

Programming Boot Camp

# Adaloでのデータベースの設計と操作、外部連携

東京工業大学 2021/11/6

Ryo Imahashi

# 目次

- 前回のふりかえりと今回のゴールの確認
- データベースについて学ぼう
- データベース設計
- データベース操作
- サンプルアプリを改善しよう
- 外部連携
- Adaloでのチーム開発のやり方
- 演習
- まとめ

# 前回のふりかえりと今回のゴールの確認

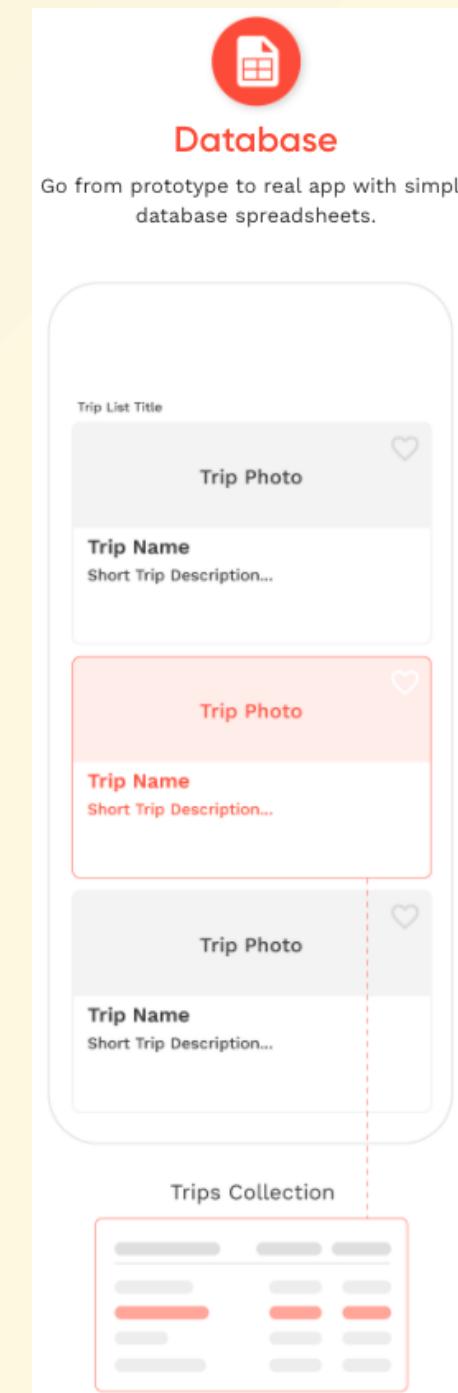
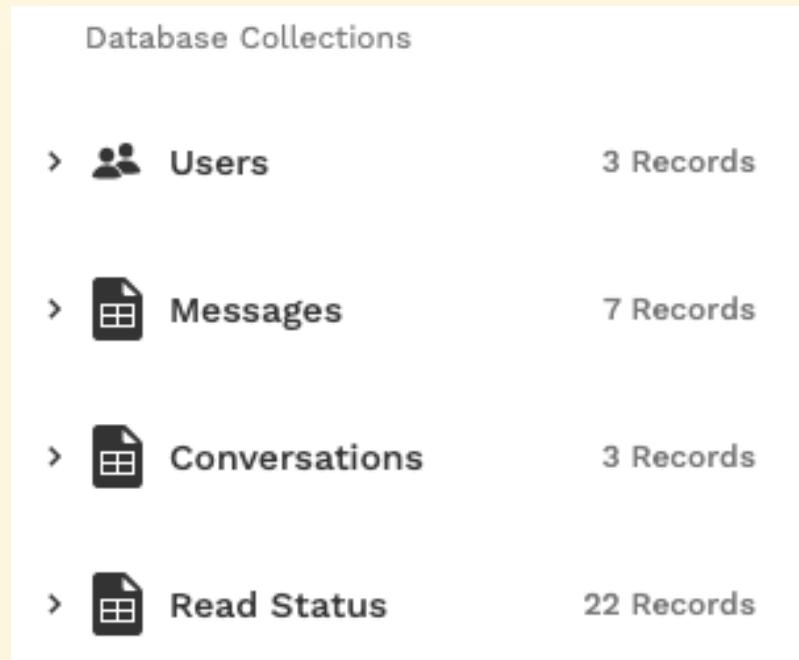
- 前回のレクチャーでは、ノーコードツールのAdaloについて紹介し、ペットの健康管理アプリを題材にアプリのUIを作成しました。
  - レクチャーでは、データベースを必要としない、シンプルなコンポーネントを使いました。
- 今回のレクチャーでは、引き続きAdaloを使って、前回作ったUIに合わせたデータベースを設計し、アプリからデータを操作できるようにしていきましょう。
- また、サンプルアプリの改善をしながらAdaloの機能をいくつか紹介したり、外部連携の方法や、チーム開発のやり方についてもお伝えします。最後には演習と成果発表も行います。

# データベースについて学ぼう

まず、これから扱うデータベースがどのようなものかを確認します。

# Database(前回の復習)

- 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						<a href="#">+ ADD USER</a>				
	<a href="#">Email</a>	Password	<a href="#">Username</a>	<a href="#">Full Name</a>						
<input type="checkbox"/>	fuga@hoge.com	[hidden]		三人目のユーザー						
<input type="checkbox"/>	hoge@fuga.com	[hidden]		東 工大						
<input type="checkbox"/>	imahashi@example.com	[hidden]		今橋 謙						

# 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 and a main table view on the right.

**Left Sidebar:**

- + Database Collections
- Database Collections
- Users (selected, 3 Records)
- Email
- Password
- Username
- Full Name
- Profile Picture
- Conversations (Creator's)
- Messages (Sender's)
- Read Statuses
- Conversations (Members)
- Friends
- + ADD PROPERTY
- Messages (7 Records)
- Conversations (3 Records)

**Main View:**

### Users

	A Email	>Password	A Username	A Full Name	Profile Picture
<input type="checkbox"/>	fuga@hoge.com	[hidden]		三人目のユーザー	
<input type="checkbox"/>	hoge@fuga.com	[hidden]		東 工大	
<input type="checkbox"/>	imahashi@example.com	[hidden]		今橋 陵	

**Top Right Buttons:**

- + ADD USER
- Upload icon
- Cloud icon
- Sync icon
- Search icon

**Bottom Right Buttons:**

- DONE

- 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					
	<a href="#">Email</a>	<a href="#">Password</a>	<a href="#">Username</a>	<a href="#">Full Name</a>	<a href="#">Profile Picture</a>
<input type="checkbox"/>	fuga@hoge.com	[hidden]		三人目のユーザー	
<input type="checkbox"/>	hoge@fuga.com	[hidden]		東 工大	
<input type="checkbox"/>	imahashi@example.com	[hidden]		今橋 膜	

[+ ADD USER](#)    

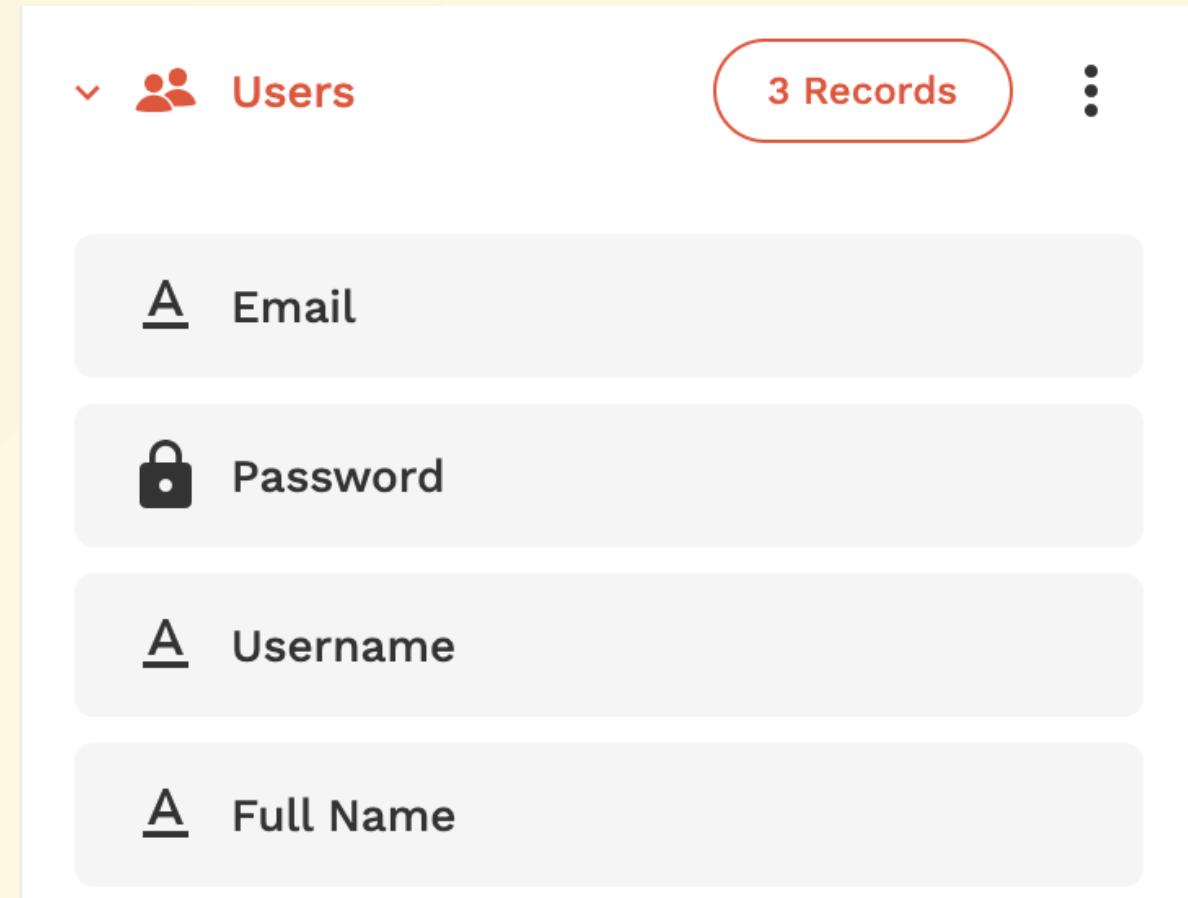
[DONE](#)

- 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 screenshot shows a mobile application's user registration screen titled 'New User'. The form consists of five input fields: 'Email' (with an '@' icon), 'Password' (with a lock icon), 'Username' (with an '@' icon), 'Full Name' (with a person icon), and 'Profile Picture' (with a camera icon). Below the profile picture field is a placeholder text 'Choose image'. At the bottom right are two buttons: 'CANCEL' and 'SAVE'.

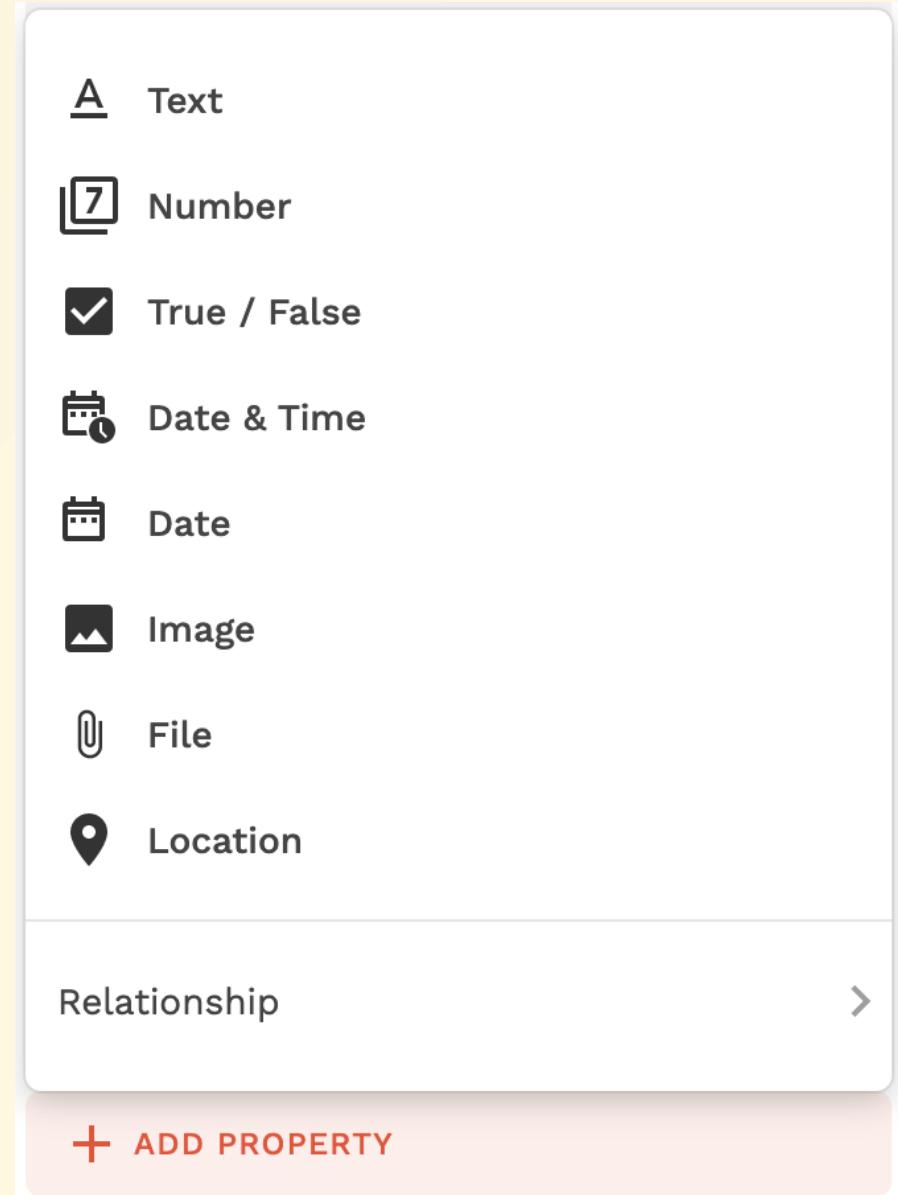
## Propertyとは

- 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

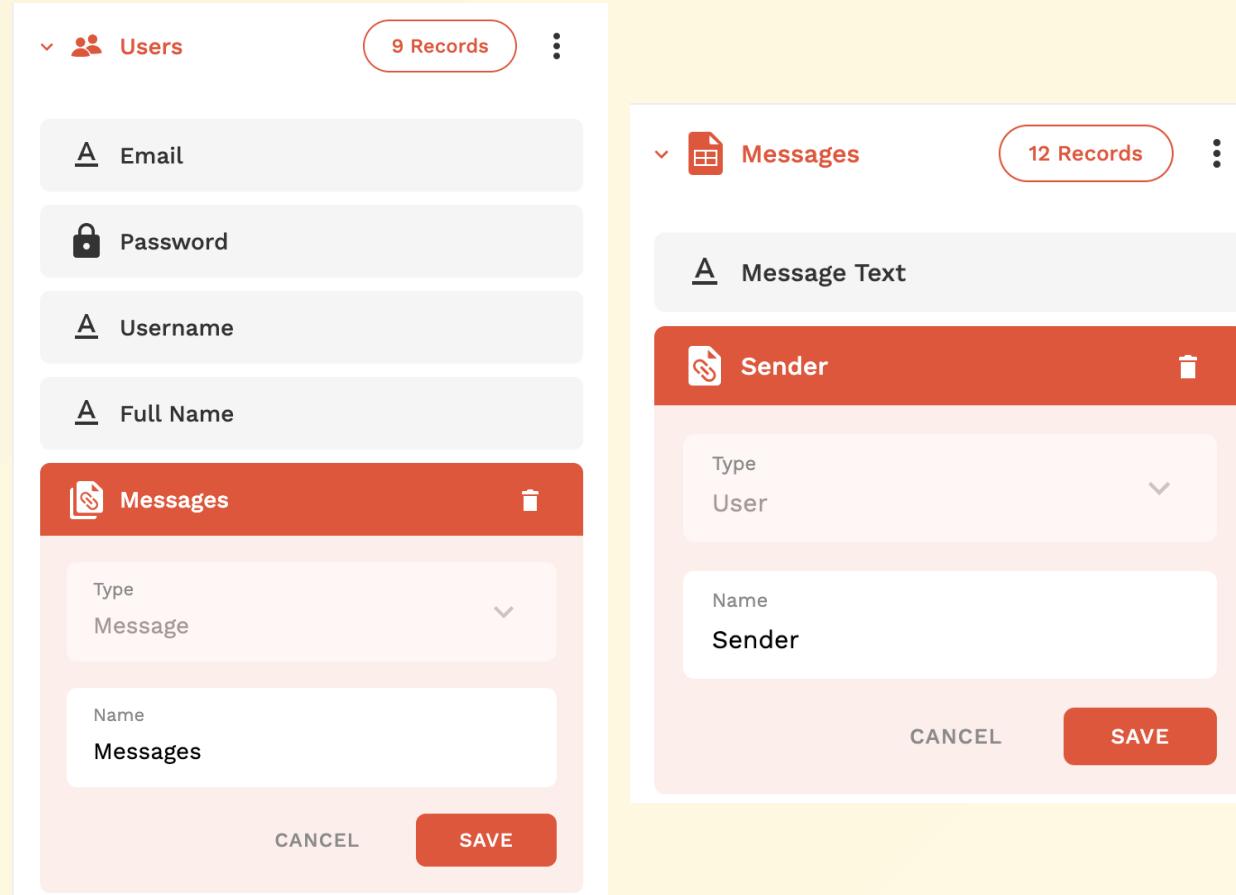


## Relationshipとは

- 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。

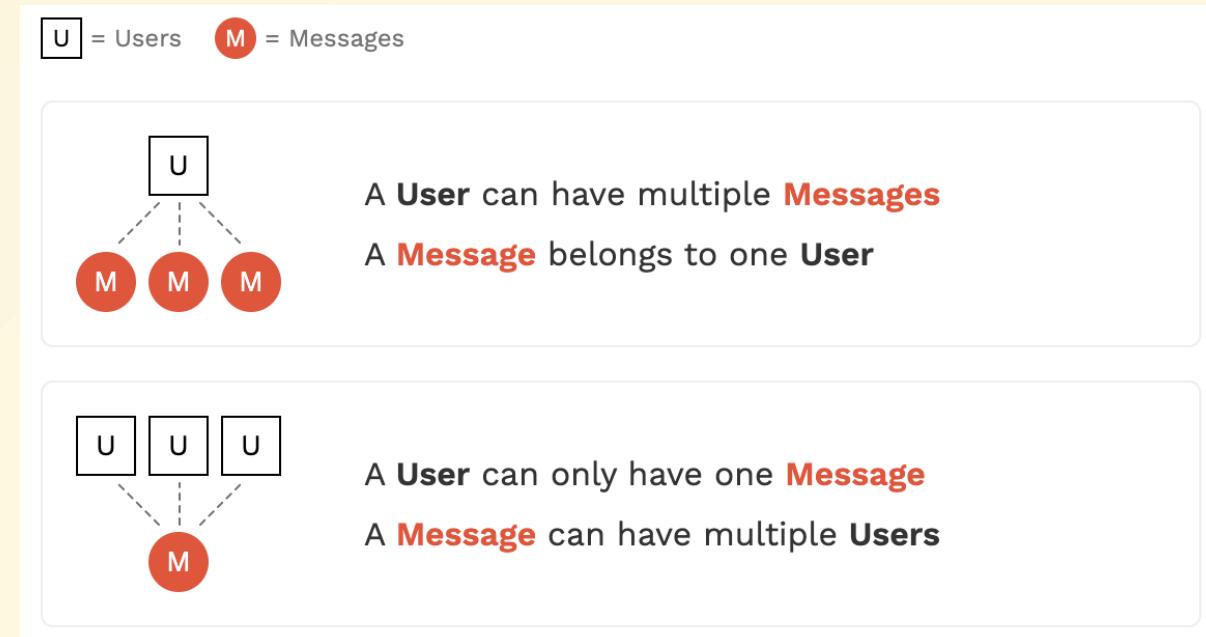


## Relationshipの種類

- 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。

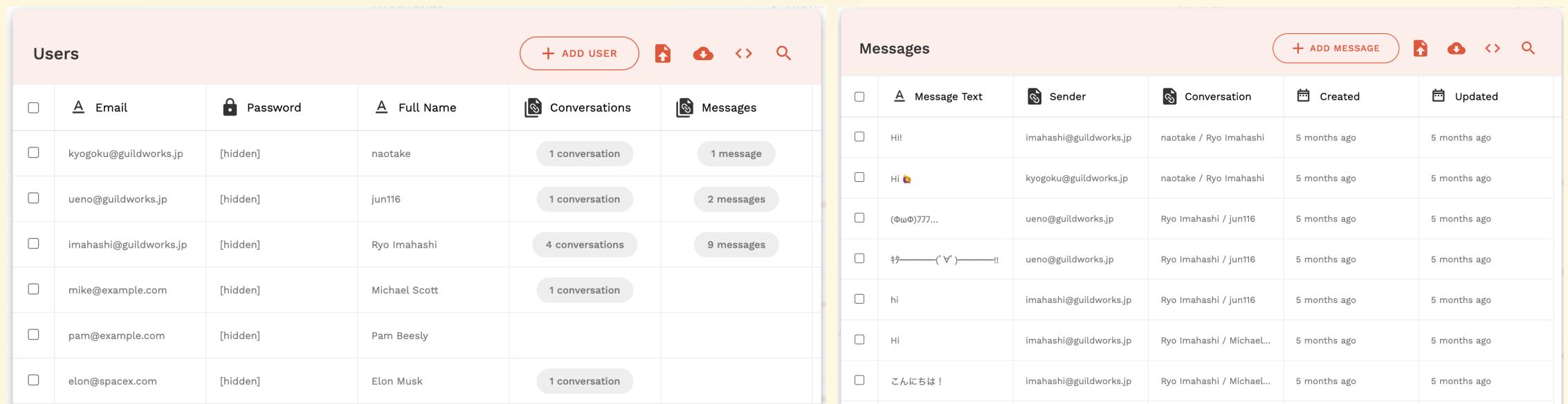
# 1対多のRelationship

- 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。



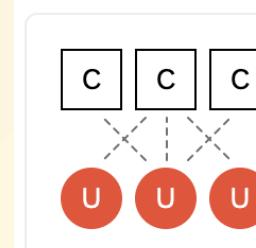
The screenshot shows a MongoDB interface with two collections displayed side-by-side:

Users					
	<input type="checkbox"/>				
	Email	Password	Full Name	Conversations	Messages
1	kyogoku@guildworks.jp	[hidden]	naotake	1 conversation	1 message
2	ueno@guildworks.jp	[hidden]	jun116	1 conversation	2 messages
3	imahashi@guildworks.jp	[hidden]	Ryo Imahashi	4 conversations	9 messages
4	mike@example.com	[hidden]	Michael Scott	1 conversation	
5	pam@example.com	[hidden]	Pam Beesly		
6	elon@spacex.com	[hidden]	Elon Musk	1 conversation	

Messages					
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	Message Text	Sender	Conversation	Created	Updated
1	Hi!	imahashi@guildworks.jp	naotake / Ryo Imahashi	5 months ago	5 months ago
2	Hi 🌻	kyogoku@guildworks.jp	naotake / Ryo Imahashi	5 months ago	5 months ago
3	(ΦωΦ)777...	ueno@guildworks.jp	Ryo Imahashi / jun116	5 months ago	5 months ago
4	タ———(` ∀` )——!!	ueno@guildworks.jp	Ryo Imahashi / jun116	5 months ago	5 months ago
5	hi	imahashi@guildworks.jp	Ryo Imahashi / jun116	5 months ago	5 months ago
6	Hi	imahashi@guildworks.jp	Ryo Imahashi / Michael...	5 months ago	5 months ago
7	こんにちは！	imahashi@guildworks.jp	Ryo Imahashi / Michael...	5 months ago	5 months ago

## 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。



A **User** can have multiple **Conversations**  
A **Conversation** can have multiple **Users**

# 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	Full Name	Conversations	Messages
<input type="checkbox"/>	kyogoku@guildworks.jp	[hidden]	naotake	<span>1 conversation</span>	<span>1 message</span>
<input type="checkbox"/>	ueno@guildworks.jp	[hidden]	jun116	<span>1 conversation</span>	<span>2 messages</span>
<input type="checkbox"/>	imahashi@guildworks.jp	[hidden]	Ryo Imahashi	<span>4 conversations</span>	<span>9 messages</span>
<input type="checkbox"/>	mike@example.com	[hidden]	Michael Scott	<span>1 conversation</span>	
<input type="checkbox"/>	pam@example.com	[hidden]	Pam Beesly		
<input type="checkbox"/>	elon@spacex.com	[hidden]	Elon Musk	<span>1 conversation</span>	

Conversations					
	Name	Members	Messages	Read Statuses	Creator
<input type="checkbox"/>	naotake / Ryo Imahashi	<span>2 users</span>	<span>2 messages</span>	<span>6 read statuses</span>	kyogoku@guildworks.jp
<input type="checkbox"/>	Ryo Imahashi / jun116	<span>2 users</span>	<span>3 messages</span>	<span>9 read statuses</span>	imahashi@guildworks.jp
<input type="checkbox"/>	Ryo Imahashi / Elon M...	<span>2 users</span>	<span>2 messages</span>	<span>5 read statuses</span>	imahashi@guildworks.jp
<input type="checkbox"/>	Ryo Imahashi / Michael...	<span>2 users</span>	<span>5 messages</span>	<span>15 read statuses</span>	imahashi@guildworks.jp

