) に変更



- ADD ACTIONからLink > New Screenを選択



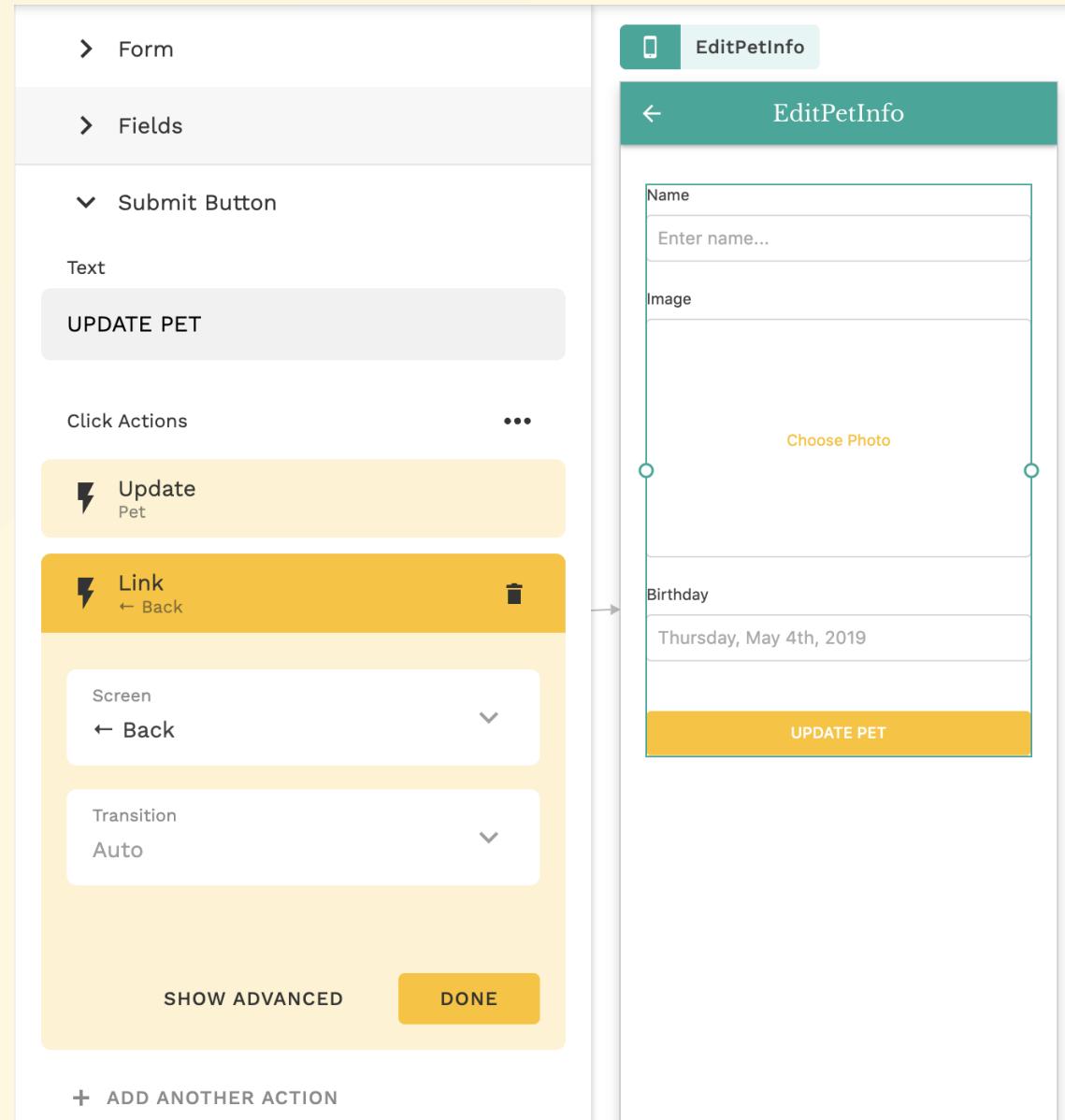
- Blank Mobile Firstを選択し、EditPetInfoスクリーンを作成



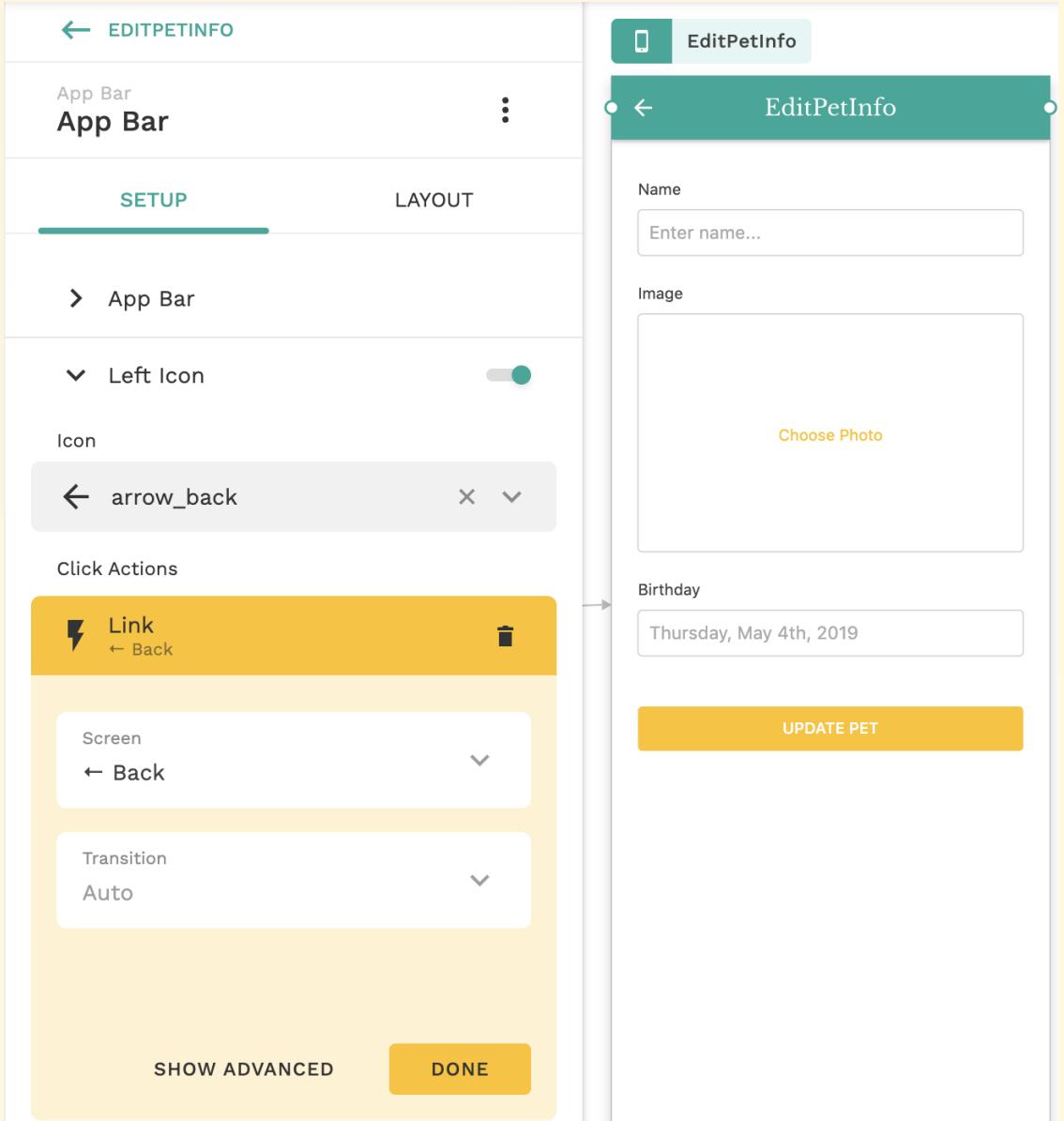
- App Barを追加してTitleをEditPetInfoに変更
- Formを追加して、以下を設定
 - Which data collection?で Petsを選択
 - What do you want the form to do?でUpdate Current Petを選択
- FieldsでBirthdayとImageの順番を入れ替え

The image displays two screenshots of a mobile application interface. On the left, the 'SETUP' screen of the 'EditPetInfo' form is shown. It includes a title bar with 'EDITPETINFO' and a back arrow. Below the title, it says 'Form' and 'Form'. Under 'SETUP', there are sections for 'Form', 'Which data collection?' (set to 'Pets'), 'What do you want the form to do?' (set to 'Update Current Pet'), 'Fields', and 'Submit Button'. On the right, the 'EditPetInfo' screen is shown on a mobile device. It has a green header bar with the title 'EditPetInfo'. The main content area contains fields for 'Name' (placeholder 'Enter name...'), 'Image' (with a 'Choose Photo' button), and 'Birthday' (set to 'Thursday, May 4th, 2019'). A large yellow 'UPDATE PET' button is at the bottom.

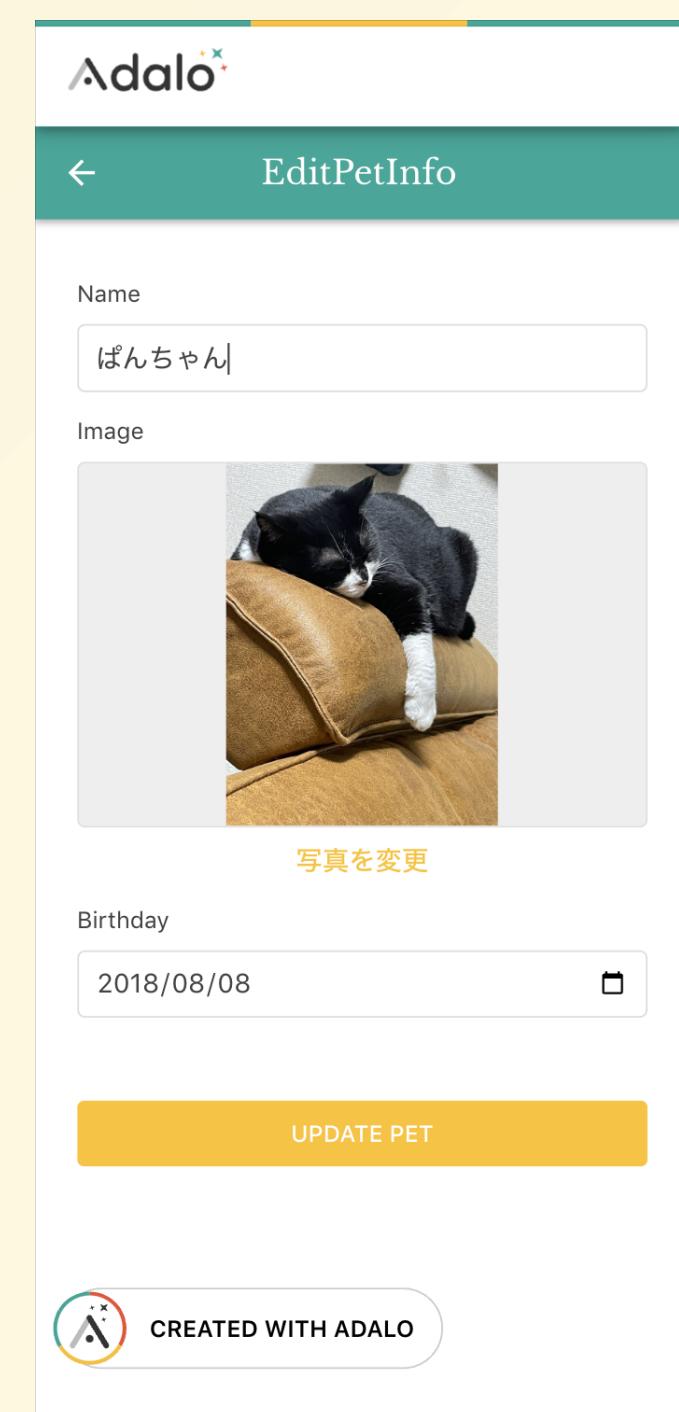
- FormのSubmit Buttonに戻るリンクを追加



- App BarのLeft Iconに戻るリンクを追加

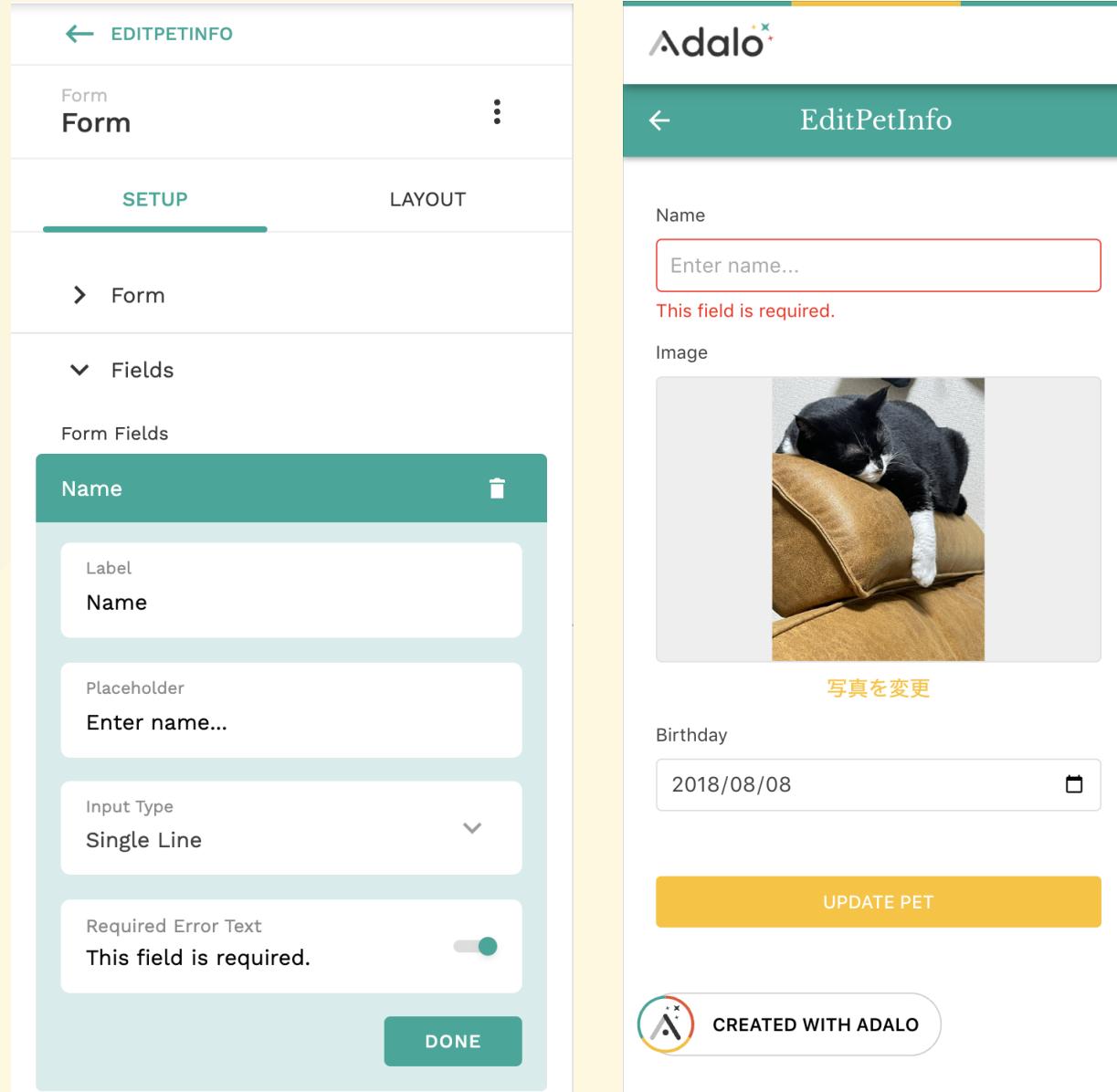


Preview機能でペット情報編集
画面が使えることを確認しま
しょう。



参考

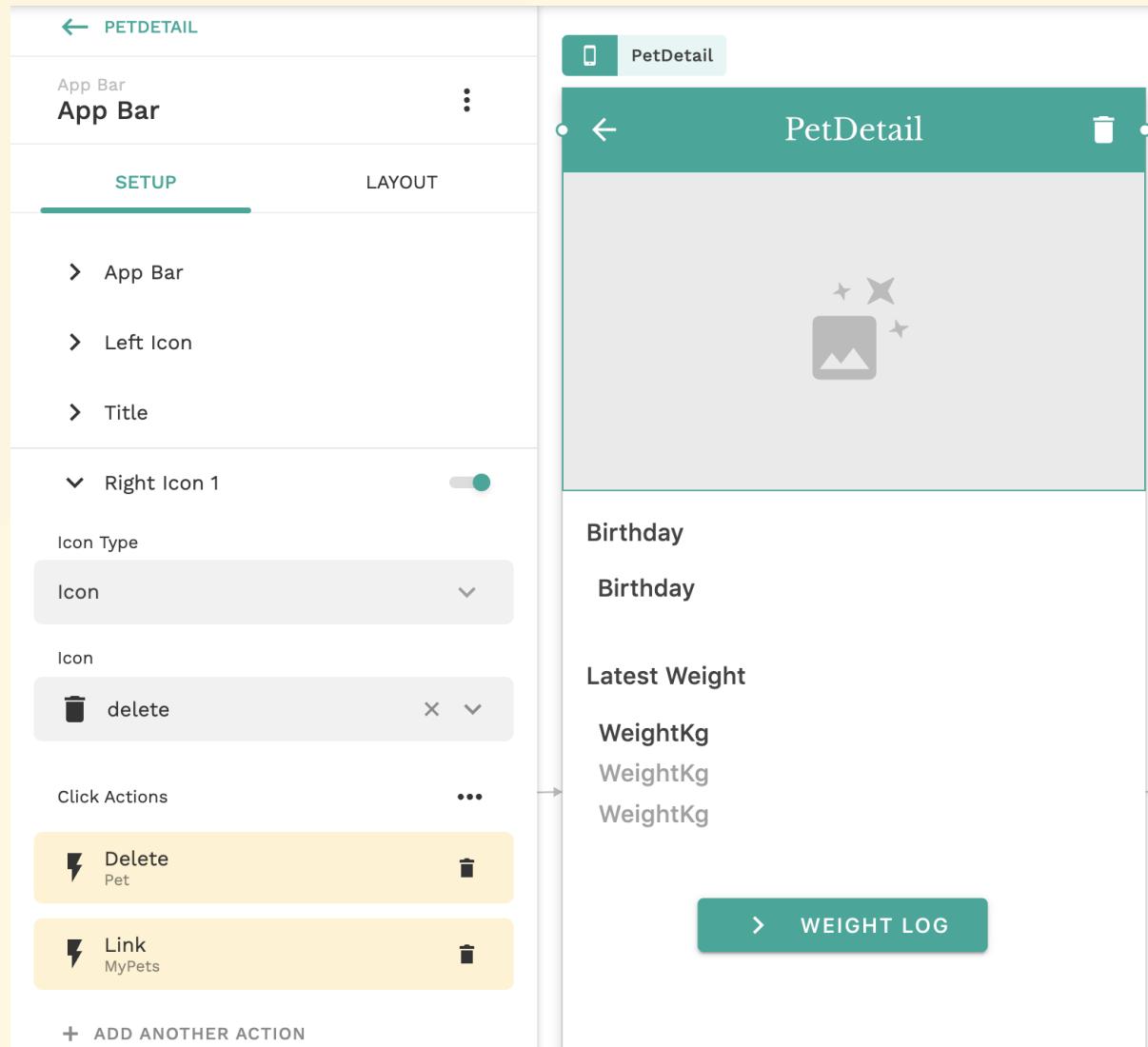
- FormコンポーネントのFieldのRequired Error Textにはデフォルトでチェックが入っており、その項目が入力されていない場合にはエラーメッセージが表示されます。
- 未入力のままUPDATE PETをクリックして確認してみましょう



ペット削除機能

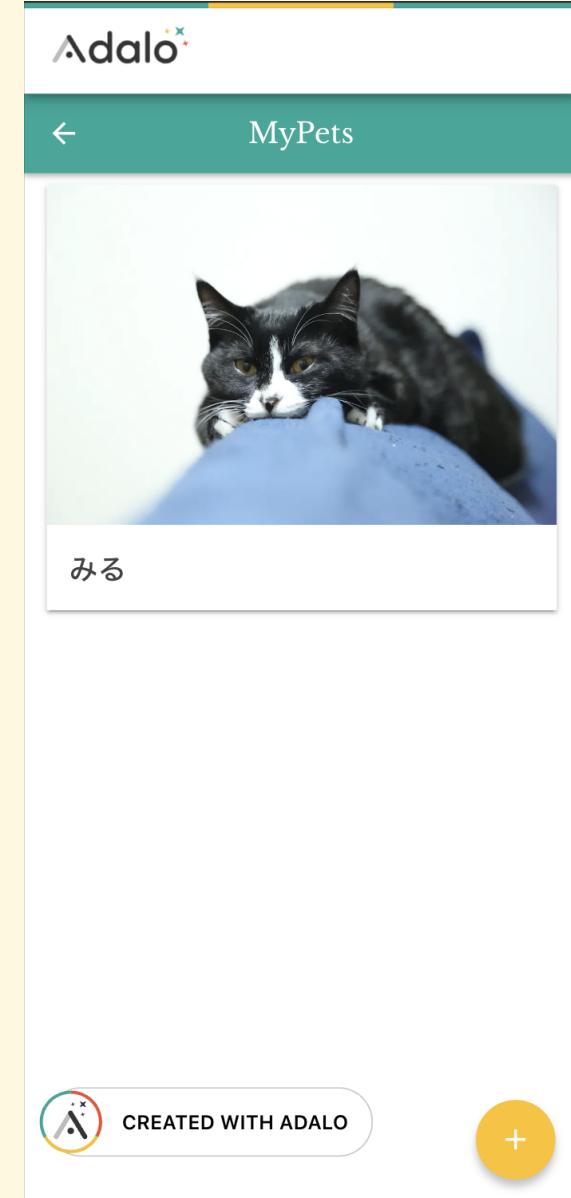
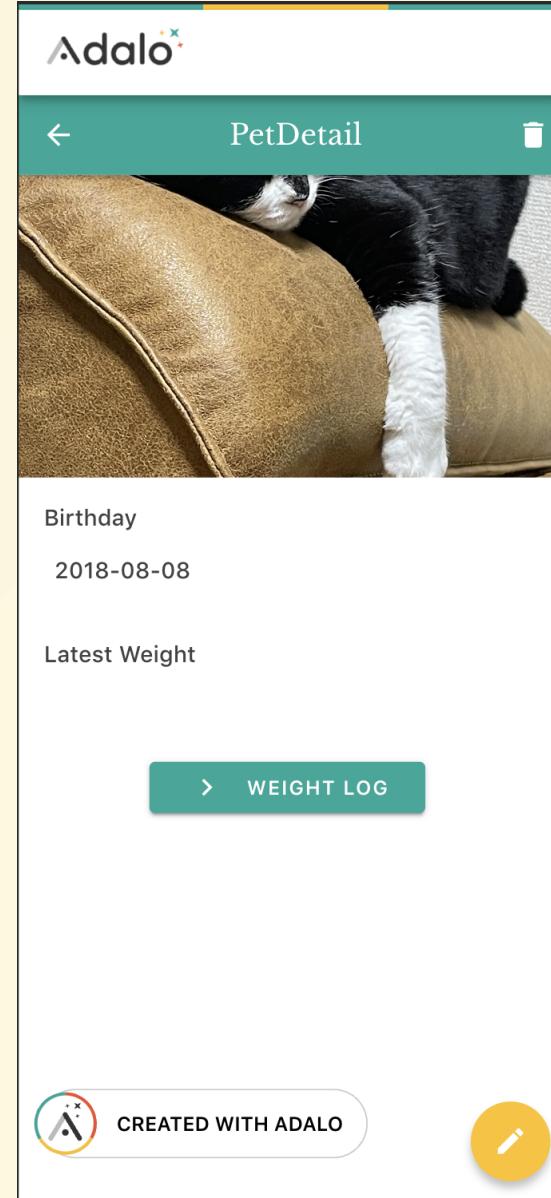
ペット詳細画面に、登録したペットを削除できるボタンを追加します。

- App Barを選択し、Right Icon 1のトグルをONにする
- Iconをdeleteに変更
- ADD ACTIONからDelete > Current Petを選択
- ADD ANOTHER ACTIONからLink > Mypetsを選択



Preview機能で削除を試してみましょう。

削除が完了してペット一覧画面に遷移すると、削除したペットは表示されません。

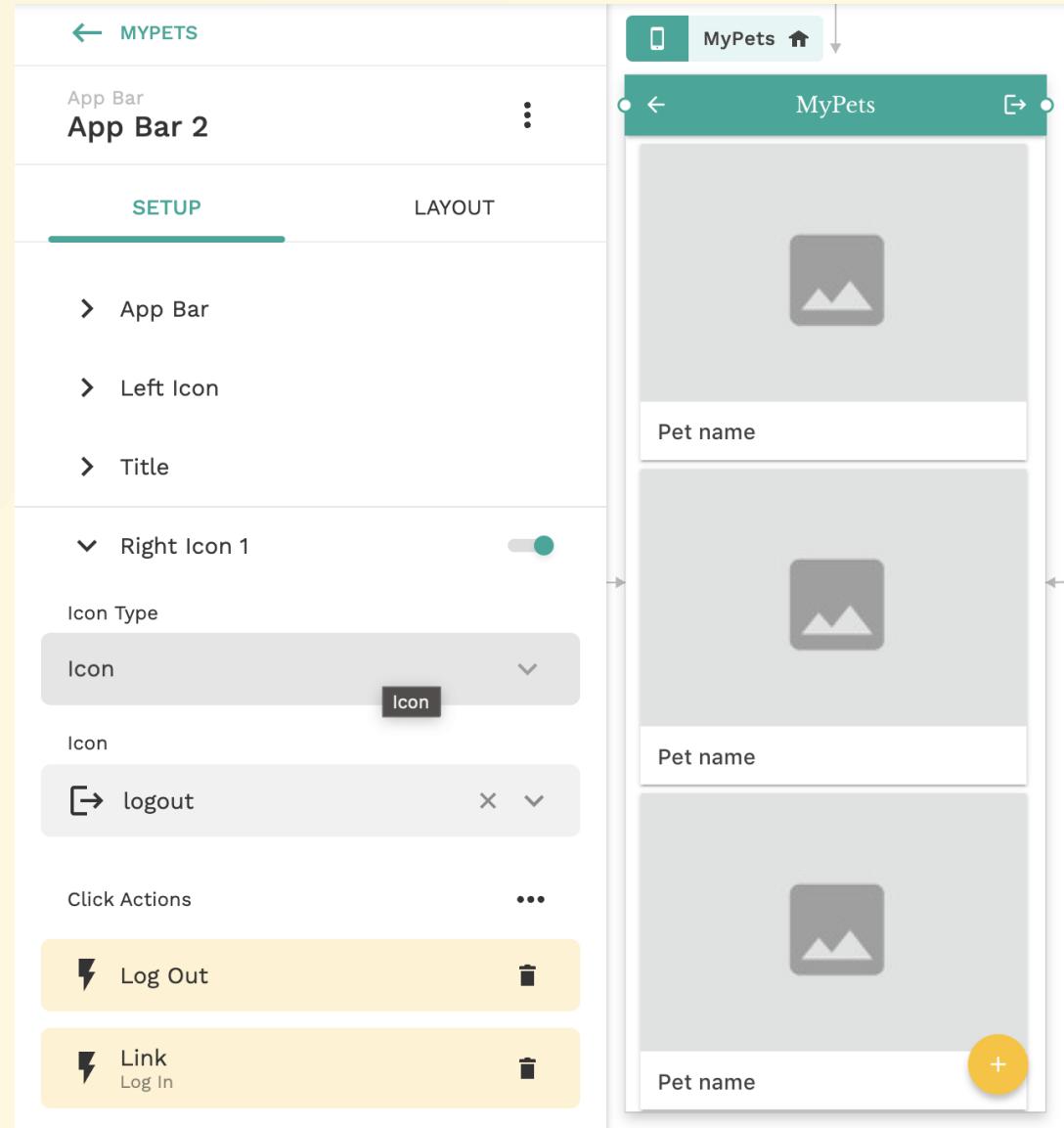


サンプルアプリを改善しよう

まだ紹介していないAdaloの機能を使いながら、サンプルアプリを改善していきます。

ログアウト

- ペット一覧画面のAppBarを選択し、Right Icon 1を有効化
- Iconをlogoutに変更する
- ADD ACTION > More... > User Login > Log Outを選択
- ADD ANOTHER ACTION > Link > Loginを選択