# データベース設計

前回のレクチャーで作成したサンプルアプリのUIを見ながら、保存が必要なデータを考えて、データベースを設計しましょう。

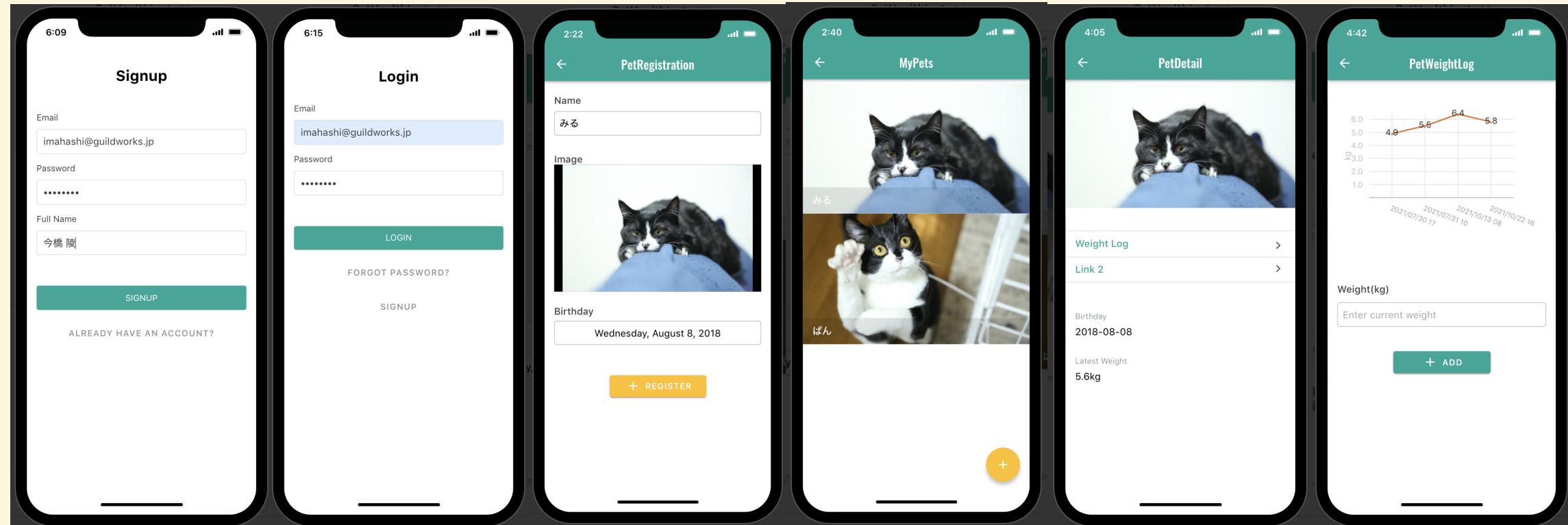
## 前回のレクチャーで作成したアプリのクローン用URL

- 以下のURLからアプリをクローンしてください。それを使ってここからのレクチャーを進めます。

<https://previewer.adalo.com/014fd9d1-80c6-4325-899a-d943e778c865>

# 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。

\* 絶対的な正解はありません。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。

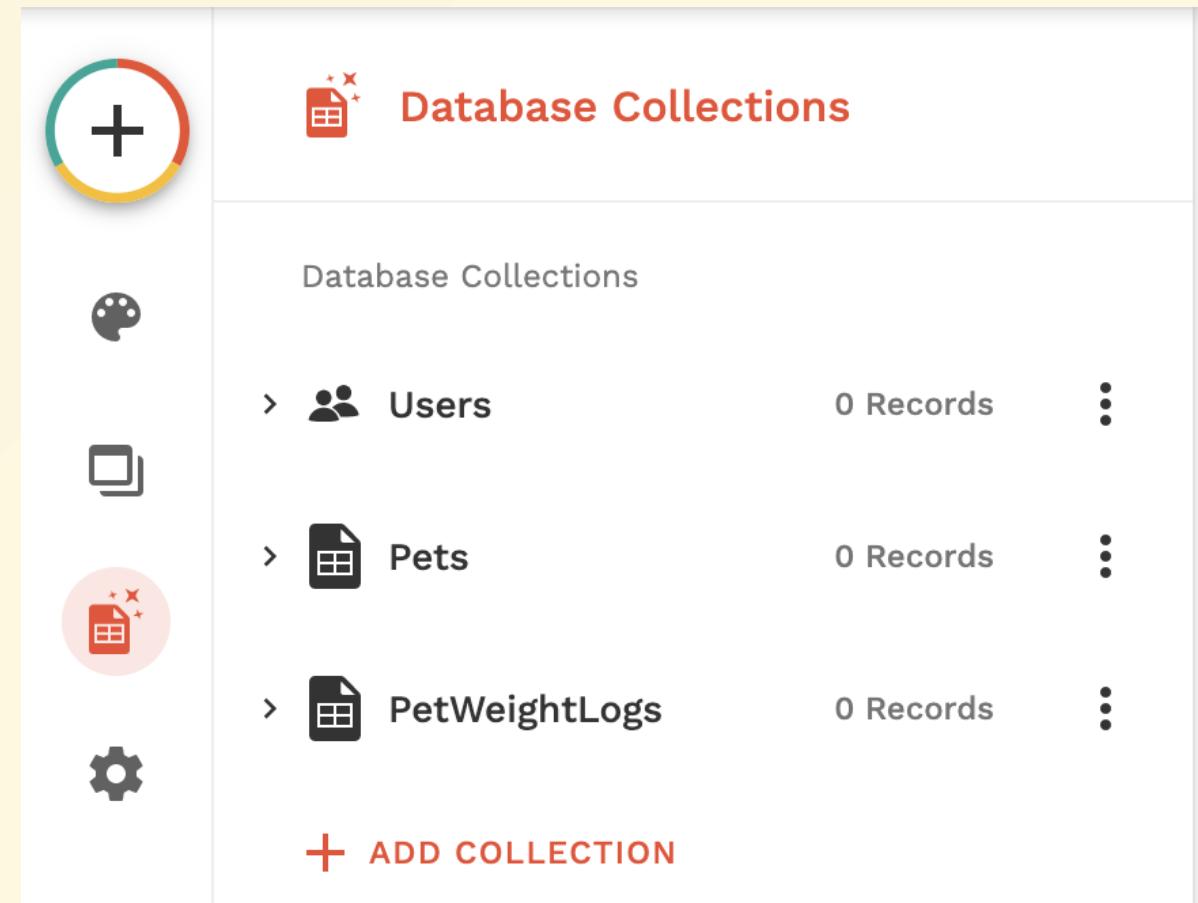
- Users
- Pets
- PetWeightLogs

- 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。
- Usersはデフォルトで作成されています。

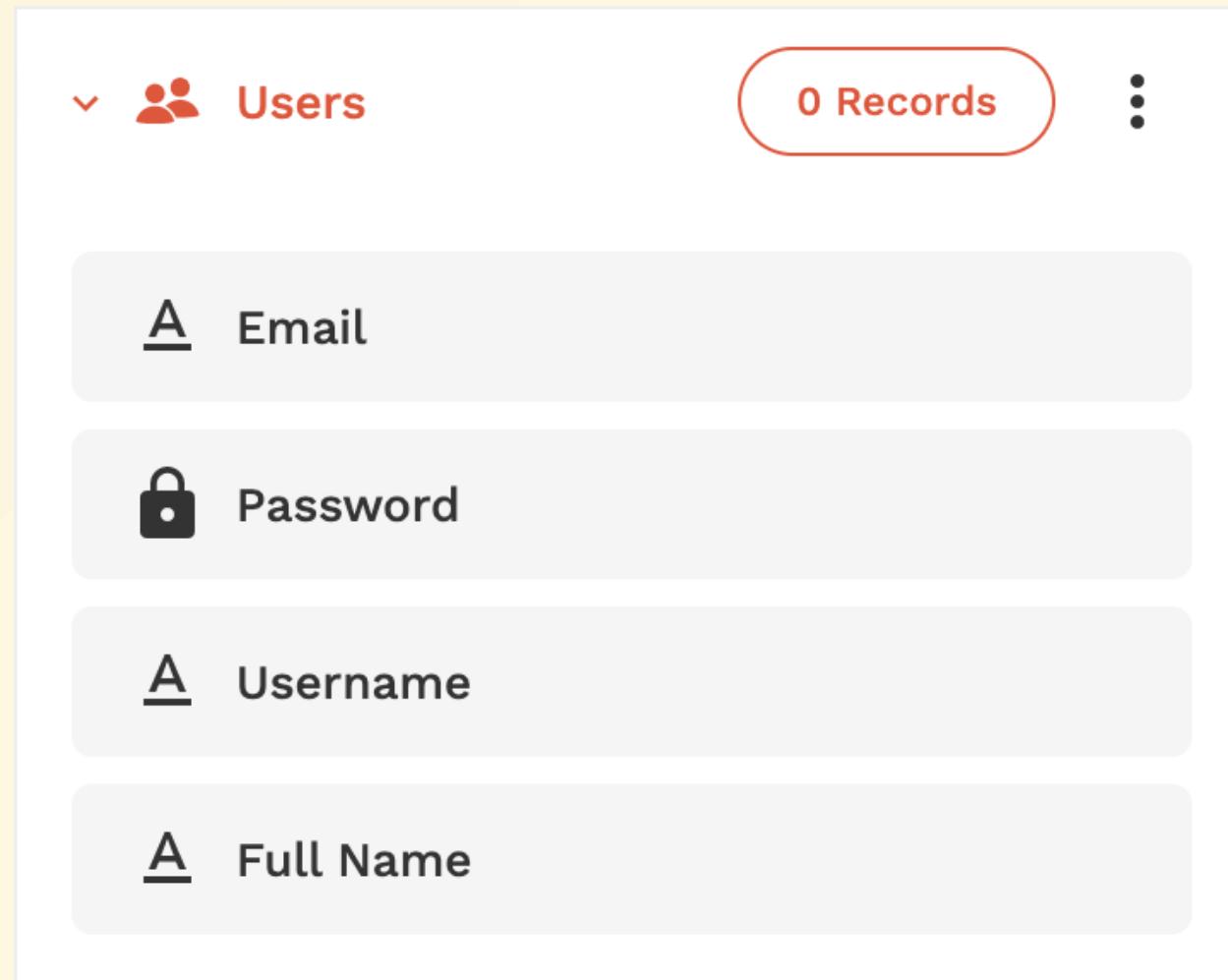


次に、1でリストアップしたデータを適切なCollectionの配下にPropertyとして追記すると、以下のようにになりました。()の中は選択するTypeです。

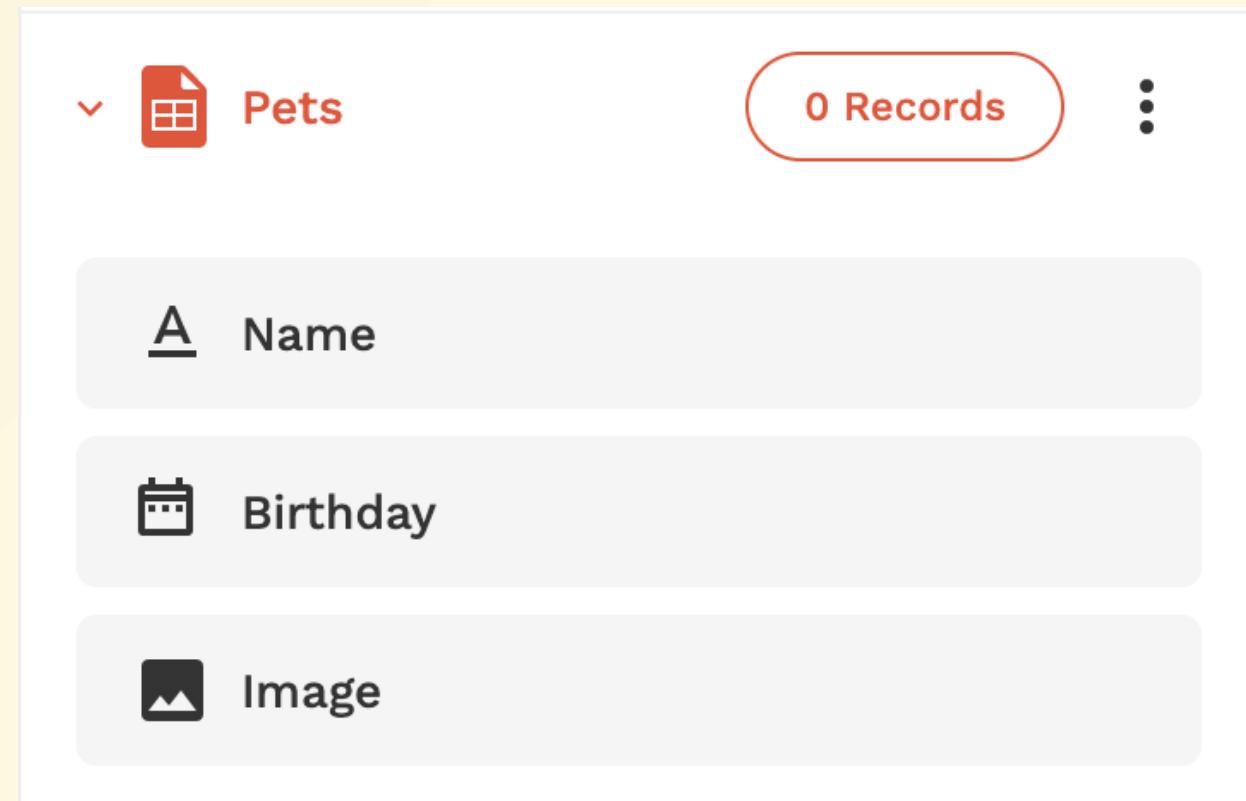
- Users
  - Email(Text)
  - Password(※Password)
  - FullName(Text)
- Pets
  - Name(Text)
  - Image(Image)
  - Birthday(Date)
- PetWeightLogs
  - WeightKg(Number)
  - RegisteredTime(Date&Time)

\* Passwordはデフォルトで設定される特殊なTypeです。

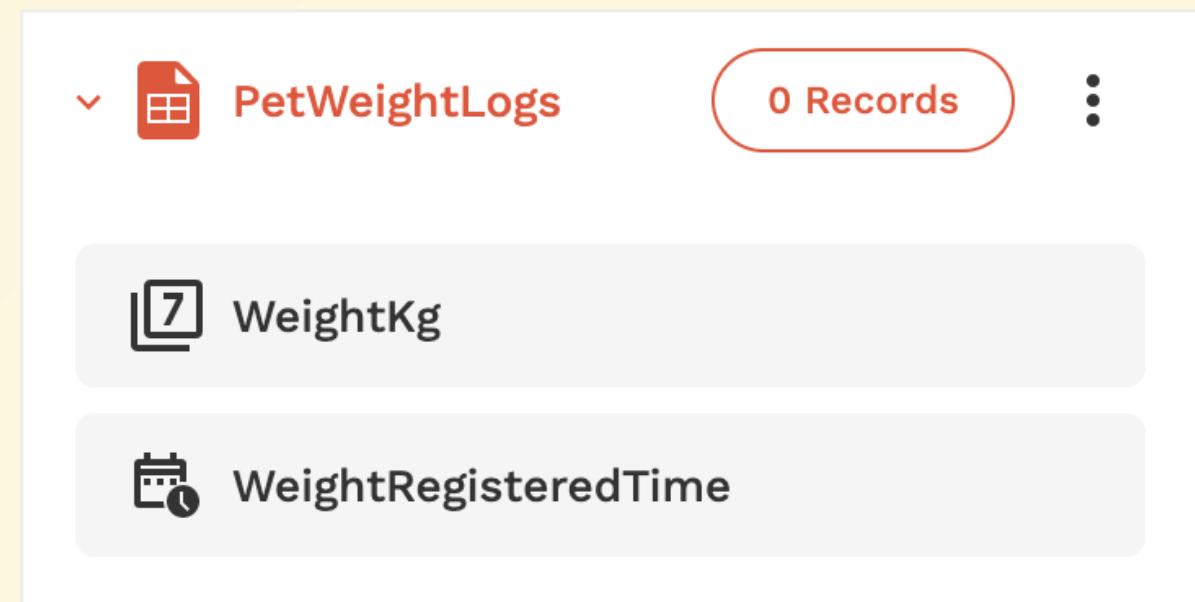
- Adaloで実際にPropertyを追加します。
- Users Collectionはデフォルトで設定済みで、必要な項目は含まれています。
- Usernameは不要ですが、削除できないのでそのままにしておきます。



- Pets CollectionのPropertyはこのようになります。

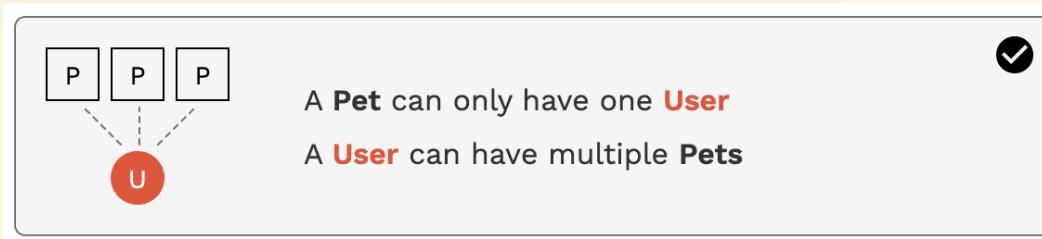


- PetWeightLogs Collection のPropertyはこのようになります。
- Collection追加時にデフォルトで設定されるName Propertyは不要なので、削除します。
  - ドラッグアンドドロップで順番がCollection内の一一番上でなくなるように移動すれば、削除できます。



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。



▼ Users 0 Records ⋮

Email

Password

Username

Full Name

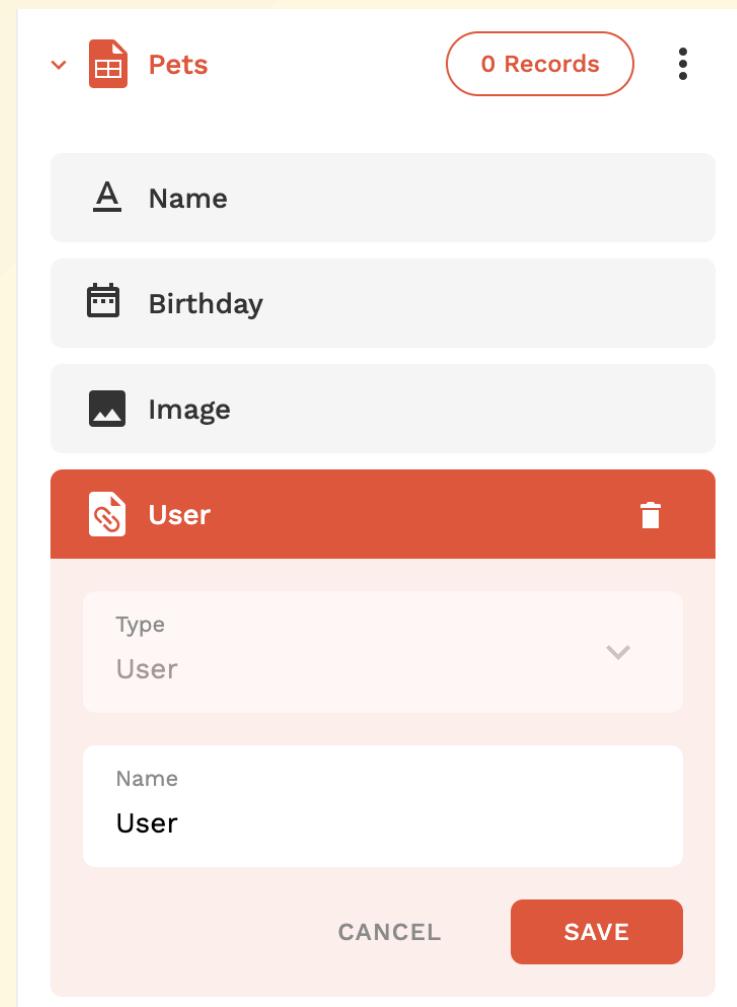
Pets trash icon

Type  
Pet

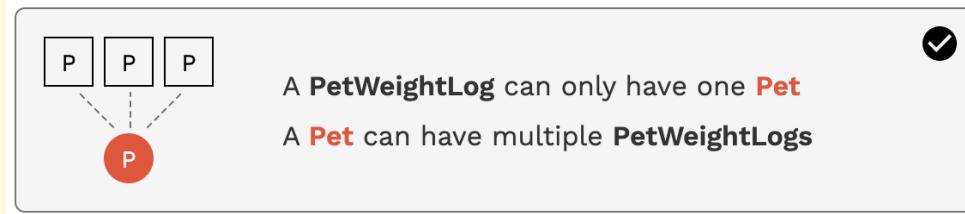
Name  
Pets

CANCEL SAVE

- 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。



Pets 0 Records ⋮

A Name

Birthday

Image

User

**PetWeightLogs** trash

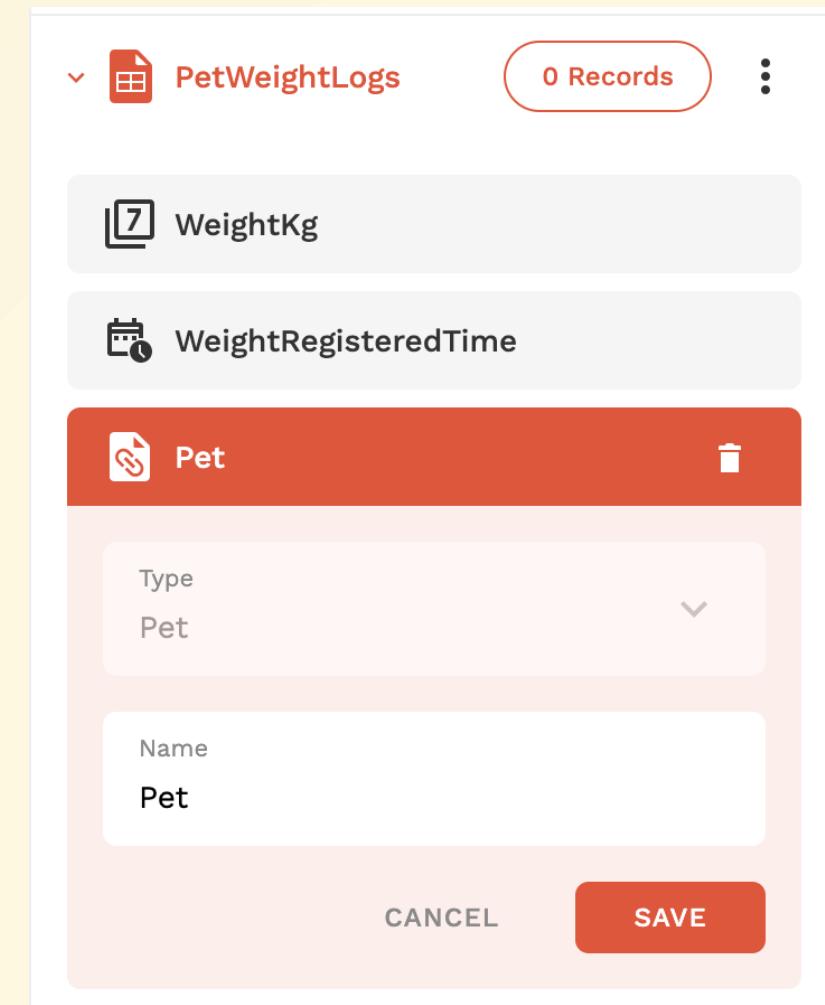
Type  
PetWeightLog

Name  
PetWeightLogs

CANCEL SAVE

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。

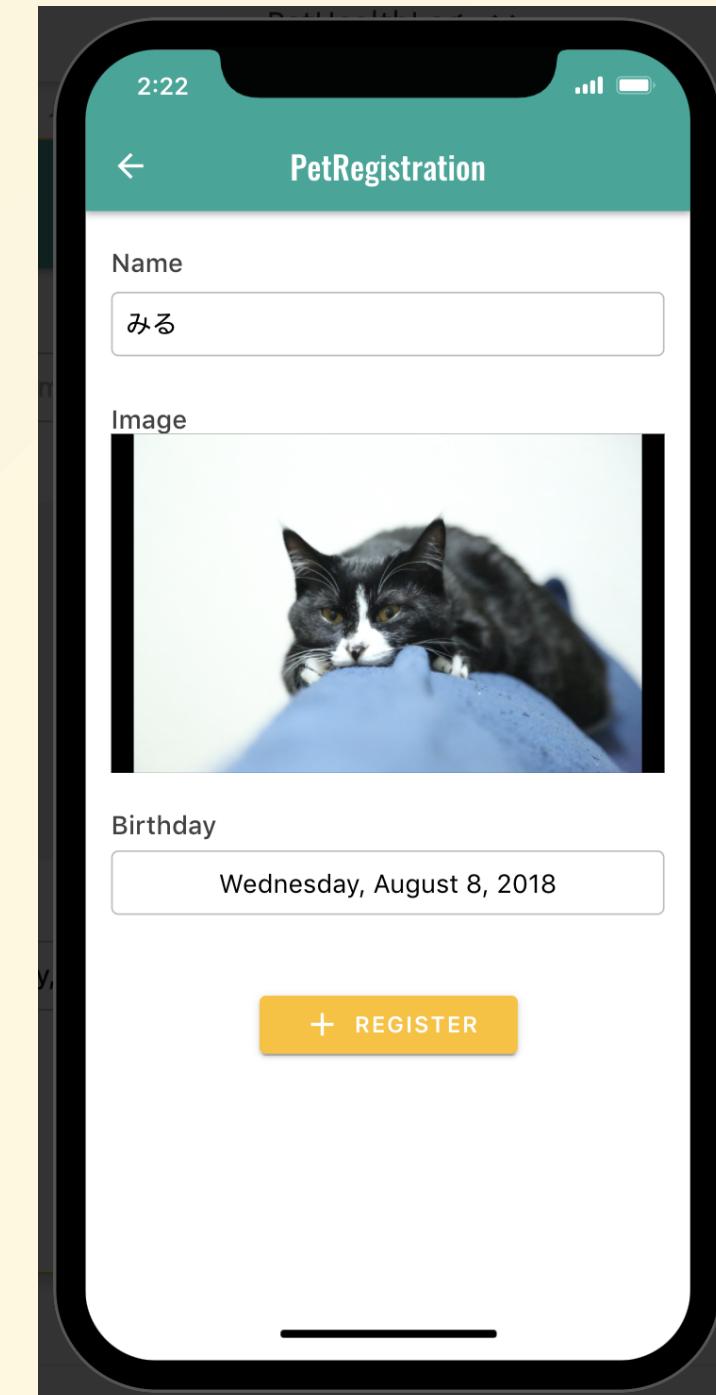
\* この後の作業で混乱しないために、Adaloのデータベースを資料と同じ状態に設定しておくことをおすすめします。

# データベース操作

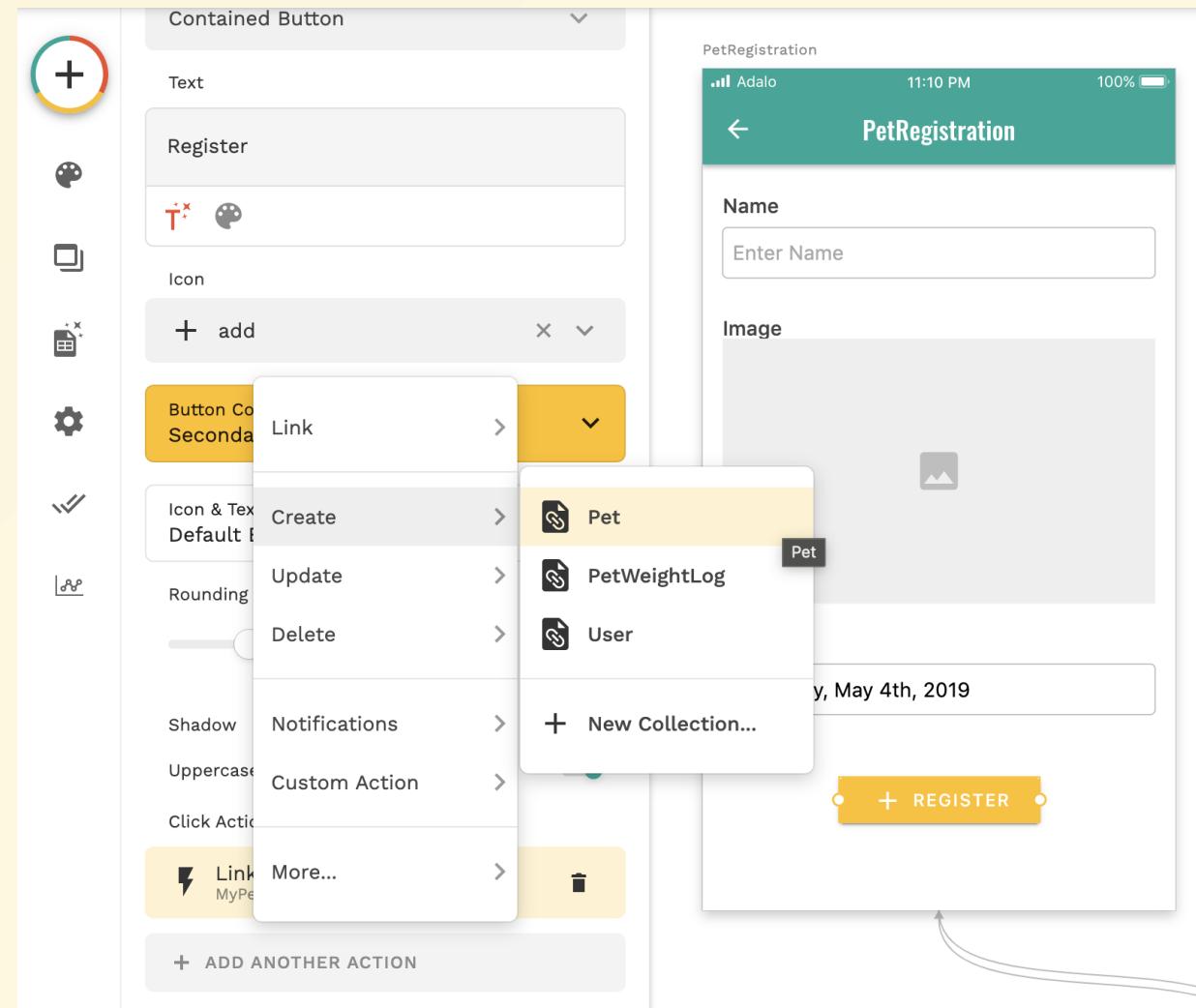
設計したデータベースを使って、サンプルアプリでデータのCRUD操作ができるようにしましょう。

## データの作成(CREATE)

まず、作成済みのペット登録画面で実際にペットのレコードを登録できるようにします。

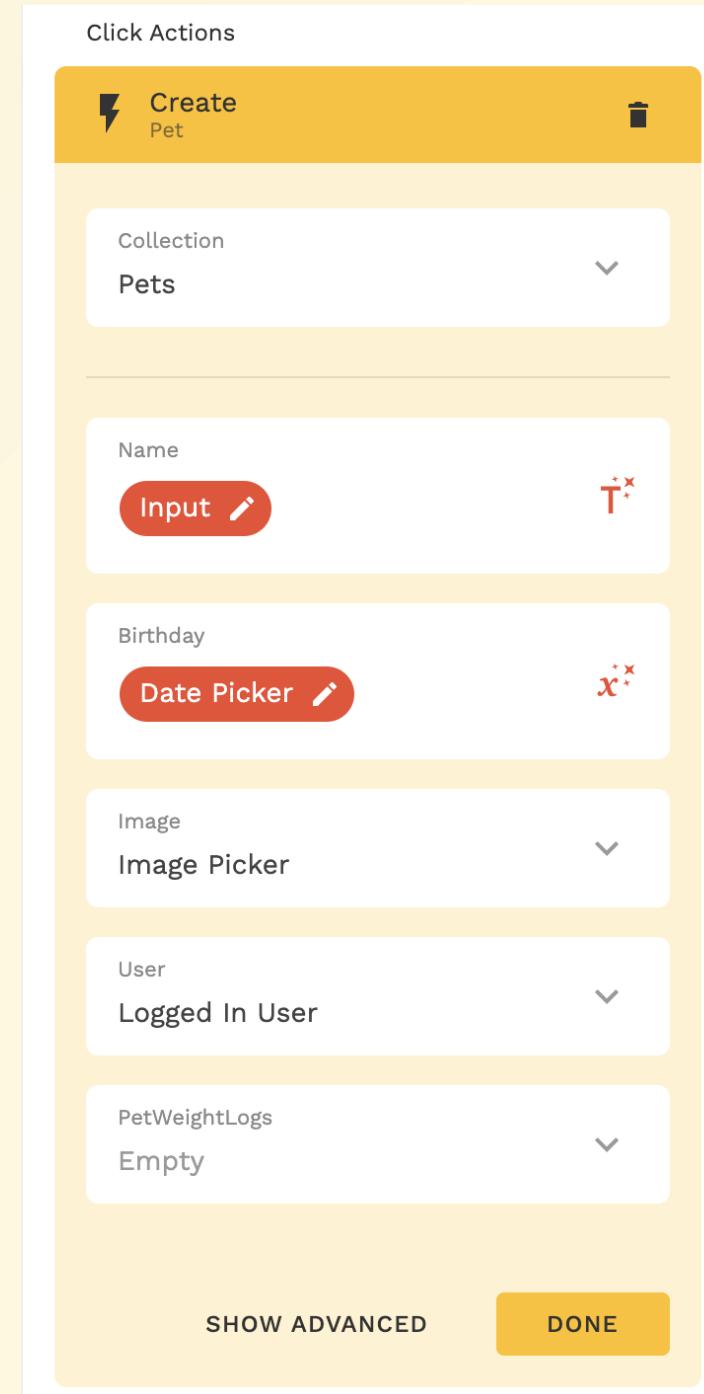


- ペット登録画面のREGISTERボタンを選択し、「ADD ANOTHER ACTION」をクリック
- Create > Pet を選択



以下を入力してDONE。

- NameはOther ComponentsのInputを選択
- BirthdayはOther ComponentsのDate Pickerを選択
- ImageはOther ComponentsのImage Pickerを選択
- UserはLogged In Userを選択
- PetWeightLogsはEmptyのまま(ペット登録時には不要)



Preview機能でペットを登録してみましょう。  
Pets CollectionにRecordが登録されたらOKです。

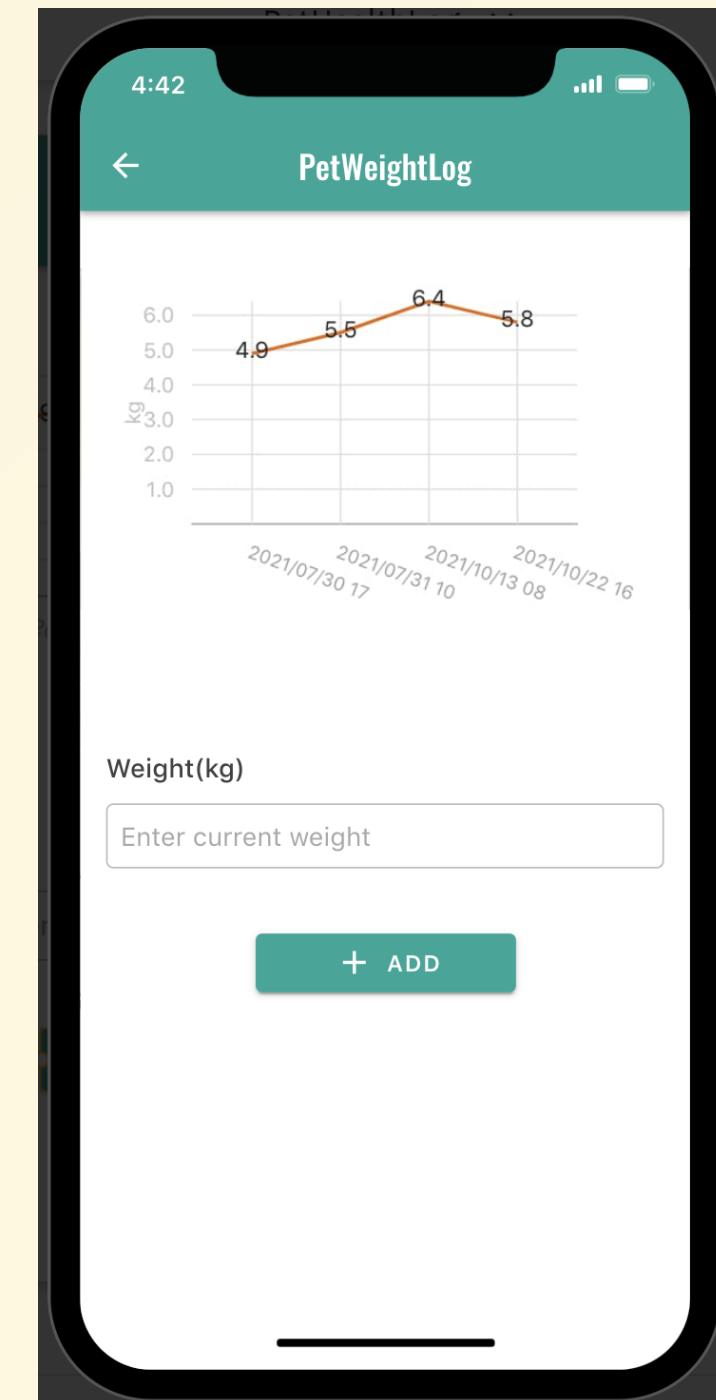
Pets

[+ ADD PET](#)    

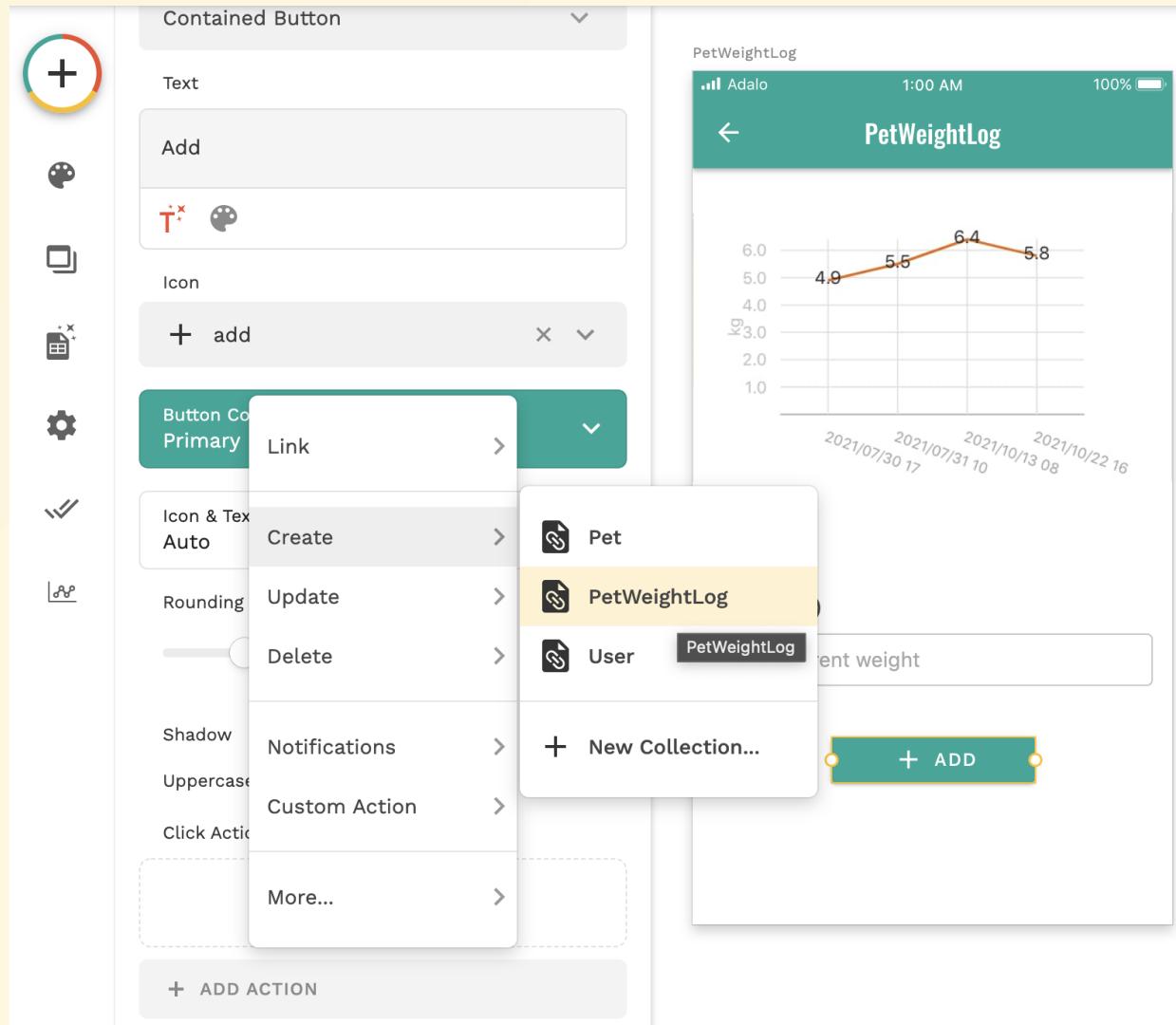
<input type="checkbox"/>	Name	Birthday	Image	User	PetWeightLogs	Created
<input type="checkbox"/>	みる(Database Record)	8/8/2018		imahashi@guildworks.jp		43 minutes ago

[DONE](#)

次に、ペットの体重管理画面で現在の体重を登録できるようになります。

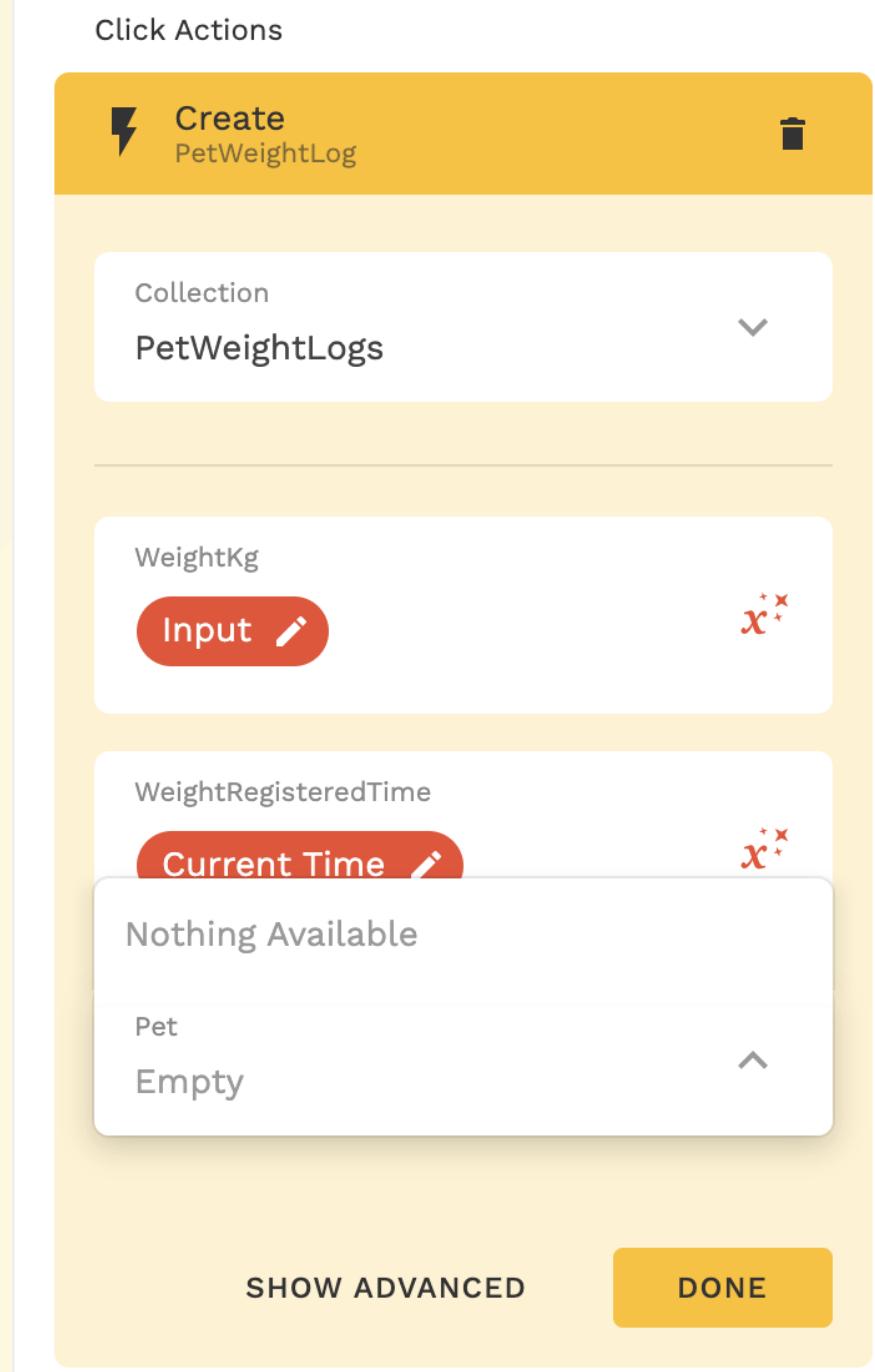


- ペットの体重管理画面で ADDボタンを選択し、「ADD ACTION」をクリック
- Create > PetWeightLog を選択



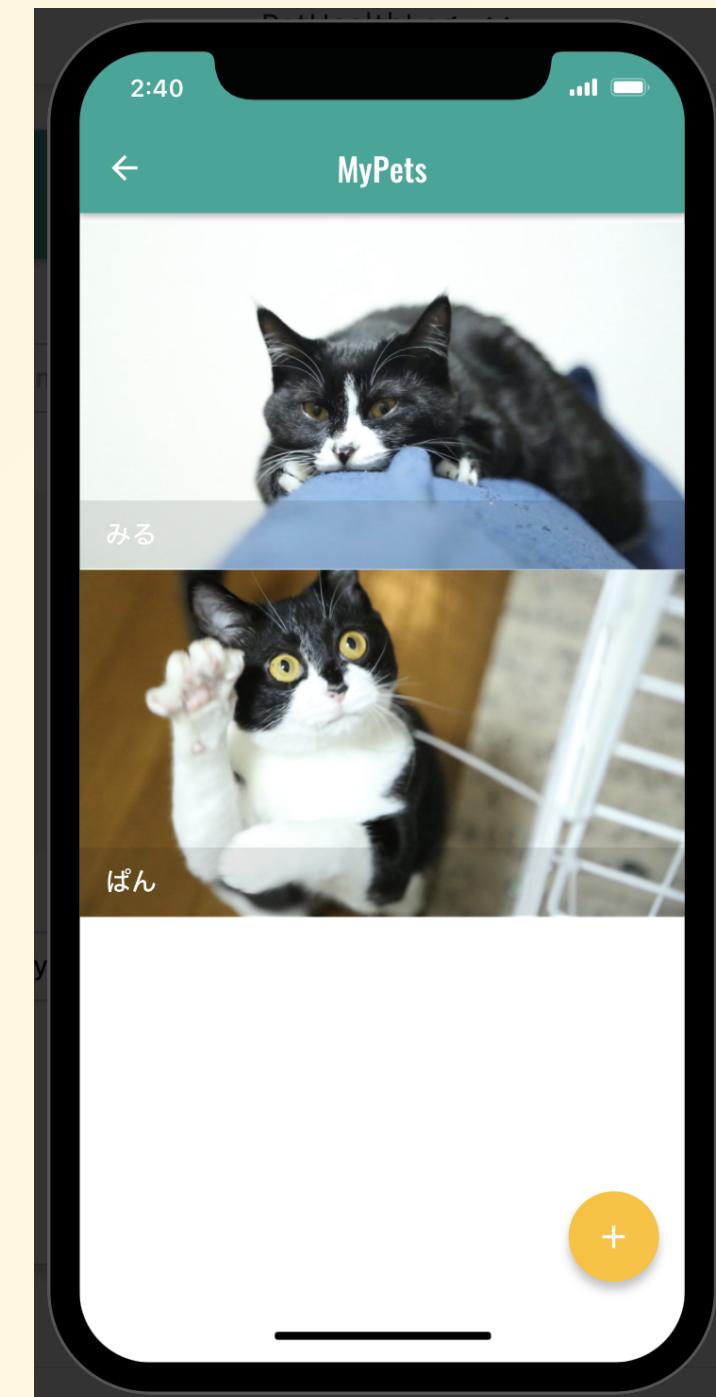
以下を入力してDONE。

- WeightKgはOther ComponentsのInputを選択
- WeightRegisteredTimeは Date&Time > Current Time を選択
- PetはNothing Availableなので、一旦Emptyのままにする  
(後ほど、選択したペットの体重を登録できるように設定します)