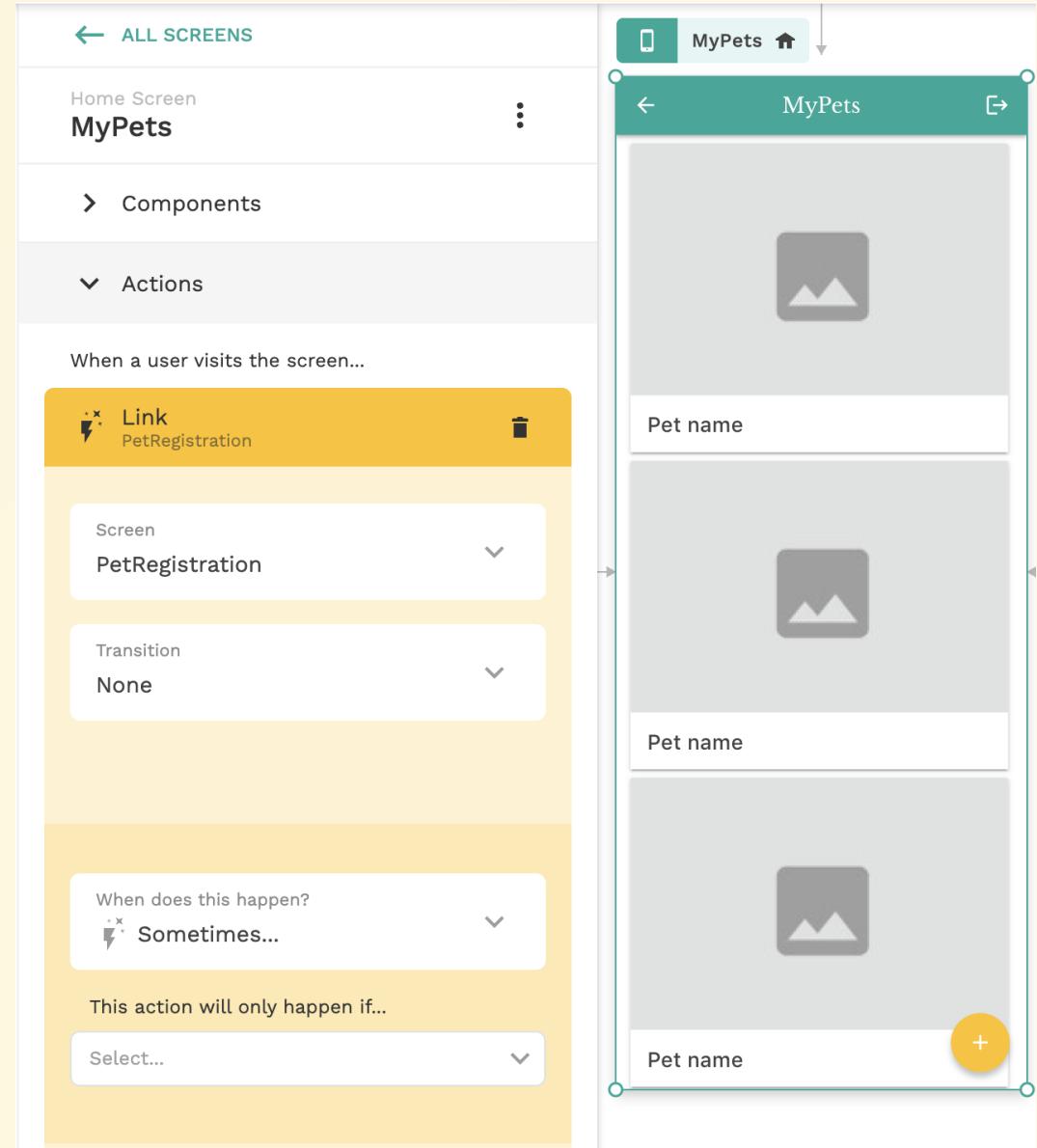
Preview機能で確認しましょう。

追加したアイコンをクリックするとログアウトし、ログイン画面に遷移するようになりました。

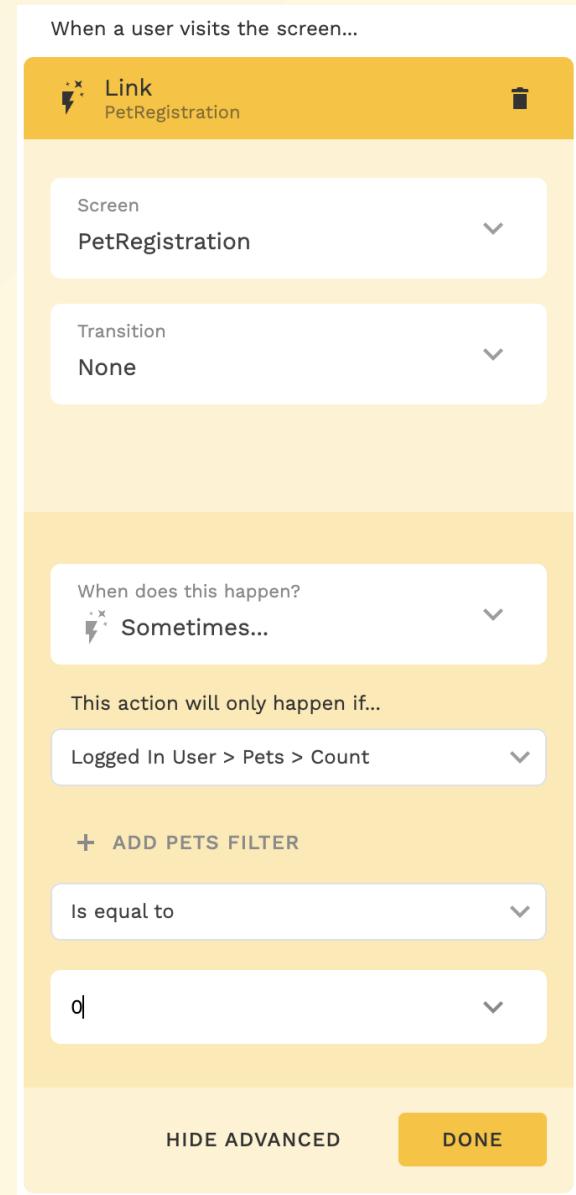
Actionの実行条件設定

ペットが未登録の場合にペット一覧画面を表示した場合、ペット登録画面へ遷移させます。

- ペット一覧画面でActions > ADD ACTION > Link > Pet Registrationを選択
- SHOW ADVANCEDをクリックし、When does this happen?をSometimesに変更



- This action will only happen if...でMore > Logged In User's > Pets' > Count を選択
- Is equal to の下の数値を0に変更



Preview機能で確認しましょう。

新しいユーザーでSignupすると、ペット一覧画面からペット登録画面に遷移します。

選択式入力フォーム

ペットの情報に性別を追加し、入力フォームで選択式入力ができるようになります。

選択式入力フォームで使う選択肢は、データベースにCollectionを追加してレコードとして用意します。

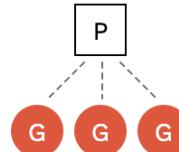
- データベースにGenders Collectionを追加(PropertyはデフォルトのままでOK)
- 0 Records > ADD GENDERとクリックし、MaleとFemaleの2つのRecordを追加

Genders					
	Name	ID	Created	Updated	
<input type="checkbox"/>	Female	2	a few seconds ago	a few seconds ago	    
<input type="checkbox"/>	Male	1	a few seconds ago	a few seconds ago	

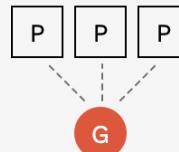
- Genders CollectionにPets Collectionとの1対多のRelationshipを追加する
 - 1匹のペットは1つの性別を持ち、1つの性別は複数のペットに対して設定されるため

How are Pets & Genders related?

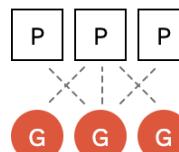
P = Pets G = Genders



A Pet can have multiple Genders
A Gender belongs to one Pet



A Pet can only have one Gender
A Gender can have multiple Pets ✓



A Gender can have multiple Pets
A Pet can have multiple Genders

CANCEL

DONE

- ペット登録画面のフォームを選択
- Fields > ADD VISIBLE FIELD > Genderを選択

PetRegistration

PetRegistration

Name
Enter name...

Image

Birthday
Thursday, May 4th, 2019

Choose Photo

Gender
Select gender...

+ ADD CUSTOM FILTER

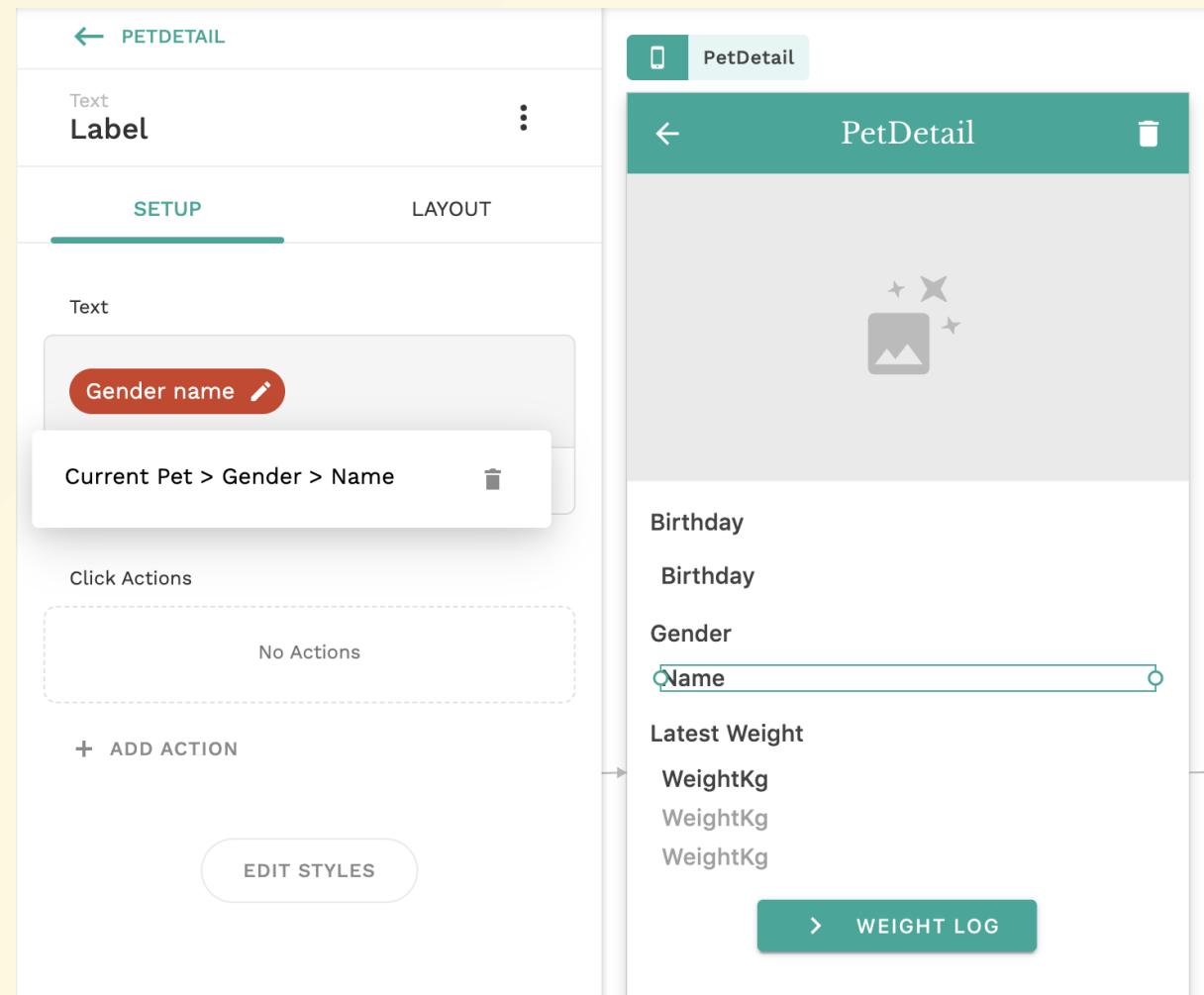
Sorting
Select...

Required Error Text
This field is required.

DONE

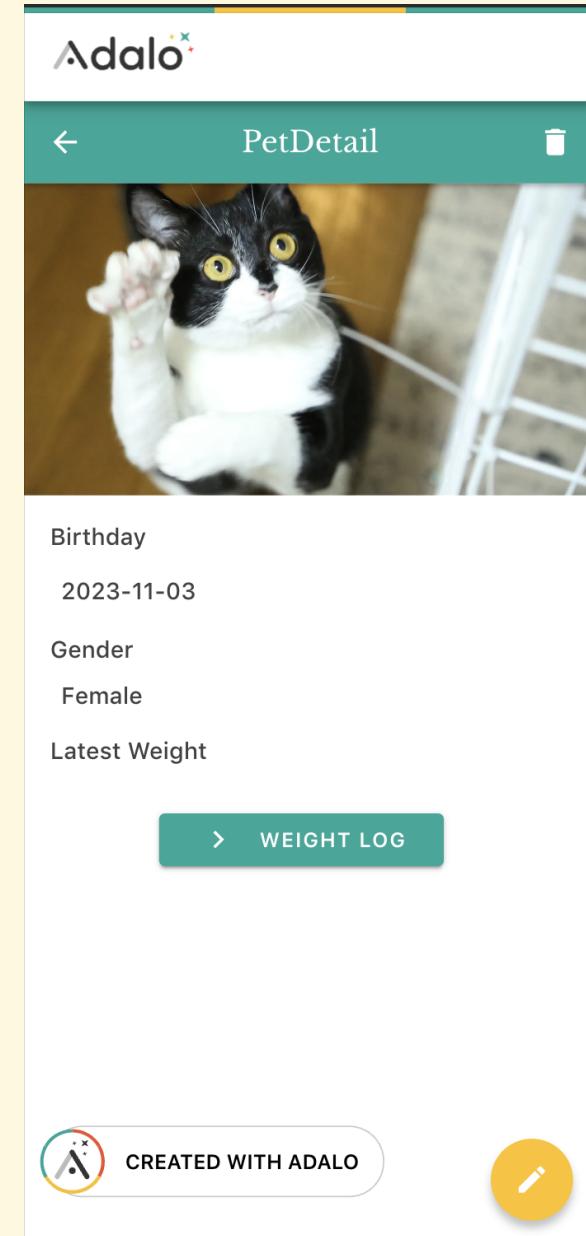
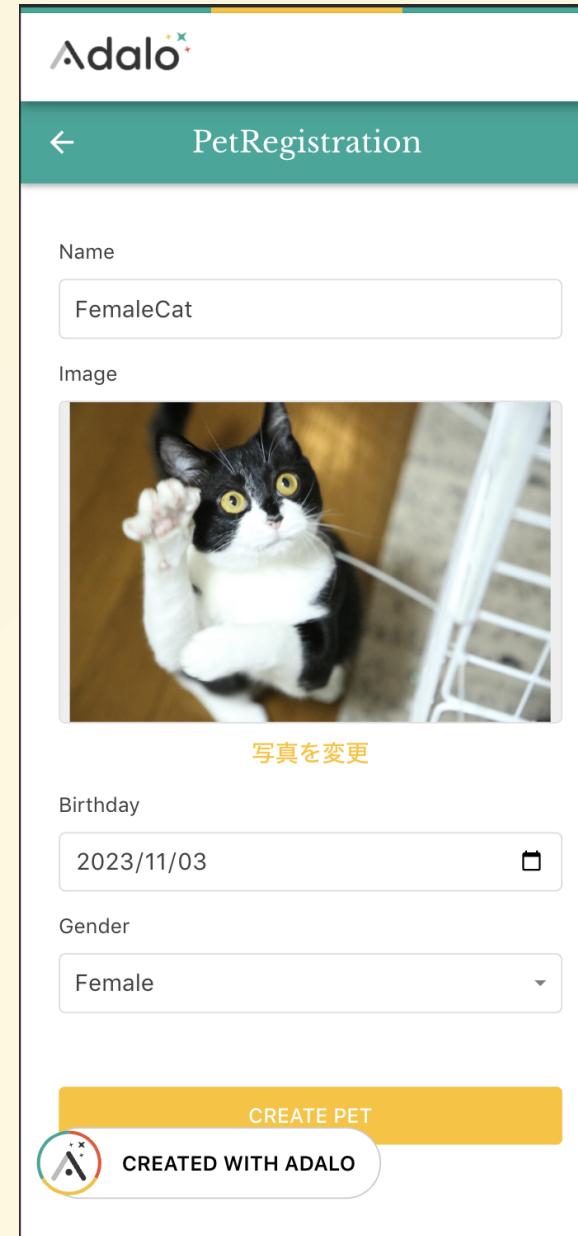
CREATE PET

- ペット詳細画面に性別の表示欄を追加



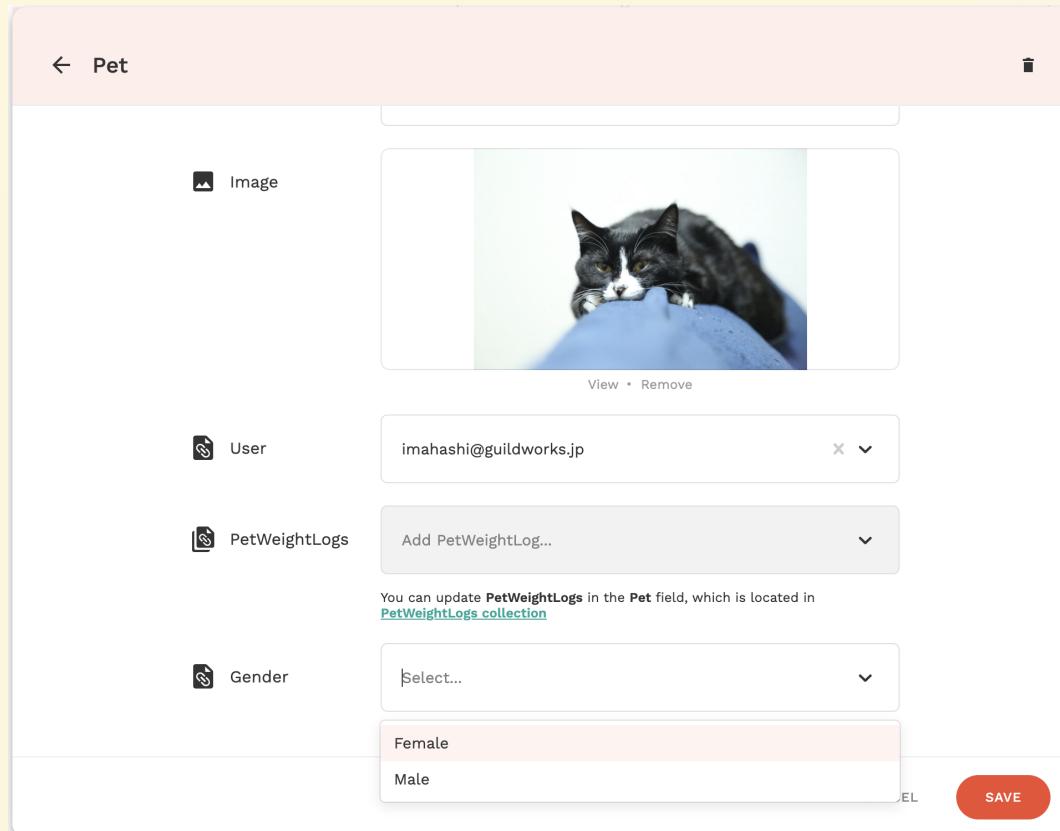
Preview機能で確認しましょう。

- ・ペット登録画面で性別が選択できます。
- ・選択した性別がペット詳細画面に表示されます。



補足

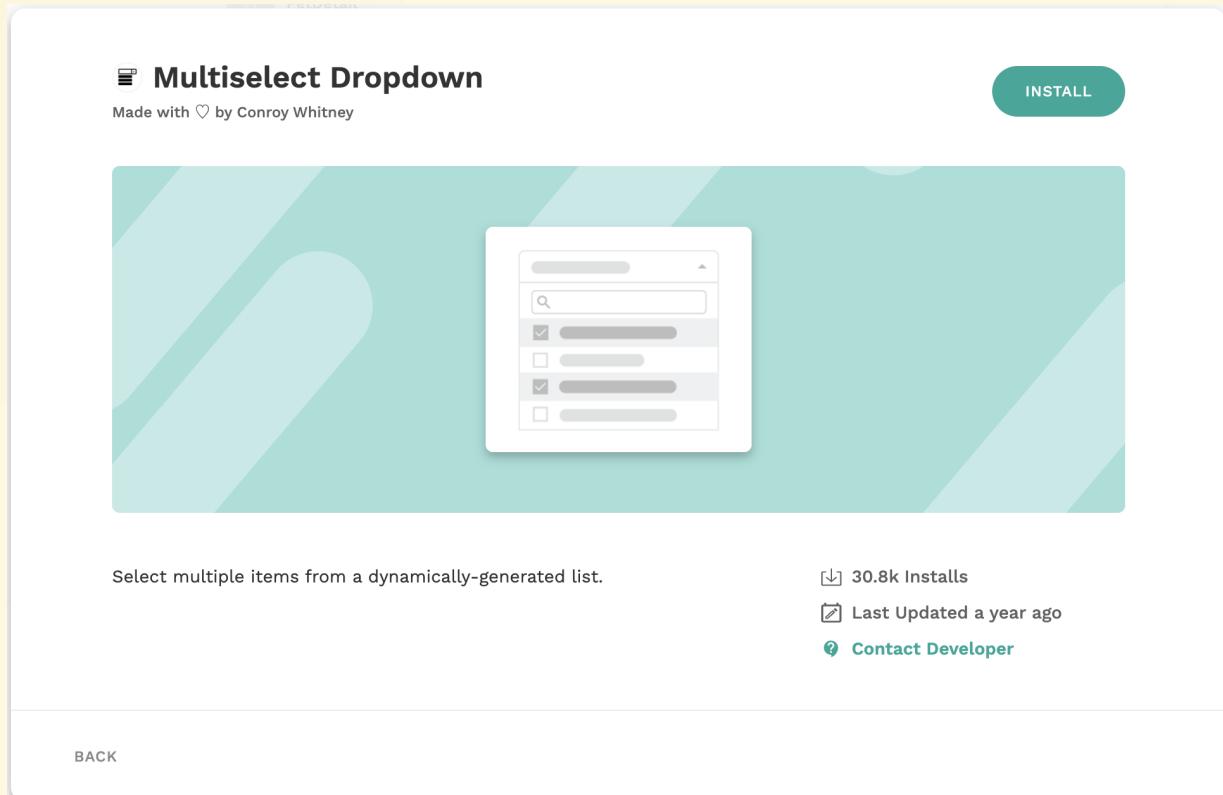
- 性別を選択できるようになる前に登録したペットで空欄が表示されることが気になるようであれば、Pet CollectionのRecordを表示してそのペットをクリックし、性別を手動で設定してもOKです。



参考

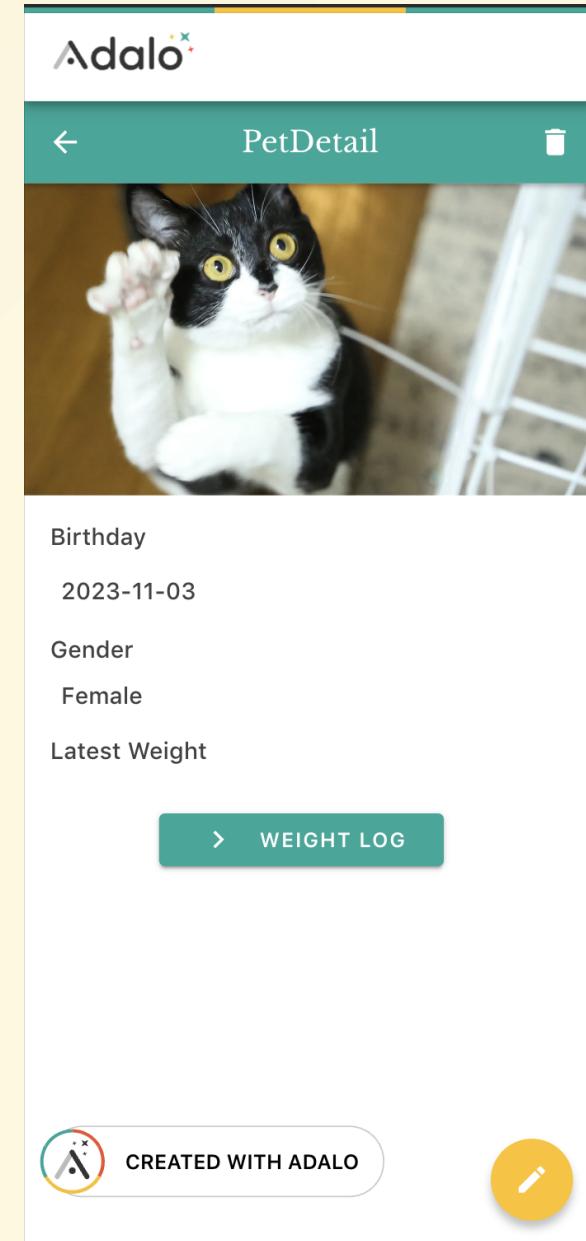
複数選択式の入力フォーム
は、Marketplaceの
MultiselectDropdownを使って
作れます。

必要になったら、試してみて
ください。

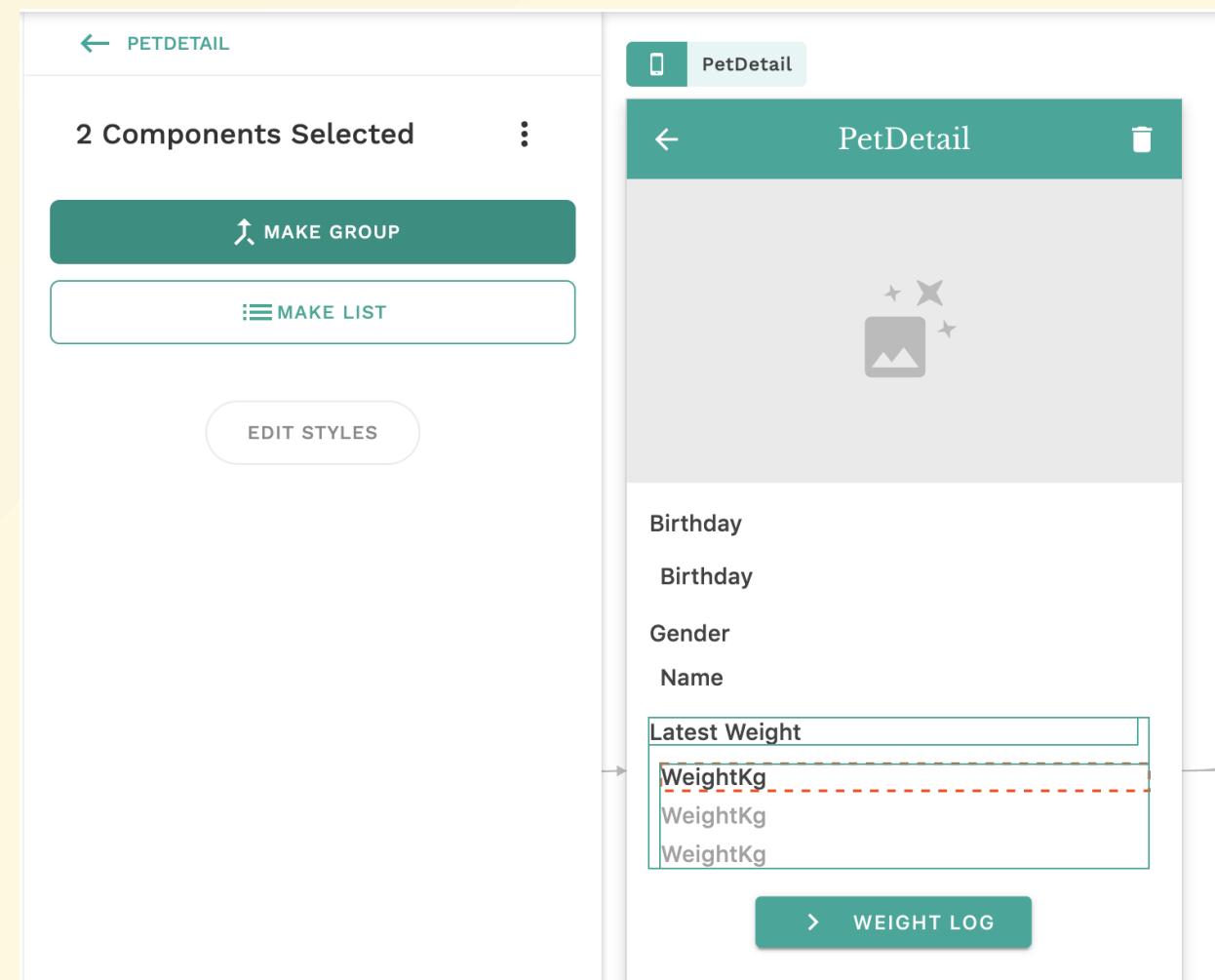


条件によってコンポーネントの表示or非表示を切り替える

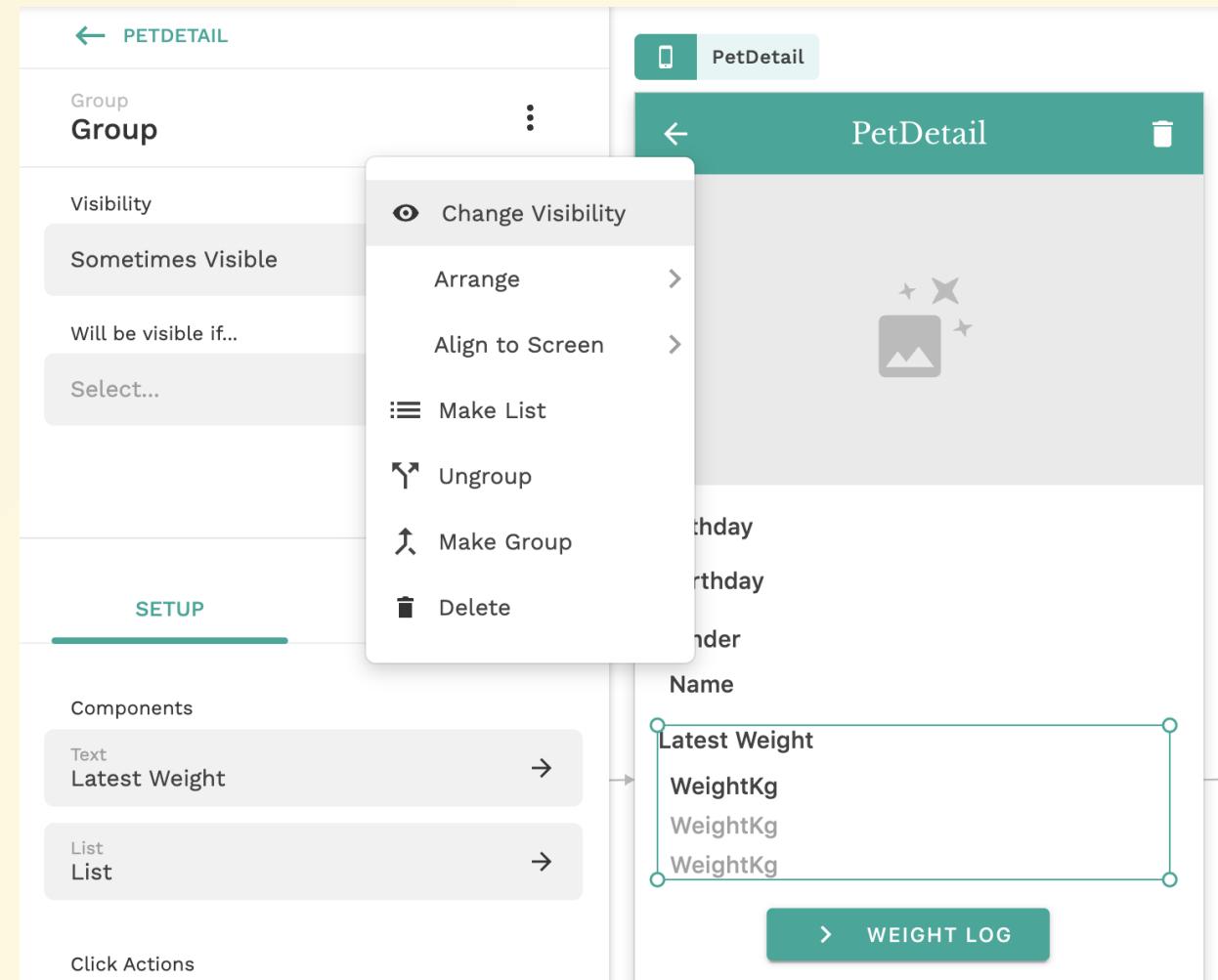
体重が未登録であれば、ペット詳細画面のLatest Weightは非表示にします。