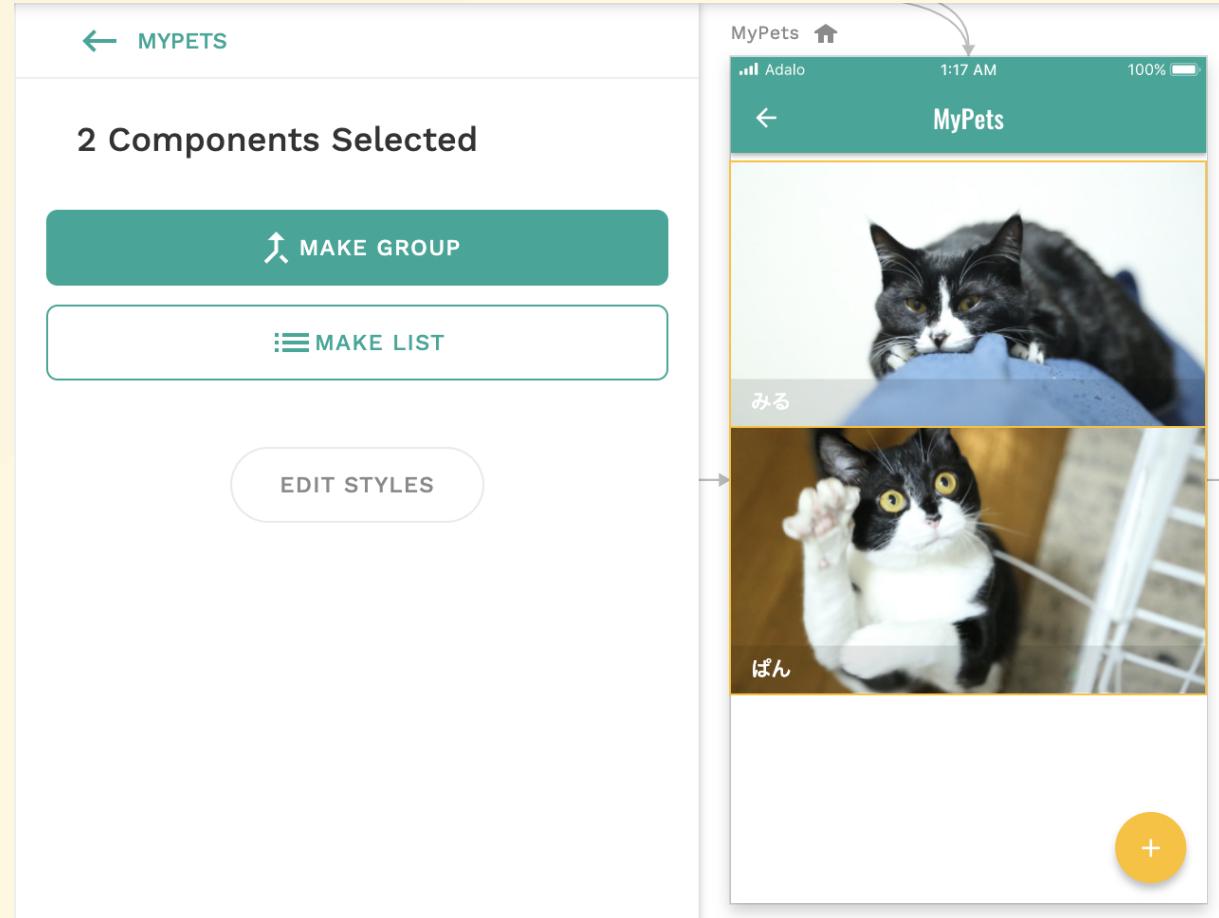


## データの表示(READ)

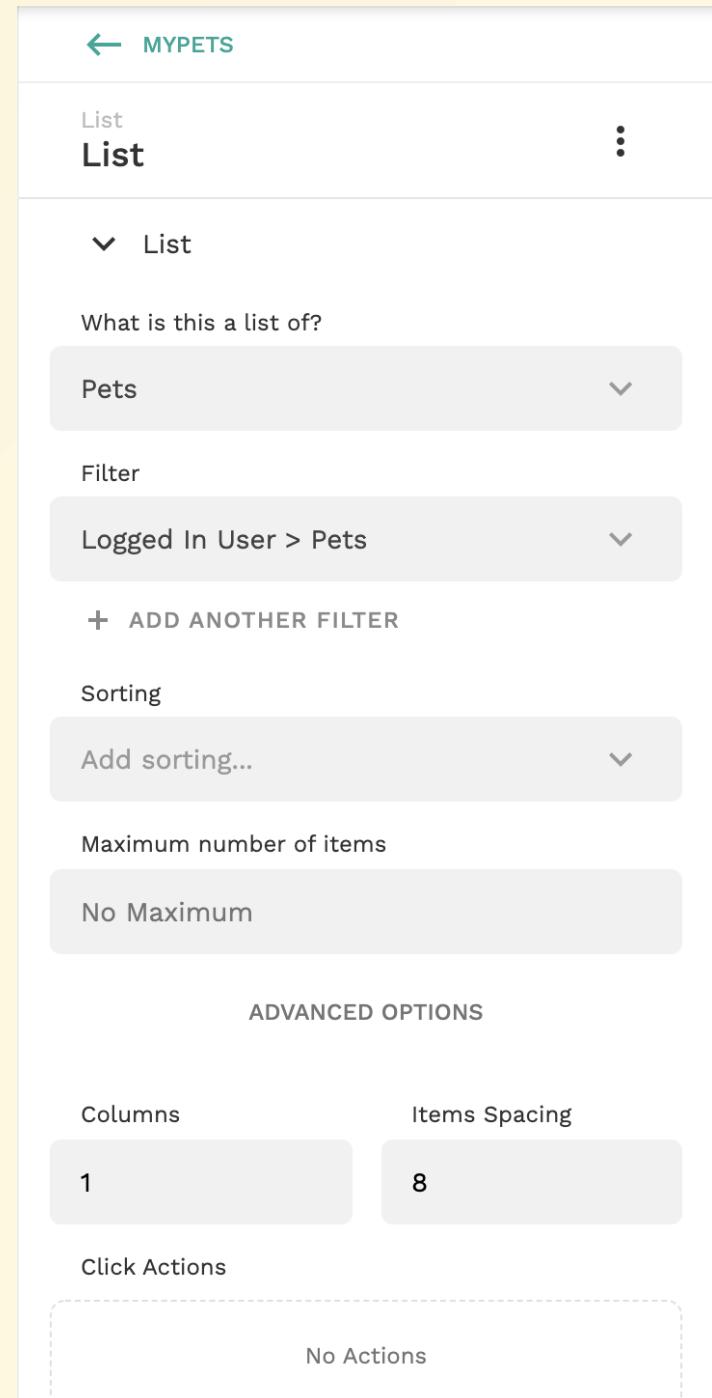
まず、作成済みのペット一覧画面に実際に登録したペットが表示されるようにします。



- ペットの画像と名前を表示している2つのコンポーネントを選択し、MAKE LISTをクリック



- What is this a list of?でPetsを選択
- FilterでLogged In User > Petsを選択



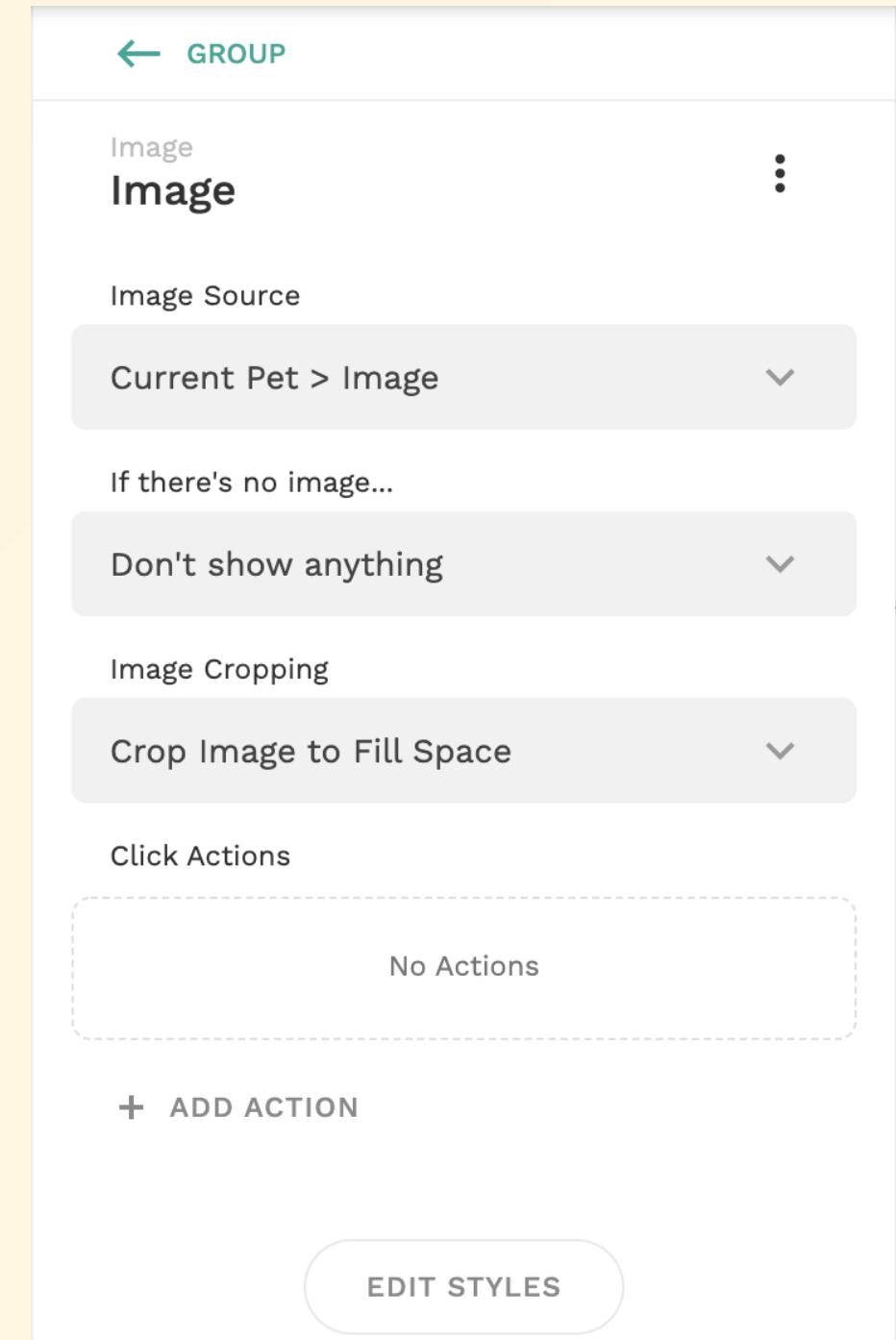
- 作成したListを構成するコンポーネントであるGroupの1つ目をクリック。
- Group内のImageコンポーネント、ペット名のコンポーネントをそれぞれ編集していきます。

The screenshot shows the Figma interface with two panels open:

- Left Panel (MYPETS):** Shows a list of components under "Components". Two items are highlighted: "Group" (with ID "Group") and "Group 3" (with ID "Group 3"). Below the list is a button labeled "+ ADD COMPONENT".
- Right Panel (LIST):** Shows the internal structure of the "Group" component. It contains three items:
  - Image**: An image component with the ID "Image".
  - Shape**: A shape component with the ID "Rectangle".
  - Text**: A text component with the ID "みる".

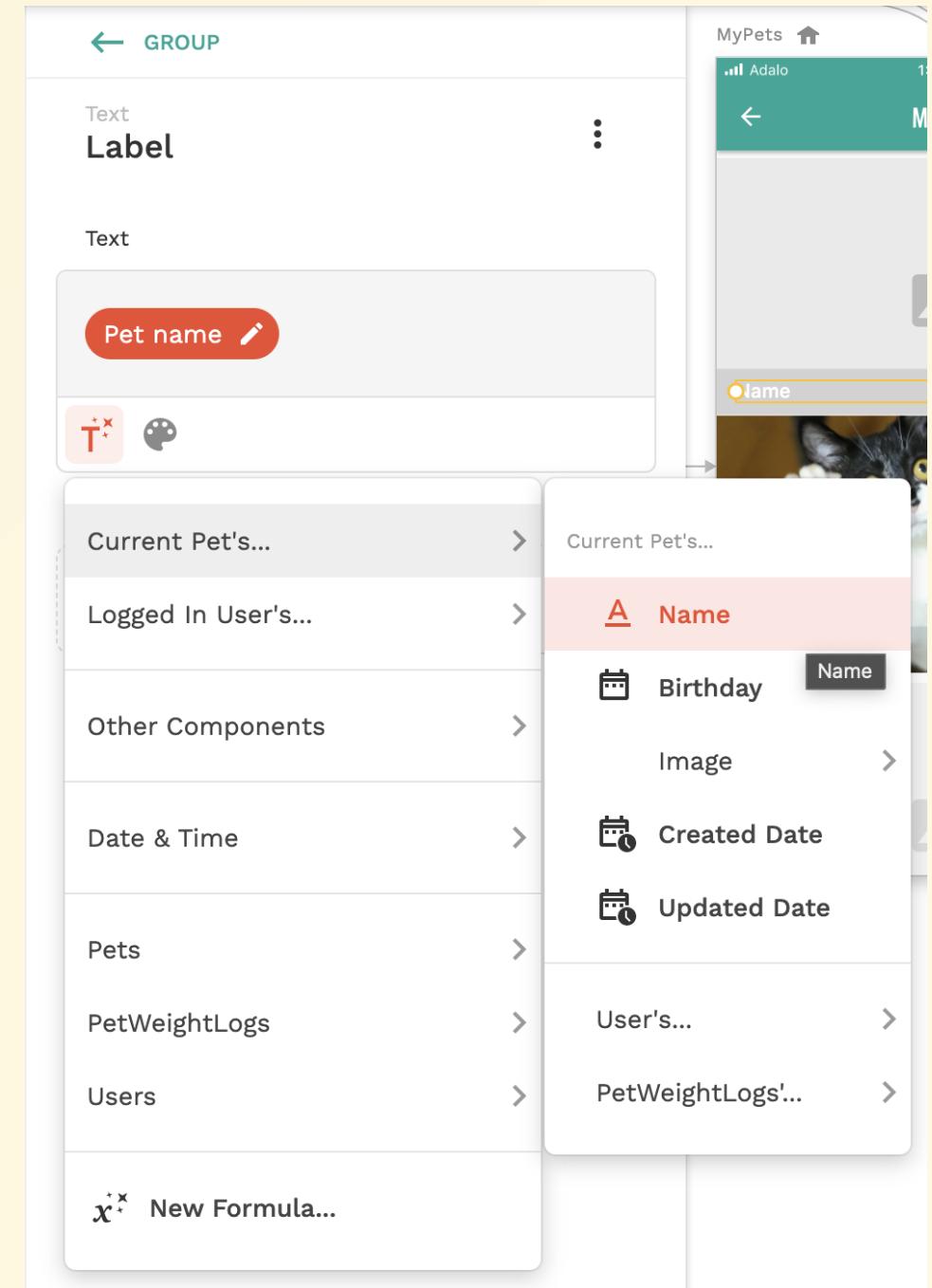
# Imageコンポーネントを編集

- Image Sourceで、Database > Current Pet's > Imageを選択
- If there's no image...でDon't show anythingを選択
  - あるいは、Show a place holder imageを選択して好きなペットのシルエット画像を設定してもOKです

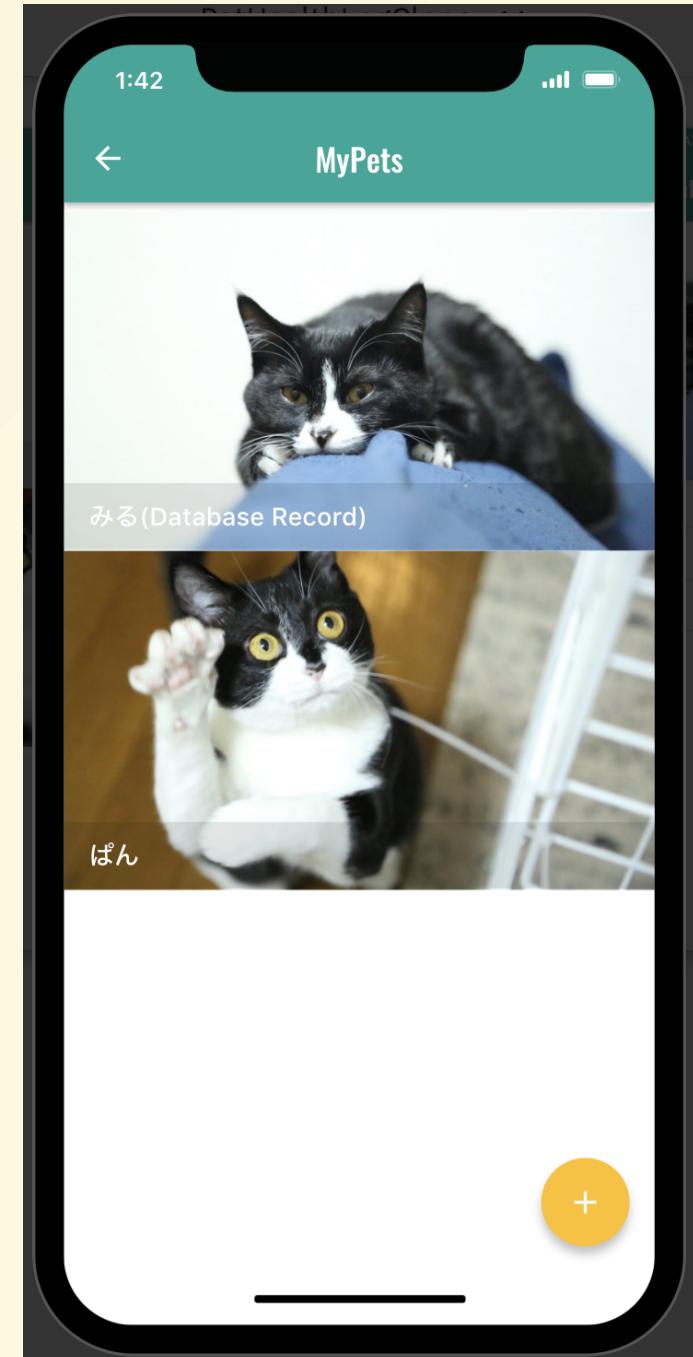


# ペット名のコンポーネントを編集

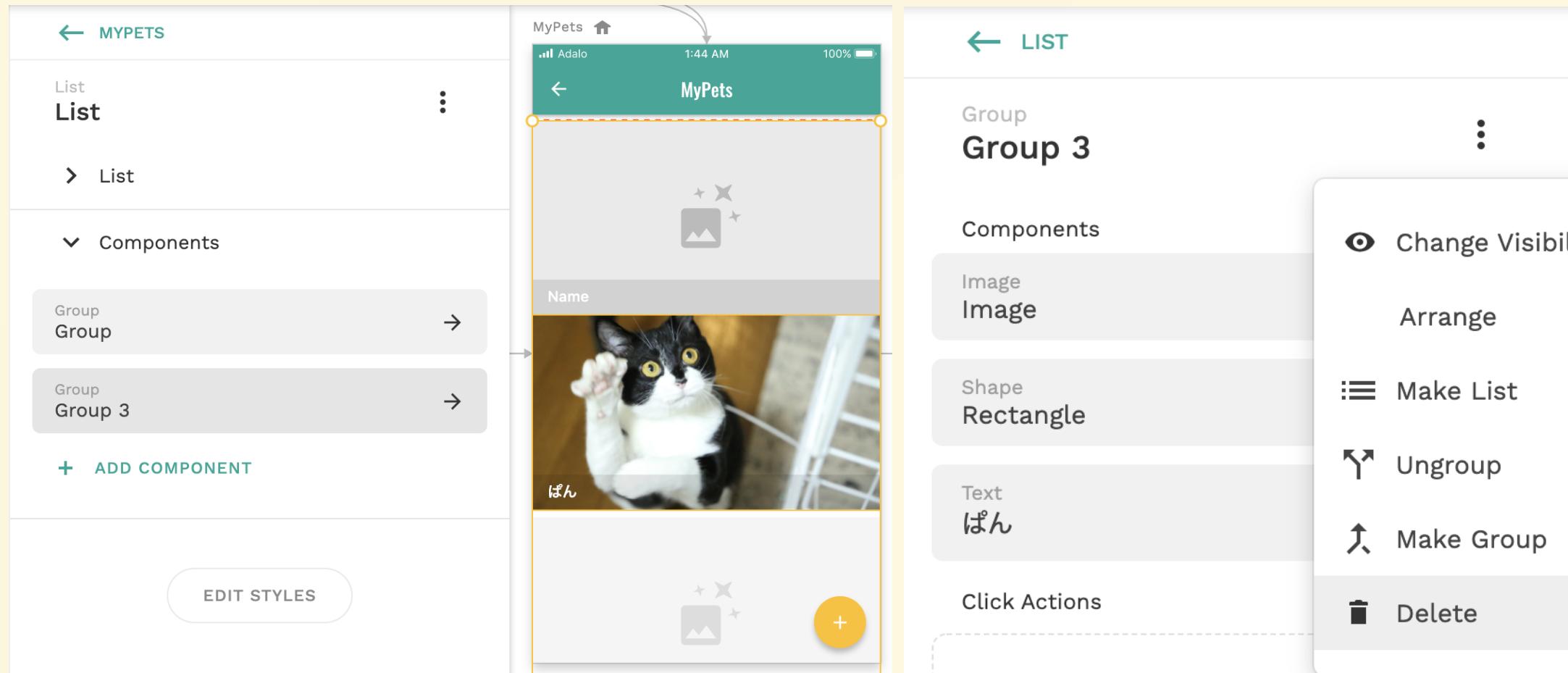
- Add Magic TextでCurrent Pet's > Nameを選択



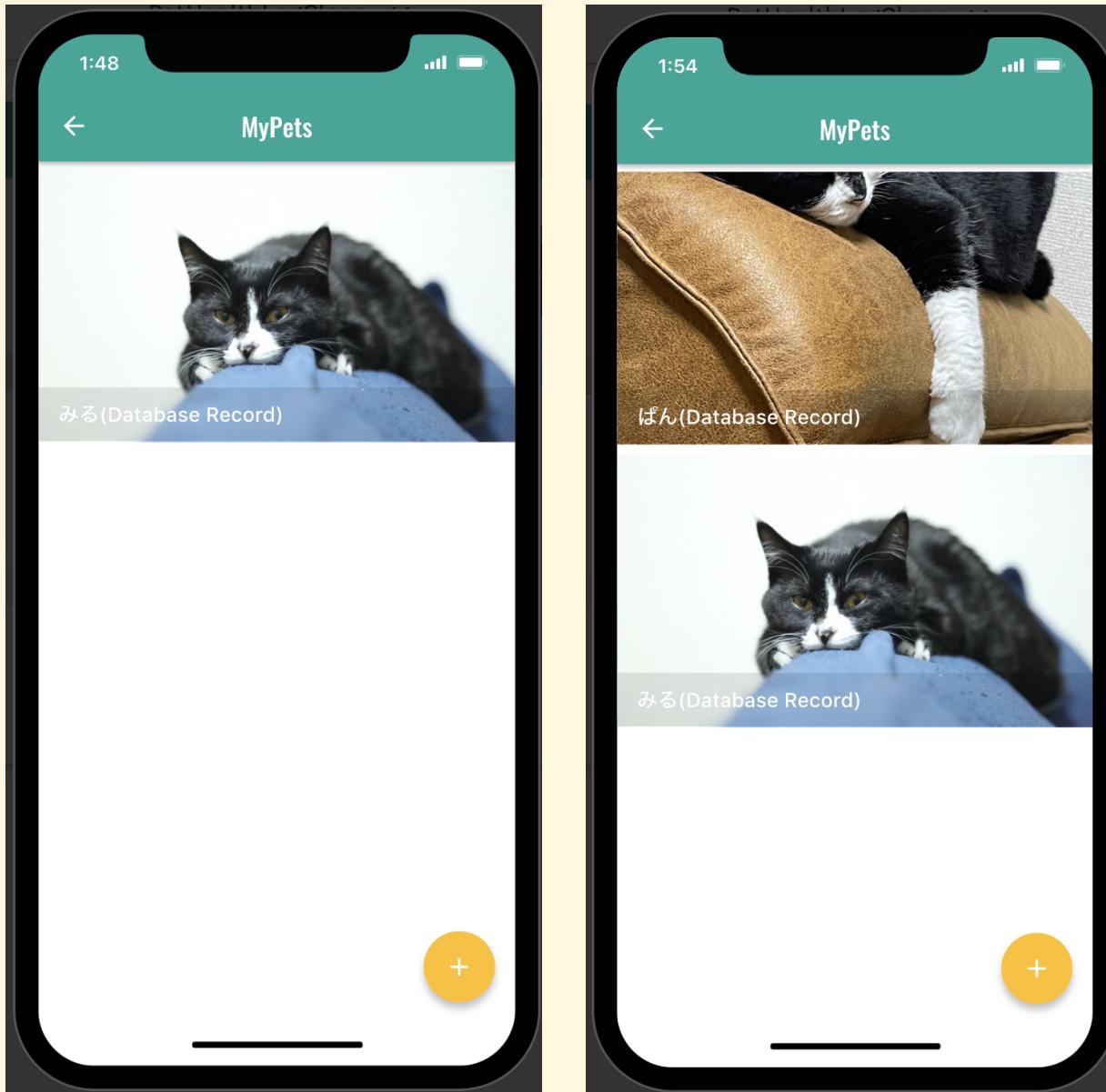
- Preview機能で確認すると、1匹目のペットとしてデータベースに登録したRecordが表示されます。



- ペットのListを構成するコンポーネントの中の2つ目のGroup(固定で表示していた2匹目のペット)は不要なので、削除しましょう

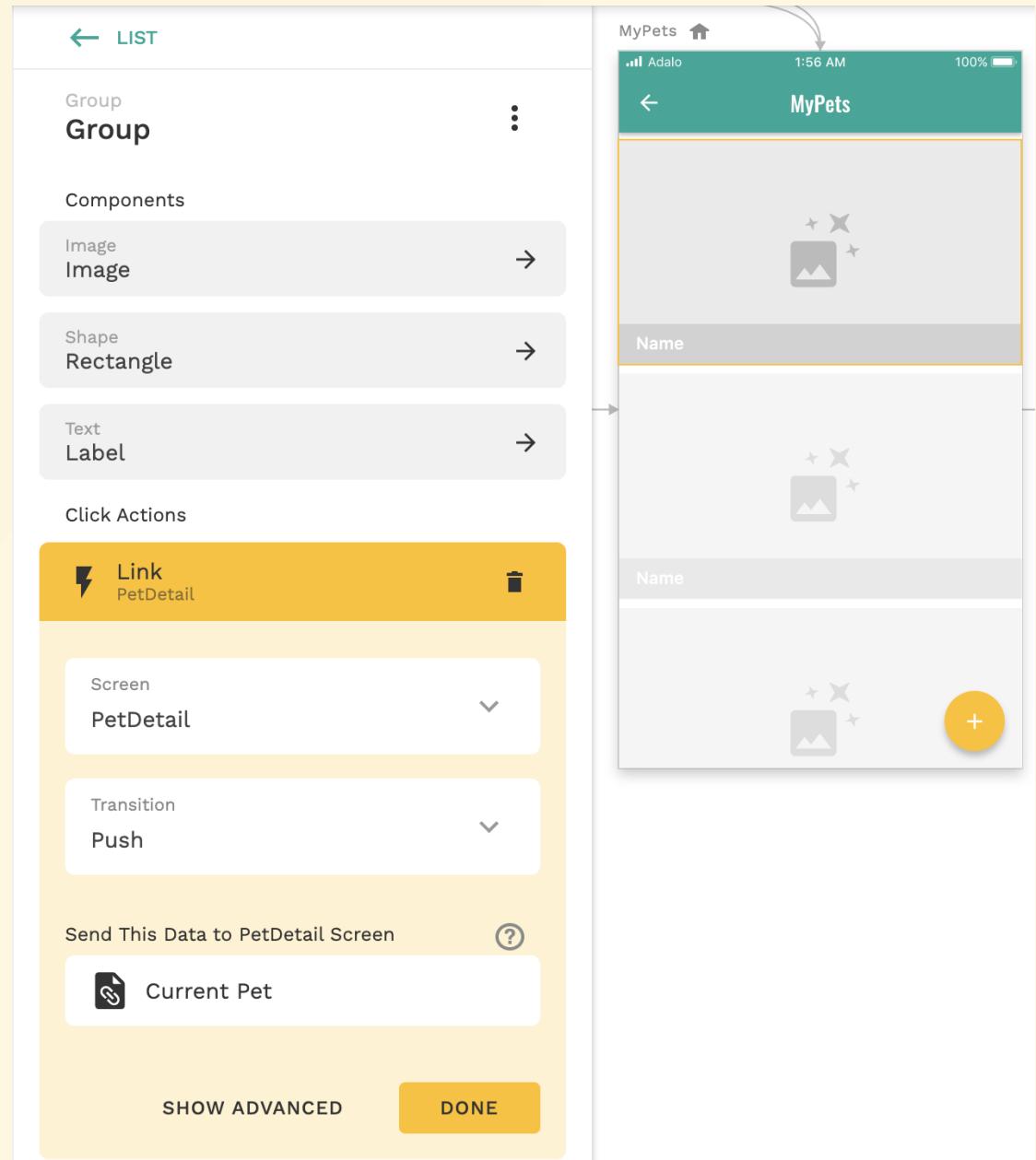


- Preview機能で確認すると、データベースに登録したペットだけが表示されるようになりました。
  - ペットを追加で登録すれば、複数のペットが表示されます。

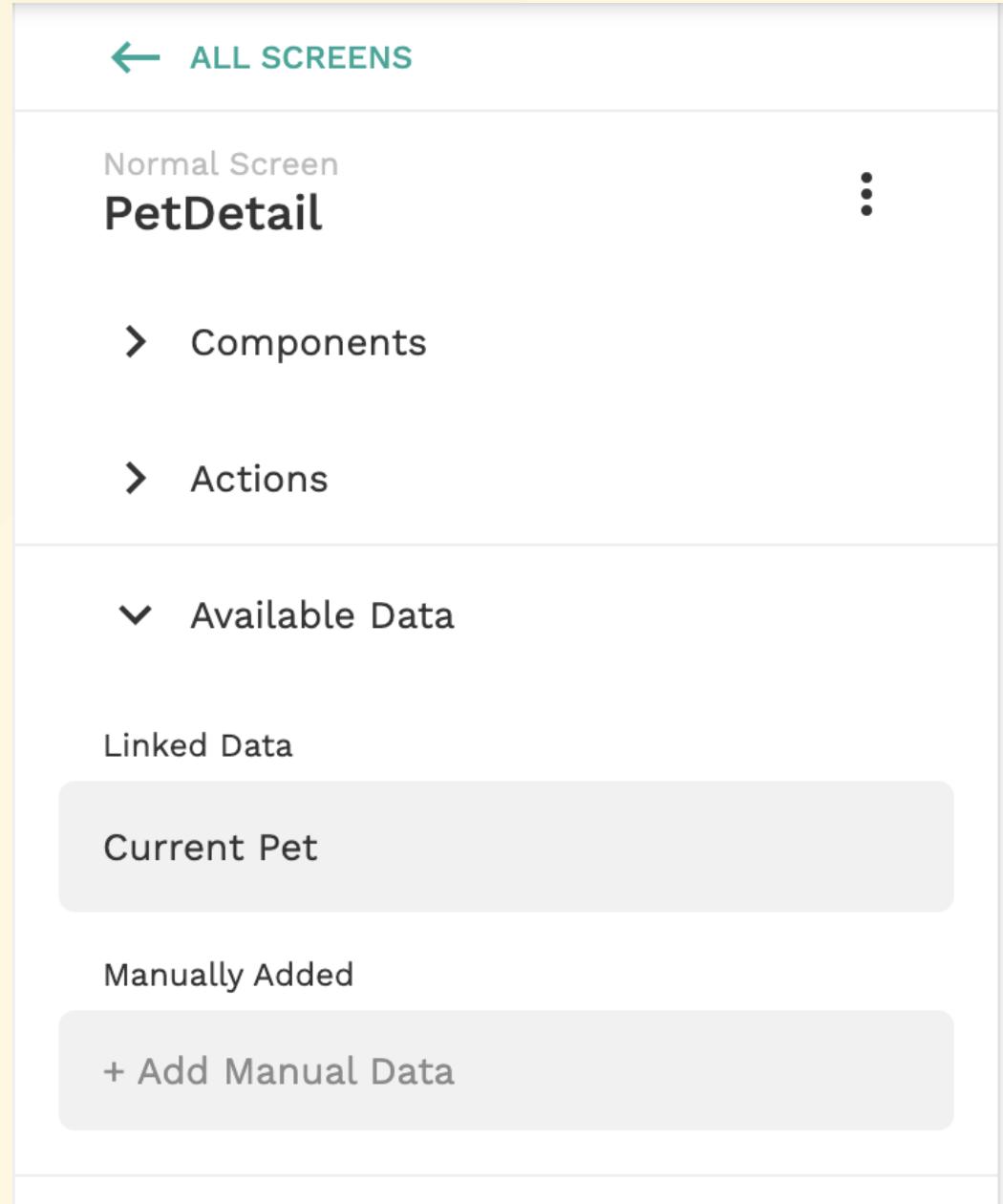


ペット一覧でペットをクリックした時に、そのペットの詳細画面に遷移できるようになっていることを確認します。

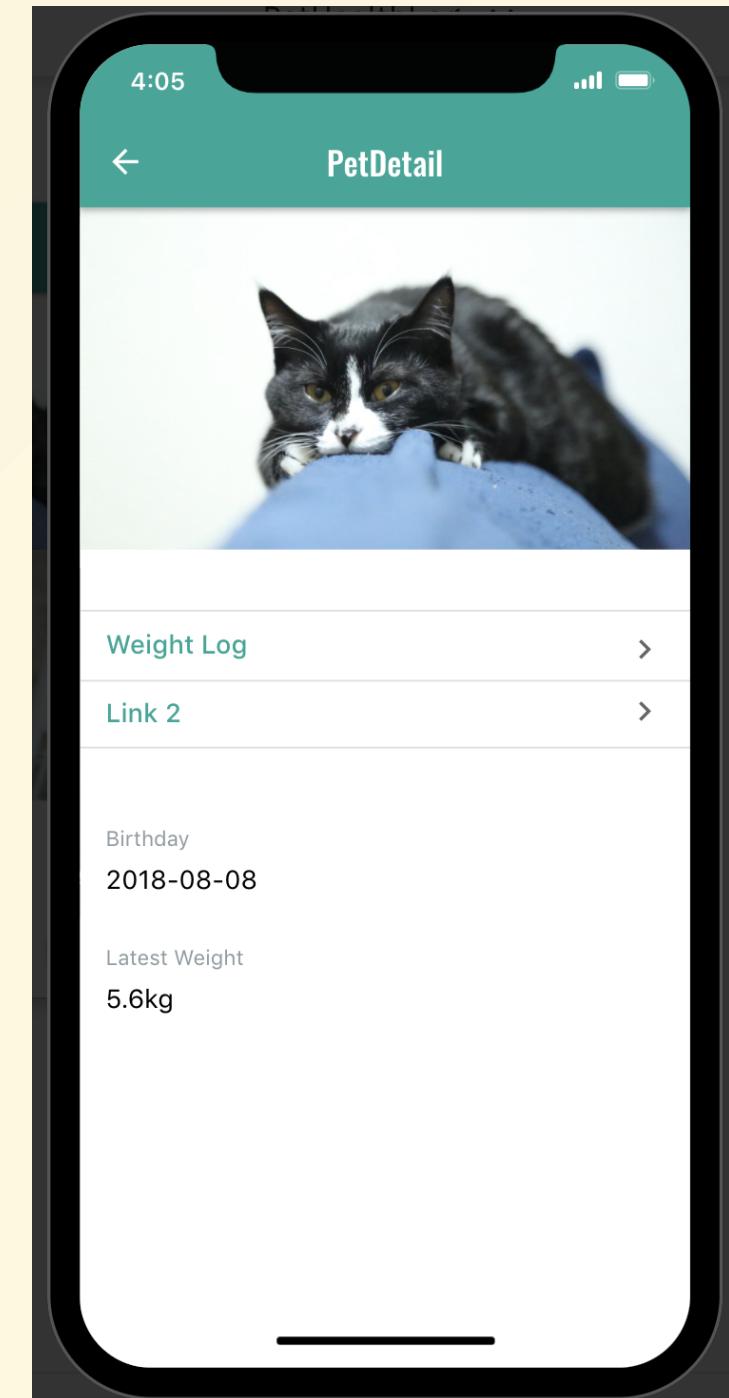
- ペットのGroupコンポーネントに設定されているLinkのSend This Data to PetDetail ScreenにCurrent Petが自動で設定されています。



- 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。

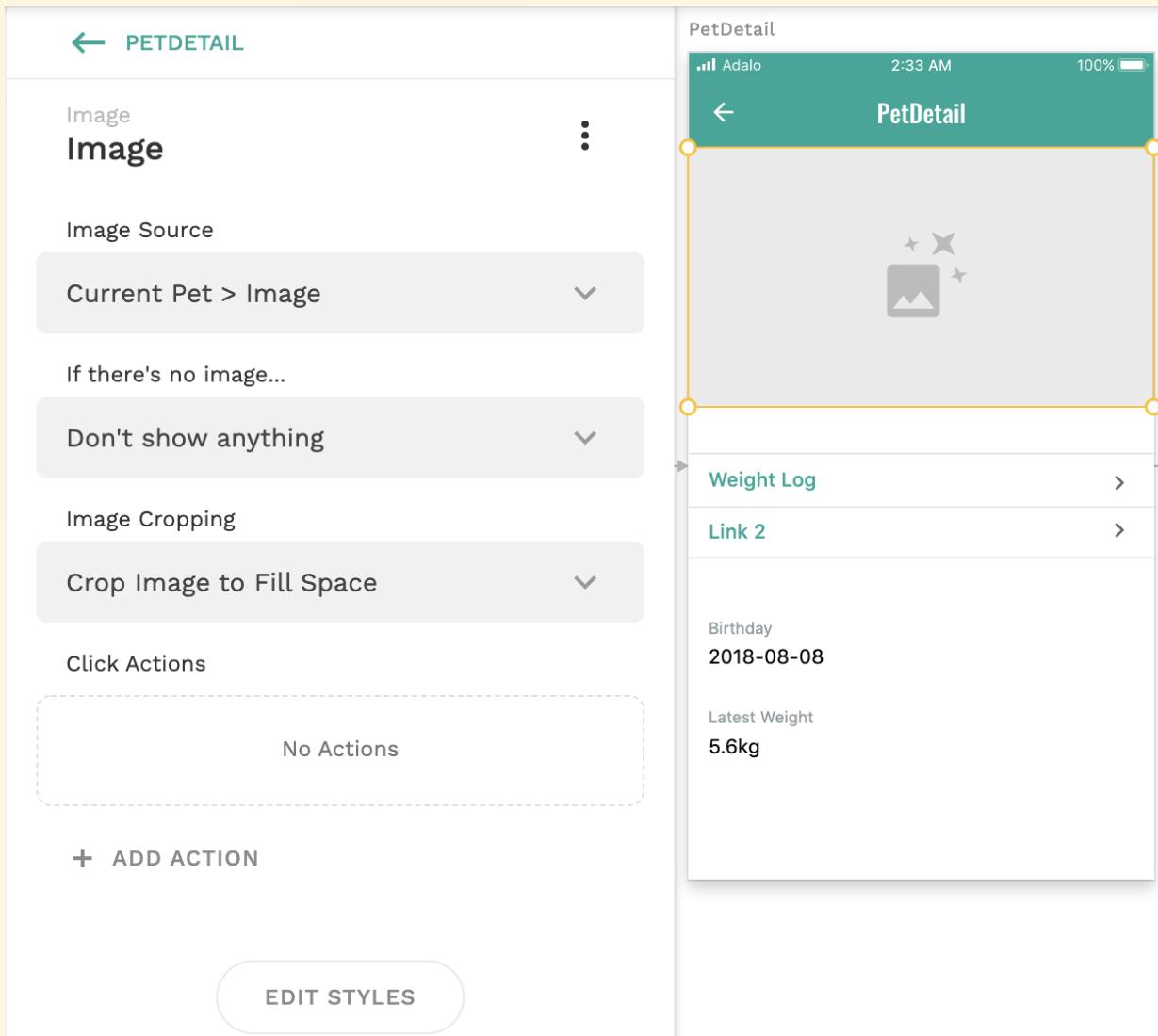


次に、ペット詳細画面にペット一覧画面で選択したペットが表示されるようにします。



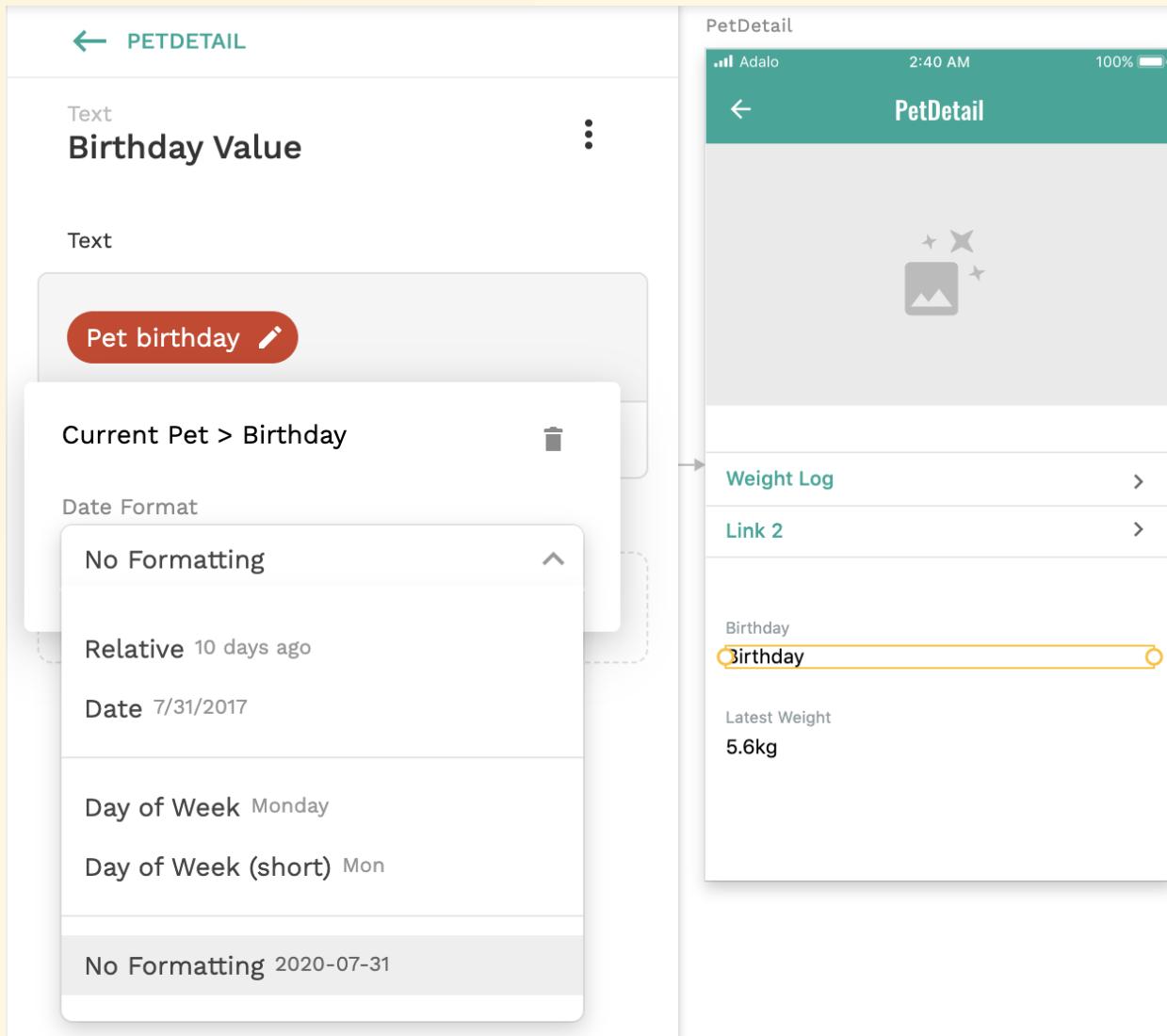
Imageコンポーネントをクリックし、

- Image SourceでDatabase > Current Pet's > Imageを選択
- If there's no image...でDon't show anythingを選択
  - あるいは、Show a place holder image を選択して好きなペットのシルエット画像を設定してもOKです



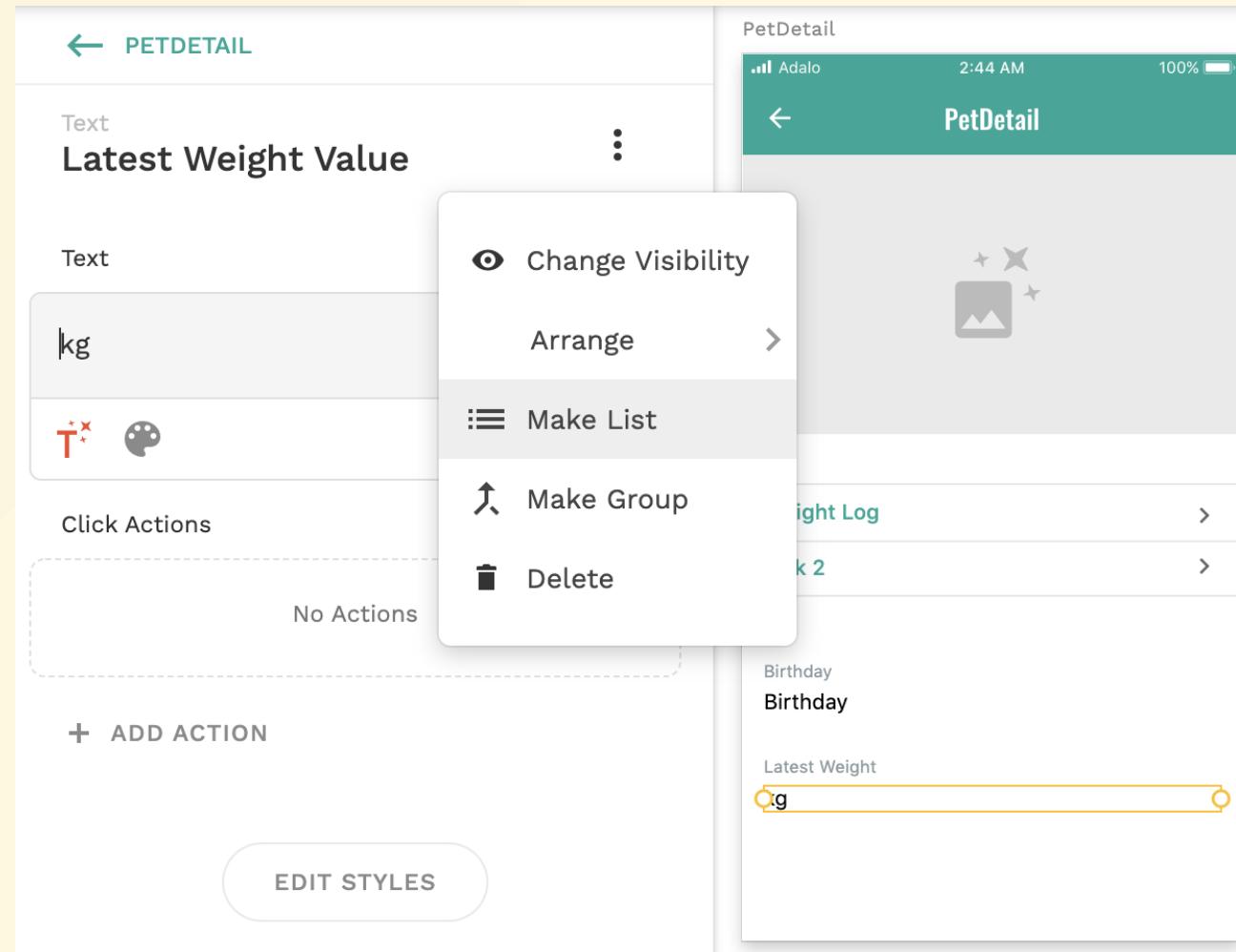
Birthday Valueコンポーネント  
をクリックし、

- TextでCurrent Pet's > Birthdayを選択
- Date FormatでNo Formattingを選択



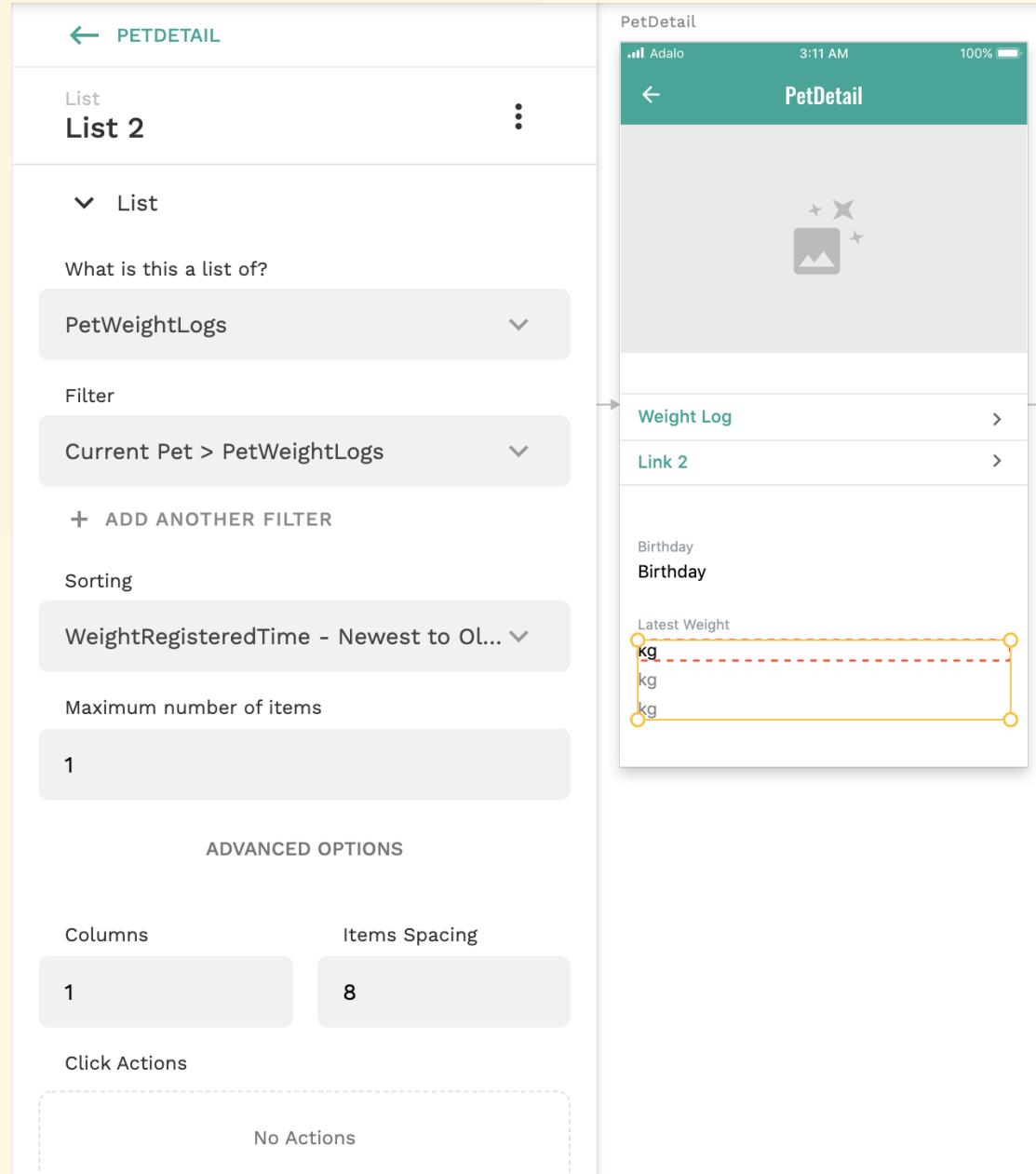
- Latest Weight Valueコンポーネントをクリックし、MAKE LISTでリストにする

\* 最新の1件を表示するためには、そのコンポーネントをListにします  
(次のページの設定で、最新の1件に絞り込みます)