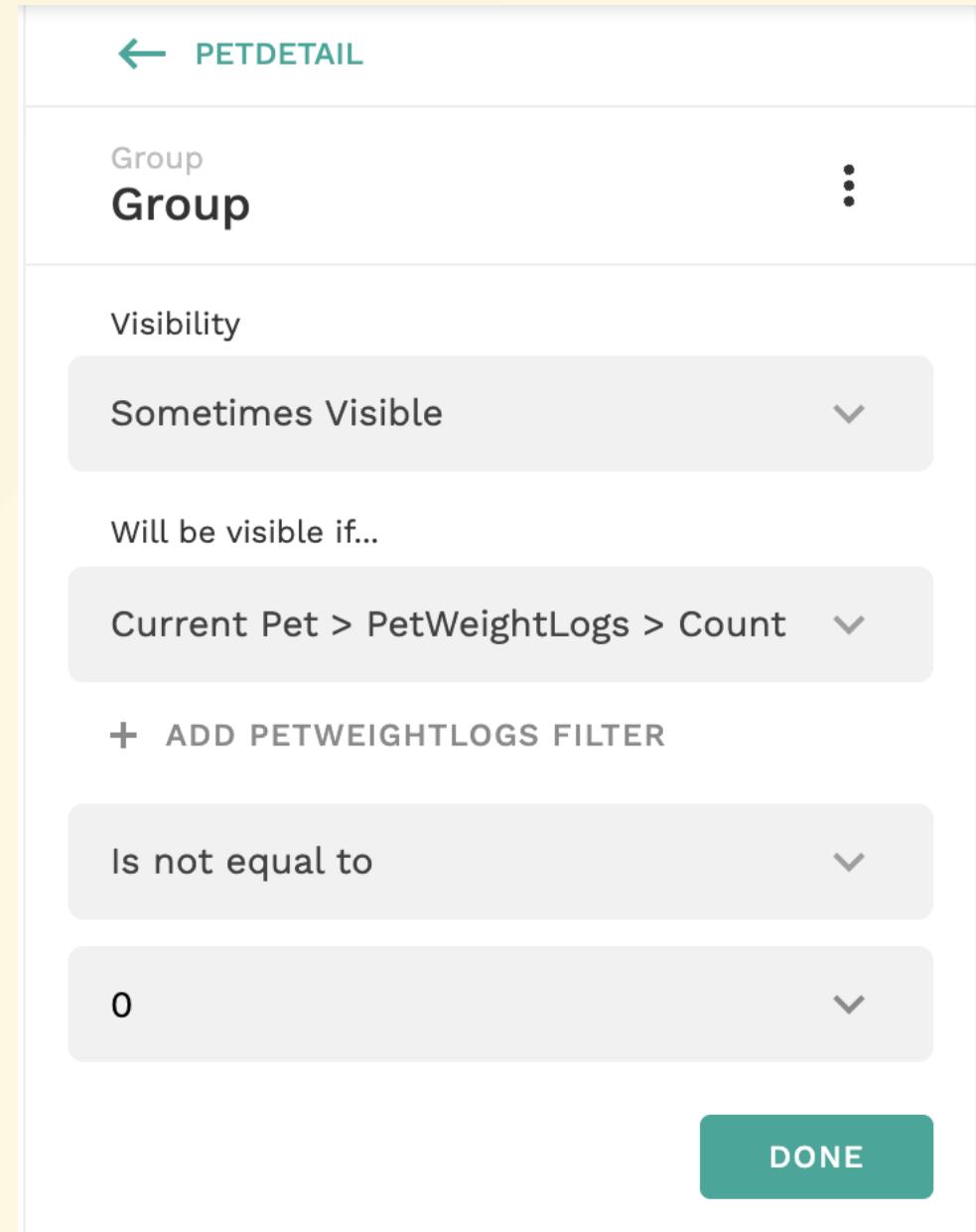
- Latest Weightというラベルとその値を選択し、グループ化します



- Change Visibilityを選択します

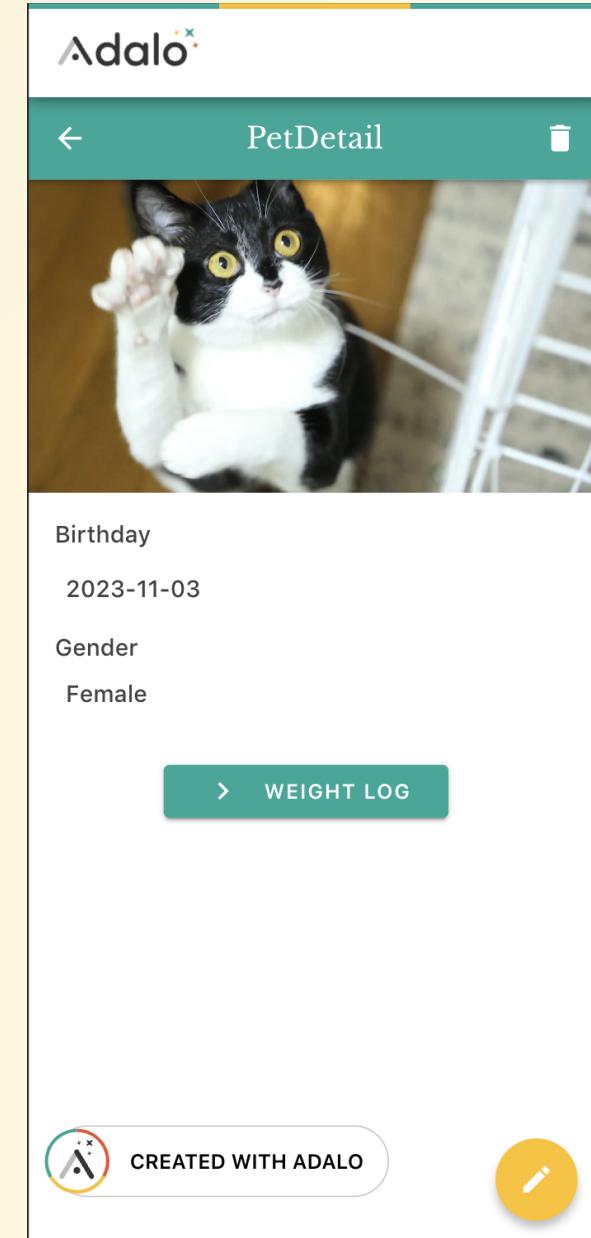


- VisibilityをSometimes Visibleに変更
- Will be visible if...でCurrent Pet > PetWeightLogs > Countを選択
- Is not equal to 0と設定



Preview機能で確認すると、体重が未登録であれば、Latest Weightが非表示になりました。

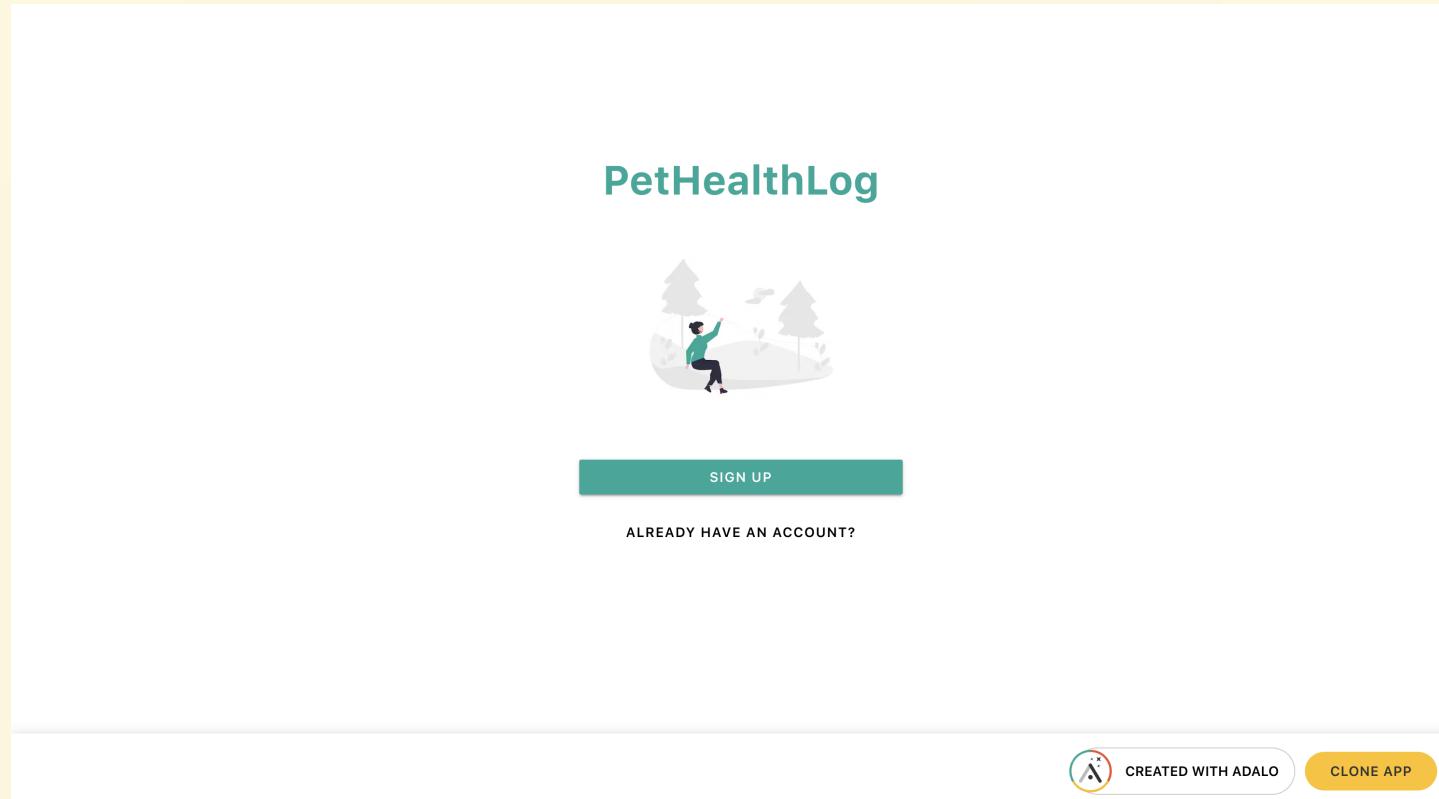
サンプルアプリの改善は一旦ここまでとします。



クローン用URL

- 以下のURLの右下のCLONE APPボタンでアプリをクローンできますので、答え合わせに使ってください

<https://ryo-imahashis-team-6.adalo.com/improvedpethealthlog>



外部連携

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

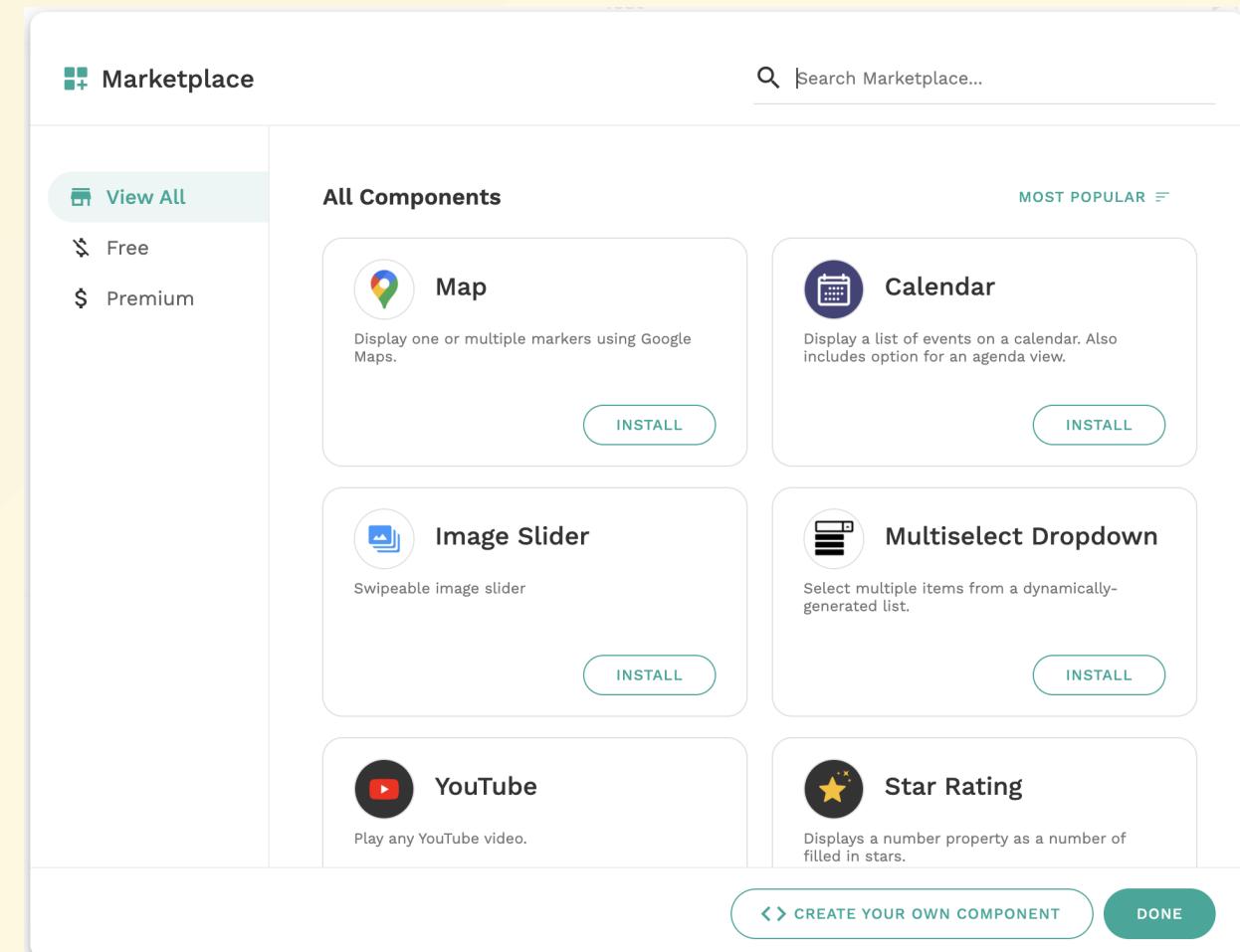
その方法をご紹介します。

以下4種類の外部連携の方法を紹介します。

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

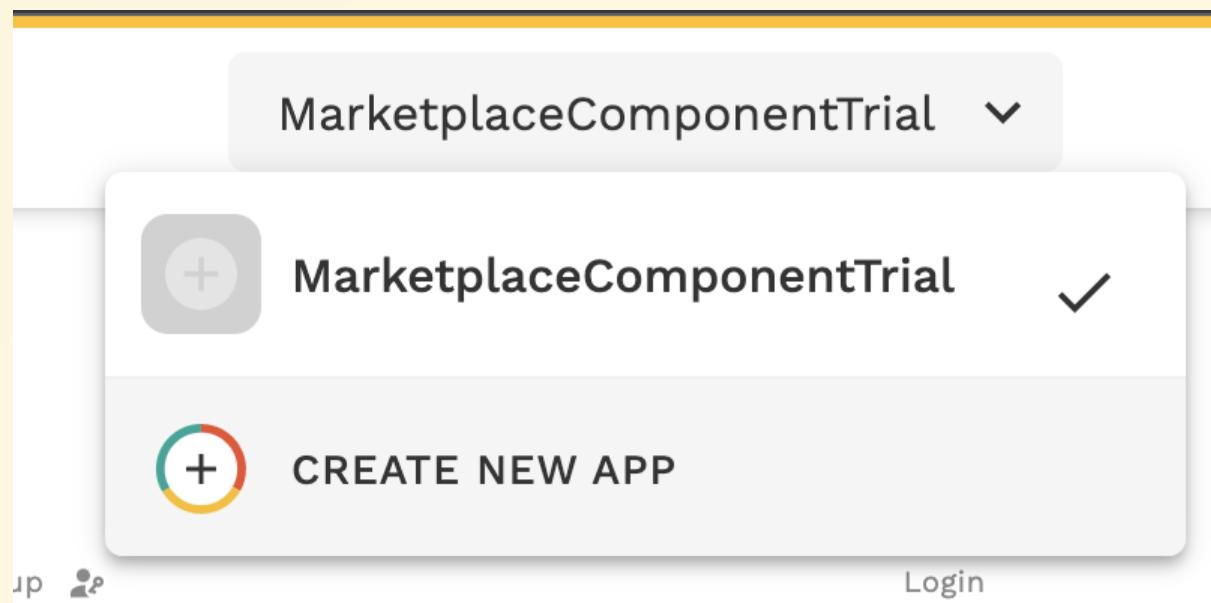
Marketplaceの外部連携コンポーネント

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



まずはアプリを新規作成しましょう。

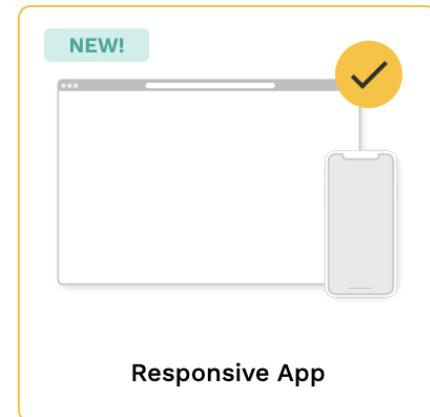
- CREATE NEW APPをクリック



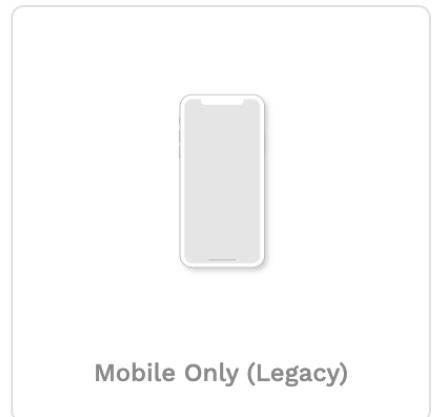
- PlatformはResponsive Appを選択

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

What kind of app do you want to make?



Responsive App



Mobile Only (Legacy)

NEX

- TemplateはBlank Mobile Firstを選択

1 Platform 2 Template 3 Branding

Blank Desktop First Blank Mobile First Learn Adalo

Todo List Chat Layout Tips & Tricks

[BACK](#) [ADVANCED OPTIONS](#) [NEXT](#)

- App Nameには MarketplaceComponentTrial を入力
- その他の項目は自由に設定してください

The screenshot shows the branding configuration step of an app creation process. At the top, there are three status indicators: 'Platform' (green checkmark), 'Template' (green checkmark), and 'Branding' (green circle with the number '3'). Below these, the 'App Name' field contains 'MarketplaceComponentTrial'. The 'Who are the users of this app?' dropdown is set to 'Mostly me, I'm just testing'. There are two color selection dropdowns: 'Primary Color' set to '#00A898' (green) and 'Secondary Color' set to '#FFC00E' (yellow). At the bottom left is an 'ACK' button, and at the bottom right is a large green 'CREATE' button.

Platform Template 3 Branding

App Name

MarketplaceComponentTrial

Who are the users of this app?

Mostly me, I'm just testing

Primary Color
#00A898

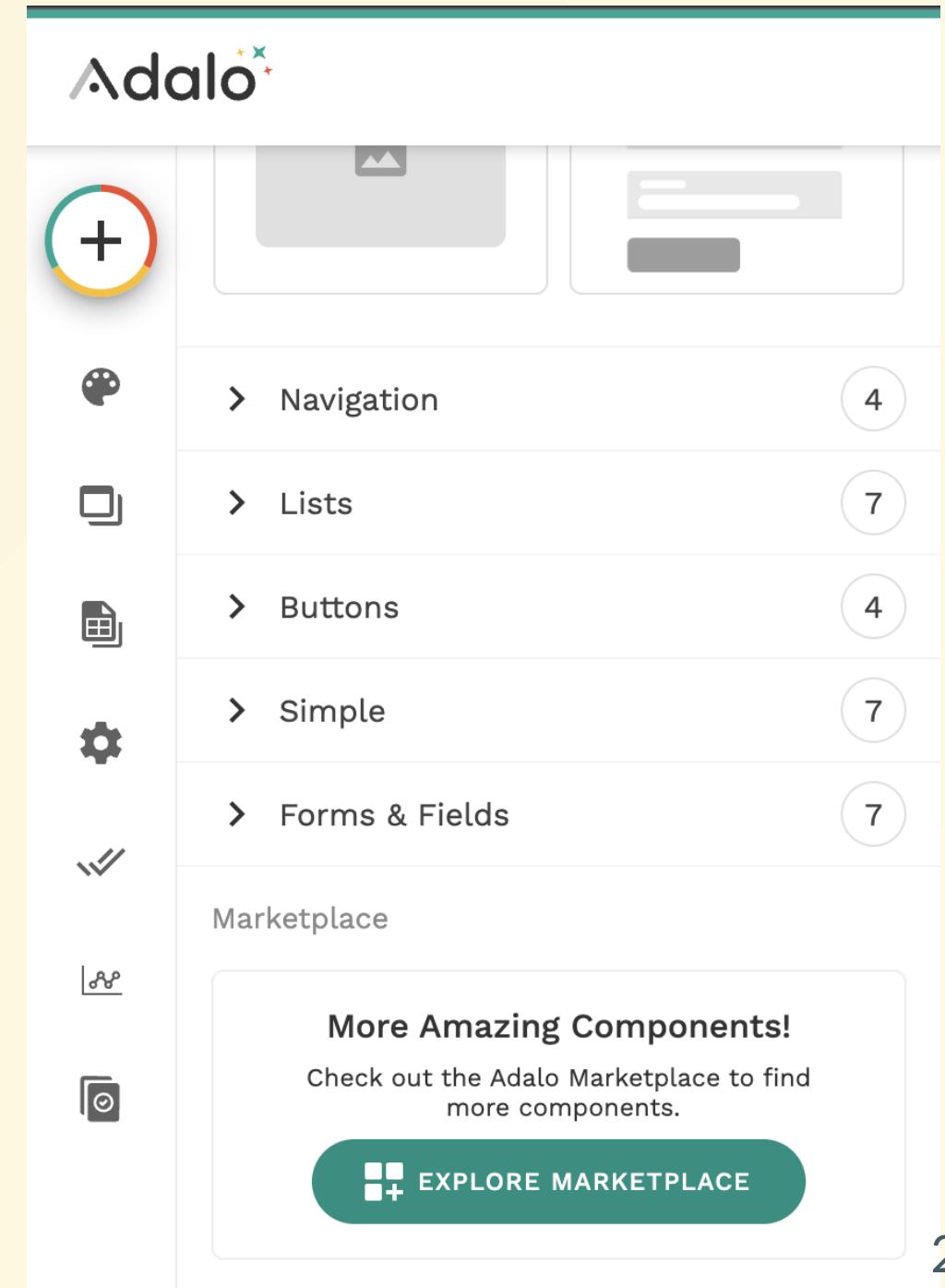
Secondary Color
#FFC00E

ACK

CREATE

Twitter Timeline コンポーネント

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



- Twitter TimelineコンポーネントをINSTALL

twitter

All Components



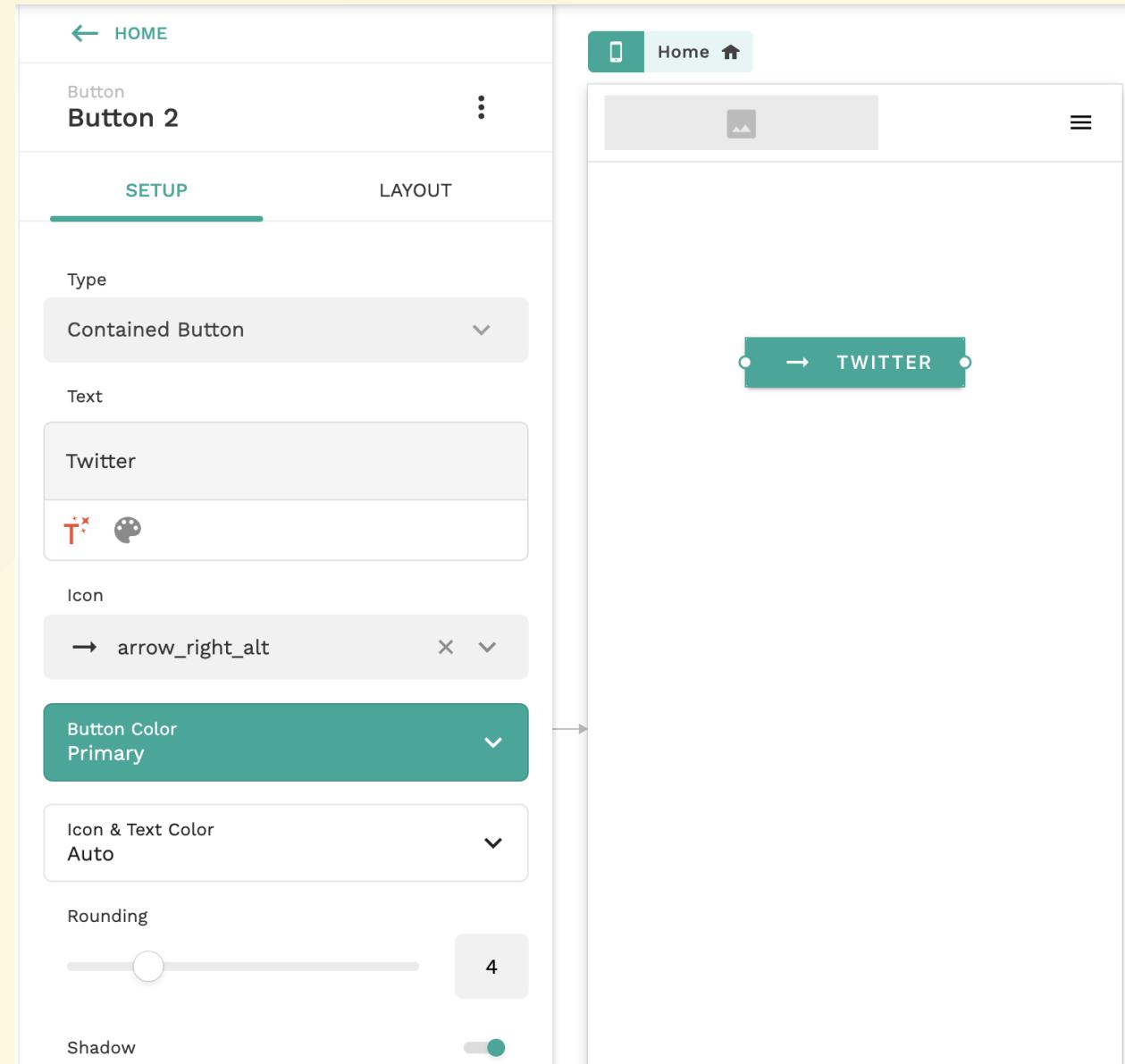
Twitter Timeline

Embed a twitter timeline with customized settings.