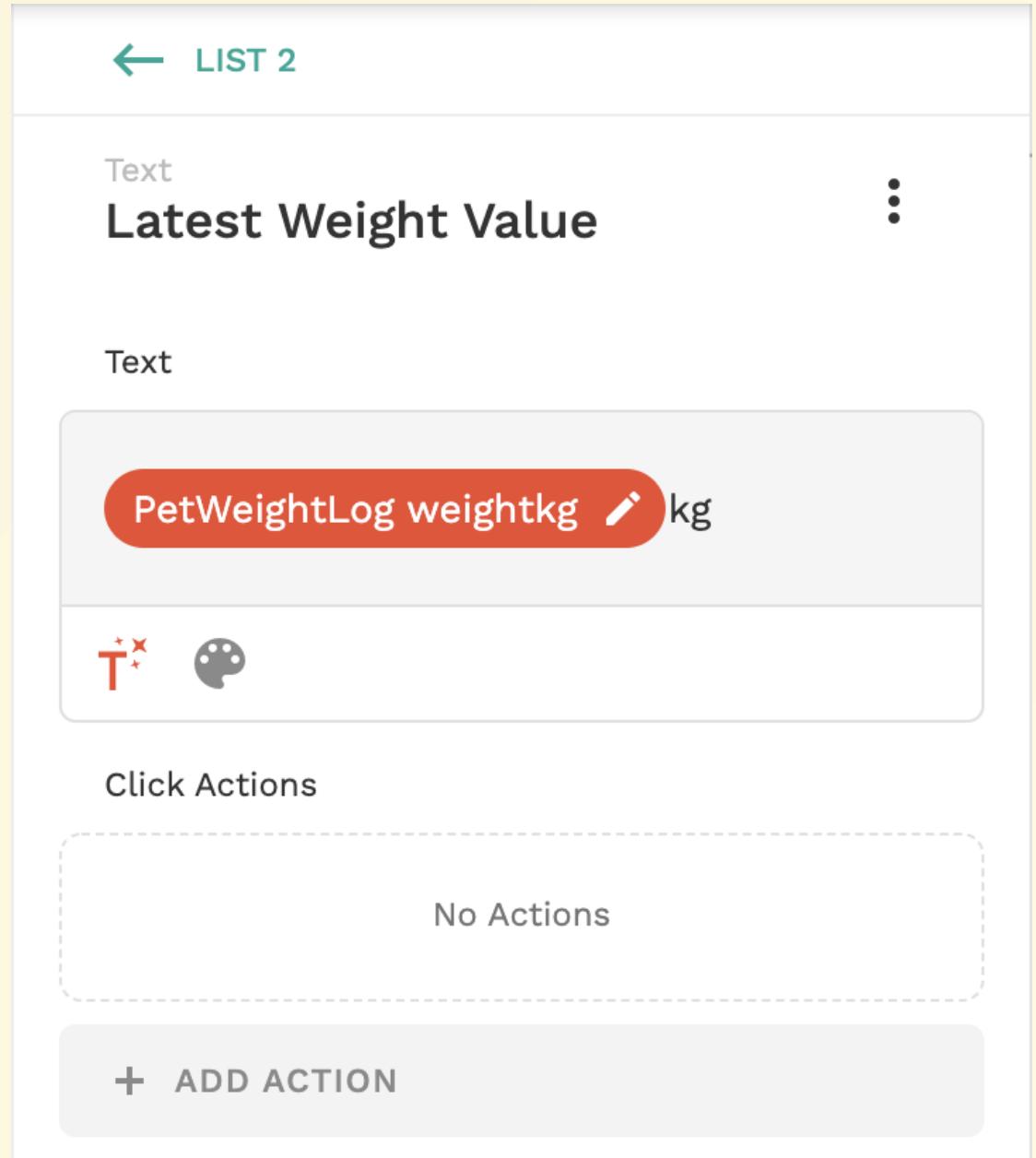


- What is this a list of?で PetWeightLogsを選択
- Filterで Current Pet > PetWeightLogsを選択
- Sortingで WeightRegisteredTime - Newest to Oldestを選択
- Maximum number of itemsに1を設定

これにより、最新の1件だけに絞り込まれます。

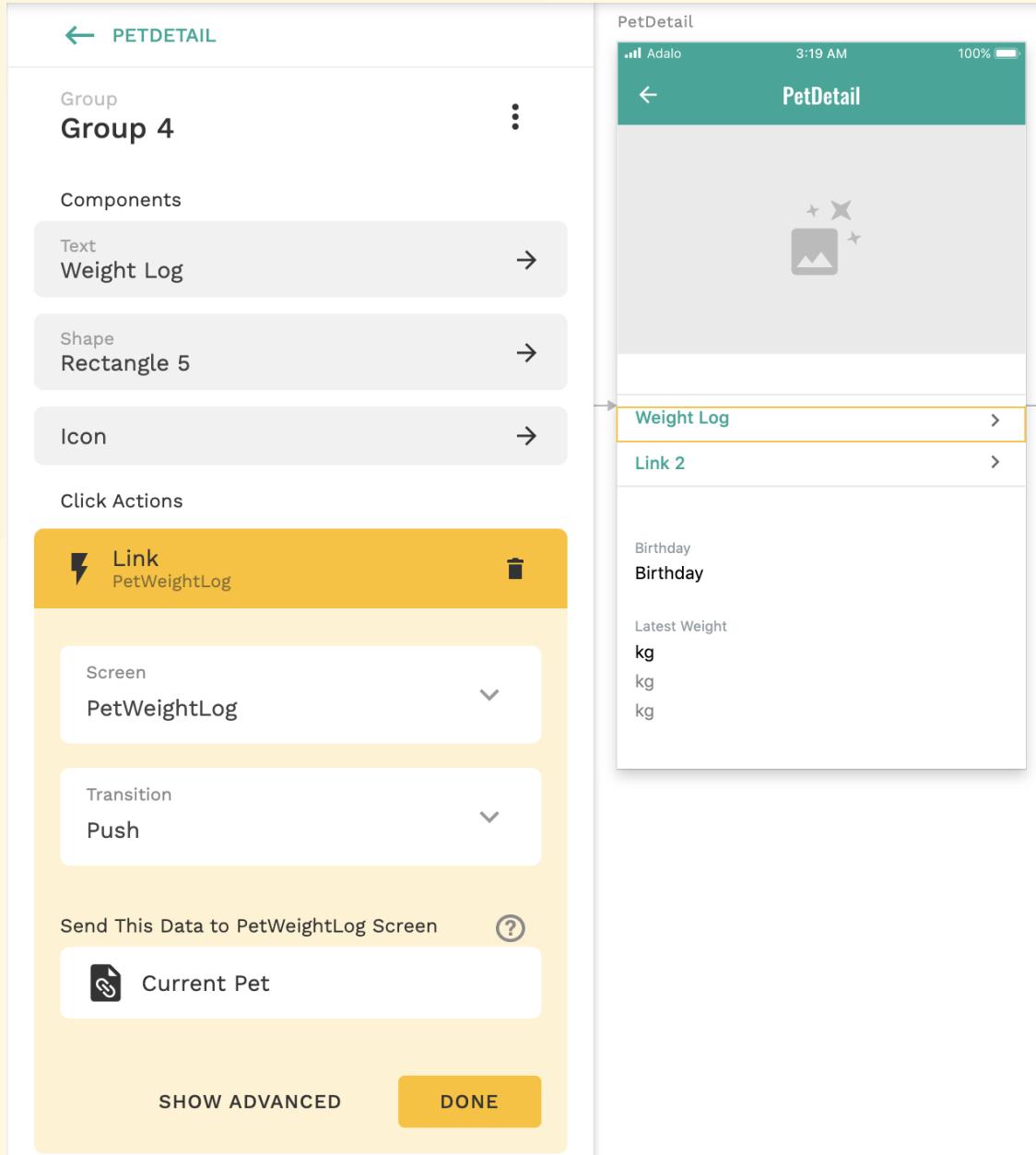


- Latest Weight Valueコンポーネントをクリックし、TextにCurrent PetWeightLog's > WeightKgを追加した後、末尾に"kg"をつける



ペット詳細画面からペットの体重管理画面に遷移する際に選択したペットが受け渡されることを確認します。

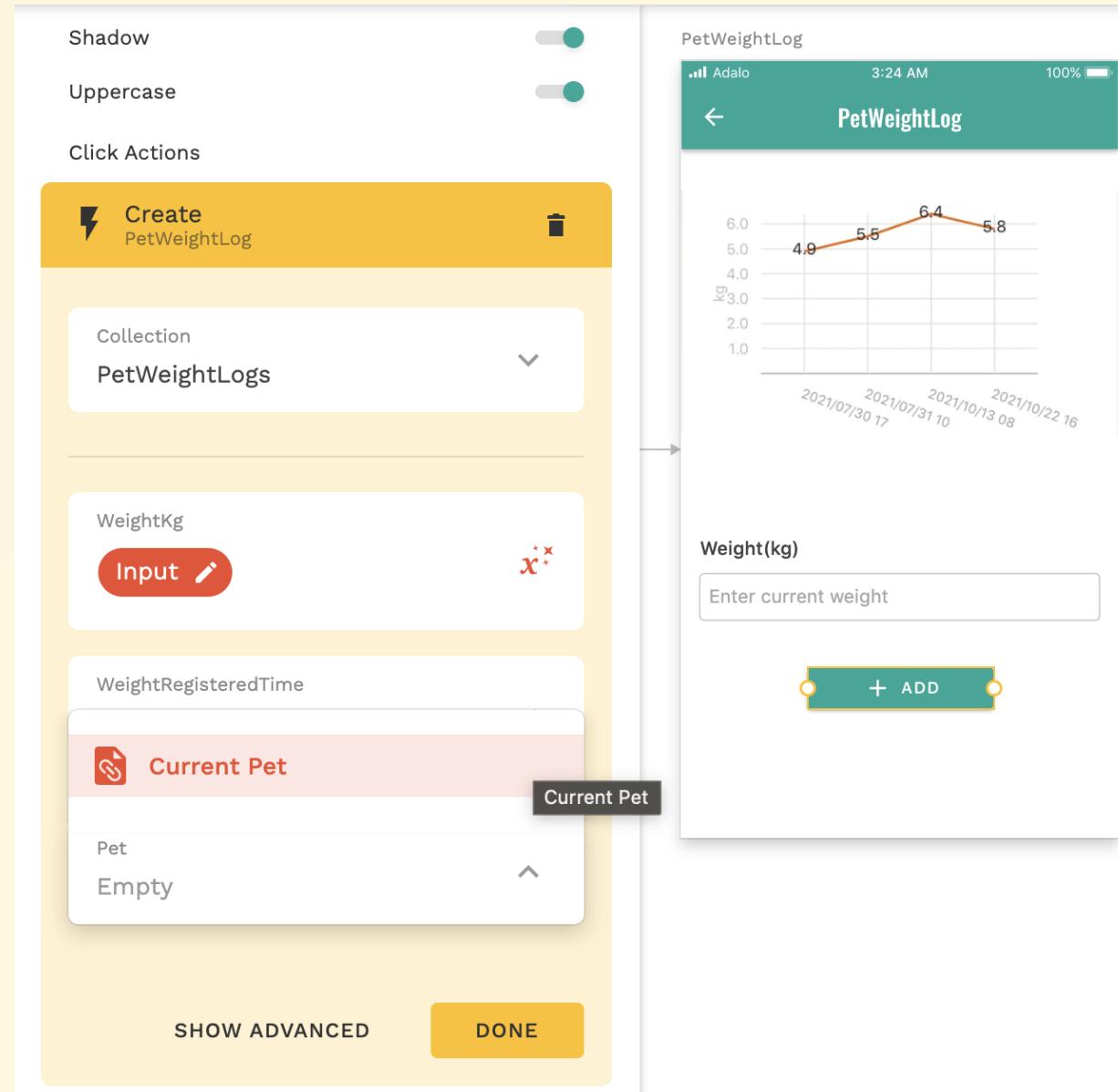
- Weight Logというリンクのコンポーネントを選択し、LinkのSend This Data to PetDetail ScreenにCurrent Petが自動で設定されています。



## \* データの作成(CREATE)の残作業

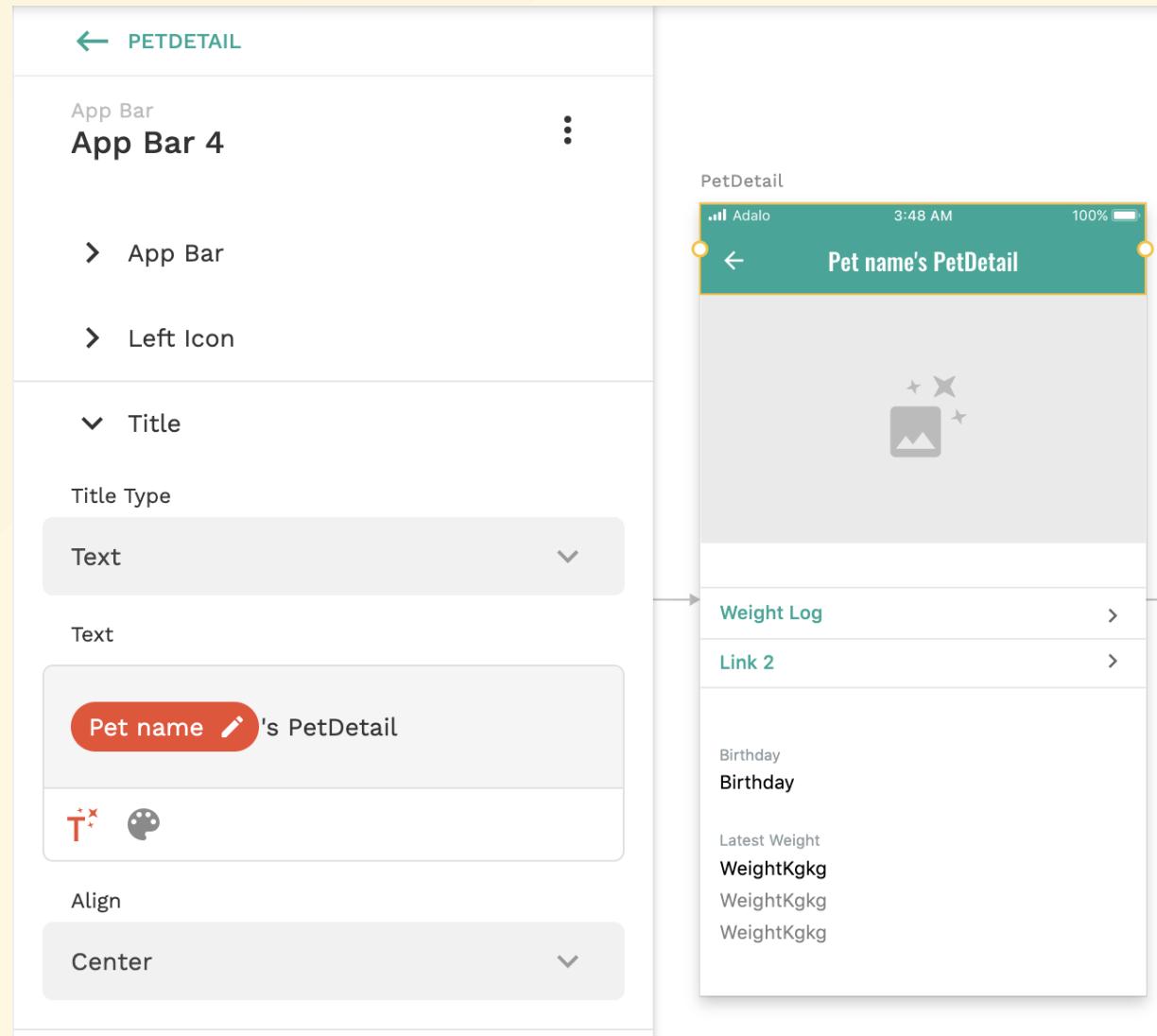
ペットの体重管理画面で選択したペットの体重をデータベースに登録できるようにします。

- ADDボタンを選択し、CreateのActionで一旦EmptyのままにしていたPetにCurrent Petを設定



ペット詳細画面、ペットの体重管理画面のヘッダーに選択したペットの名前を表示しましょう。それぞれの画面で、

- 画面上部のAppBarコンポーネントを選択し、TitleのTextに Current Pet's > Name を追加
- その後ろに、"'s PetDetail"という文字列を追加

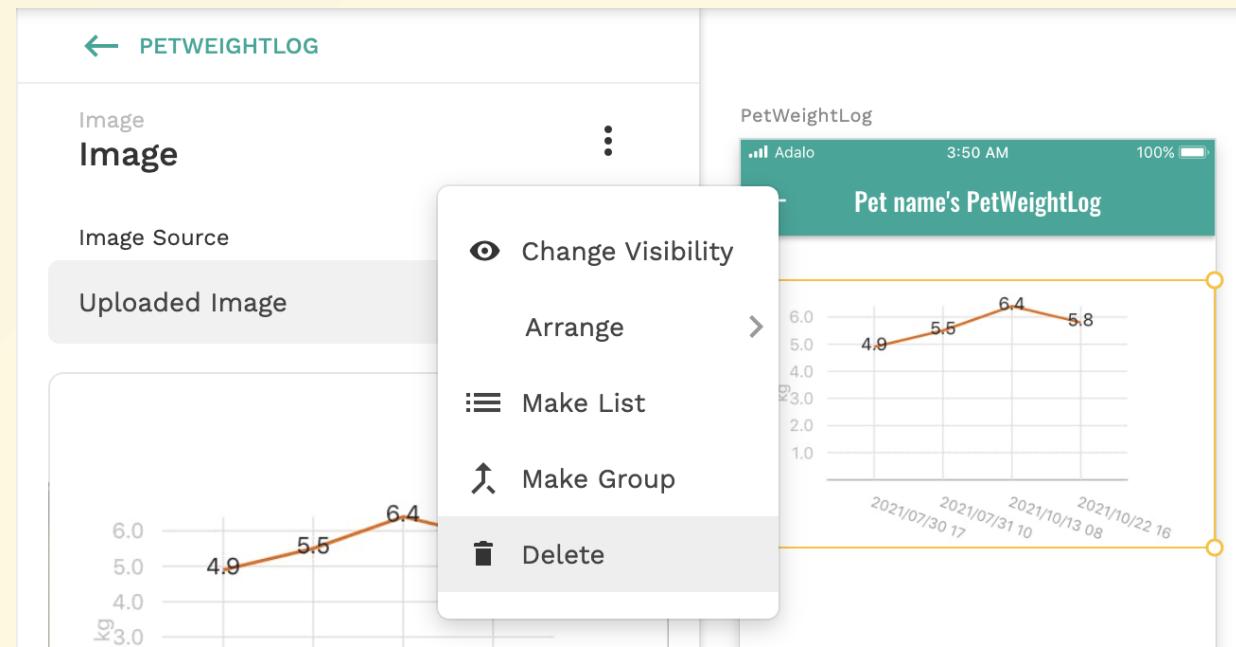


Preview機能で、以下を確認しましょう。

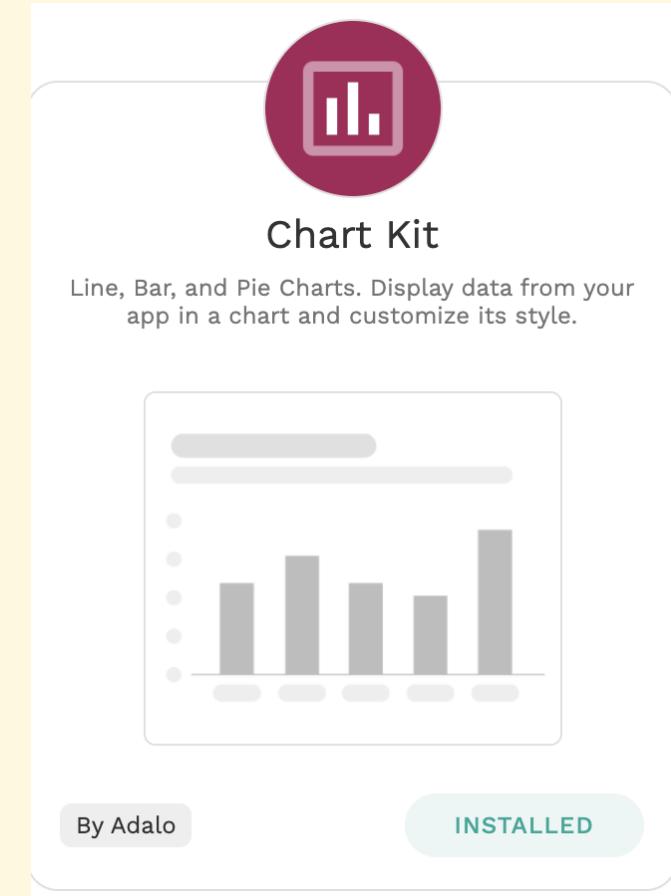
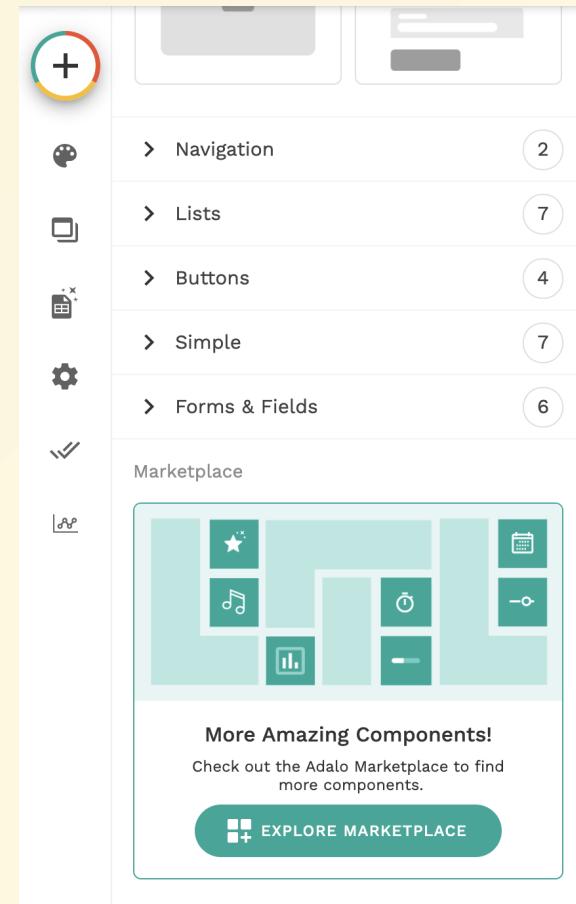
- ペット一覧画面から選択したペットの詳細画面に遷移できること
- ペット詳細画面から選択したペットの体重管理画面に遷移できること
- ペットの体重を登録すると、データベースのPetWeightLogs CollectionにRecordが登録されること
- ペット詳細画面に最新の体重が表示されること

次に、ペットの体重管理画面で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ます。

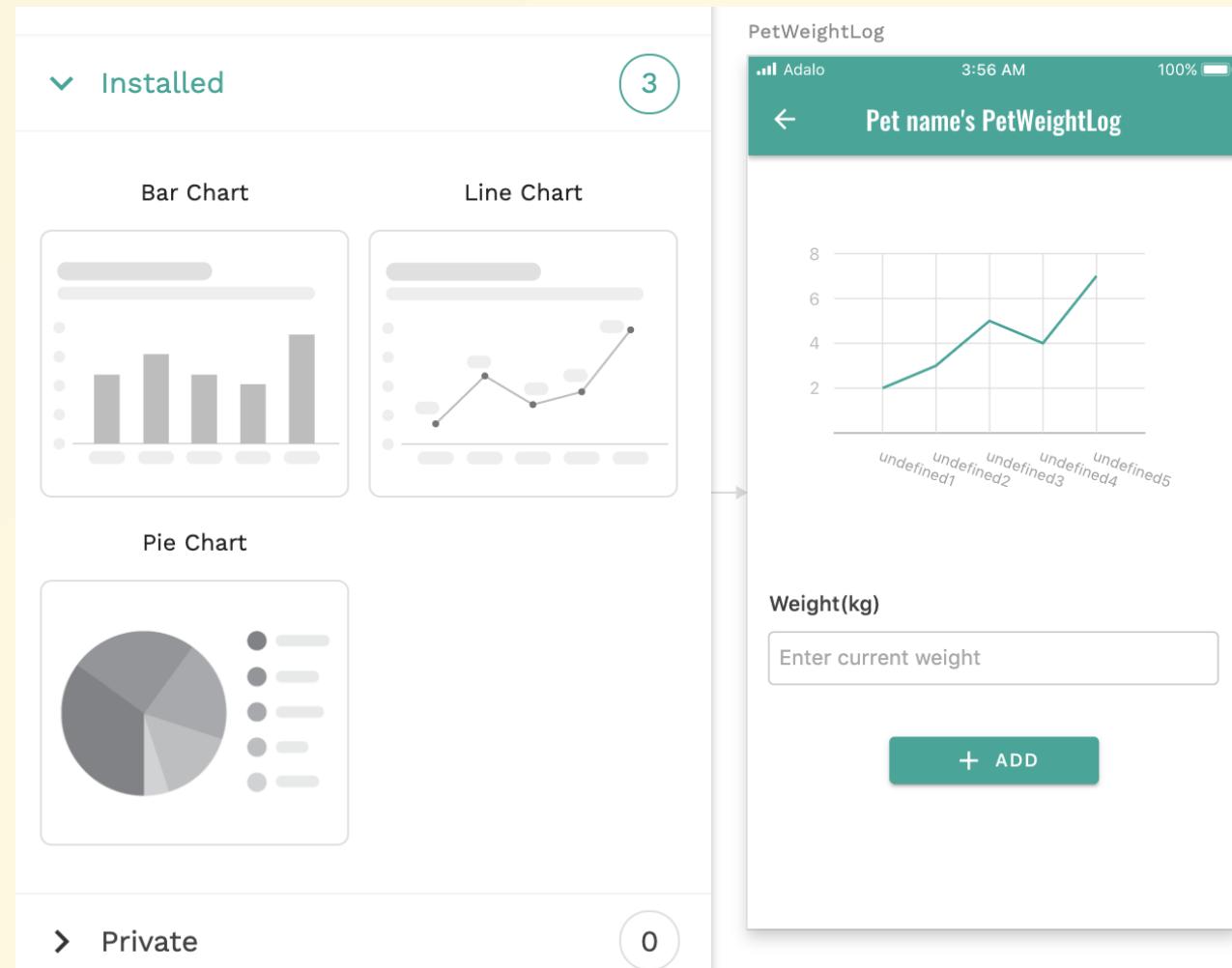
- まず、画像で貼り付けていたグラフを削除



- 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

The screenshot shows the configuration interface for a Line Chart. At the top, it says "Line Chart" and "Line Chart". Below that, there's a section titled "What is this a chart of?" with a dropdown menu showing "PetWeightLogs". Under "Filter", there's another dropdown menu showing "Current Pet > PetWeightLogs". A "Custom Filter" section is expanded, containing three dropdown menus: "WeightRegisteredTime" (set to "Is after"), "Is after" (set to "30 days ago"), and a date range selector. Below this, there's a "+ ADD ANOTHER FILTER" button. Under "Sorting", there's a dropdown menu showing "WeightRegisteredTime - Oldest to Ne...". At the bottom, there's a section for "Maximum number of items" with a dropdown menu set to "No Maximum". At the very bottom, there's a link "ADVANCED OPTIONS".