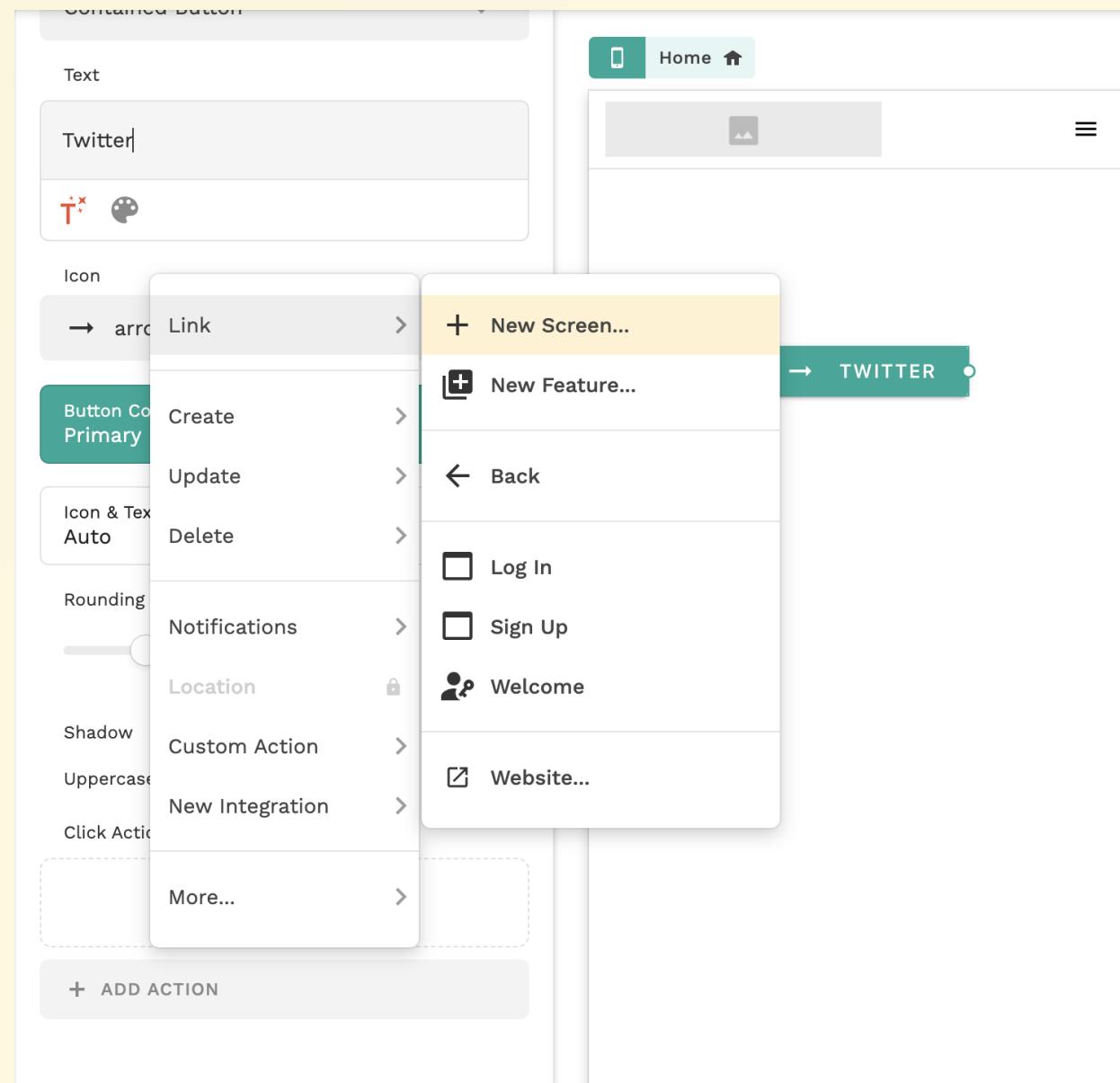
INSTALL

Twitter Timelineコンポーネントを配置します。

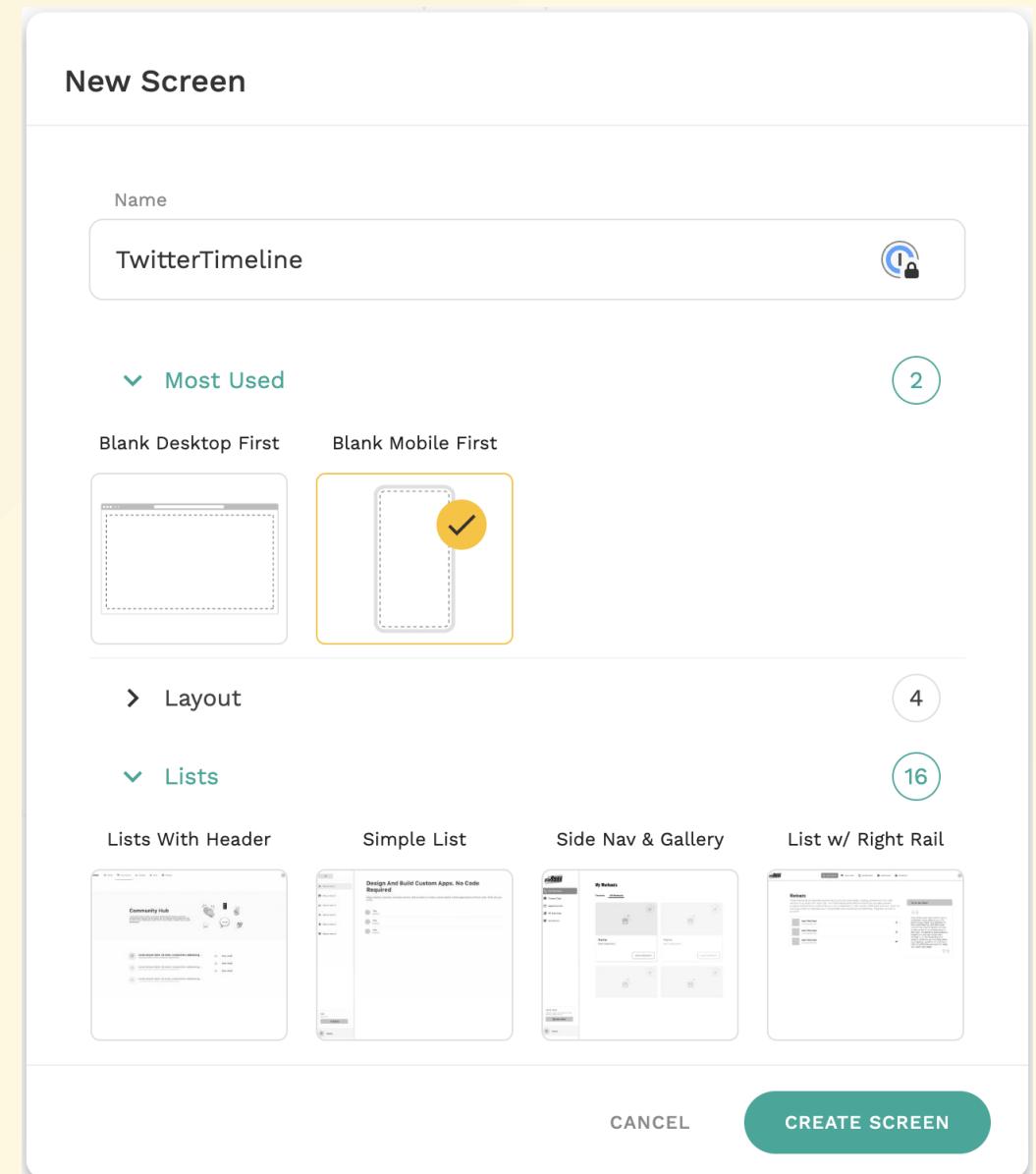
- Home ScreenにTwitter用画面へのLinkボタンを追加



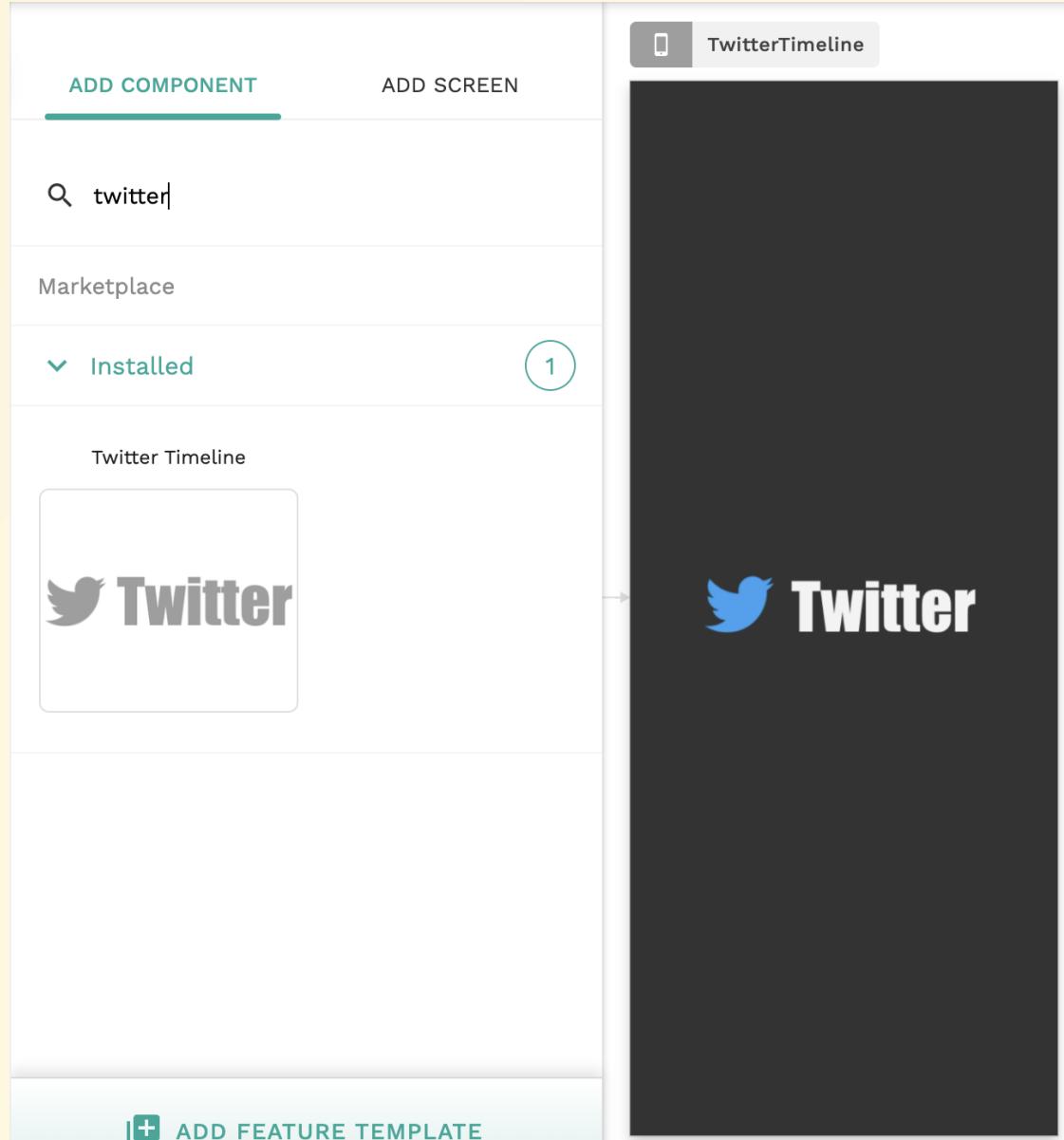
- ADD ACTIONからNew ScreenへのLinkを追加



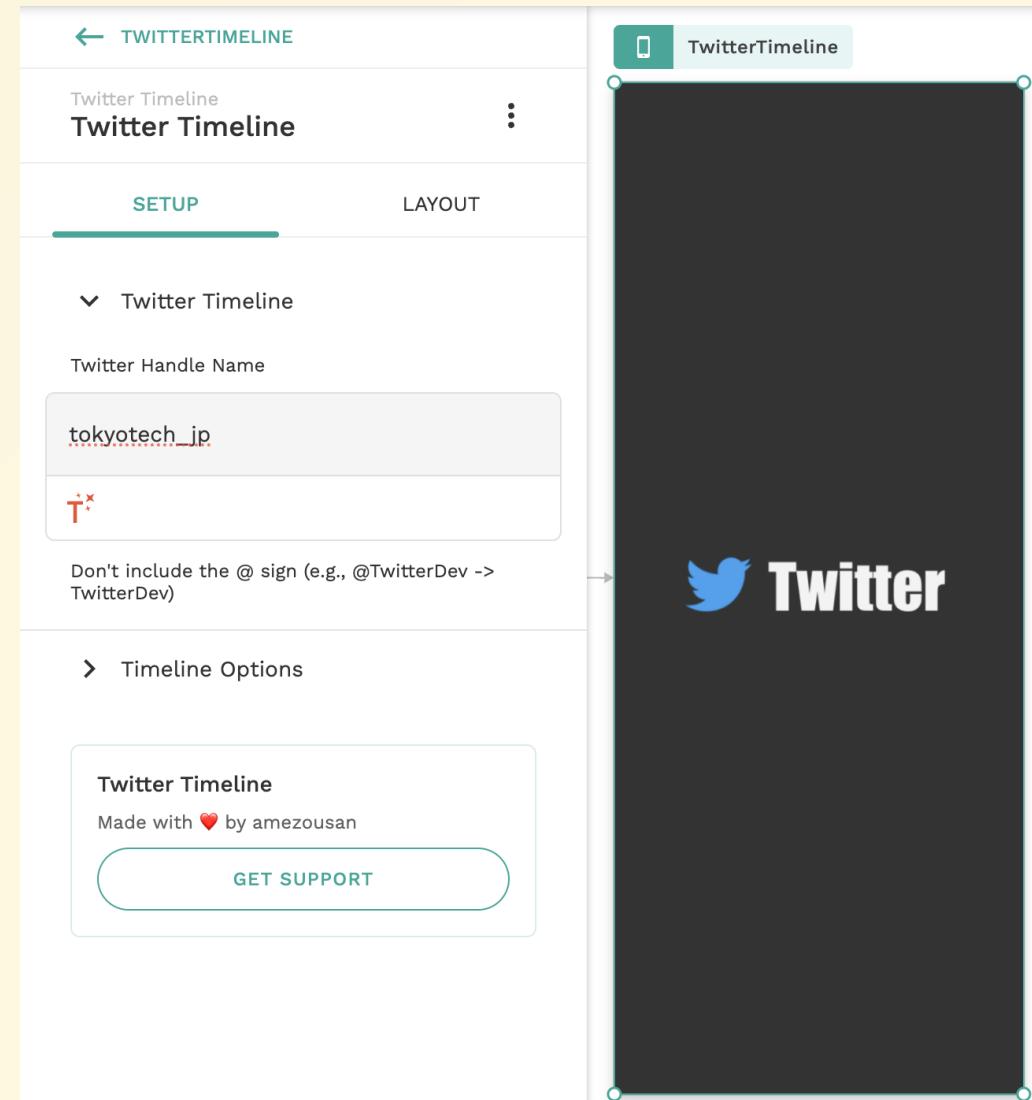
- Blank Mobile Firstを選択し、TwitterTimeline画面を作成



- Twitter Timelineコンポーネントを配置



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



Preview機能でSignupして、
HomeのTwitterボタンをクリ
ックすると、TwitterTimeline
画面に入力したHandle Name
のTwitterアカウントの投稿が
一覧表示されます

Adalo

Tweets from @tokyotech_jp [Follow](#)

東京工業大学 @tokyotech_jp · Oct 30 X
2023年度末松賞「革新的価値創造の基礎と展開」を4人が受賞: 東京工業大学は、2023年度末松賞「革新的価値創造の基礎と展開」の受賞者を決定し、9月12日に授賞式を行いました。末松賞「革新的価値創造の基礎と展開」は、科学技術の各分野における「革新的価値創造」につ...



titech.ac.jp
2023年度末松賞「革新的価値創造の基礎と展開」を4人が受賞

1 20

東京工業大学 @tokyotech_jp · Oct 30 X
東工大史資料館提供「今月の一枚」: エジプトに輸出された日本の高等教育システム facebook.com/tokyotech.jp/p...



CREATED WITH ADALO

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

例:

- Youtube(無料)
- Google Map(クレジットカードの登録が必要だが、無料枠で利用可能)

Custom Action

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

参考

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

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

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

<https://thecatapi.com/>

The screenshot shows the homepage of The Cat API. At the top, there's a navigation bar with links for PRICING, ABOUT, MORE APIs, DOCUMENTATION, and SHOWCASE. Below the header, the main title "The Cat API - Cats as a Service." is displayed, followed by the tagline "Because everyday is a Caturday." A brief description states it's a public service API for cats, free to use, and mentions over 80K developers. It also highlights that billions of requests are delivered monthly. Below this, there are icons for various technologies: React, jQuery, Vue.js, Node.js, and Angular. Two buttons at the bottom are "GET YOUR API KEY" and "READ OUR GUIDES". A note indicates a free allowance of 10,000 requests per month. On the right side of the page, there's a large image of a fluffy cat looking directly at the camera, with some cherry blossoms in the background. Above the cat image is a black navigation bar with links for VOTE, BREEDS, IMAGES/SEARCH, FAVOURITES, and UPLOAD. Below the image are two buttons: a green "LOVE IT" button with a thumbs-up icon and a red "NOPE IT" button with a thumbs-down icon.

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

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

The screenshot shows the homepage of The Dog API. At the top, there's a blue navigation bar with links for PRICING, ABOUT, MORE APIs, DOCUMENTATION, and SHOWCASE. Below the header, the main title "The Dog API - Dogs as a Service" is displayed, followed by the tagline "Because everyday is a Dog Day.". A subtext explains it's a public service API for Dogs, free to use, and links to a license. It also mentions billions of requests from over 30K developers. Below this, there are icons for various frameworks: React, jQuery, Vue.js, Node.js, and Angular. Two prominent buttons at the bottom left are "GET YOUR API KEY" and "READ OUR GUIDES". In the center, there's a large image of two dogs in a grassy field, with "LOVE IT" and "NOPE IT" buttons above it. The overall design is clean and modern.

APIドキュメントでAPIの使い方を確認しましょう。

- 以下のURLへアクセス

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

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

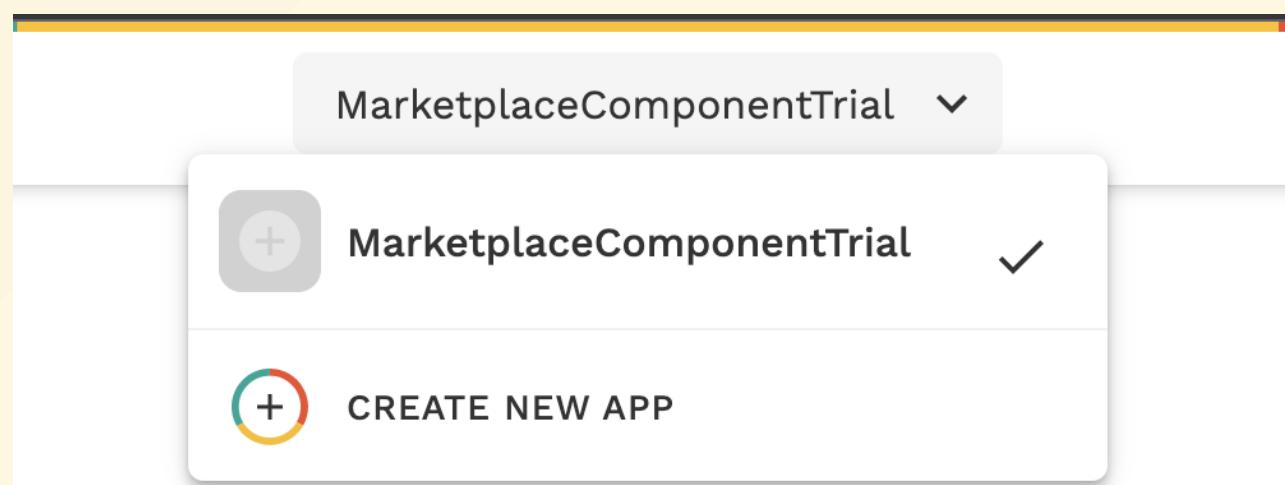
<https://api.thecatapi.com/v1/images/search>

The screenshot shows the documentation portal for TheCatAPI. On the left, there's a sidebar with various links: Authentication, Basics: Getting Images (which is selected and highlighted in grey), Basics: Filter By Breed, TheDogAPI.com, Basics: Voting, Discord Server, NPM node library installation, Basics: Favouriting, Code Samples, OpenAPI Spec Doc, Postman API Docs (with a small icon next to it), and More APIs... At the top right, it says 'TheCatAPI - Documentation Portal'. The main content area has a 'Quick start' section with three buttons: one to get a random image, one to get 10 random images, and one to copy a link for 10 Bengal images. Below that is a 'Getting Images' section with instructions for getting more than 10 images and using an API key. It also includes sections for 'What's in the Array of Images' and 'Example Response (Array of Image Objects)' which shows a JSON snippet:

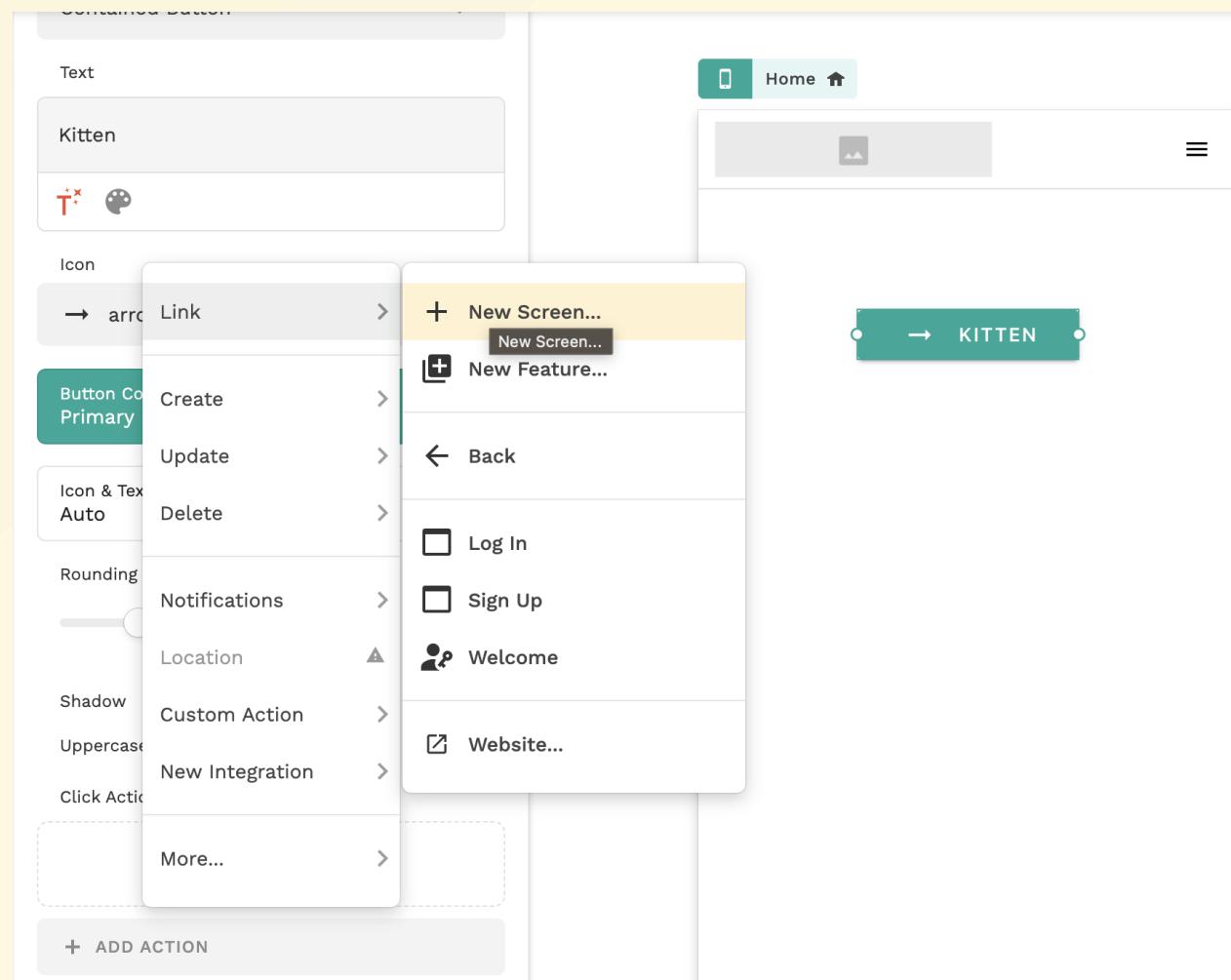
```
[{"id": "ebv", "url": "https://cdn2.thecatapi.com/images/ebv.jpg", "width": 176, "height": 540, "breeds": [], "favourite": {}}]
```

At the bottom, there's a 'Query Parameters (only used with a valid API Key)' section.

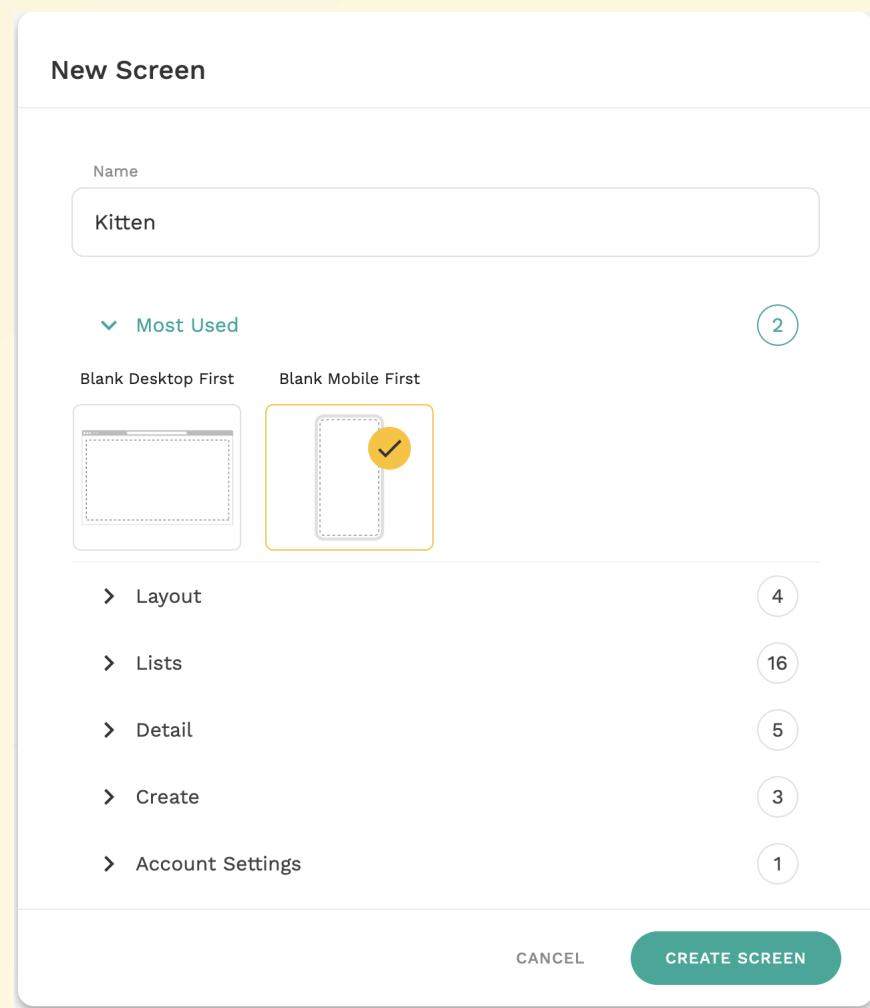
- Adaloの管理画面でCREATE NEW APP
- 設定は以下の通り
 - Platform: Responsive App
 - Template: Blank Mobile First
 - App Name: ApilntegrationTrial



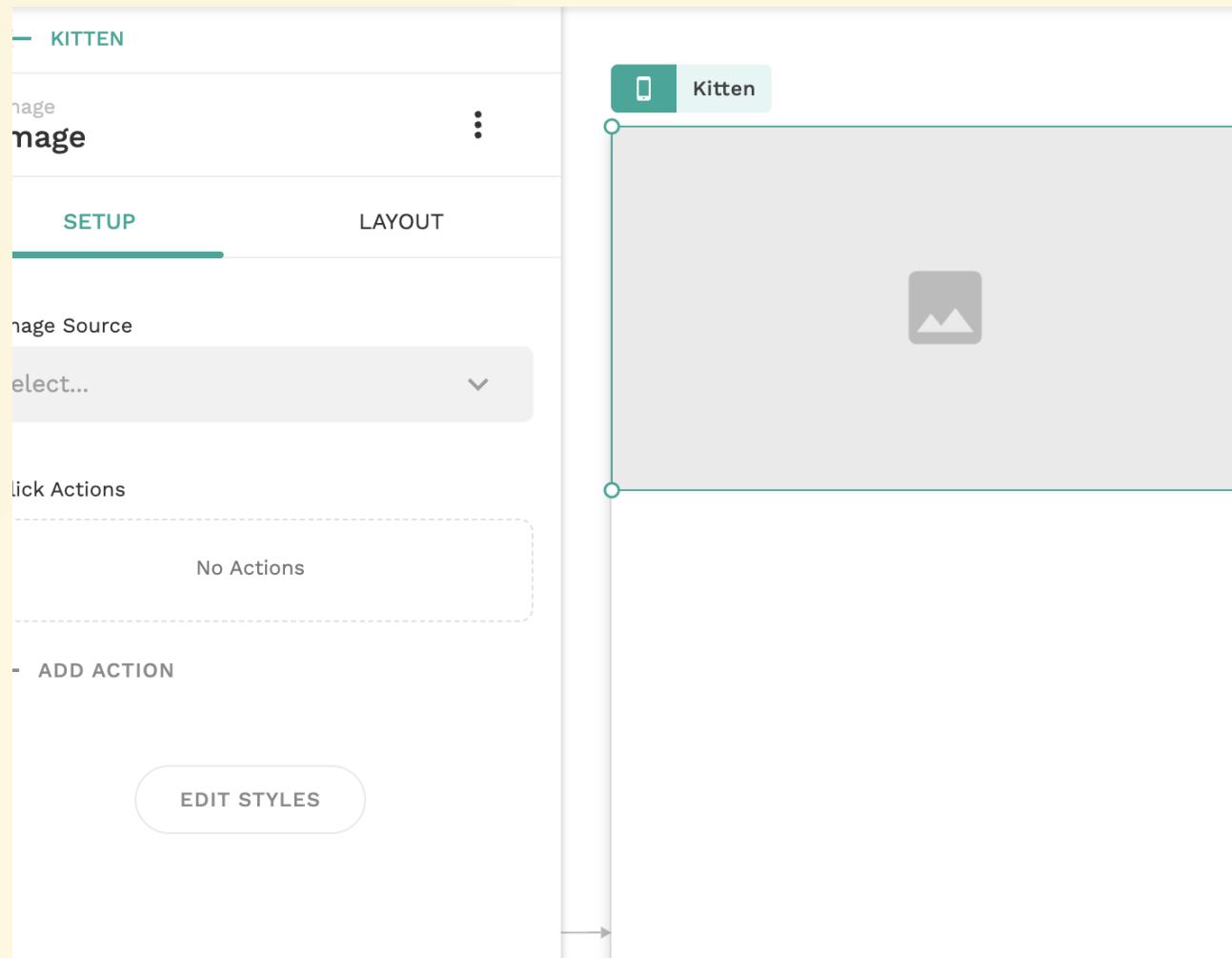
- Home画面に子猫画像表示画面へのリンクボタンを追加
- ADD ACTIONからNew ScreenへのLinkを設定



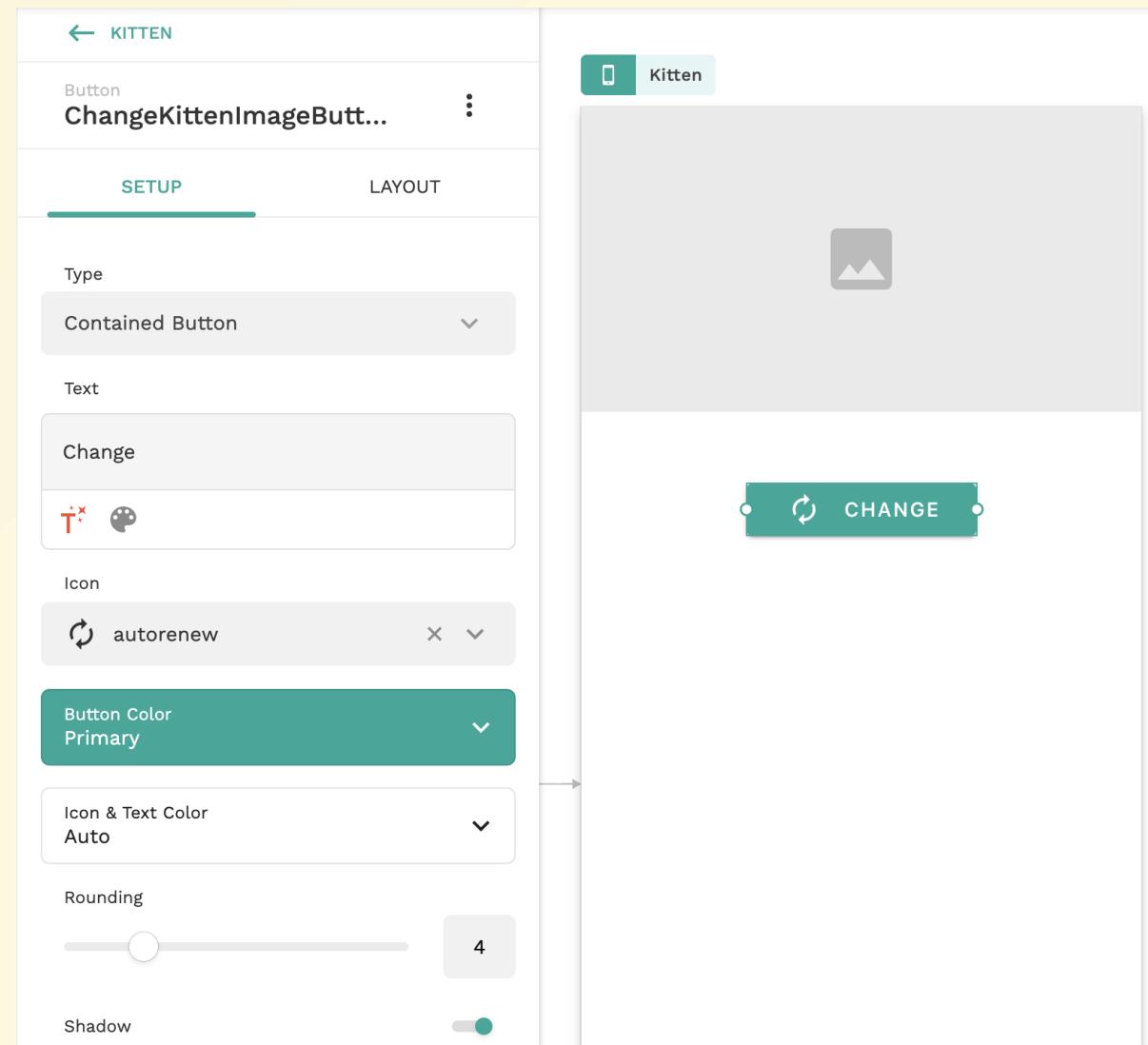
- TemplateにBlank Mobile Firstを選択し、Kitten画面を作成



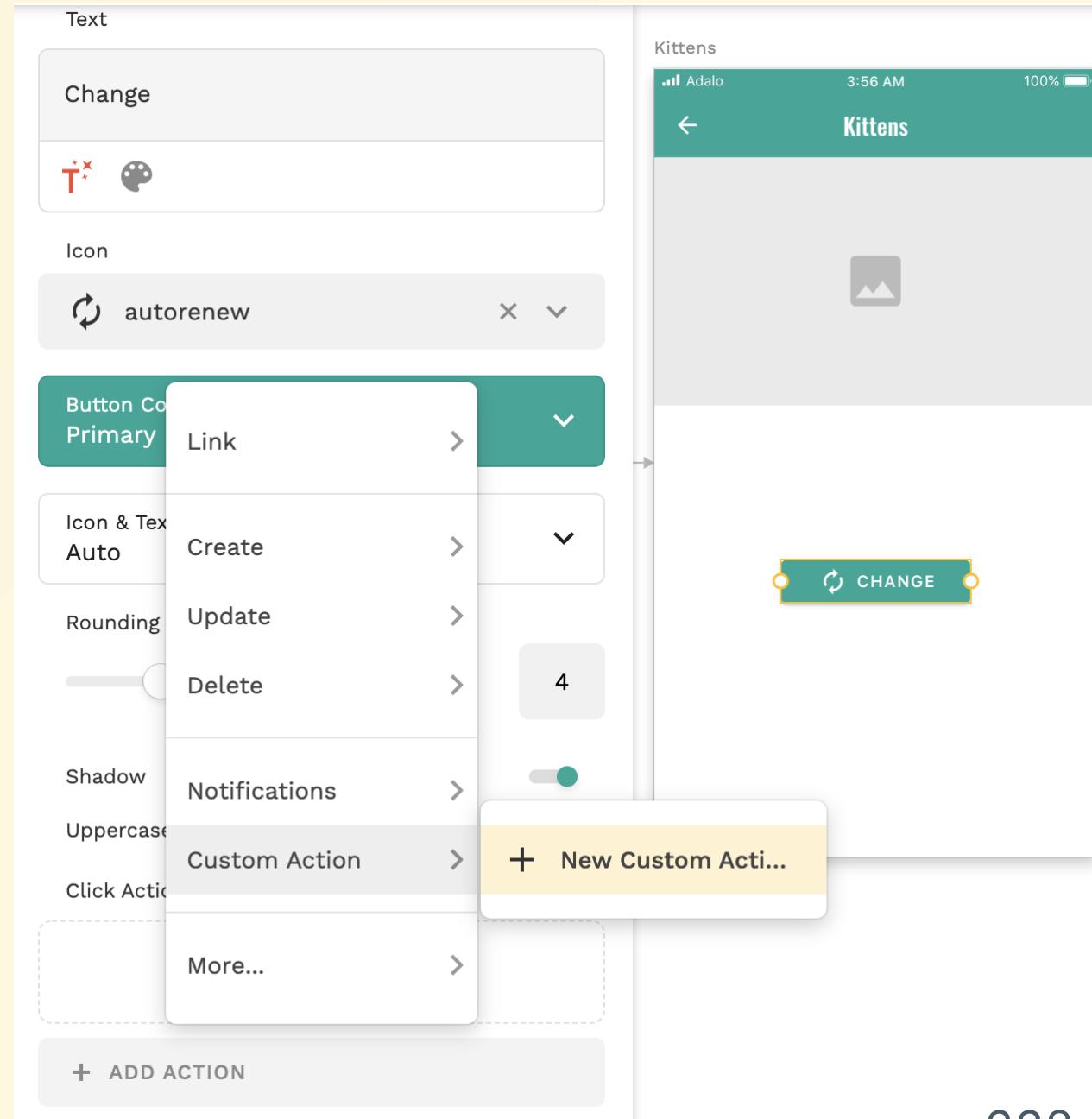
- Imageコンポーネントを画面上に配置
- コンポーネントの設定はそのままにしておく(後で設定します)



- Change Kitten Image
Buttonを追加



- ADD ACTIONからNew Custom Actionを選択



14日間のフリートライアル(無料)の開始を促されます。

- "START FREE TRIAL" をクリック

Unlock Premium Features for 14 Days with a Free Trial

Begin your 14-Day Free trial today & experience premium features including Custom Fonts, External Collections, Custom Actions, Versions, Geolocation & more!

[LEARN MORE](#)

External Collections

Connect & use the data you already have inside your Adalo-made app.

[LEARN MORE](#)

Geolocation NEW

Utilize this popular feature to access your users' locations to customize their experience.

[LEARN MORE](#)

Design Versions NEW

Make changes, save designs, and restore versions with this feature.

[LEARN MORE](#)

Collections API

Make your Collection data accessible through your app's very own custom API.

[LEARN MORE](#)

Custom Actions

Trigger unique actions to extend the functionality available in your app.

[LEARN MORE](#)

Google Fonts

Access over 1000 custom fonts perfect for customizing the look & feel of your app.

[NOT RIGHT NOW](#)[START FREE TRIAL](#)

トライアルが開始しました。

- "CREATE NEW CUSTOM ACTION" をクリック



Enjoy External Collections for the next 14 days!

Don't forget, until November 16th 2023, you have unlimited access to all these great features: External Collections, Custom Actions, Design Versions, Geolocation, Custom Fonts, & Collections API.

DONE

CREATE NEW CUSTOM ACTION

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

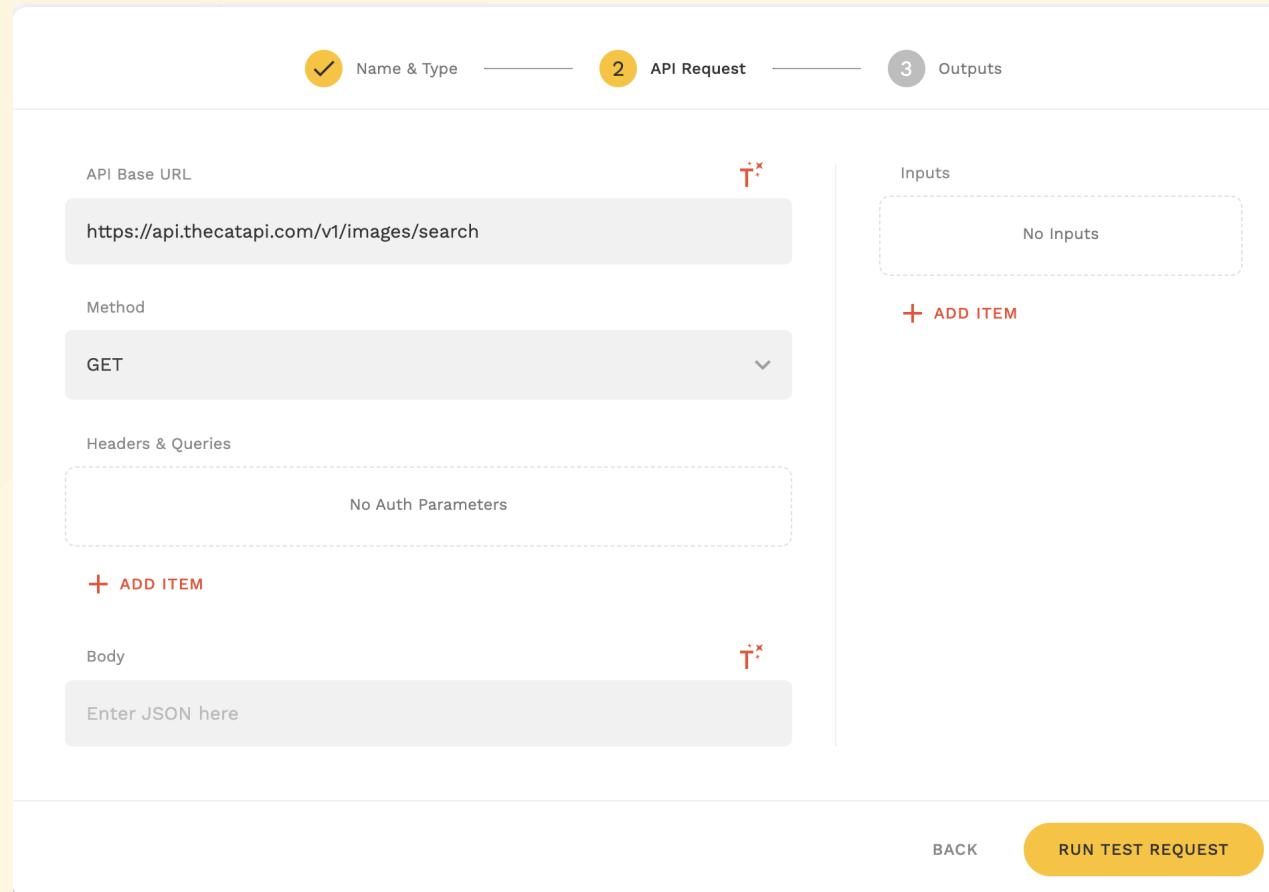
1 Name & Type ————— 2 API Request ————— 3 Outputs

Name
GetRandomKitten

Type
Create

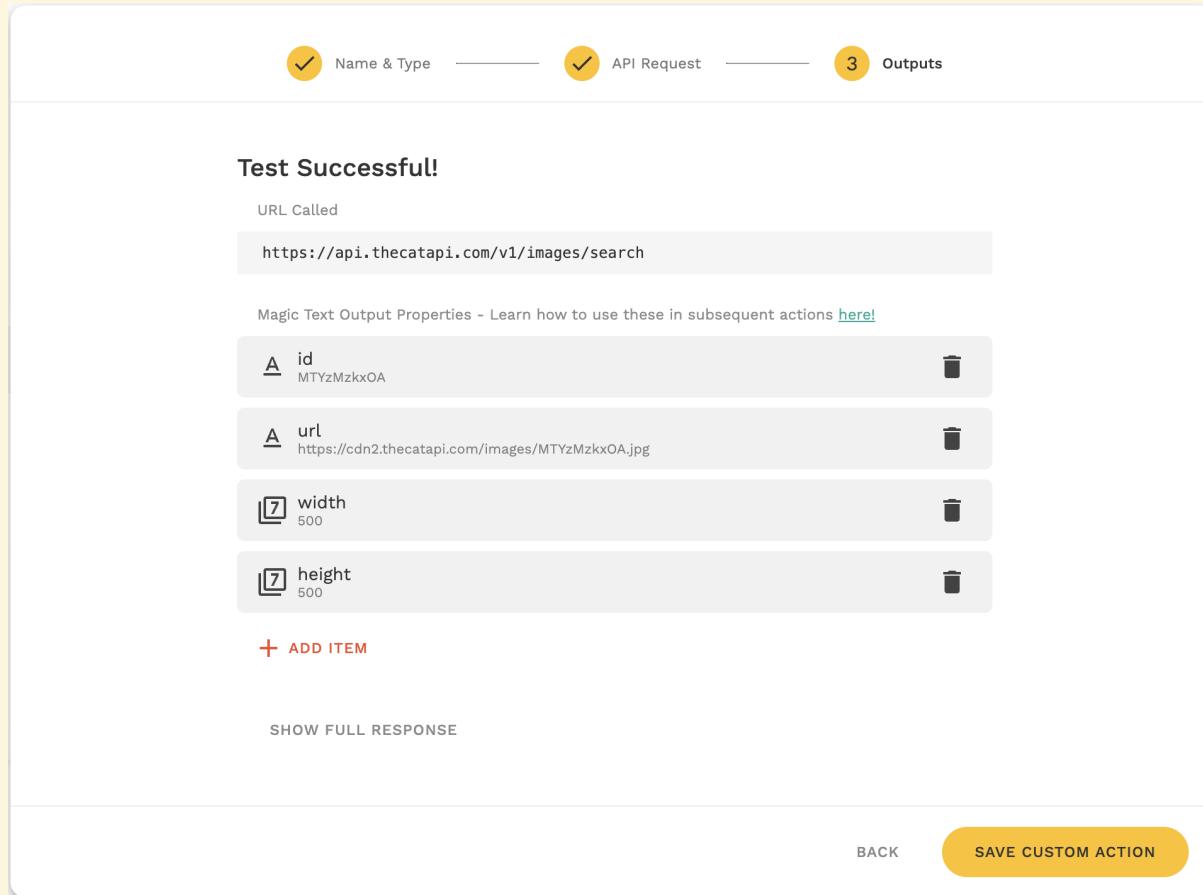
CANCEL NEXT

- 送信するAPI Requestを以下のように設定します。
 - API Base URL:
<https://api.thecatapi.com/v1/images/search>
 - Method: GET
- 設定後、"RUN TEST REQUEST" をクリック



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

- "SAVE CUSTOM ACTION" をクリック

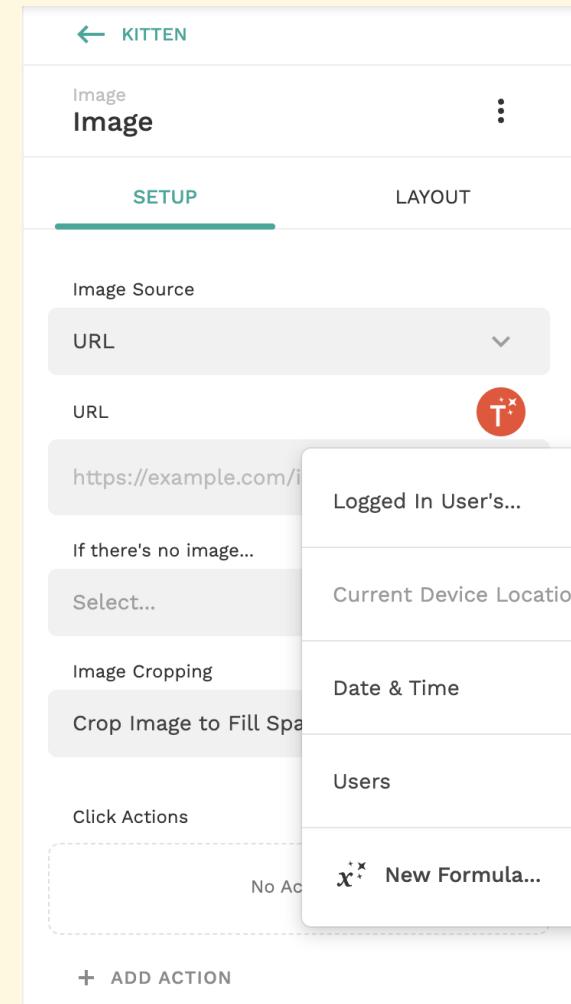
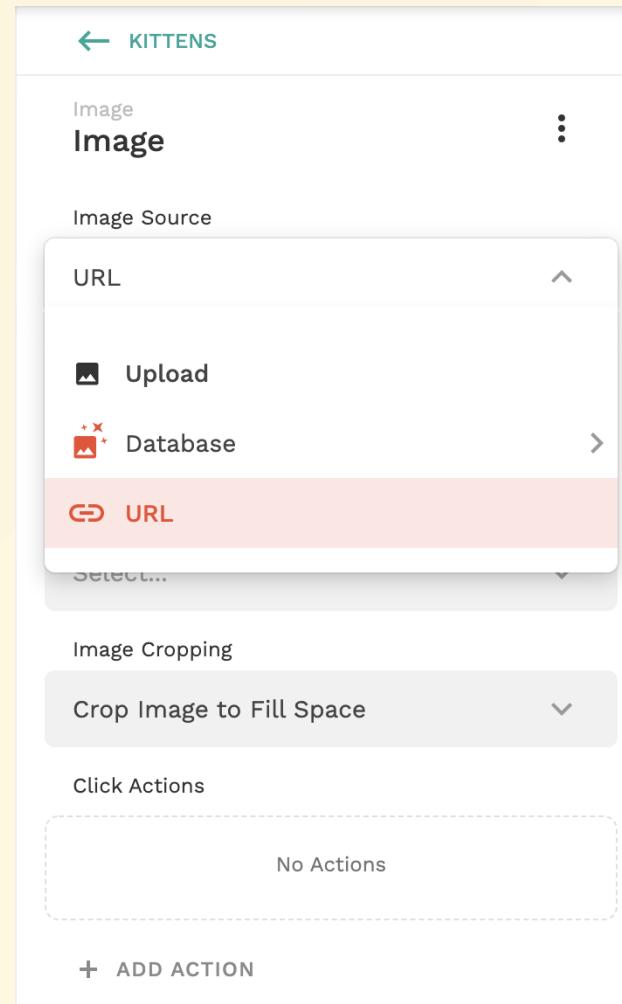


次に、APIから取得した子猫の画像のURLをImageコンポーネントのImage Sourceに設定します。

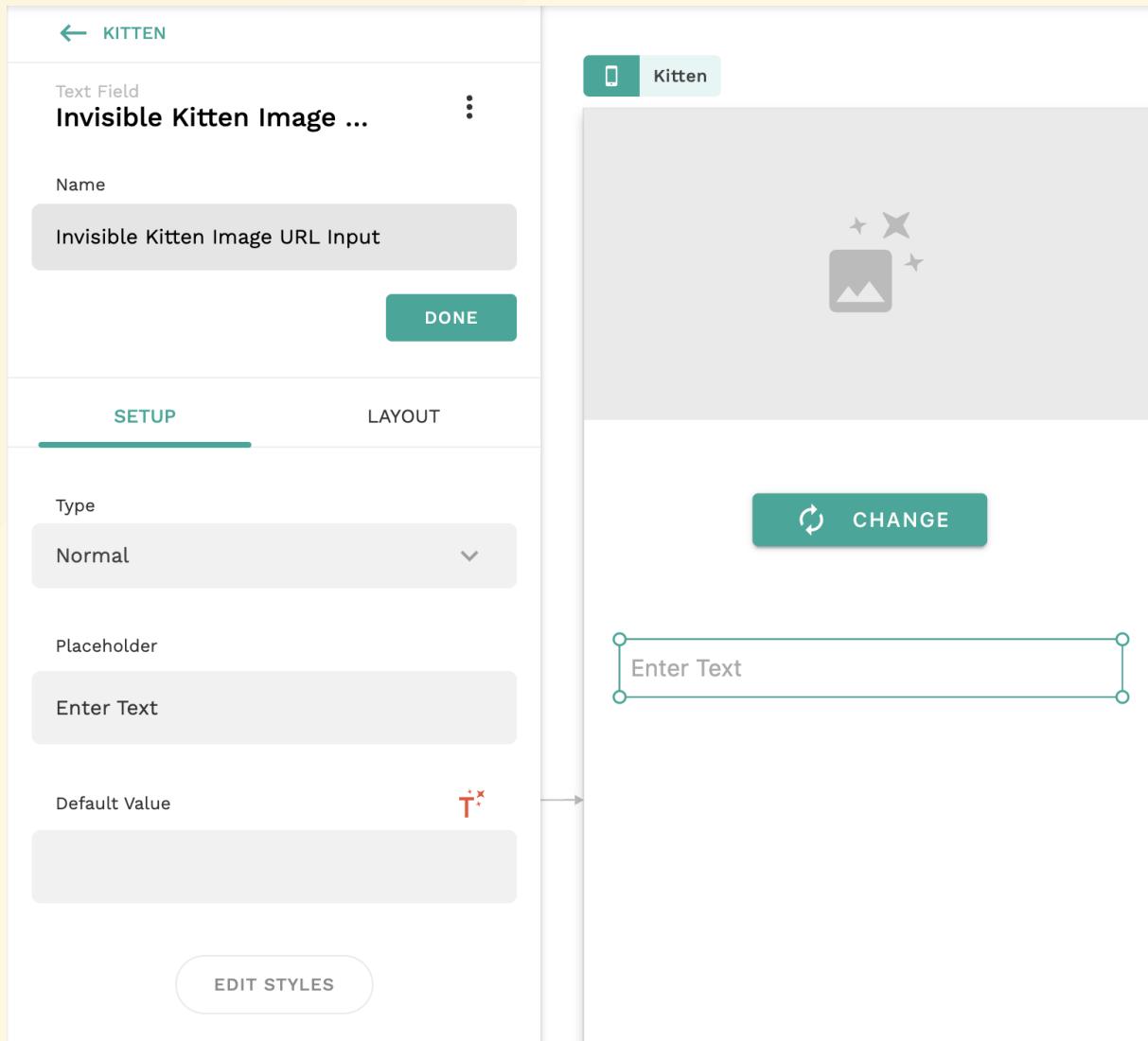
そのままでは、選択肢の中にAPIから取得したデータは出てきません。

A url

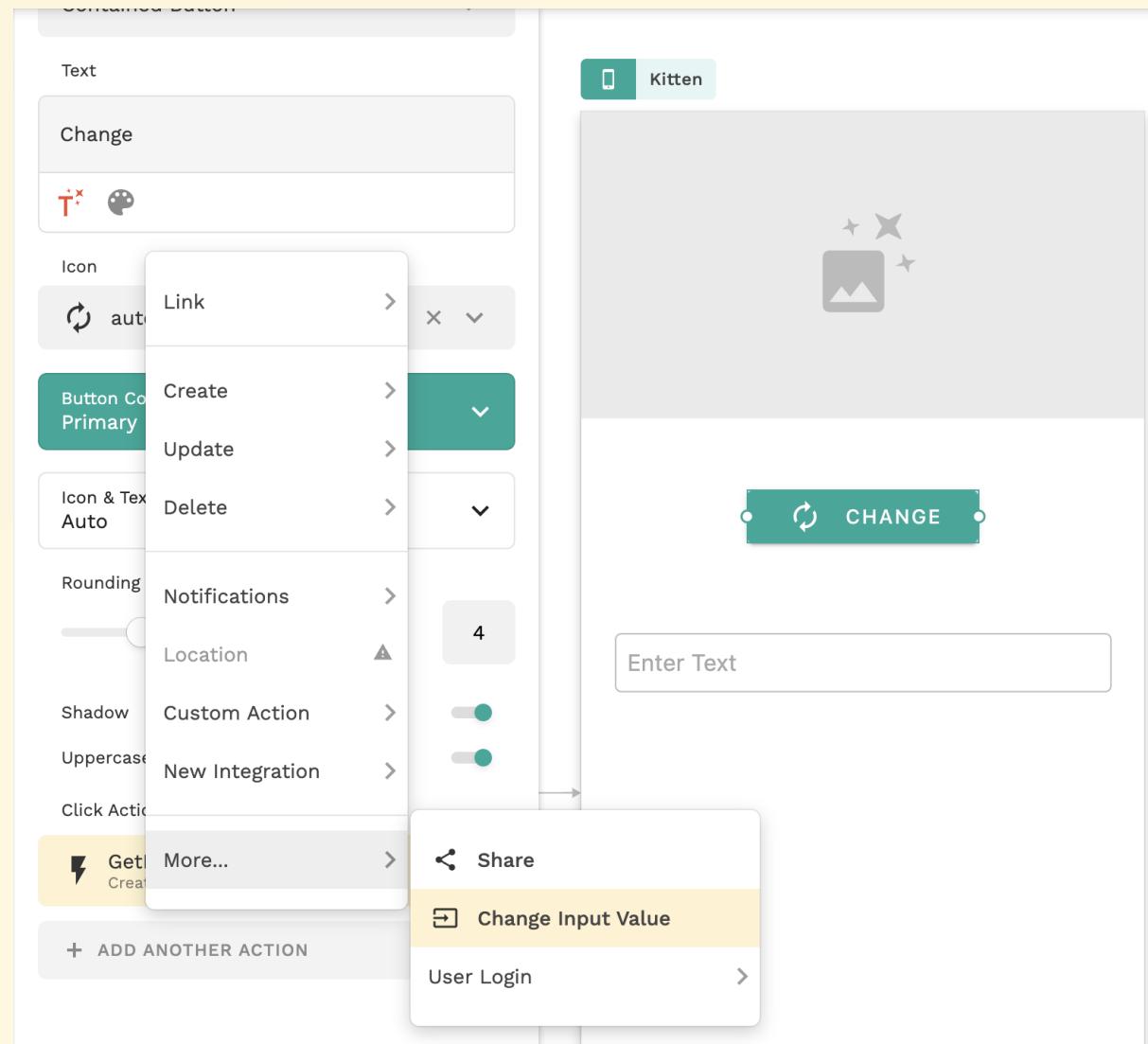
<https://cdn2.thecatapi.com/images/6k7.jpg>



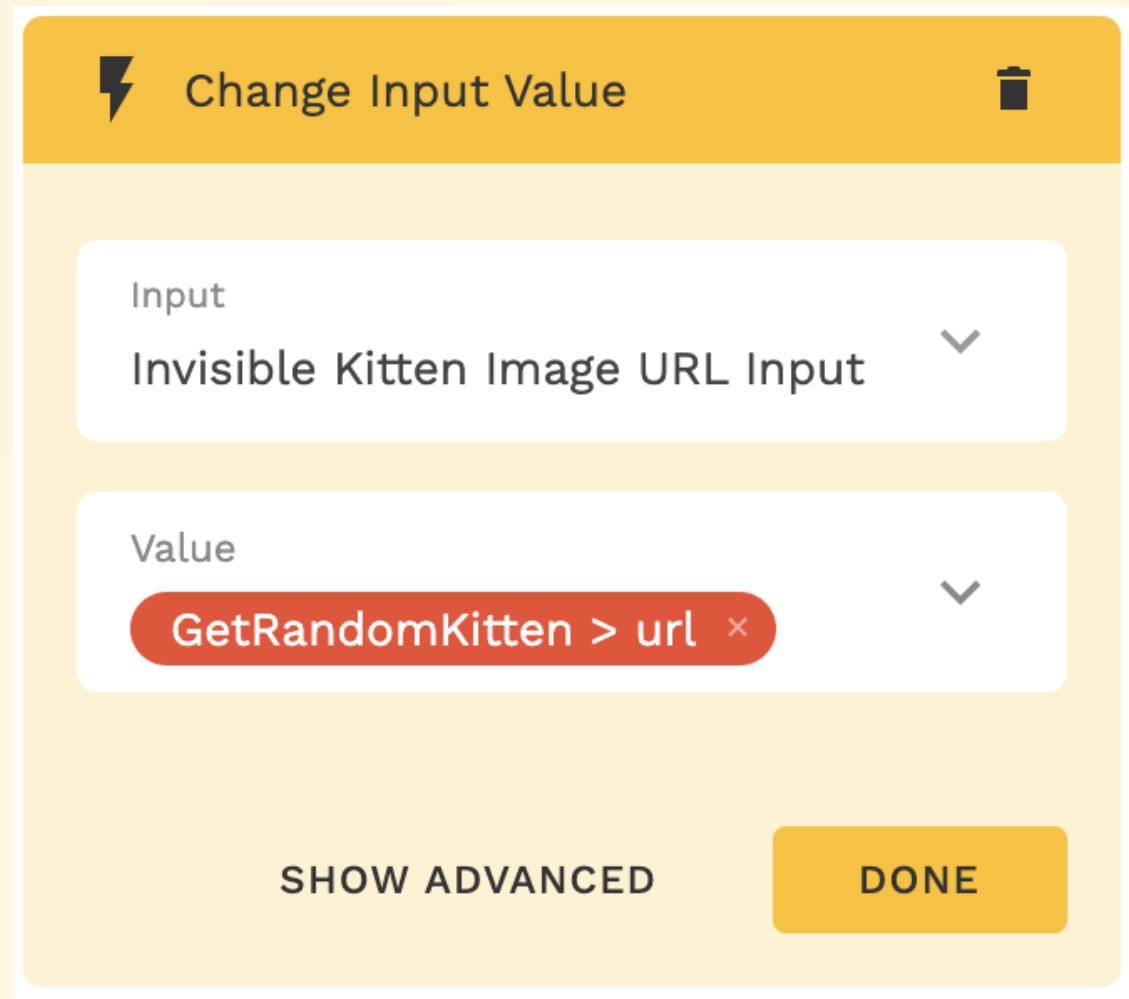
- TextInputコンポーネントを画面上に追加
- Nameを "Invisible Kitten Image URL Input" に変更