- 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 for configuring chart axes.

**Left Panel (X Axis Value):**

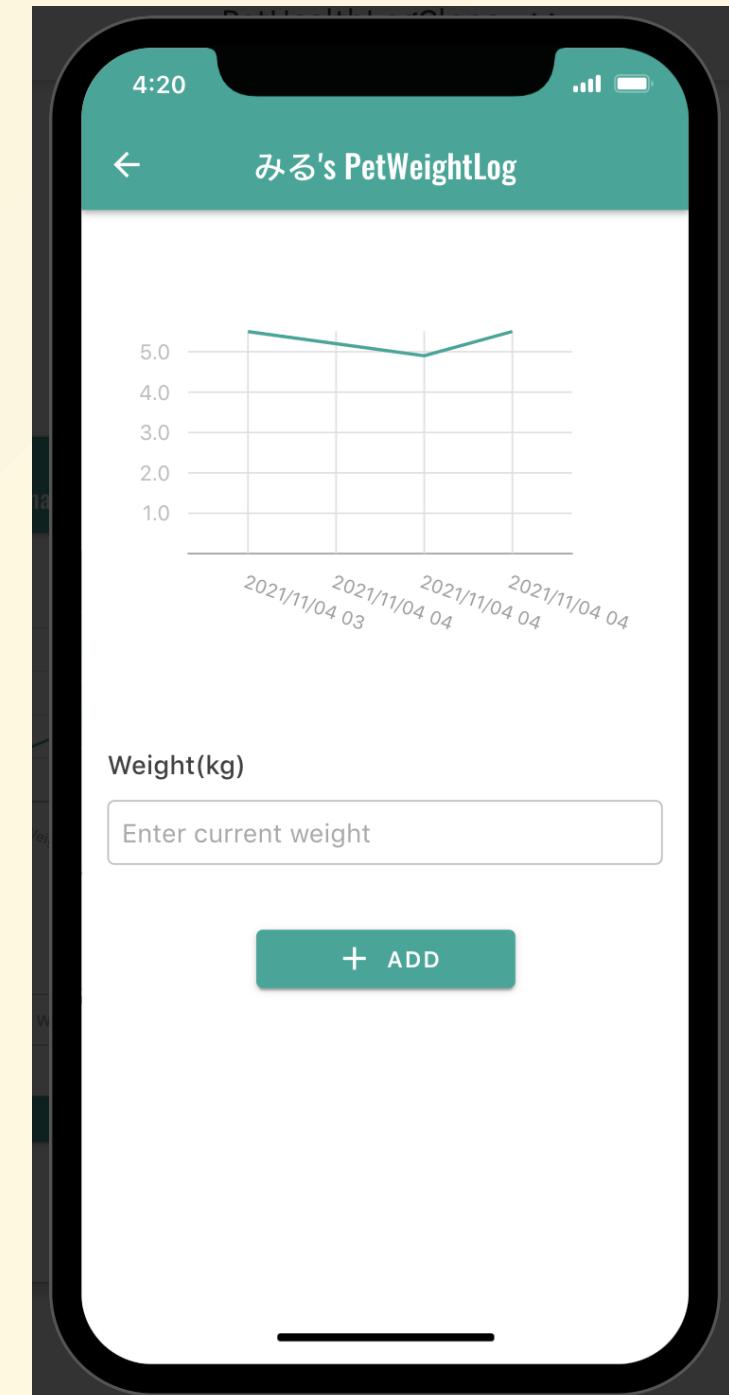
- X Axis Value:** Set to "PetWeightLog weightregisteredtime" (with edit icon).
- Y Axis Value:** Set to "PetWeightLog weightkg" (with edit icon).
- Point Action:** Set to "No Actions".
- Add Action:** A button labeled "+ ADD ACTION".
- Styles:** A section with a right-pointing arrow icon.

**Right Panel (Current PetWeightLog > WeightRegisteredTime):**

- X Axis Value:** Set to "PetWeightLog weightregisteredtime" (with edit icon).
- Date Format:** Set to "Date / Time".

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。

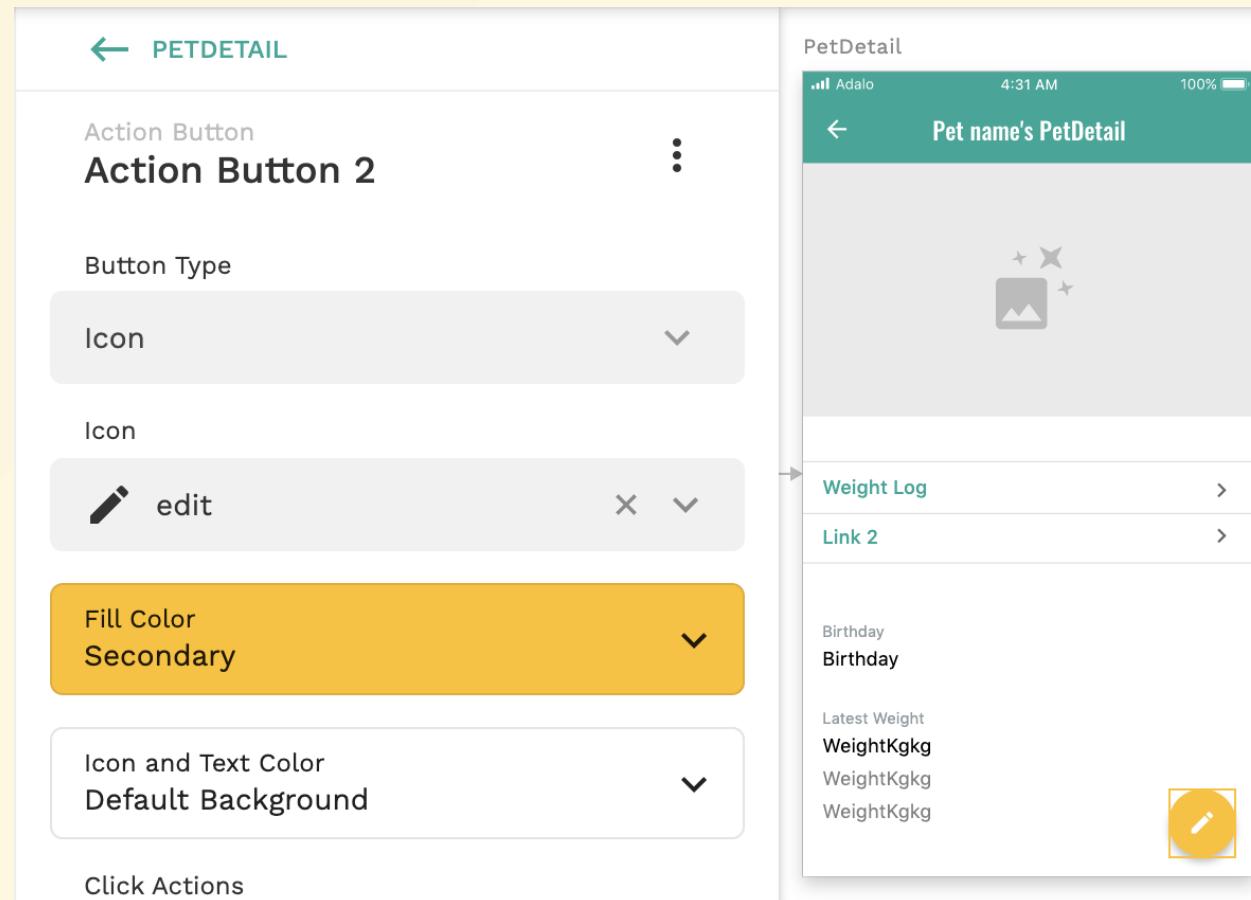


## データの更新(UPDATE)

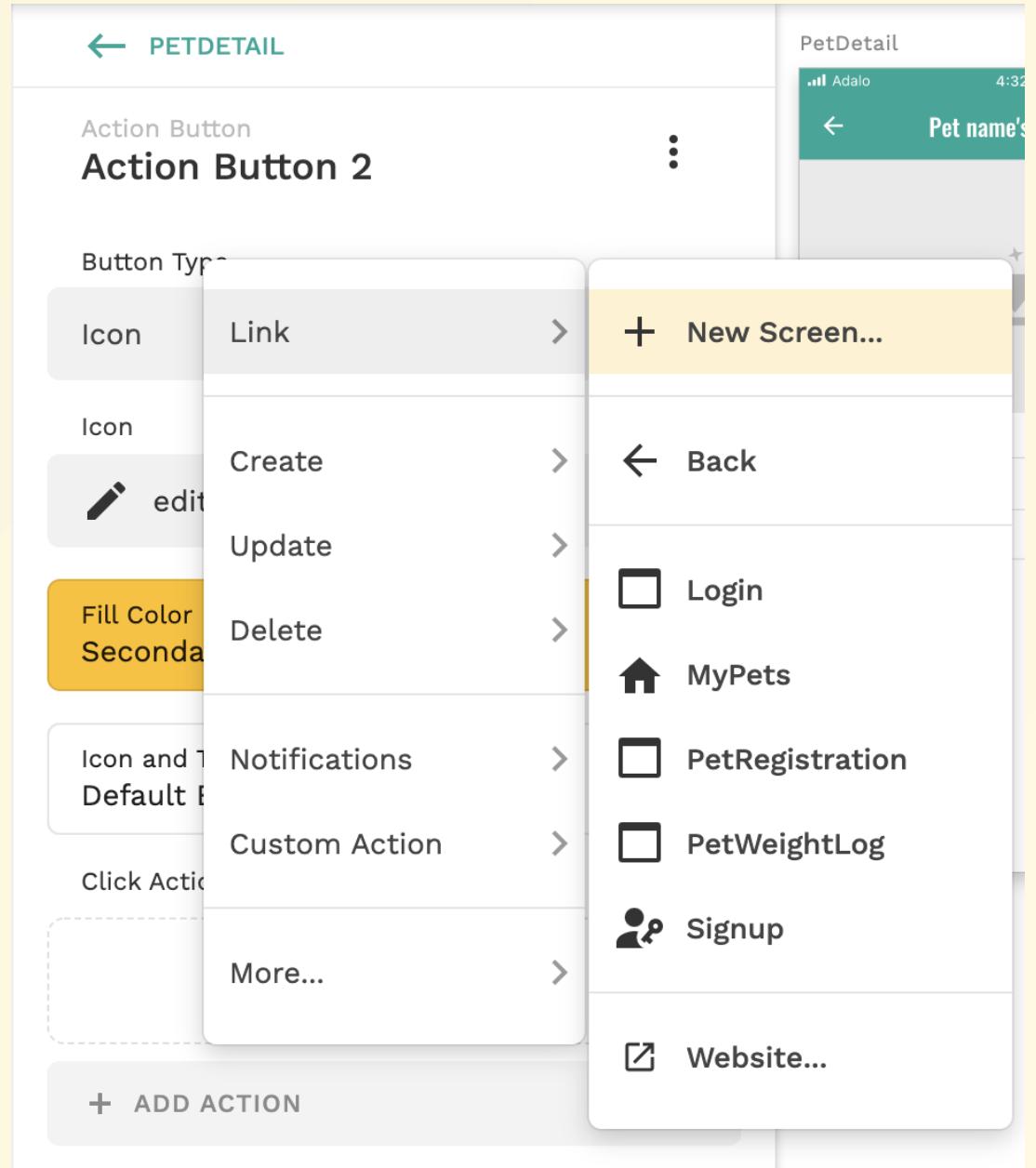
登録したペットの情報を更新できるペット情報編集画面を新たに作ります。

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。

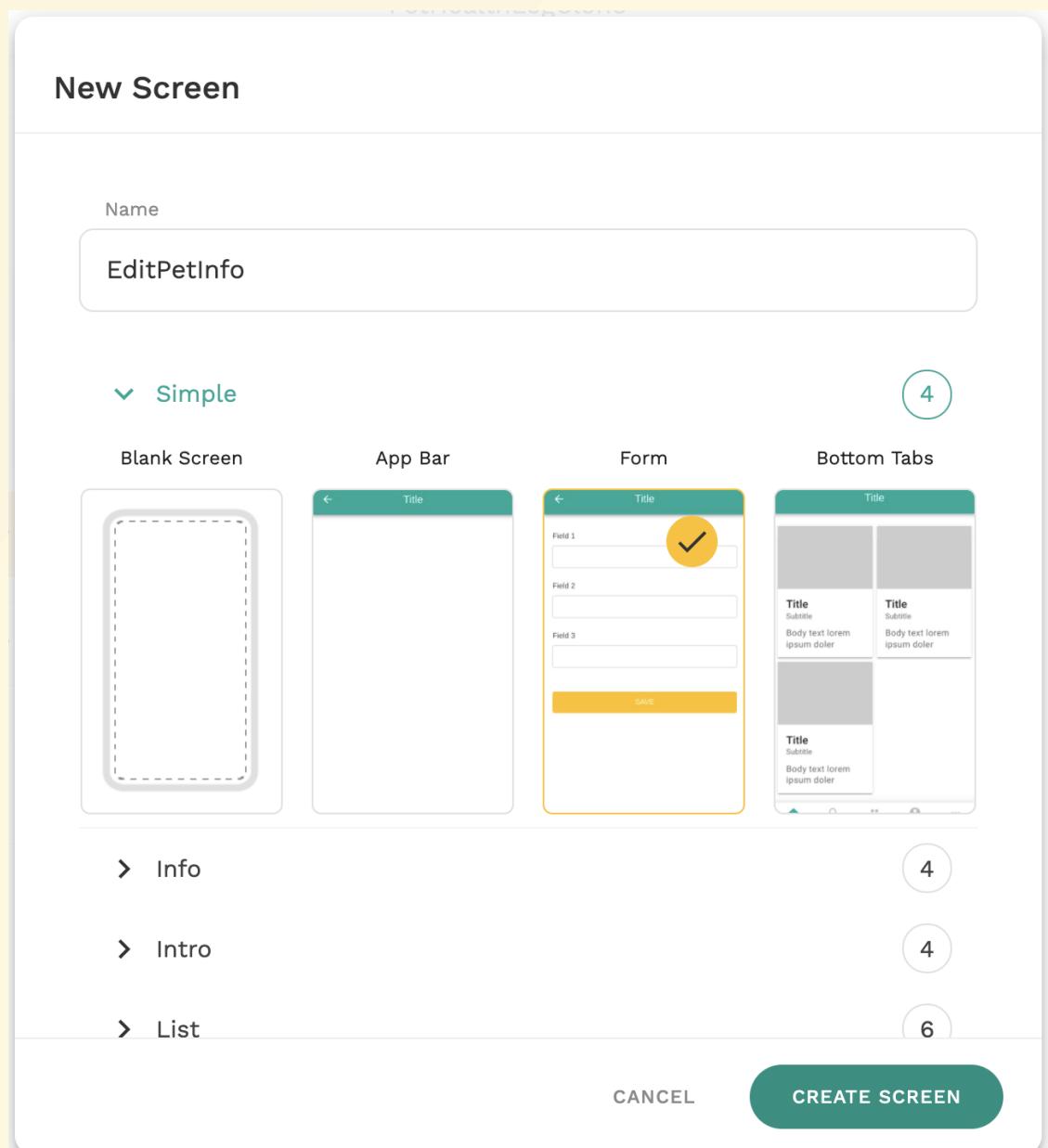
- ADD COMPONENTから Action Buttonを追加
- Iconをeditに変更
- Icon and Text Colorを Default Background(white) に変更



- ADD ACTIONからLink > New Screenを選択

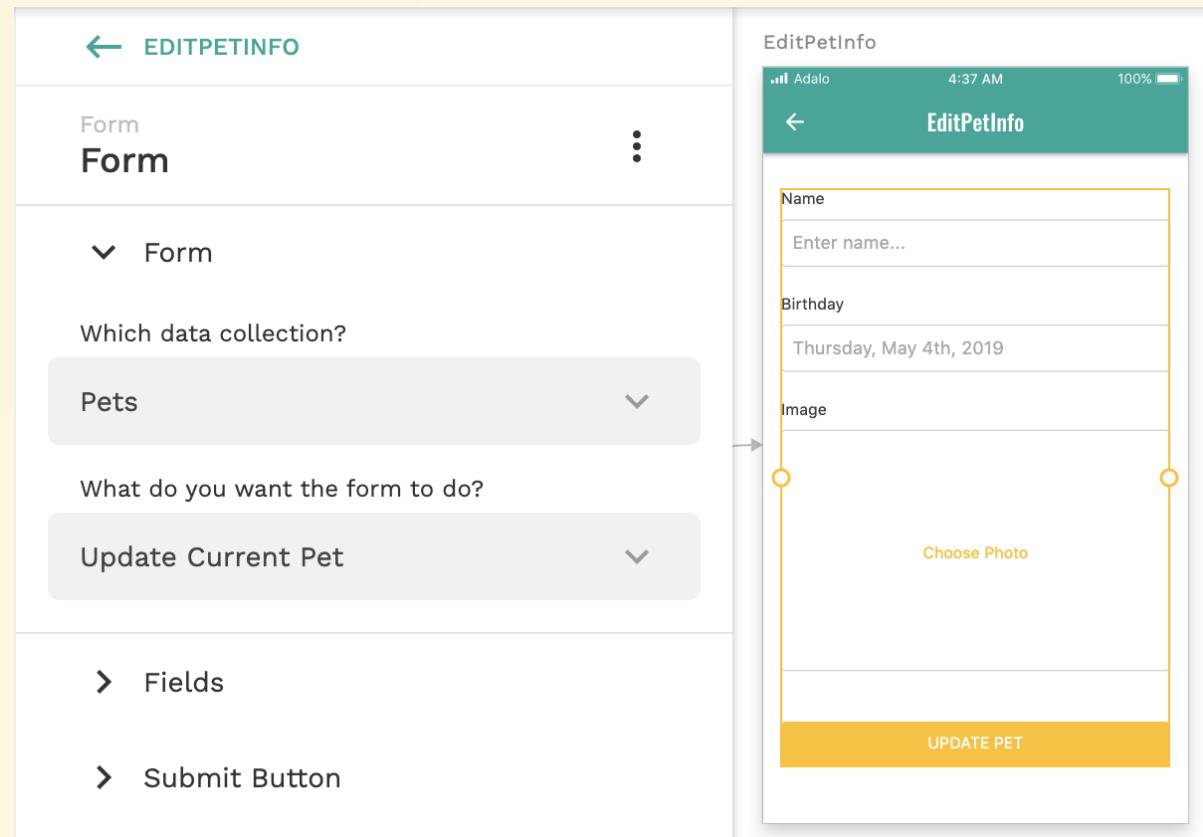


- Screen Nameを入力
- Formを選択
- CREATE SCREENをクリック

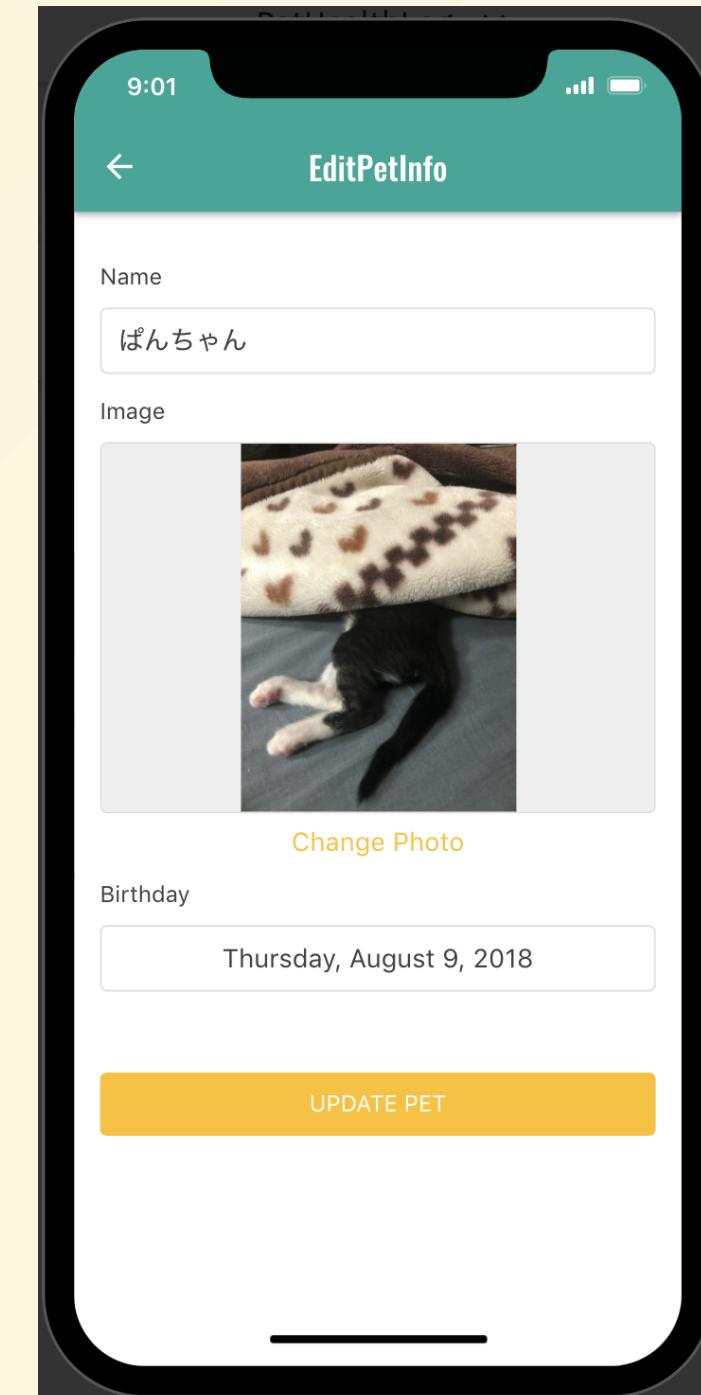


以下の設定をするだけでペット情報編集画面は完成です。

- Which data collection?で Petsを選択
  - 選択したCollectionに合わせたフォームを自動生成してくれる
- 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



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。



## 補足

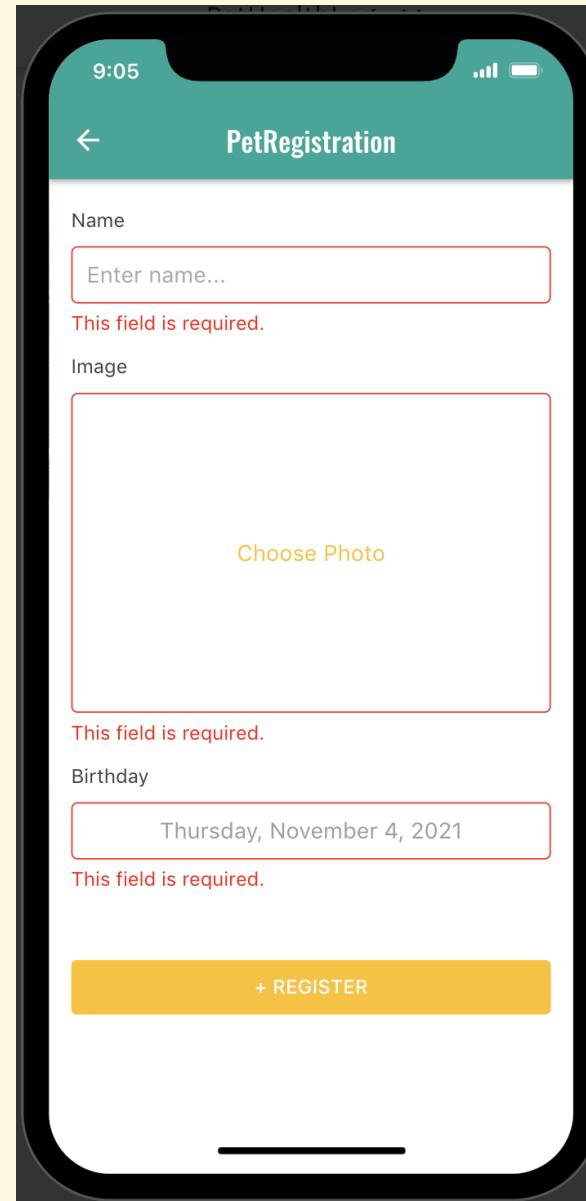
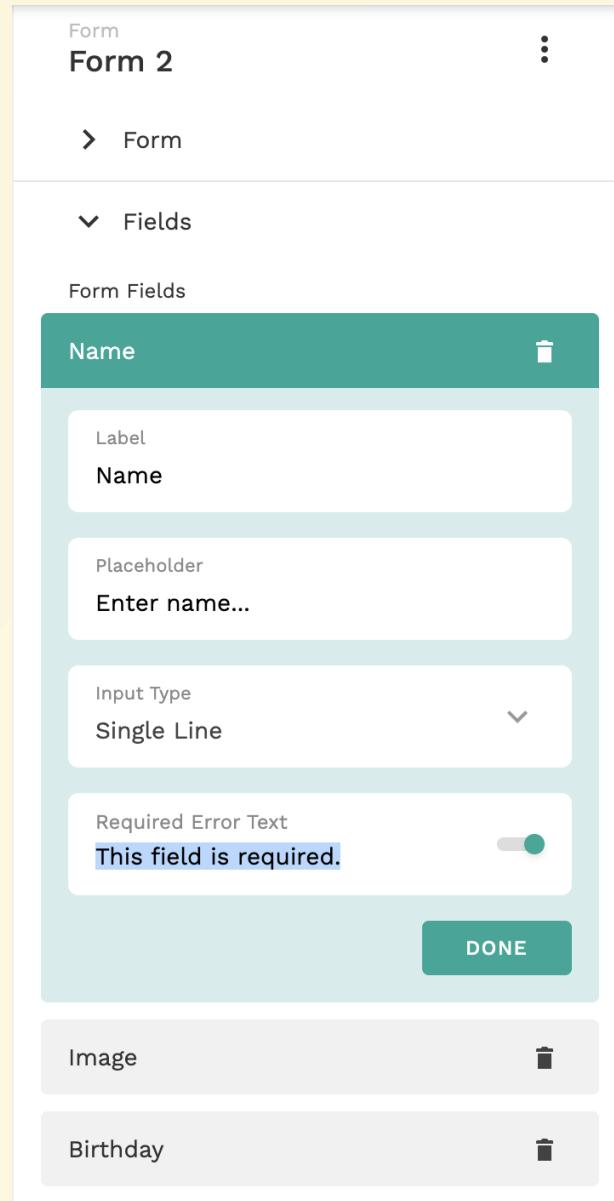
- 前回はペット登録画面の入力フォームをコンポーネントを一つずつ追加して作成しましたが、今回ペット編集画面を作ったのと同様の手順で自動生成できます。