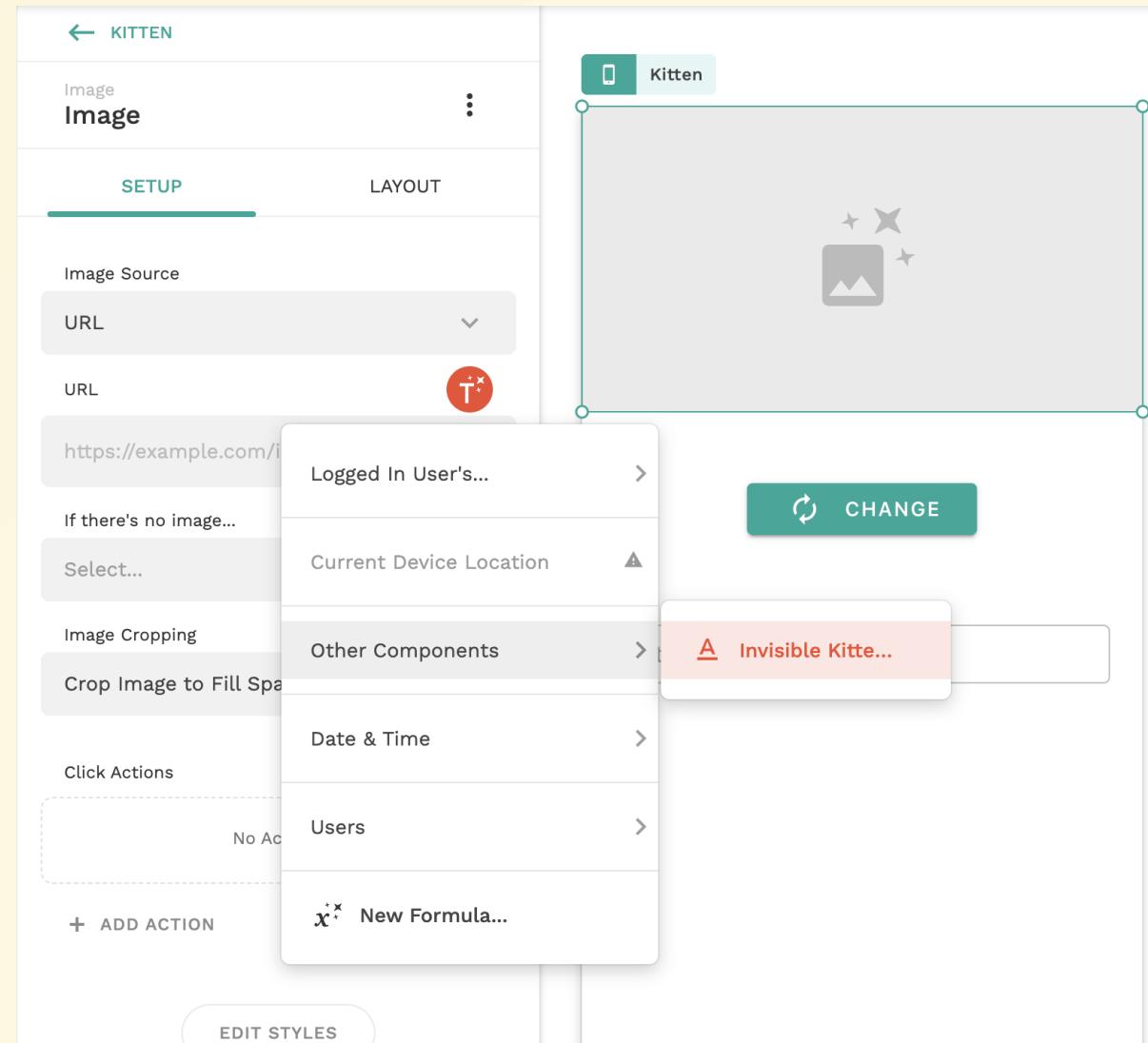
- "Change Kitten Image Button" をクリック
- ADD ANOTHER ACTION から More > Change Input Value を選択



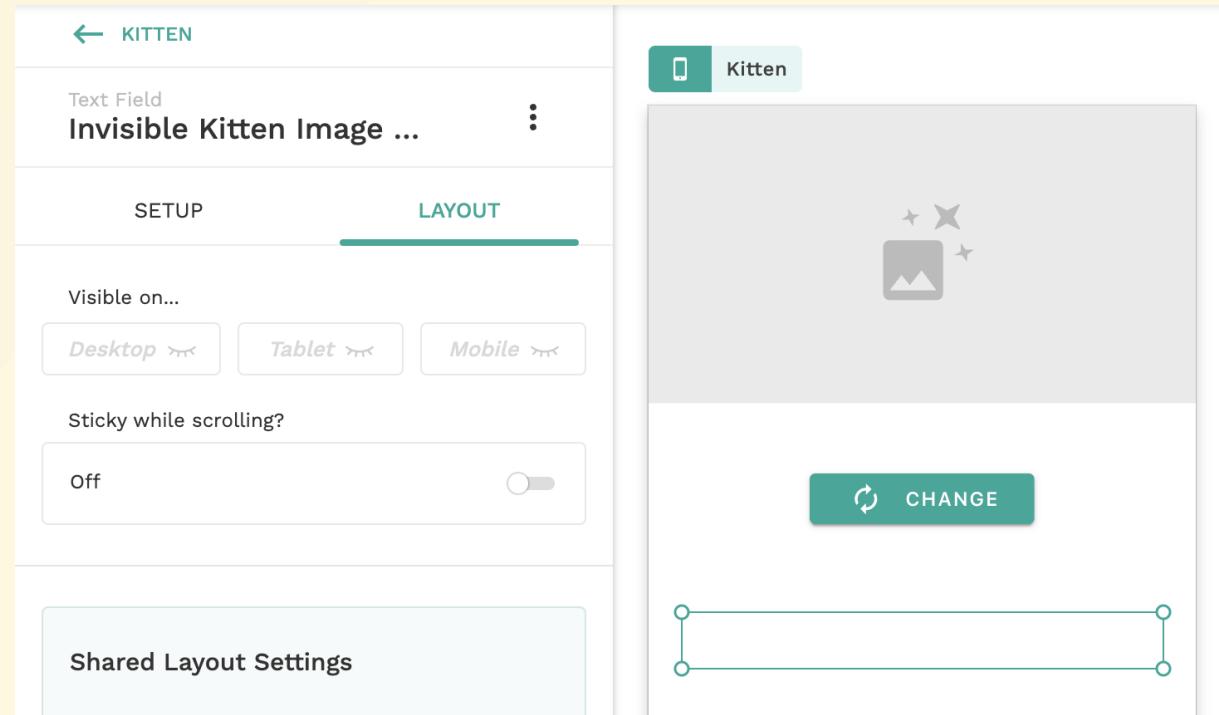
- Inputに "Invisible Kitten Image URL Input" を設定
- Valueに "GetRandomKitten > url" を設定
- "DONE" をクリック



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

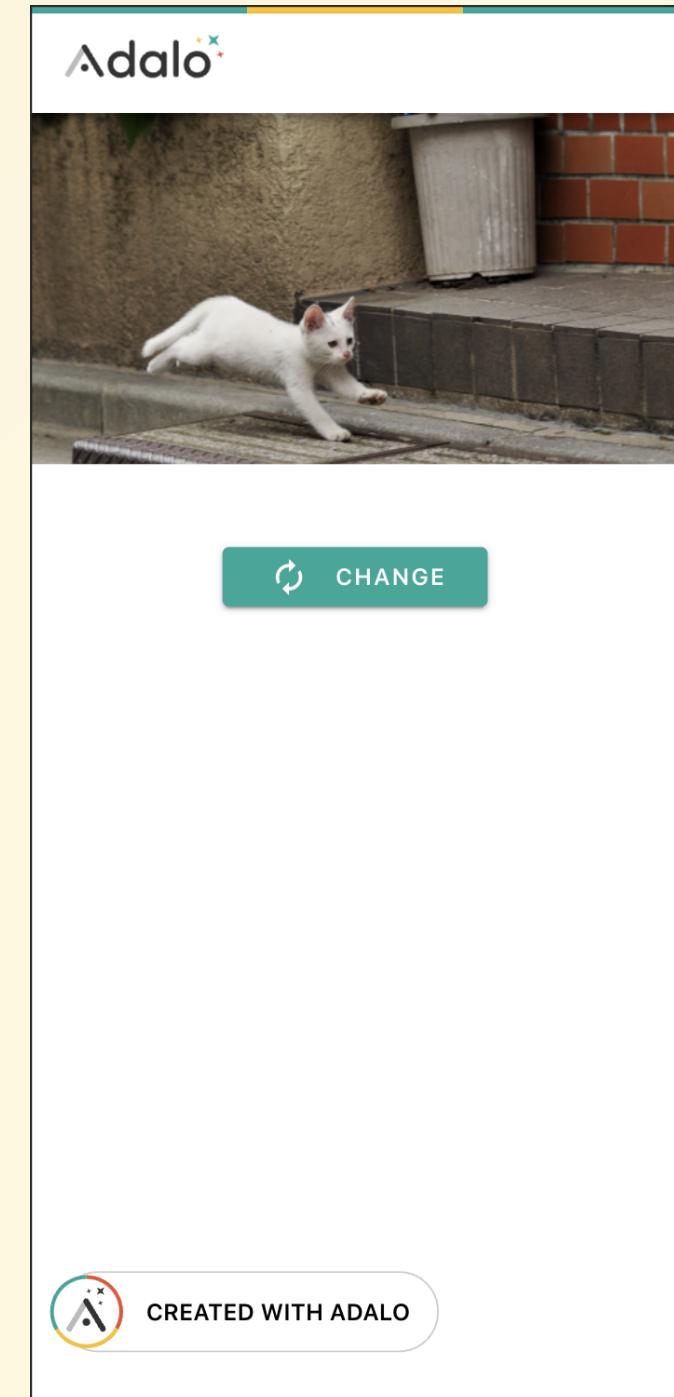


- "Invisible Kitten Image URL Input" のLAYOUTタブを開き、Visible On...で3つある目のアイコンをそれぞれクリックして、非表示にする



Preview機能で確認します。

CHANGEボタンをクリックすると、子猫の画像が表示されました。



補足

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

注意事項

現状、Custom Actionにはいくつかの制限があります。

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

External Collection

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

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

先ほどのAPIの末尾に ?limit=10 を付け加えて、猫の画像を10件取得します。

<https://api.thecatapi.com/v1/images/search?limit=10>

The screenshot shows the 'TheCatAPI - Documentation Portal' website. On the left, there's a sidebar with various links: Authentication, Basics: Getting Images (which is highlighted), Basics: Filter By Breed, TheDogAPI.com, Basics: Voting, Discord Server, NPM node library installation, Basics: Favouriting, Code Samples, OpenAPI Spec Doc, Postman API Docs, and More APIs... A blue arrow icon is next to the Postman API Docs link.

Quick start

Click on this to get a random Image <https://api.thecatapi.com/v1/images/search>

Click this to get 10 random images <https://api.thecatapi.com/v1/images/search?limit=10>

Copy this link, add your own API Key to get 10 bengal images https://api.thecatapi.com/v1/images/search?limit=10&breed_ids=beng&api_key=REPLACE_ME

Getting Images

To get more than 10, and additional fields then **be sure to** use your API Key from the welcome email as the `'x-api-key'` header, or `?api_key=` query string parameter to access all the images and data.

What's in the Array of Images

Each contains the `url` for the image file, along with its `width`, `height` and `breed` information (if available), any `favourite` or `vote` you create would also be returned.

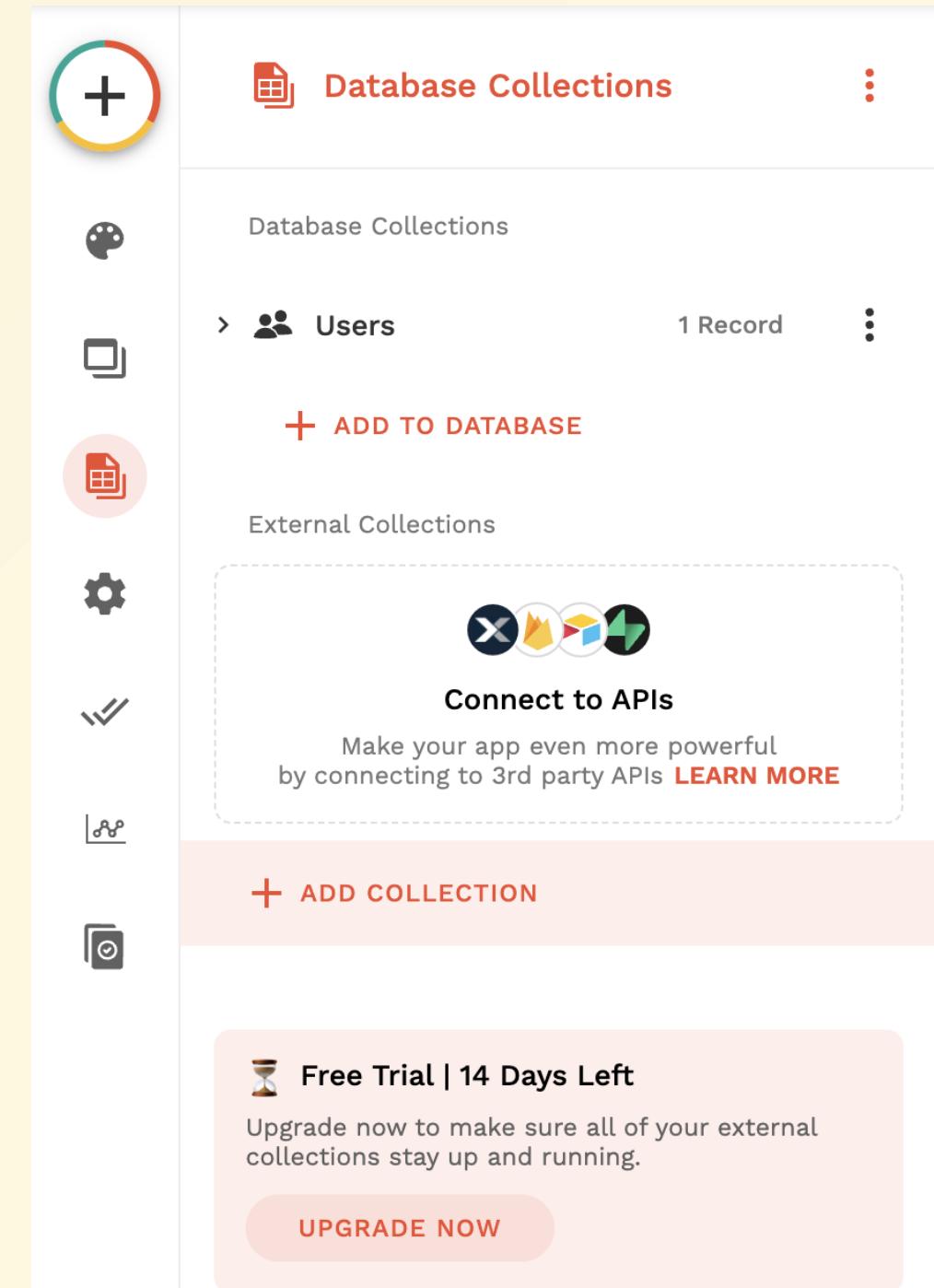
Example Response (Array of Image Objects)

```
[{"id": "ebv", "url": "https://cdn2.thecatapi.com/images/ebv.jpg", "width": 176, "height": 540, "breeds": [], "favourite": {}}]
```

Query Parameters (only used with a valid API Key)

Powered by  reportwrapper

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



- 以下を設定して、NEXTをクリック

- Collection Name: Kittens
- Base URL:
[https://api.thecatapi.com/v1/images/search?
limit=10](https://api.thecatapi.com/v1/images/search?limit=10)

The screenshot shows a user interface for setting up a collection. At the top, there are three tabs: 1. Name & URL (highlighted in red), 2. Endpoints, and 3. Test Connection. The first tab contains fields for 'Collection Name' (Kittens) and 'API Base URL' (https://api.thecatapi.com/v1/images/search?limit=10). The second tab shows 'Auth Setup' with a note 'No Auth Parameters'. A red 'ADD ITEM' button is visible. At the bottom, there are 'CANCEL' and 'NEXT' buttons.

1 Name & URL 2 Endpoints 3 Test Connection

Collection Name
Kittens

API Base URL
https://api.thecatapi.com/v1/images/search?limit=10

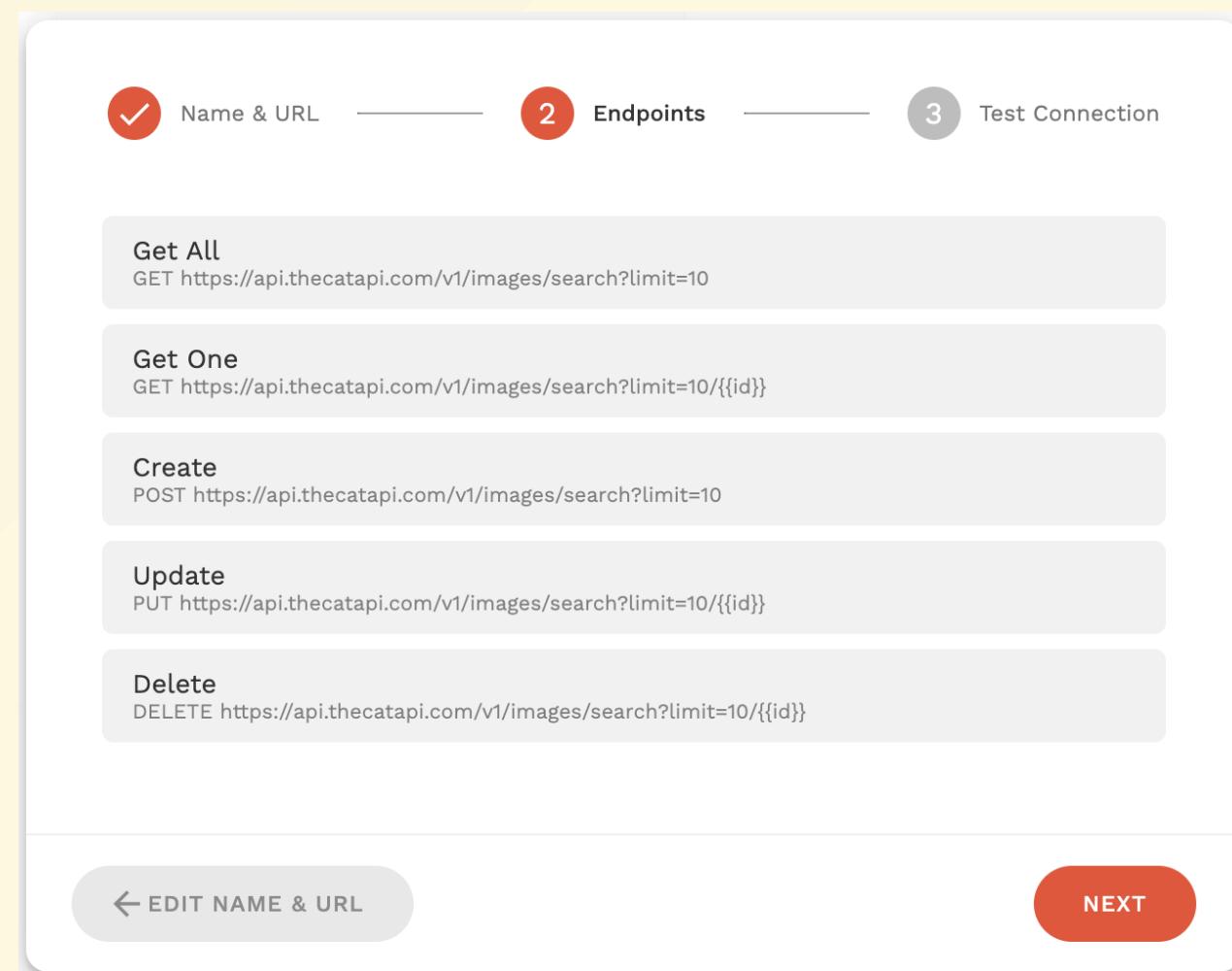
Auth Setup
No Auth Parameters

+ ADD ITEM

CANCEL NEXT

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

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



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

Name & URL Endpoints 3 Test Connection

Test Successful!

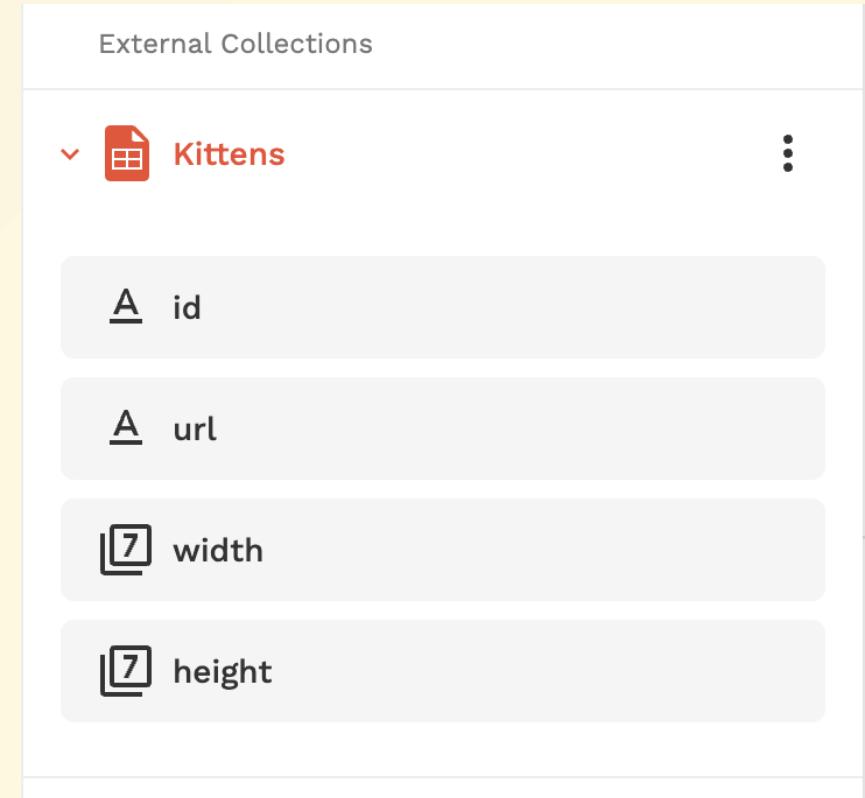
Here's the full response we got back:

```
[  
  {  
    "id": "1v8",  
    "url": "https://cdn2.thecatapi.com/images/1v8.jpg",  
    "width": 500,  
    "height": 333  
  },  
  {  
    "id": "398",  
    "url": "https://cdn2.thecatapi.com/images/398.jpg",  
    "width": 631,  
    "height": 1024  
  },  
  {  
    "id": "4g1",  
    "url": "https://cdn2.thecatapi.com/images/4g1.gif",  
    "width": 225,  
    "height": 300  
  },  
  {  
    "id": "5pm",  
    "url": "https://cdn2.thecatapi.com/images/5pm.jpg",  
    "width": 600,  
    "height": 415  
  },  
  {  
    "id": "8er",  
    "url": "https://cdn2.thecatapi.com/images/8er.jpg",  
    "width": 500,  
    "height": 462  
  },  
  {  
    "id": "8rj",  
    "url": "https://cdn2.thecatapi.com/images/8rj.jpg",  
    "width": 917,  
    "height": 610  
  }]
```

[← EDIT ENDPOINTS](#) [SAVE COLLECTION](#)

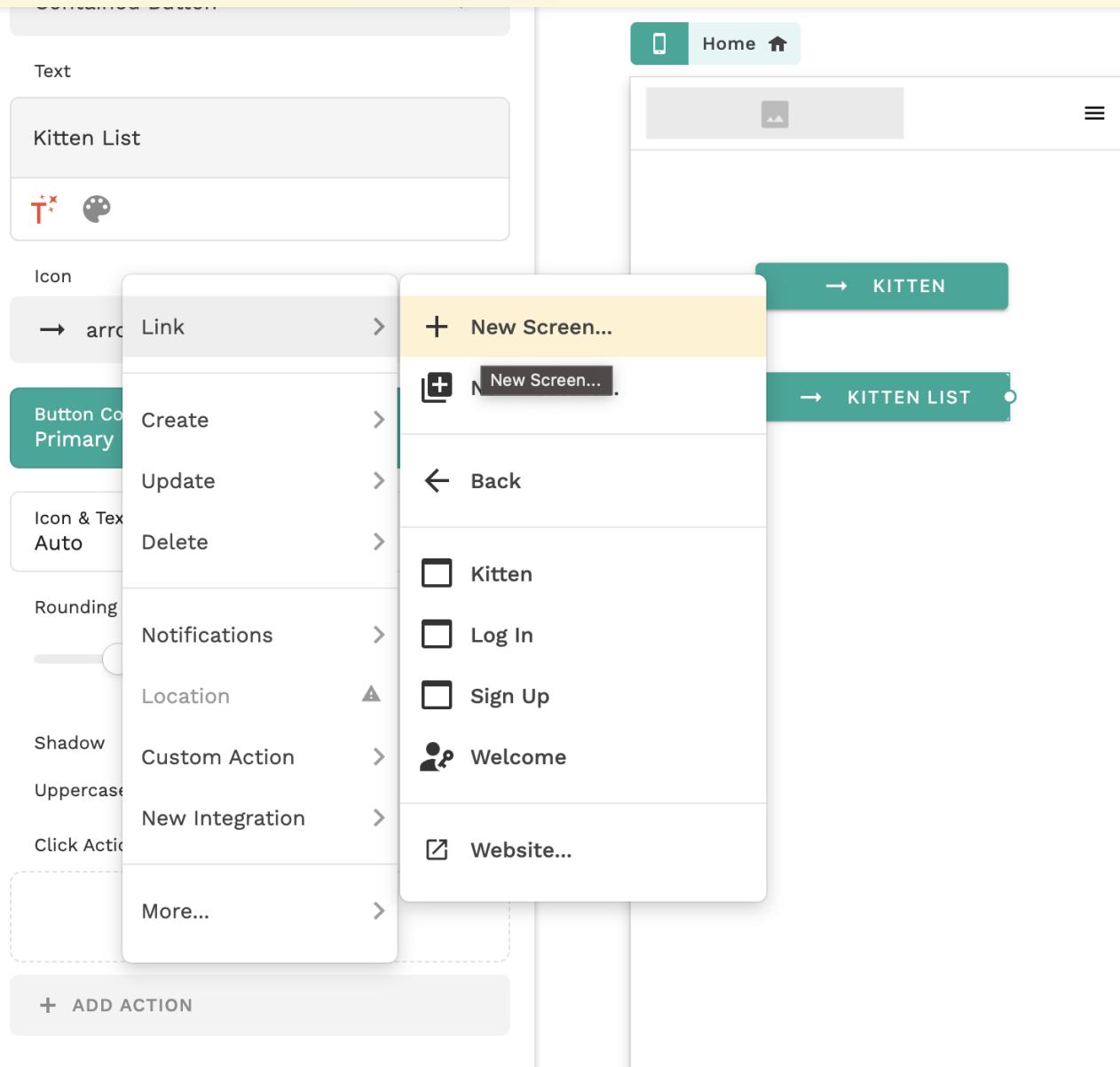
External Collectionが作成されました。

APIから取得するデータが全てプロパティとして設定されています。

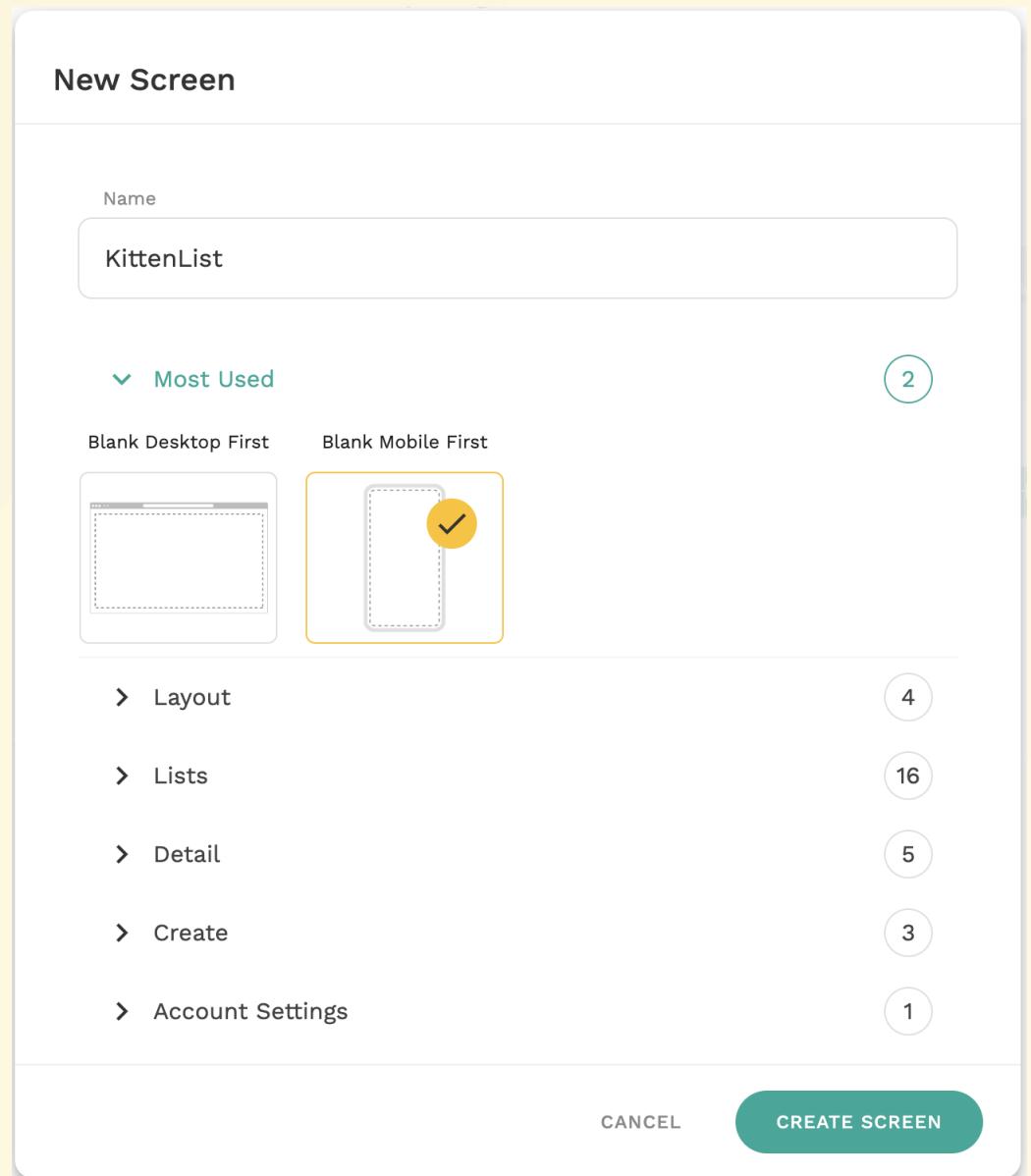


取得したデータを一覧表示しましょう。

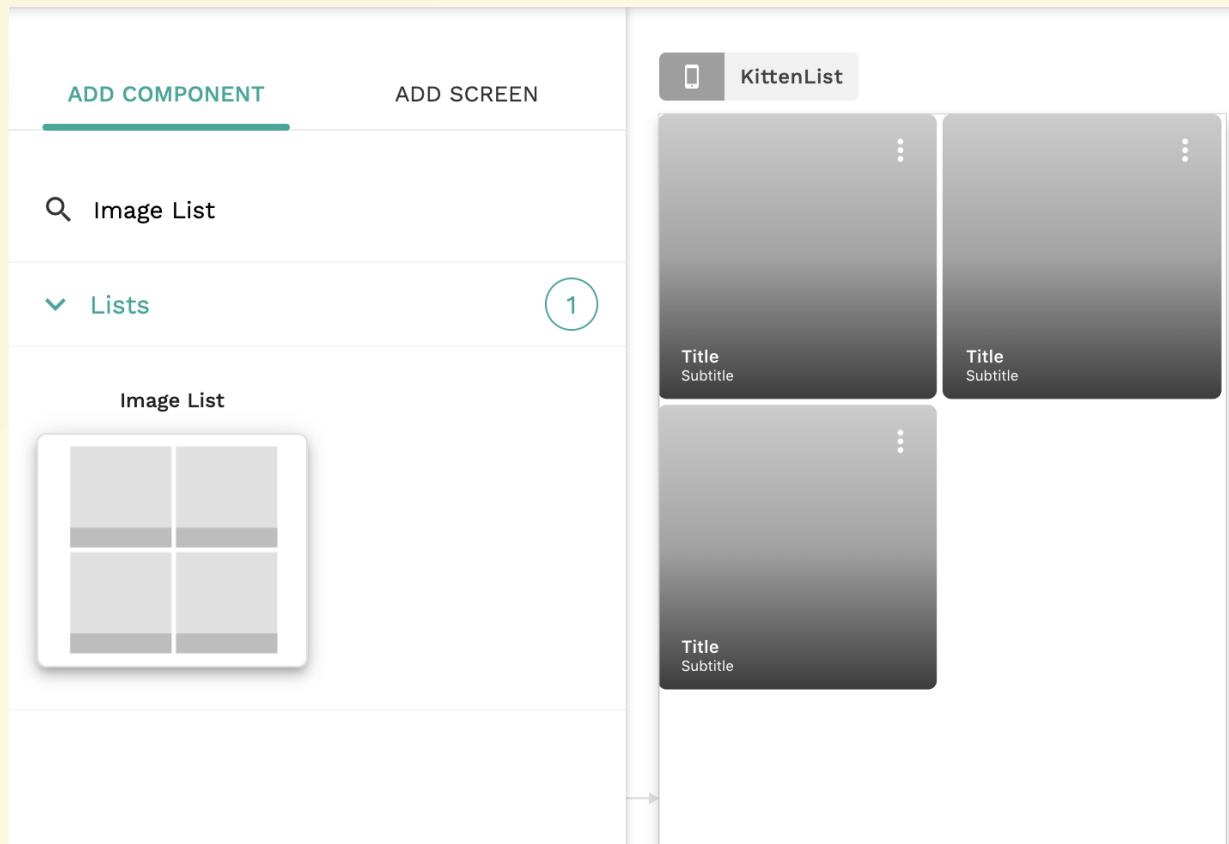
- Home画面に "Kitten List Link" ボタンを追加
- ADD ACTIONからNEW SCREENへのLINKを追加



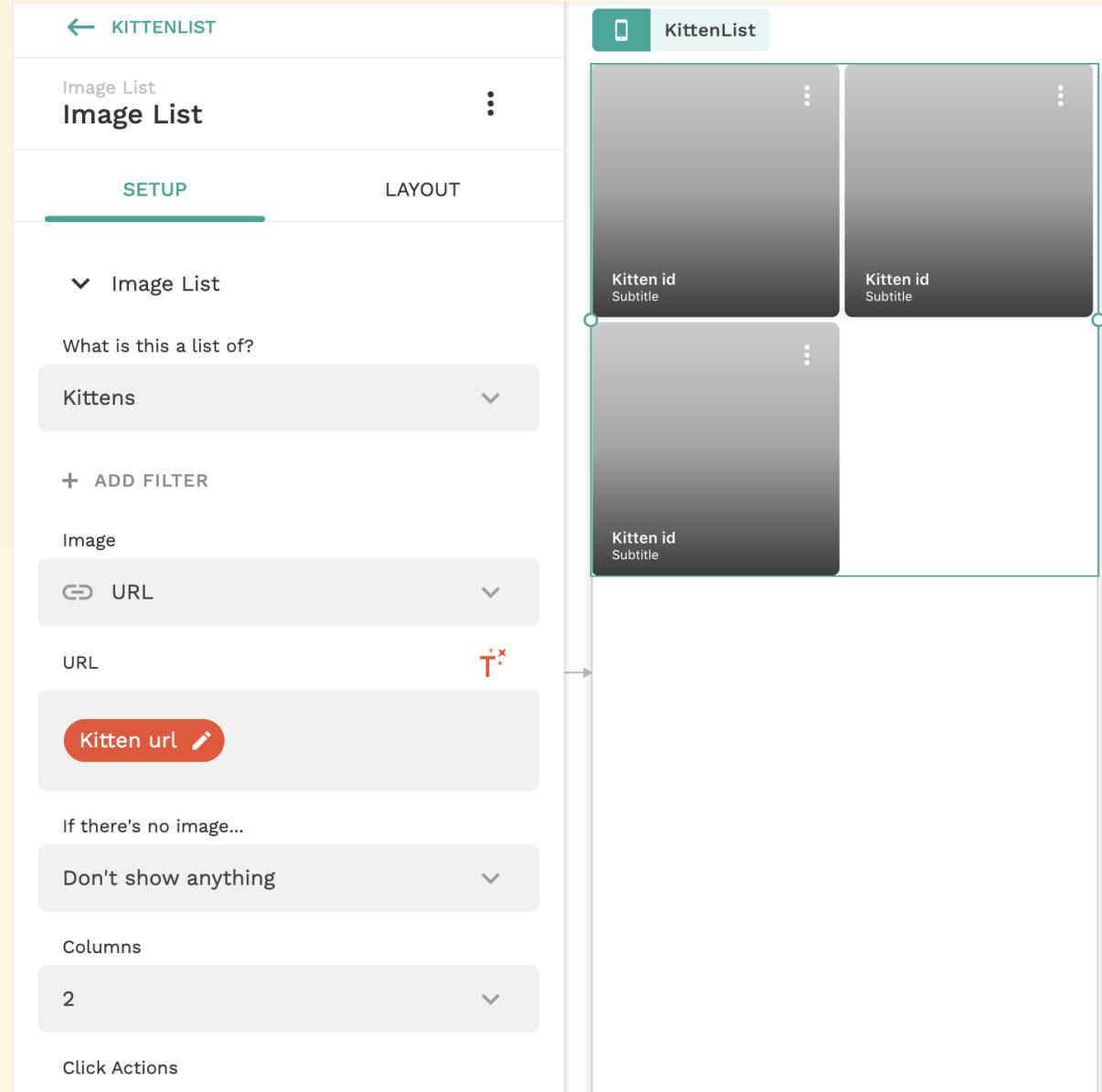
- BlankMobileFirstを選択して、"KittenList"スクリーンを作成



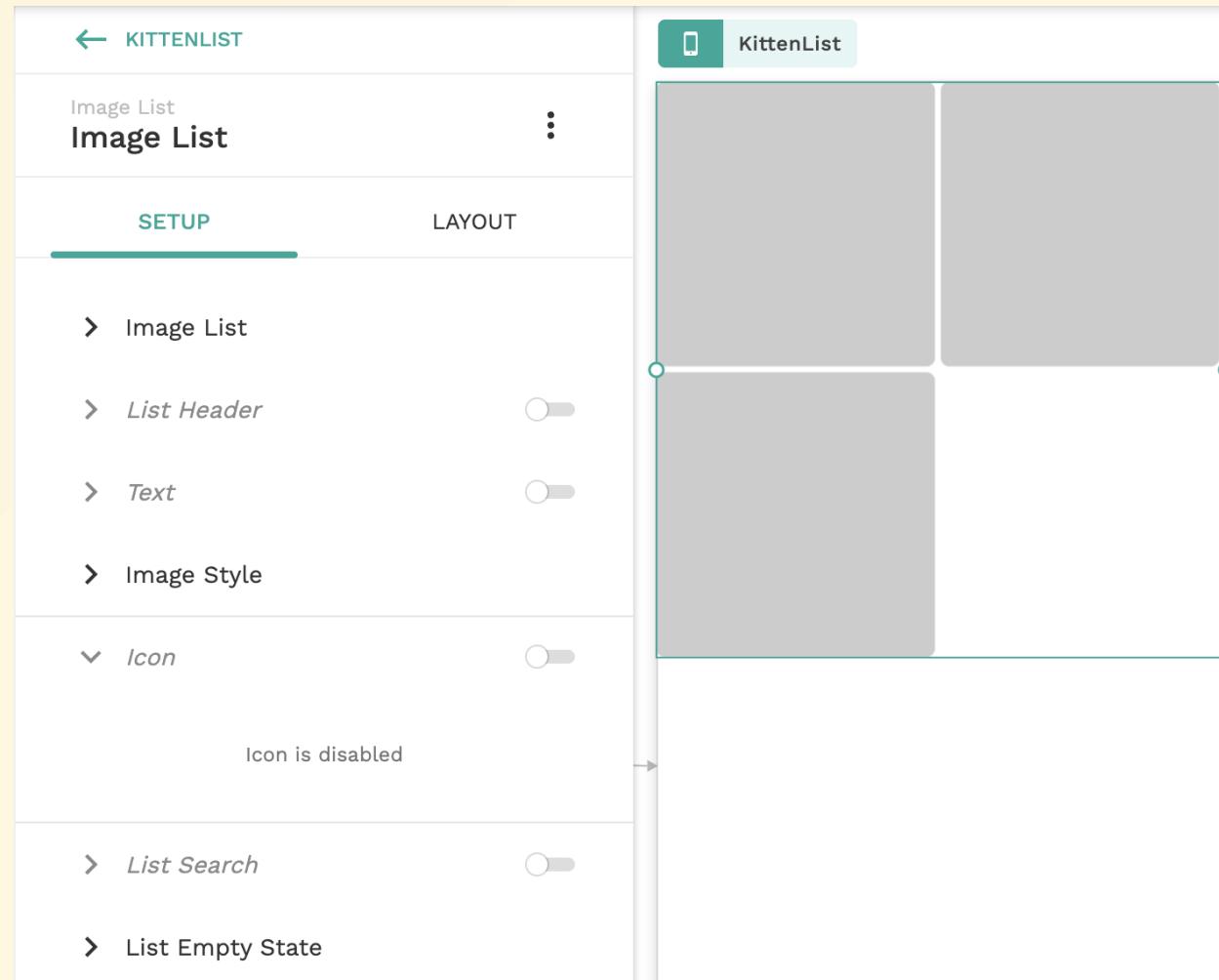
- ADD COMPONENTから
Image Listを追加



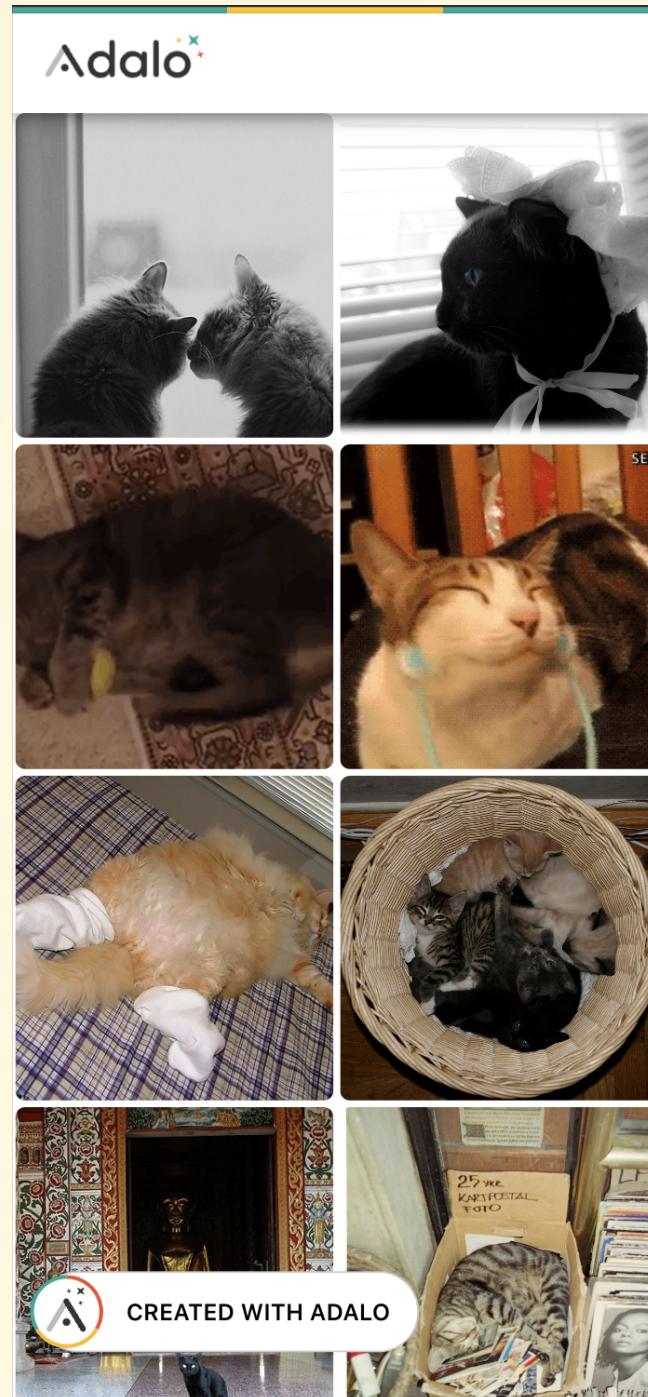
- Kittens Collectionのリストとして設定
- ImageのURLに "kitten url" を設定
- "If there's no image..." に "Don't show anything" を設定



- TextとIconは不要なので非表示にする



Preview機能で確認すると、猫のリストが表示されました。

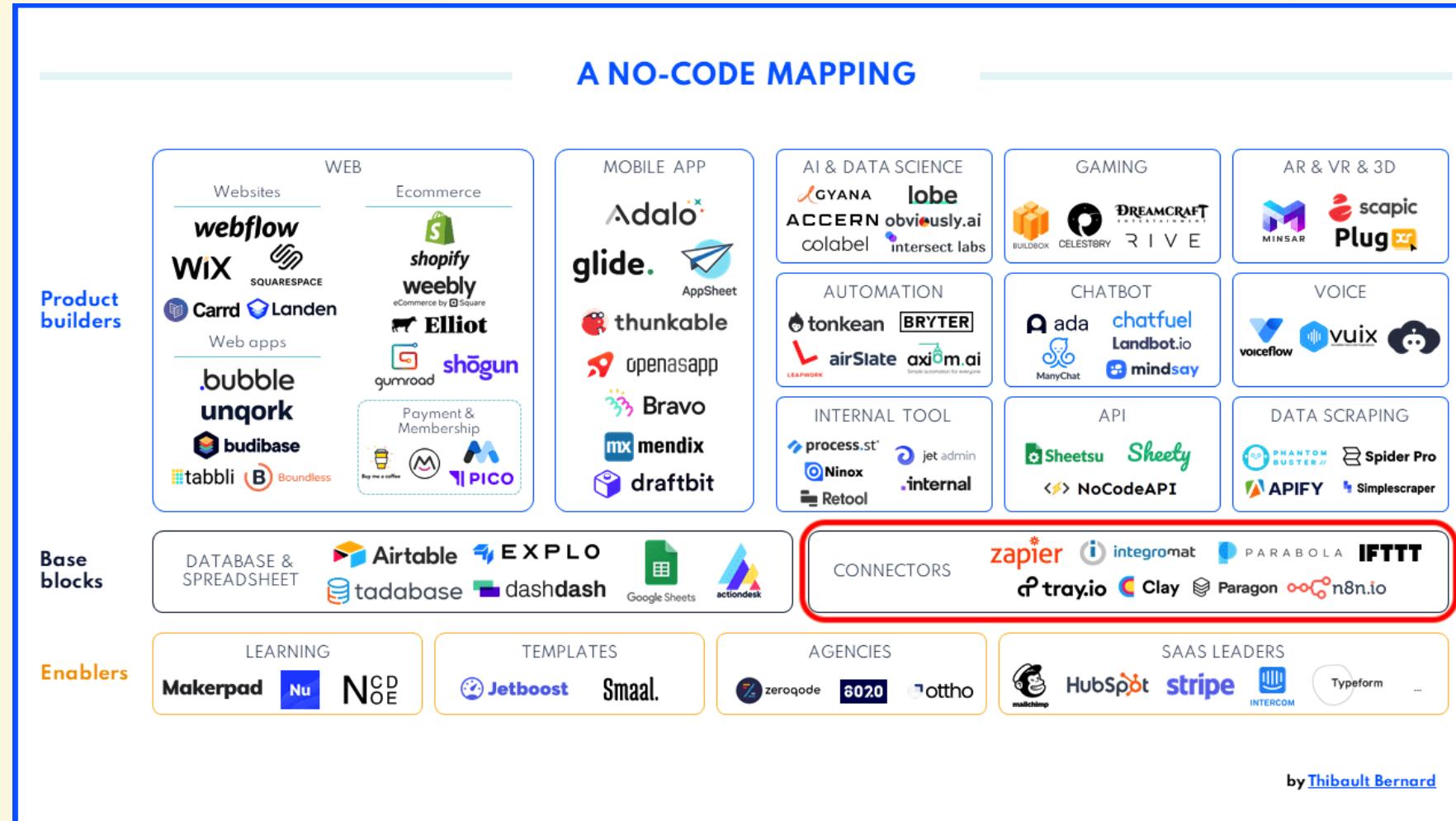


連携サービス

Custom ActionやExternal Collectionでは、連携先のサービスのAPIの仕組みをドキュメントから理解するのが大変かもしれません。

次は、もっと簡単に外部サービスとの連携ができるサービスをご紹介します。

連携サービスにも色々なものがありますが、今回はAdaloに組み込まれているZapierというサービスを紹介します。



Adaloにはメール送信機能がありません。

今回は、Zapierを使って、
AdaloとGmailを連携させることで、アプリへSignUpした人に自動でWelcomeメールが送信されるようにしていきます。

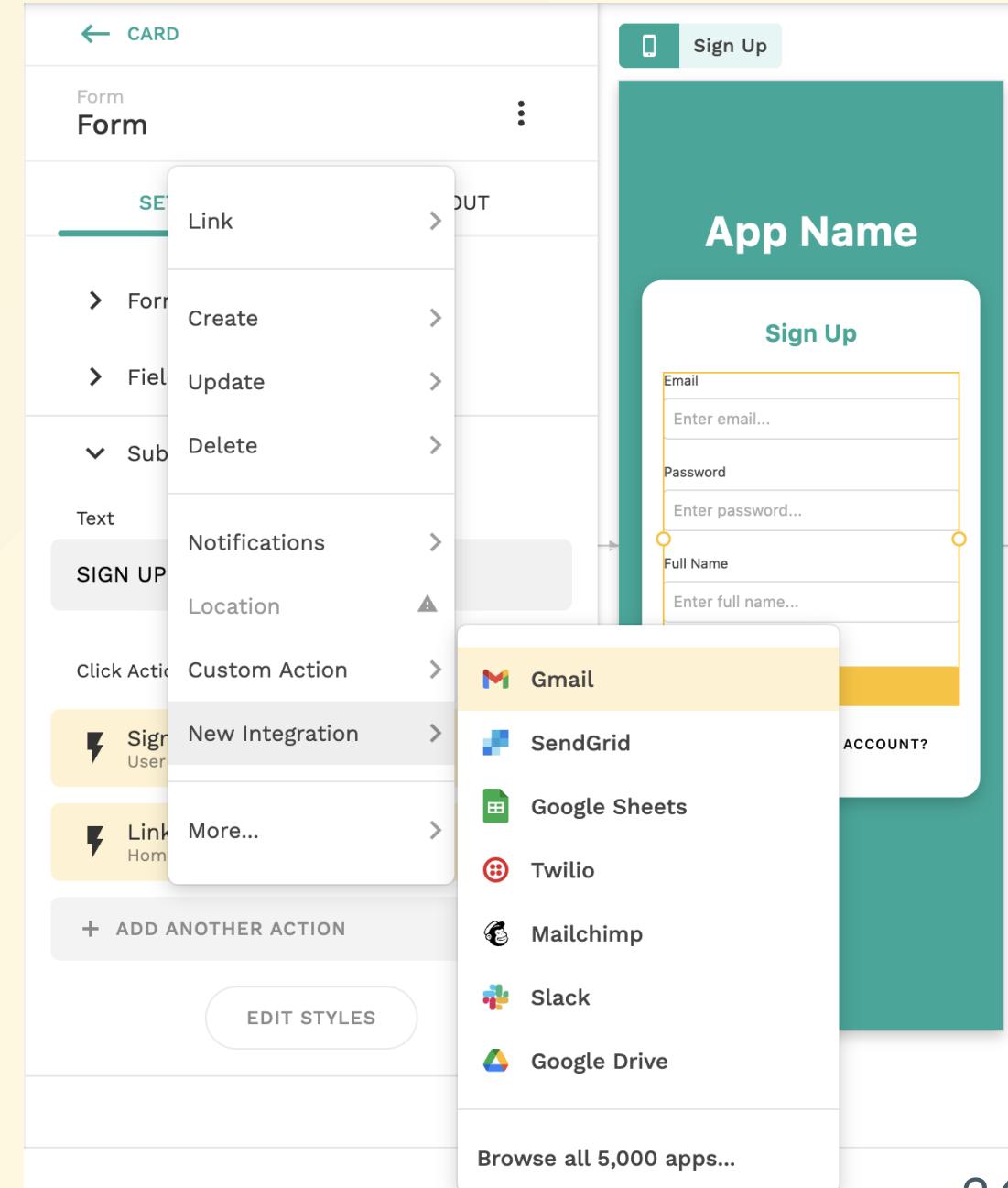


- Gmailを使うために必要なので、Googleアカウントを作成します。
 - 既に持っている場合はそのアカウントを使えば良いので、作成は不要です。
 - 持っていない人は、こちらのURLから私と一緒に作成してください。

[https://accounts.google.com/signup/v2/webcreateaccount?
continue=https%3A%2F%2Faccounts.google.com%2FManageAc
count%3Fnc%3D1&dsh=S50453738%3A1637917137418951&
biz=false&flowName=GlifWebSignIn&flowEntry=SignUp](https://accounts.google.com/signup/v2/webcreateaccount?continue=https%3A%2F%2Faccounts.google.com%2FManageAccount%3Fnc%3D1&dsh=S50453738%3A1637917137418951&biz=false&flowName=GlifWebSignIn&flowEntry=SignUp)

Googleアカウントが用意できたら、Zapierとの連携を設定していきます。

- ApilntegrationTrialアプリのSign Up画面でSubmit Buttonを選択
- ADD ANOTHER ACTION > New Integration > Gmail を選択



- 一番上のSend Gmail emails for new Adalo recordsのUse this Zapをクリック

Connect Adalo to Gmail with Zapier

[LEARN MORE](#)

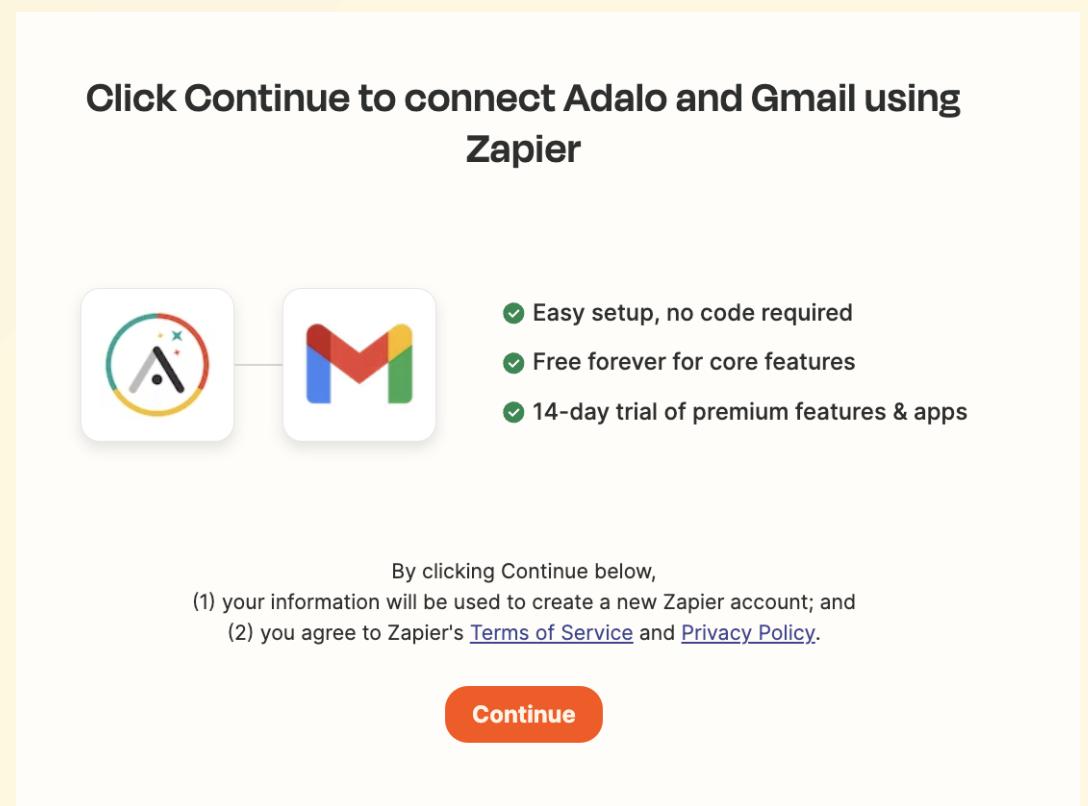
The screenshot shows three search results for "Adalo + Gmail" on Zapier:

- Send Gmail emails for new Adalo records**
Adalo + Gmail [Use this Zap](#)
- Send emails in Gmail for updated records in Adalo**
Adalo + Gmail [Use this Zap](#)
- Add labels to emails in Gmail when new records are created in Adalo**
Adalo + Gmail [Use this Zap](#)

Below the results is a link: "Not seeing what you're looking for? [Create from scratch](#) or [learn more](#)".

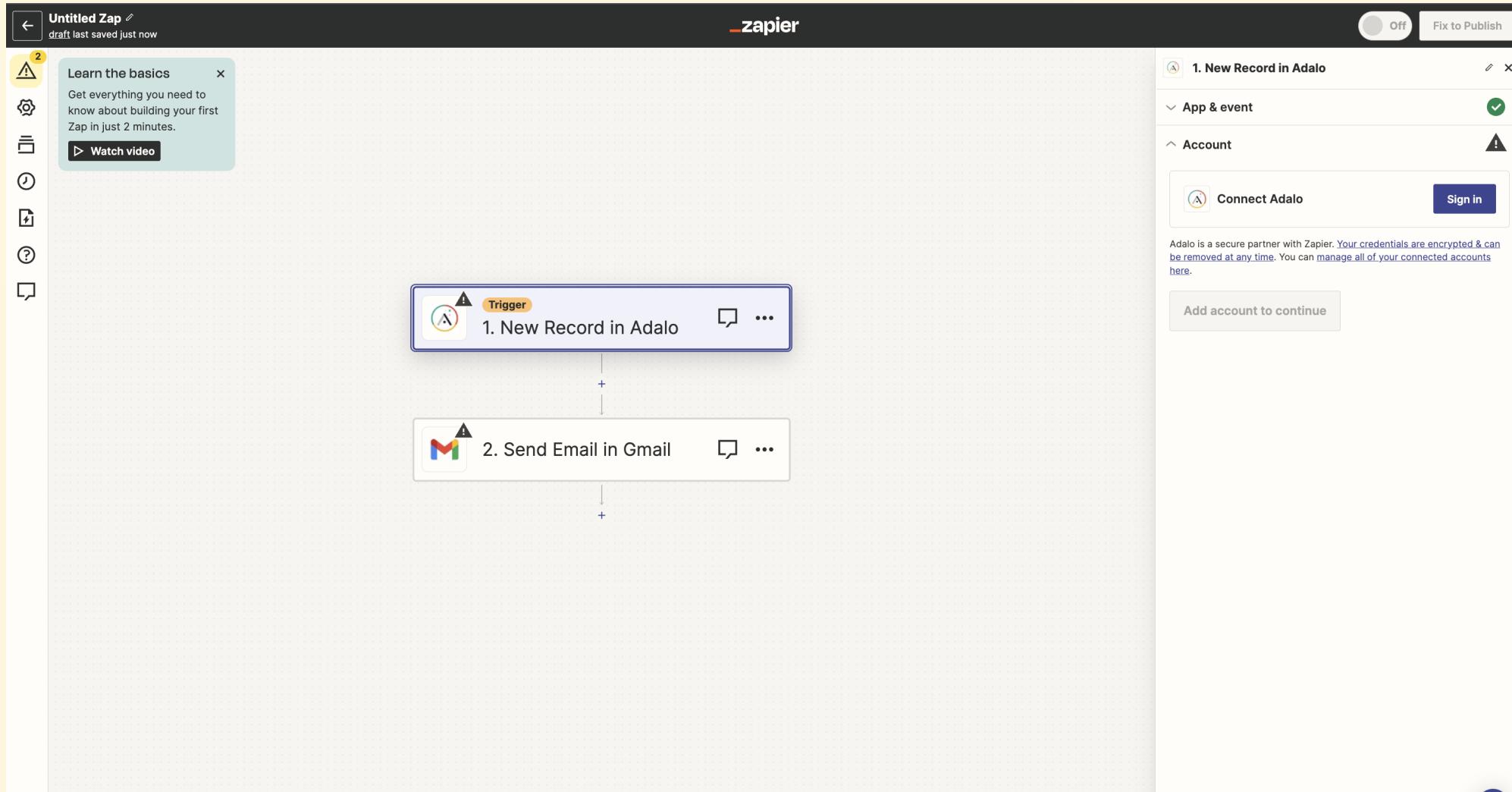
BACK

- Continueをクリック

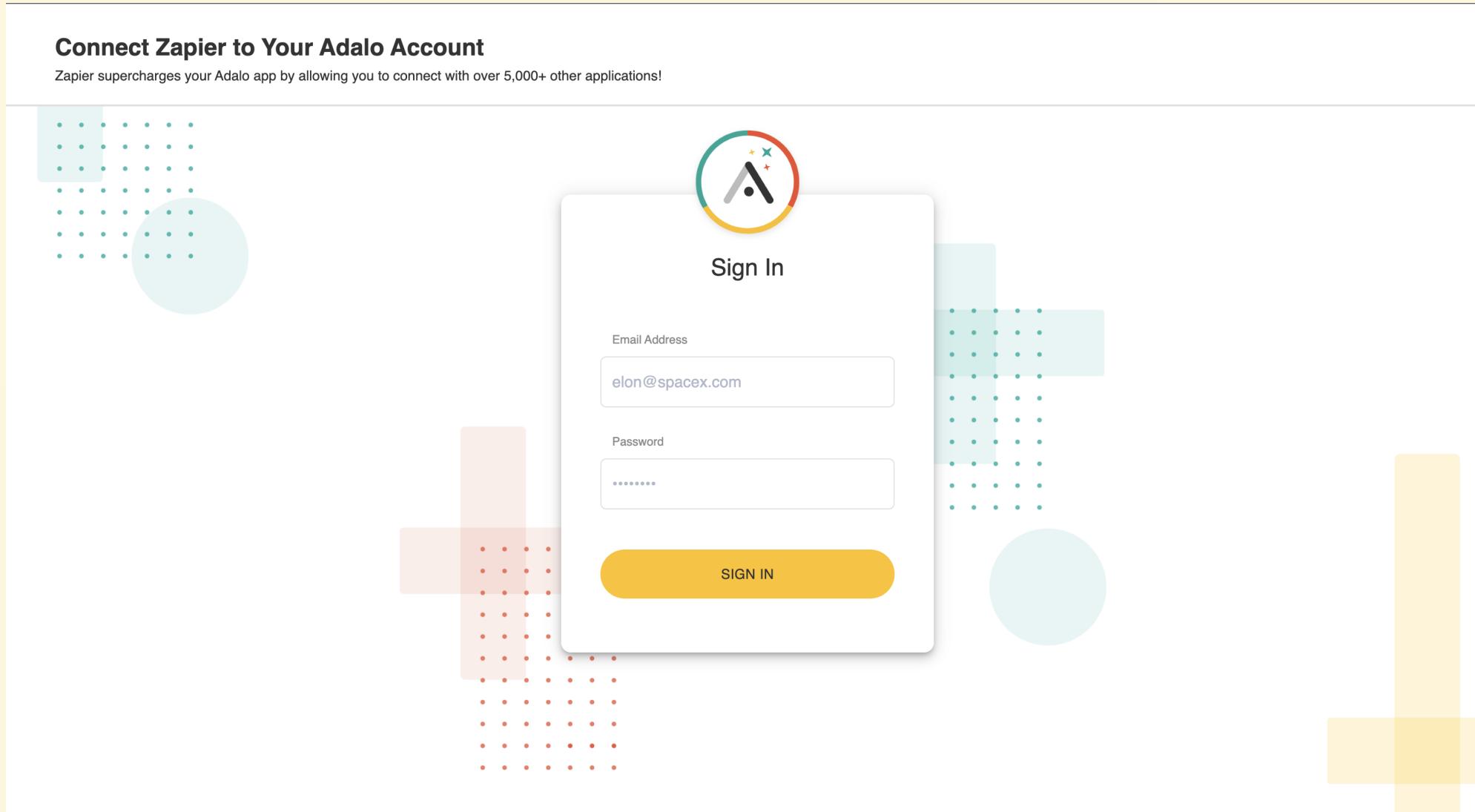


Zapierの設定画面が開きます

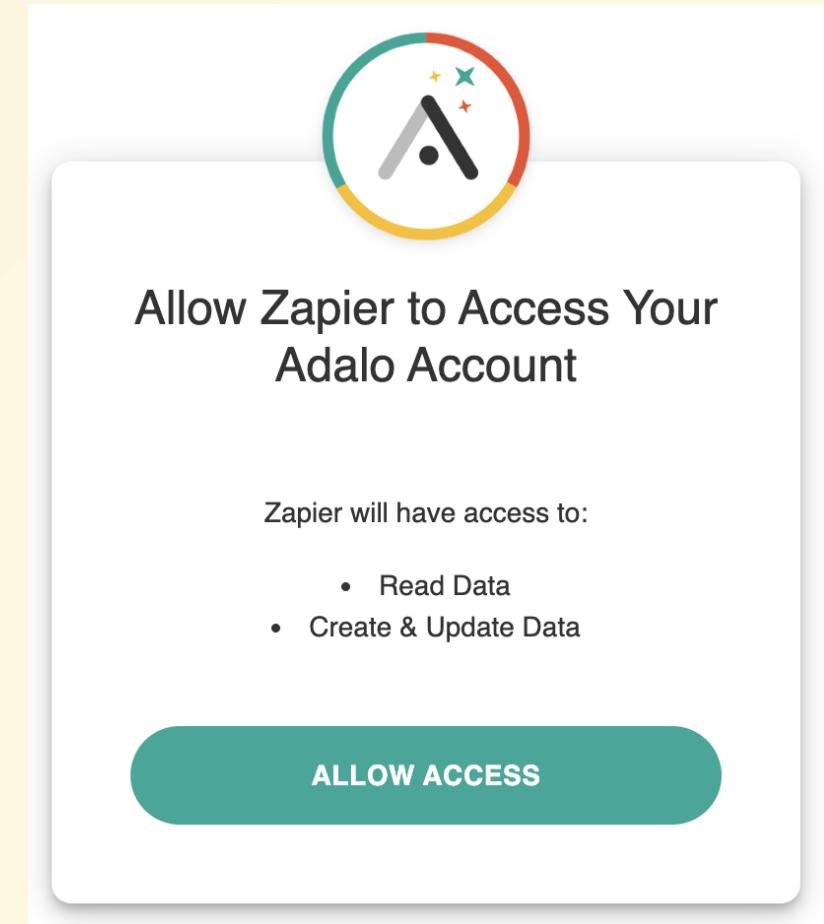
- 右側のConnect AdaloのSign Inをクリック



- Adaloにログイン

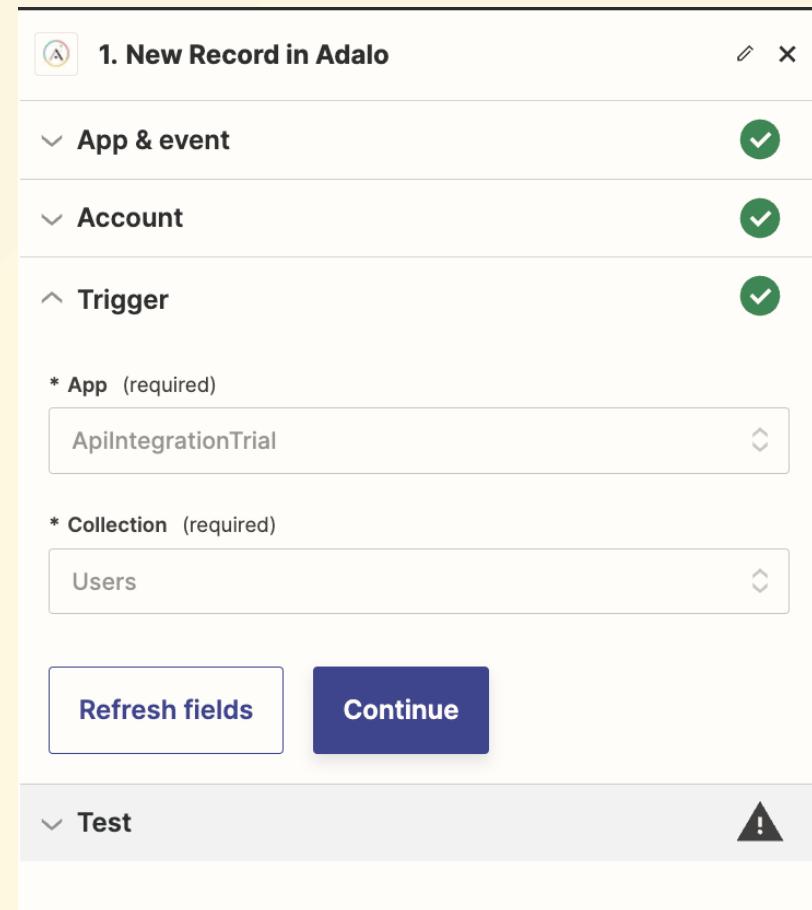


- ALLOW ACCESSをクリック

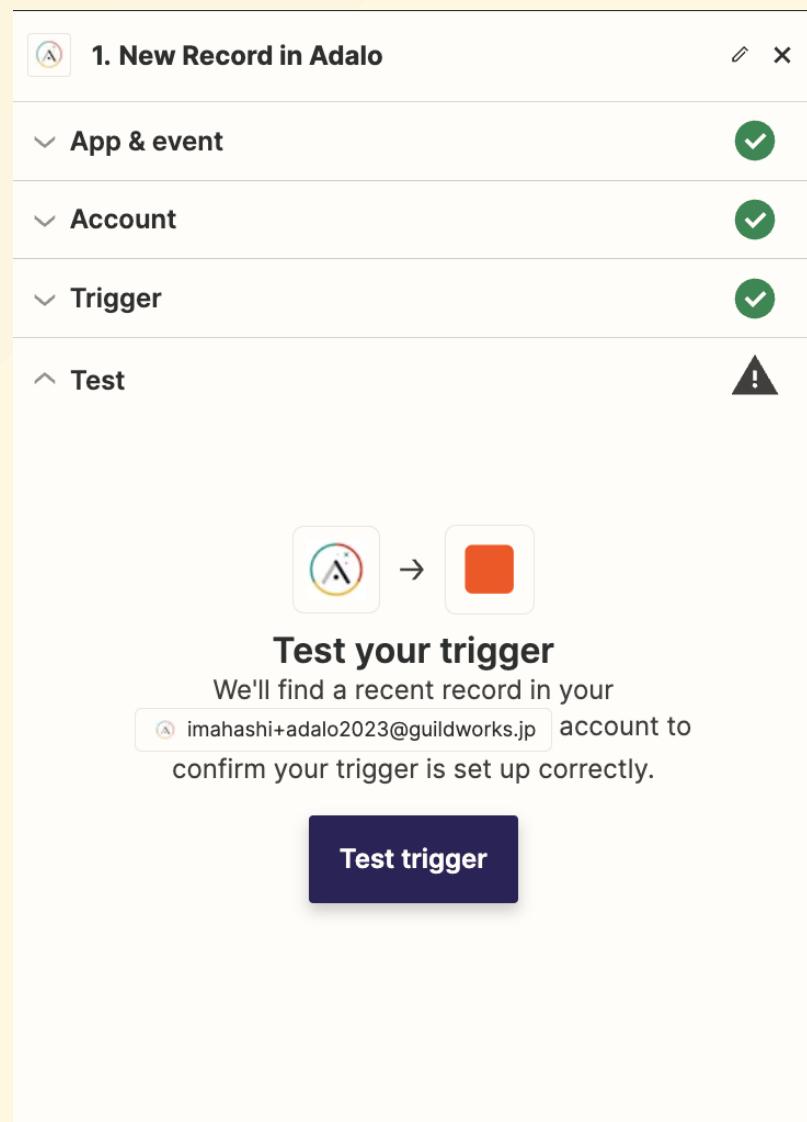


- Triggerを設定

- App: ApilntegrationTrial
- Collection: Users



- Test triggerをクリック



- レコードが見つかったら、
Continue with selected
recordをクリック

1. New Record in Adalo

✓ App & event ✓

✓ Account ✓

✓ Trigger ✓

^ Test ✓

We found records in your Adalo account. We will load up to 3 most recent records, that have not appeared previously. [Learn more about test records.](#)

Search

Record A	id 1 created_at 2023-11-02T13:18:47.193Z updated_at 2023-11-02T13:18:47.193Z Email imahashi@guildworks.jp Username Full Name 今橋 陵
----------	--

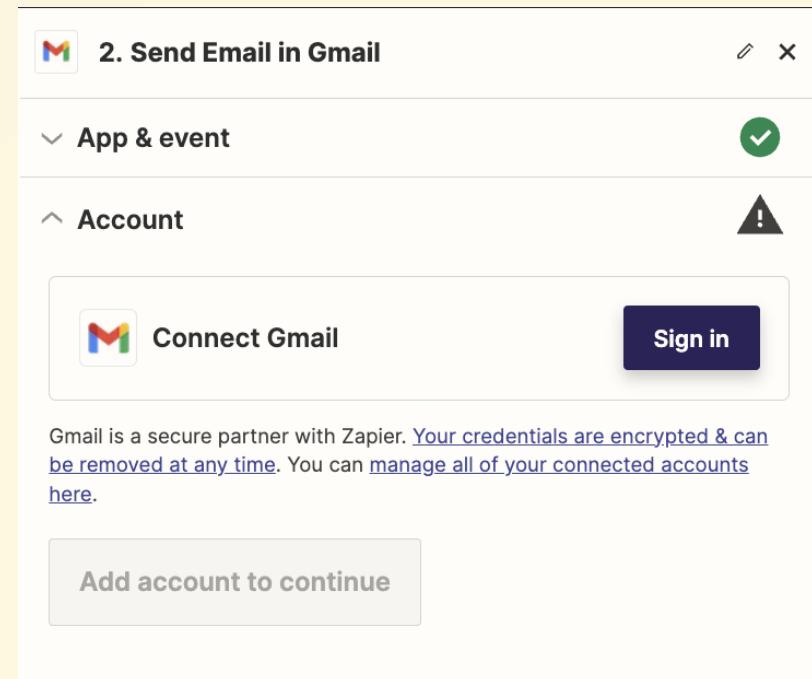
Find new records

4

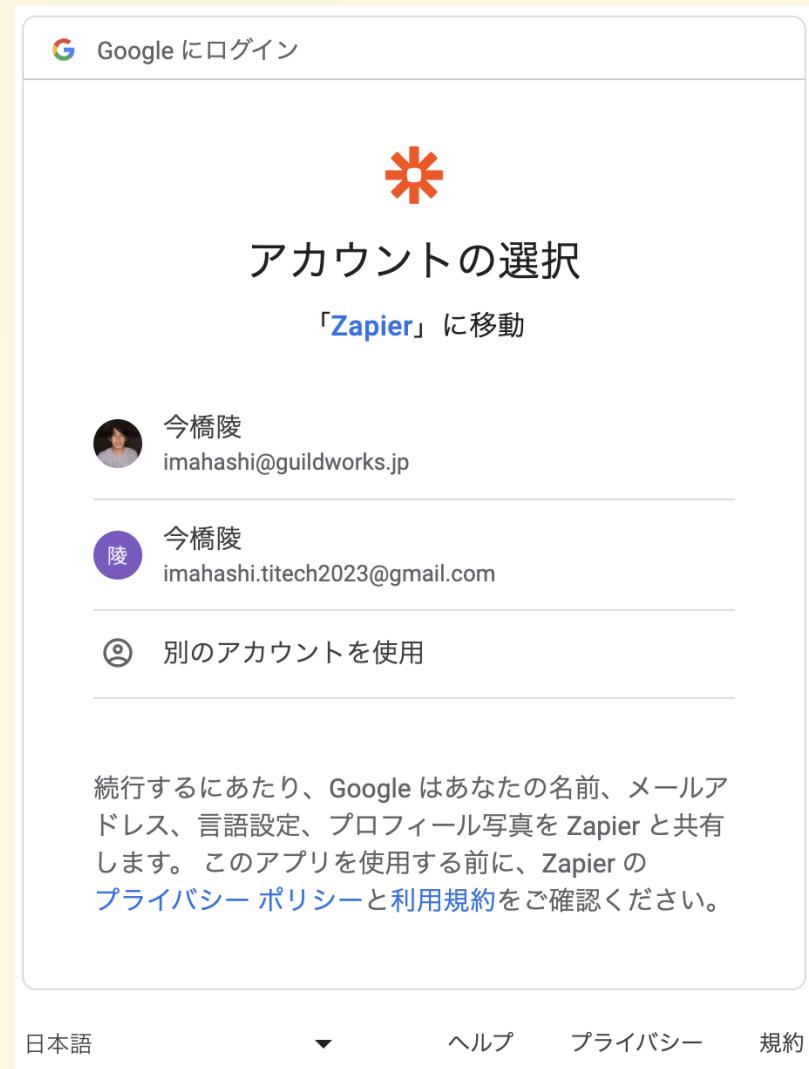
Continue with selected record

The screenshot shows the Adalo interface with a success message: "We found records in your Adalo account. We will load up to 3 most recent records, that have not appeared previously. Learn more about test records." Below this, there is a search bar and a table displaying a single record named "Record A". The record was pulled on Nov 02, 2023. The table includes columns for id, created_at, updated_at, Email, Username, and Full Name. At the bottom left is a button labeled "Find new records", and at the bottom right is a small icon with the number "4". A large blue button at the bottom center says "Continue with selected record".

- Connect GmailのSign inをクリック



- アカウントを選択



- 許可をクリック



Actionを設定します。

- Fromに自分のGmailアドレスを設定
- From Nameにアプリ名を設定
- Subject, Bodyを自由に入力
- Continueをクリック

2. Send Email in Gmail

^ Action

To

1. Email: imahashi@guildworks.jp

Enter text or insert data...

Who will this email be sent to? Multiple email addresses can be entered either individually, or as a comma separated list, eg.... more

Cc

Enter text or insert data...

Who should be cc'd on this email? Multiple email addresses can be entered either individually, or as a comma separated list, eg.... more

Bcc

Enter text or insert data...

Who should be bcc'd on this email? Multiple email addresses can be entered either individually, or as a comma separated list, eg.... more

From

imahashi.titech2023@gmail.com

Select an email address or alias from your Gmail Account. Defaults to the primary email address. more

From Name

ApilIntegrationTrial

Reply To

Enter text or insert data...

Specify a single reply address other than your own.

* Subject (required)

Welcome to ApilIntegrationTrial App

Body Type

plain

If using the HTML option, you must add any and all formatting (paragraphs, lists, etc) directly using HTML. [Learn more.](#) more

* Body (required)

Hello, 1. Full Name: 今橋 陵
Thank you for signing up my app.
I hope you enjoy it.

2

- to: のアドレスが実際の自分のメールアドレスになっていれば、"Test step"をクリックしてメール受信を確認する
 - 実際の自分のメールアドレスになっていなければ、"Skip test"をクリック(testをするとメール送信が失敗するため)

2. Send Email in Gmail

✓ App & event
✓ Account
✓ Action
⚠ Test

→ **Send Send Email to Gmail**
To test Gmail, we need to create a new send email. This is what will be created:

Search item data...

```

body Hello, 今橋 陵
      Thank you for signing up my app.
      I hope you enjoy it.
body_type plain
from imahashi.titech2023@gmail.com
from_name ApilIntegrationTrial
subject Welcome to ApilIntegrationTrial App
to
1 imahashi@guildworks.jp

```

Empty fields:

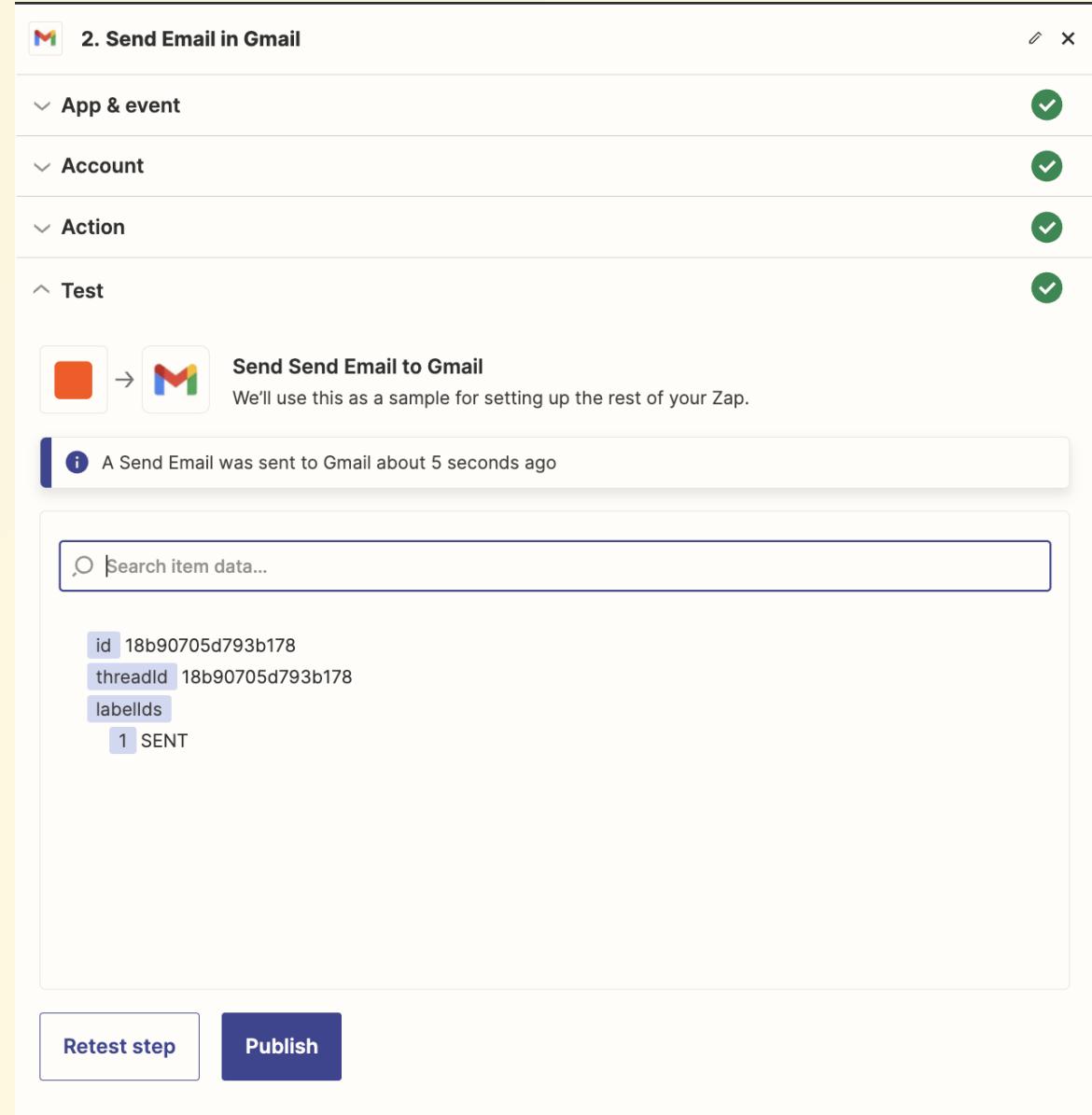
- Cc empty (optional)
- Bcc empty (optional)
- Reply To empty (optional)
- Signature empty (optional)
- Label/Mailbox empty (optional)
- Attachments empty (optional)

[Skip test](#) [Test step](#)

3

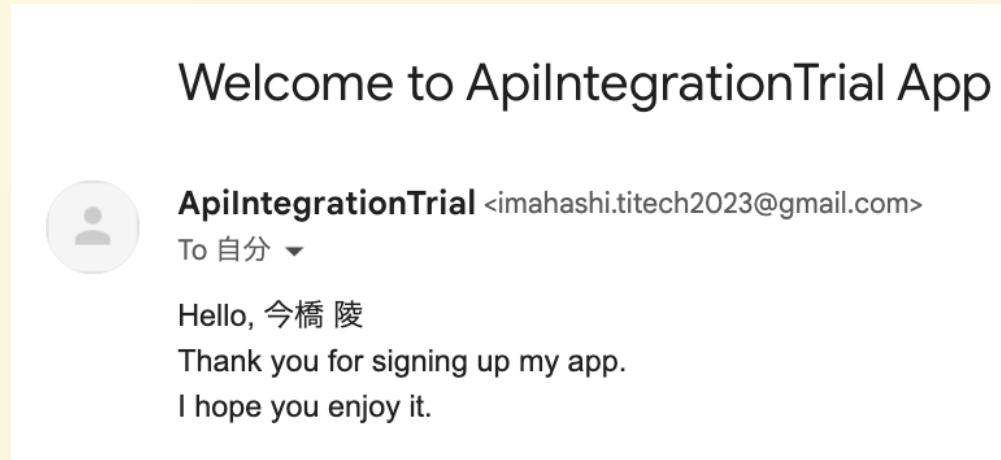
- Publishをクリック

これで、Zapierとの連携設定
ができました。



AdaloのApilntegrationTrialアプリでWelcomeメールが送信されることを確認しましょう。

- Preview機能で、実際の自分のメールアドレスを入力してSignupすると、2分以内にメールが届きます



参考: 2分待たなくても、Zapの一覧画面から、手動ですぐにZapを実行することもできます。

- <https://zapier.com/app/zaps> にアクセス
- 実行したいZapを選んで"Run Zap"をクリック

The screenshot shows the Zapier Home page. On the left, there's a sidebar with icons for creating a new Zap, managing teams, and more. The main area displays two "Untitled Zap" entries. On the right, a detailed view of a specific Zap is shown, which is currently active ("Zap is on"). The Zap details include:

- Owned by: Ryo I.
- in Home
- Published: November 02, 2023 at 11:30 pm
- View in Editor
- Buttons for Duplicate, Run Zap, History, and Share template.
- Apps section listing Adalo and Gmail integrations.
- A red "Turn off and delete Zap" button at the bottom.

Zapierを使えば、他にも様々なサービスを連携させることができます。

Adaloだけでうまく実現できないことが出てきた時には、他のサービスと組み合わせることで実現できないかを考えてみると良いかもしれません。

Most Popular Integrations using Zapier [LEARN MORE](#)

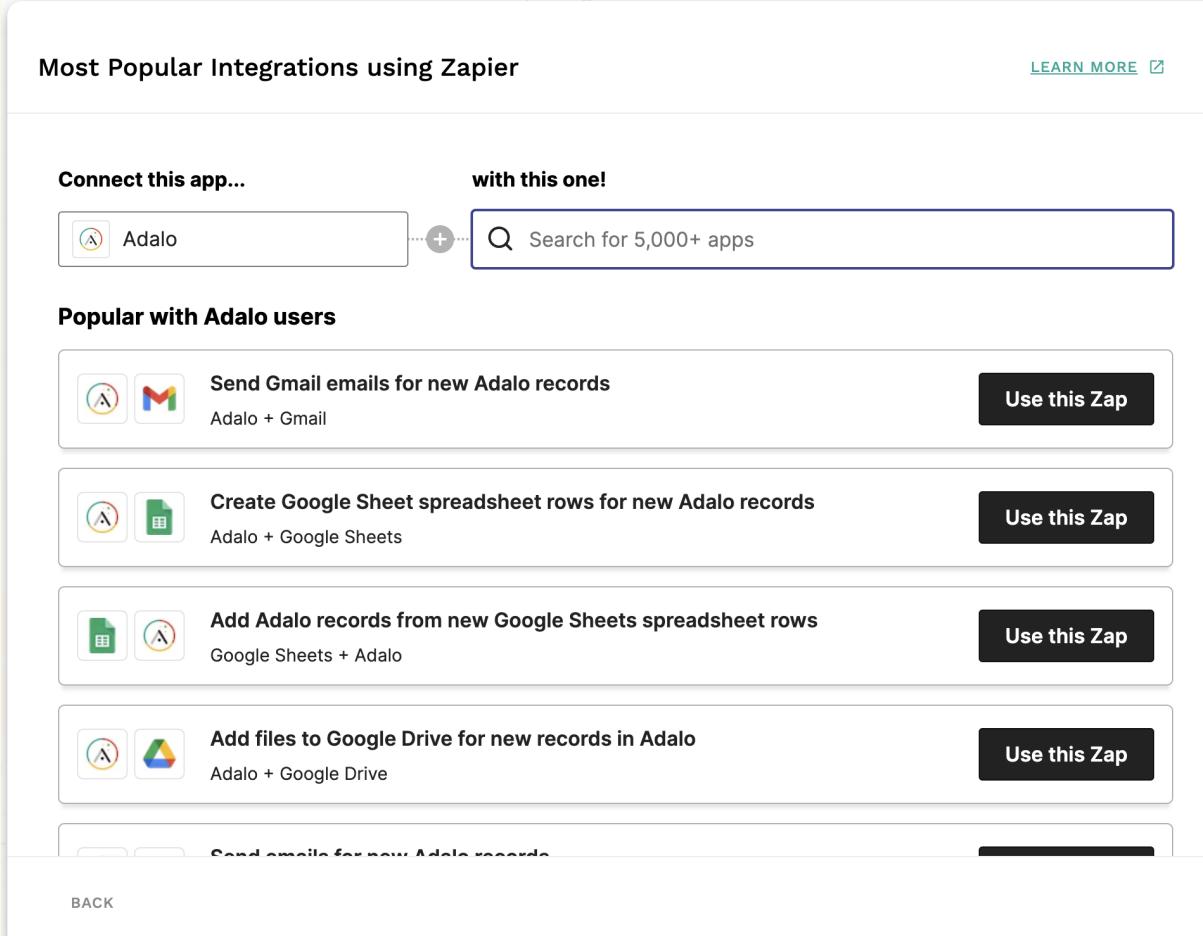
Connect this app... with this one!

Adalo + Search for 5,000+ apps

Popular with Adalo users

-   Send Gmail emails for new Adalo records
Adalo + Gmail [Use this Zap](#)
-   Create Google Sheet spreadsheet rows for new Adalo records
Adalo + Google Sheets [Use this Zap](#)
-   Add Adalo records from new Google Sheets spreadsheet rows
Google Sheets + Adalo [Use this Zap](#)
-   Add files to Google Drive for new records in Adalo
Adalo + Google Drive [Use this Zap](#)
-   Send emails for new Adalo records

BACK



Adaloでの外部連携についての紹介は以上です。

演習2

- アプリを1つ自由に開発してください。
- 作りたいアプリが思い浮かばない人は、以下の機能を持つチームメンバー管理アプリを開発しても良いです。
 - チームメンバーの登録
 - メンバー一覧の表示
 - メンバー詳細の表示
 - メンバー情報の更新
 - メンバー削除
 - 自分で考えたオリジナル機能(いくつあっても良い)

- ※ (時間が許せば)最後は全員に発表していただきたいと考えています。
- ※ アプリが使えるようになつたら、SlackでURLを共有して、みなさん見てもらいましょう。

演習における注意事項

- NoCodeツールでは、簡単にアプリを作れる反面、複雑なUIや機能を実現できない場合があります
 - 行き詰った時は、どうすれば自分がやりたいことをシンプルなUI、機能で実現できるか考えてみてください
 - 例: 1画面に多くのコンポーネントを含めず、画面を分ける 等

参考

- クローンできるAdaloのアプリが公開されているので、やりたいことに近いものがないか探してみると良いかもしれません。
 - App Templates
<https://www.adalo.com/app-templates>
 - UI & Functional Kits
<https://www.adalo.com/cloneable-kits>

クローンできるアプリの例

- Eventカレンダー <https://www.adalo.com/cloneables/event-calendar>
- SNSのフォロー機能 <https://www.adalo.com/cloneables/follow-function>
- Facebookのクローン <https://www.adalo.com/cloneables/facebook-clone>
- ブログアプリ <https://www.adalo.com/cloneables/minimal-blog-app>
- 商品販売アプリ <https://www.adalo.com/cloneables/ecommerce-app>



演習結果の発表

演習で作ったアプリについて発表してください。

まとめ

- Adaloについて紹介し、データベースの設計方法を学んだ上で、ペットの健康管理アプリを題材にアプリを開発しました。
- 外部サービスとの連携方法として以下4つを紹介しました。
 - Marketplaceの外部連携コンポーネント
 - Custom Action
 - External Collection
 - 連携サービス

- ここまで内容を踏まえて、Development Phaseで自分たちが作りたいアプリがAdaloで実現できそうかは、チームで考えてみると良いと思います。
- 次回はノーコードツールBubbleのレクチャーです。お楽しみに！

以上です！

お疲れさまでした！