## やってみよう

- ペット登録画面のAppBar以外のコンポーネントを削除して、Formコンポーネントを追加し、Pet Collectionを指定してフォームを自動生成してみましょう。
  - 登録後のペット一覧画面への遷移だけは、手動でSubmit ButtonにActionを追加する必要があるので、注意してください。

## 補足

- Formコンポーネントを使えば、FieldのRequired Error Textにチェックを入れるだけで、必須項目が入力されているかどうかをチェックすることができます。
- 入力フォームを作る際は、できるだけFormコンポーネントを使いましょう。



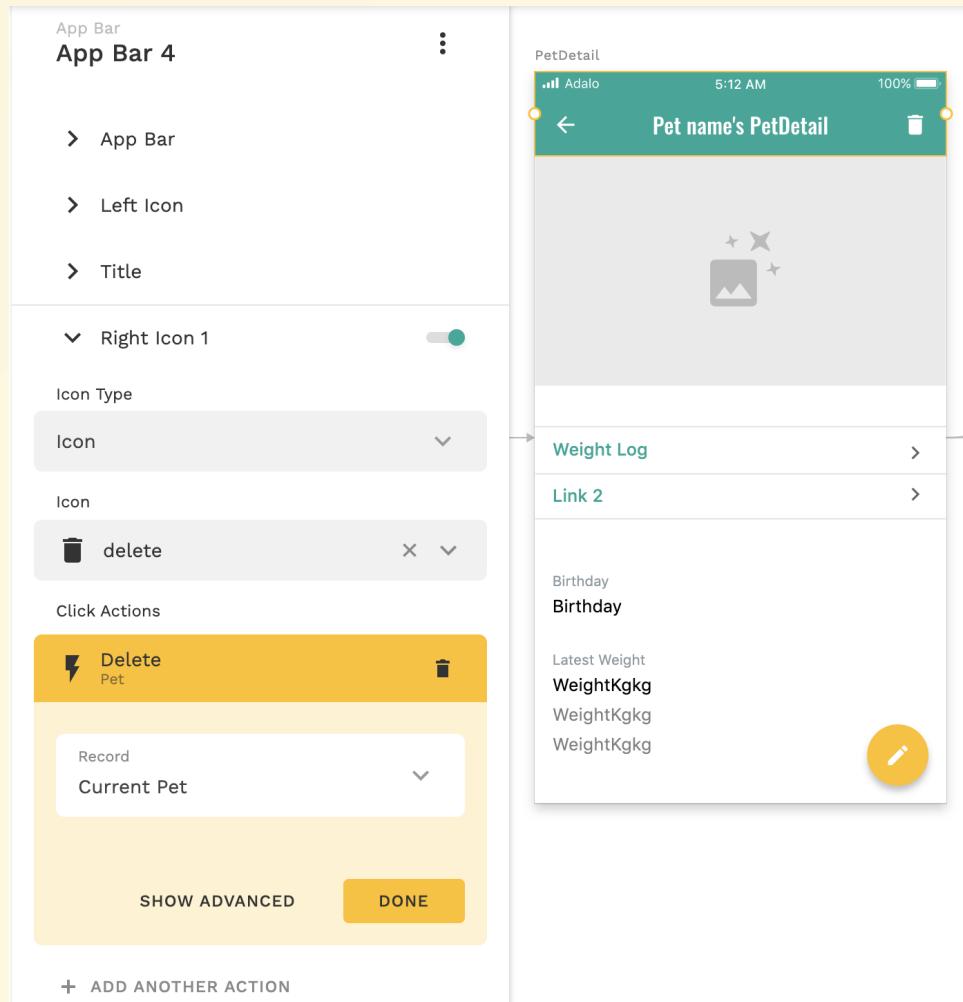
## 参考: 実際のAdaloでのアプリ開発の流れについて

- サンプルアプリでは、UI作成 > データベース設計 > それらの連携という流れで作成を進めてきました。
  - これは、どのような画面かが分からないと必要なデータも分からず、データベースの設計も難しいと考えたためです。
- 実際にAdaloでアプリ開発をする際は、まずUIスケッチ等を描いて必要なデータを洗い出し、その次にデータベースの設計を行うことをおすすめします。そうすることで、その後のUIの作成に自動生成を活用できます。
  - とはいっても、実際の開発は一方的な流れではなく、データベースとUIを交互に変更しながら試行錯誤することになると思います。

## データの削除(DELETE)

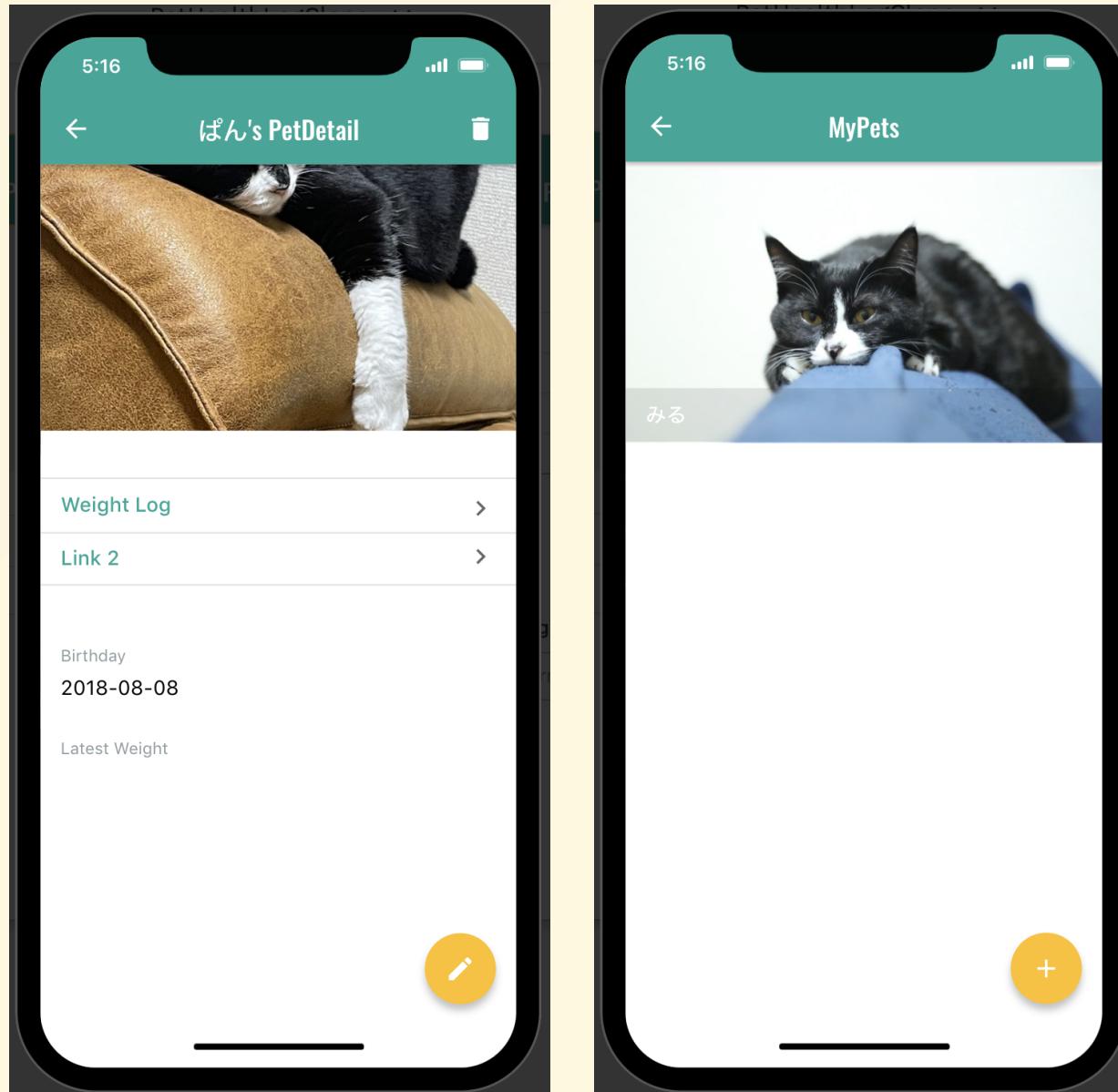
ペット詳細画面に、登録したペットを削除できるボタンを作ります。

- 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。



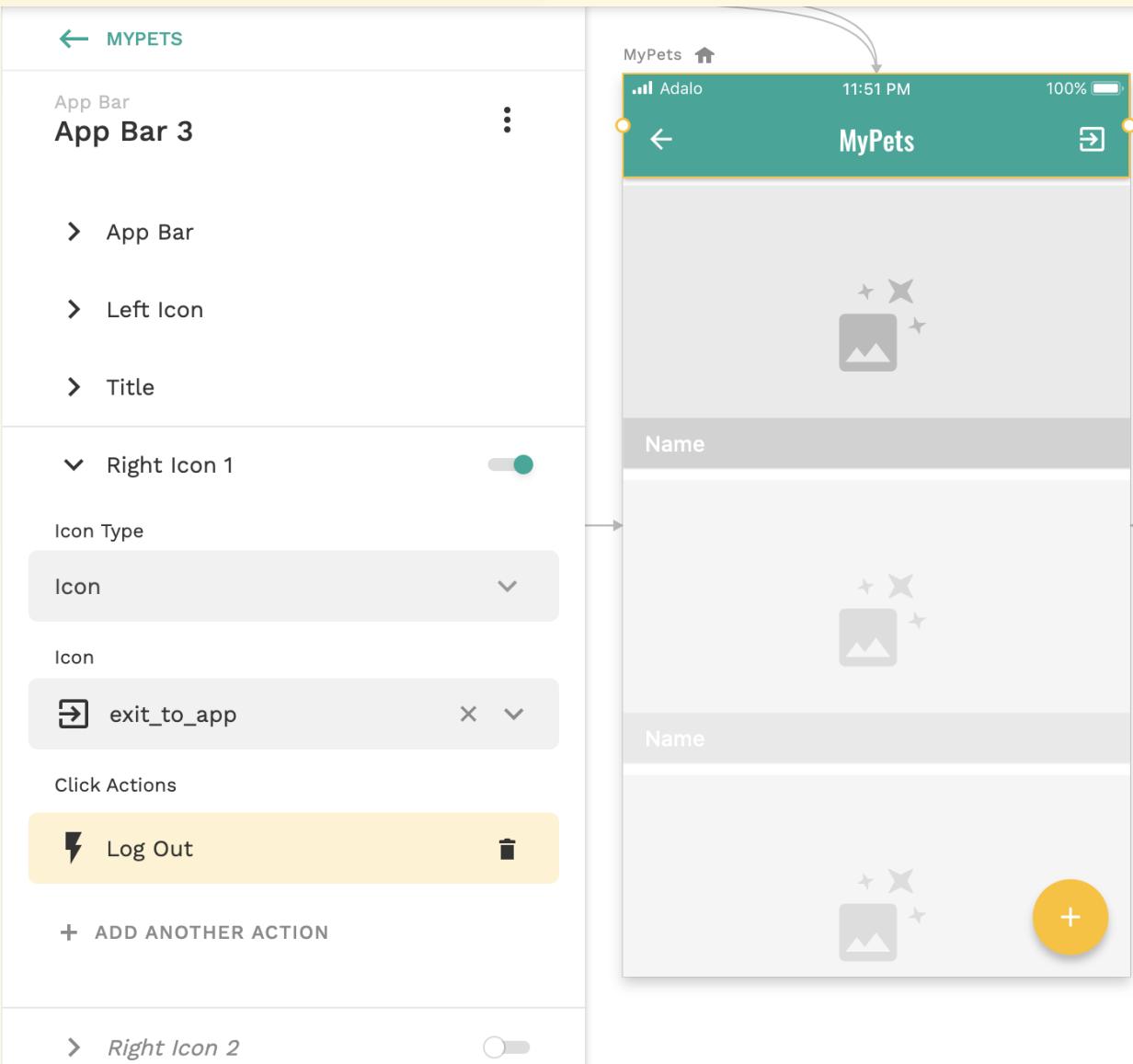
以上で、データベースに対するCRUD操作(CREATE, READ, UPDATE, DELETE)を全種類実装することができました。

# 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exit\_to\_appに変更する
- ・ADD ACTION > More... > User Login > Log Outを選択
- ・ADD ANOTHER ACTION > Link > Loginを選択



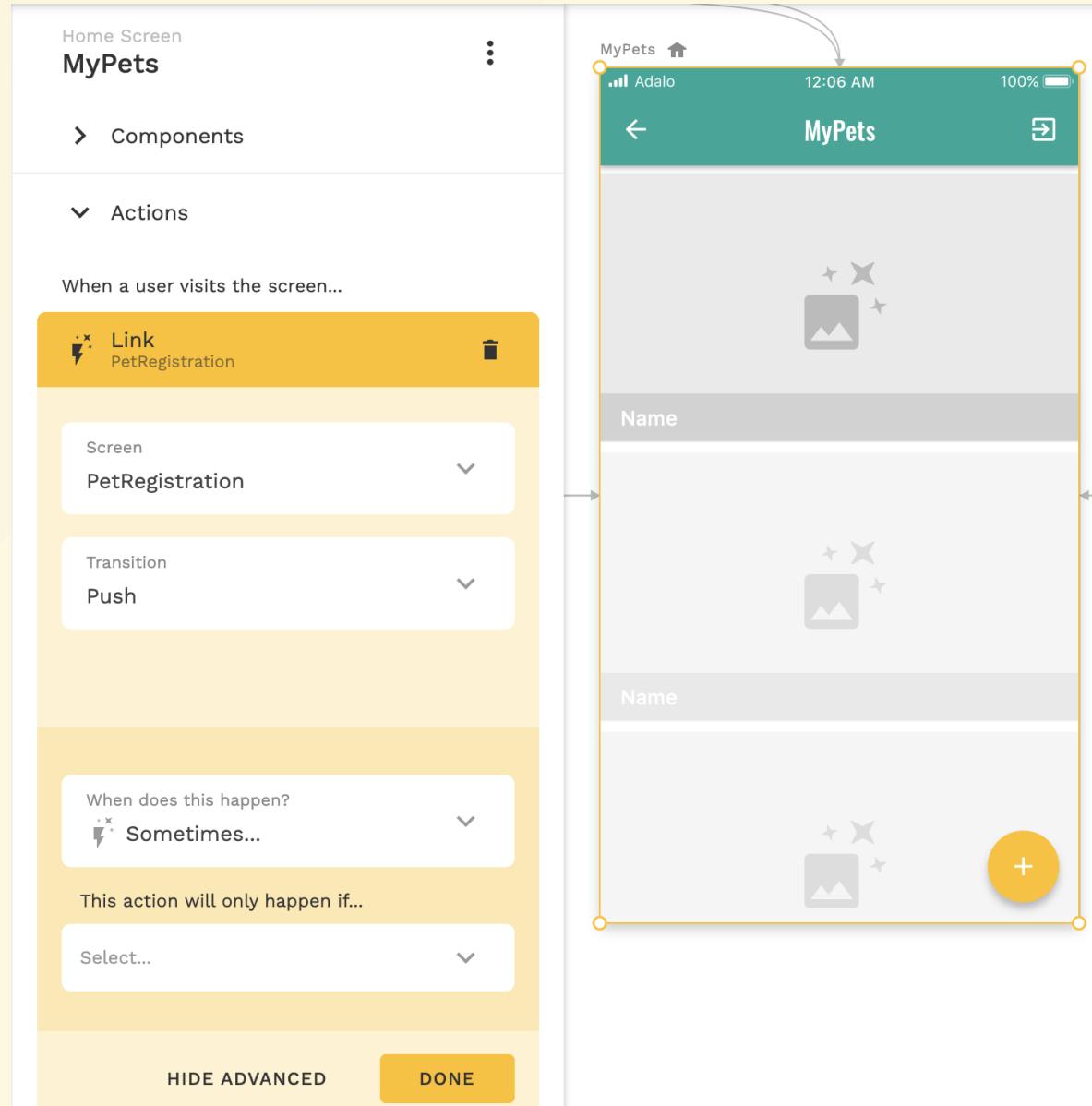
Preview機能で確認しましょう。

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。

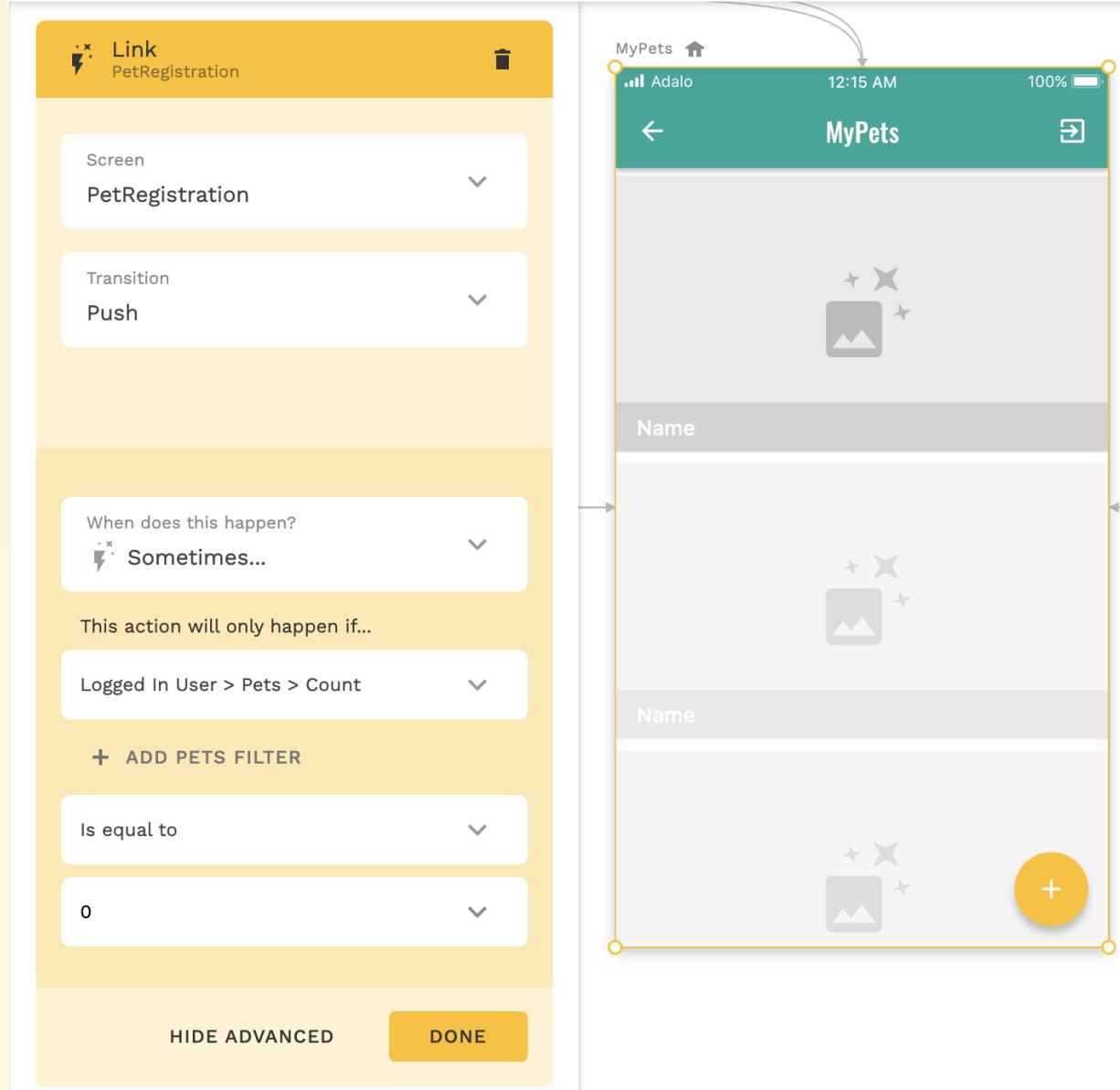
# Actionの実行条件設定

ペットが未登録の場合にペット一覧画面を表示したら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。

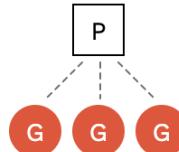
- データベースにGenders Collectionを追加(PropertyはデフォルトのままでOK)
- 0 Records > ADD GENDERとクリックし、MaleとFemaleの2つのRecordを追加

Genders				<a href="#">+ ADD GENDER</a>				
	Name	Created	Updated					
<input type="checkbox"/>	Female	8 minutes ago	8 minutes ago					
<input type="checkbox"/>	Male	8 minutes ago	8 minutes ago					

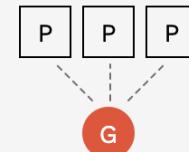
- Genders CollectionにPets Collectionとの1対多のRelationshipを追加する
  - 1匹のペットは1つの性別を持ち、1つの性別は複数のペットに対して設定されるため

How are Pets & Genders related?

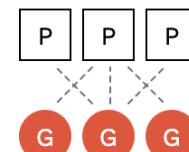
P = Pets    G = Genders



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

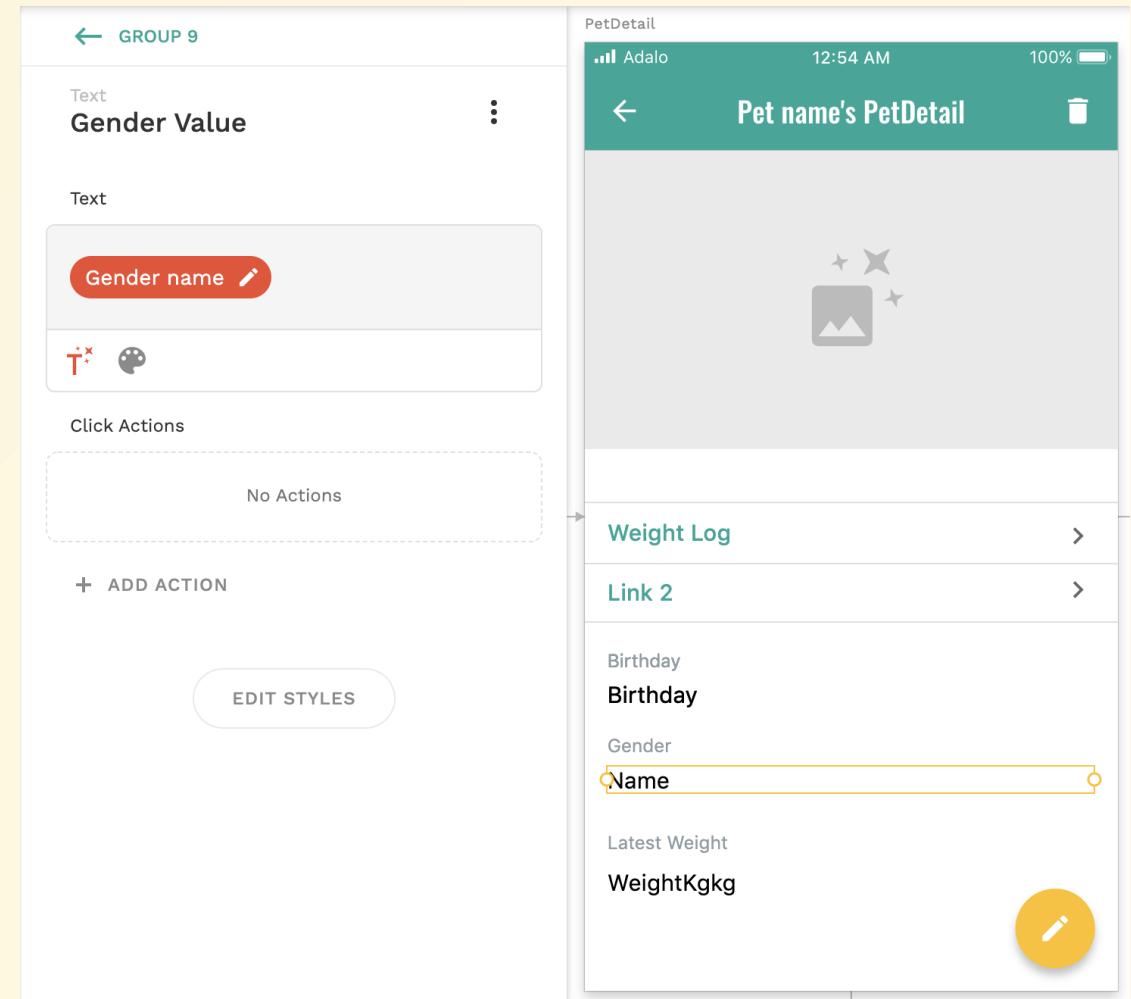
CANCEL

DONE

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

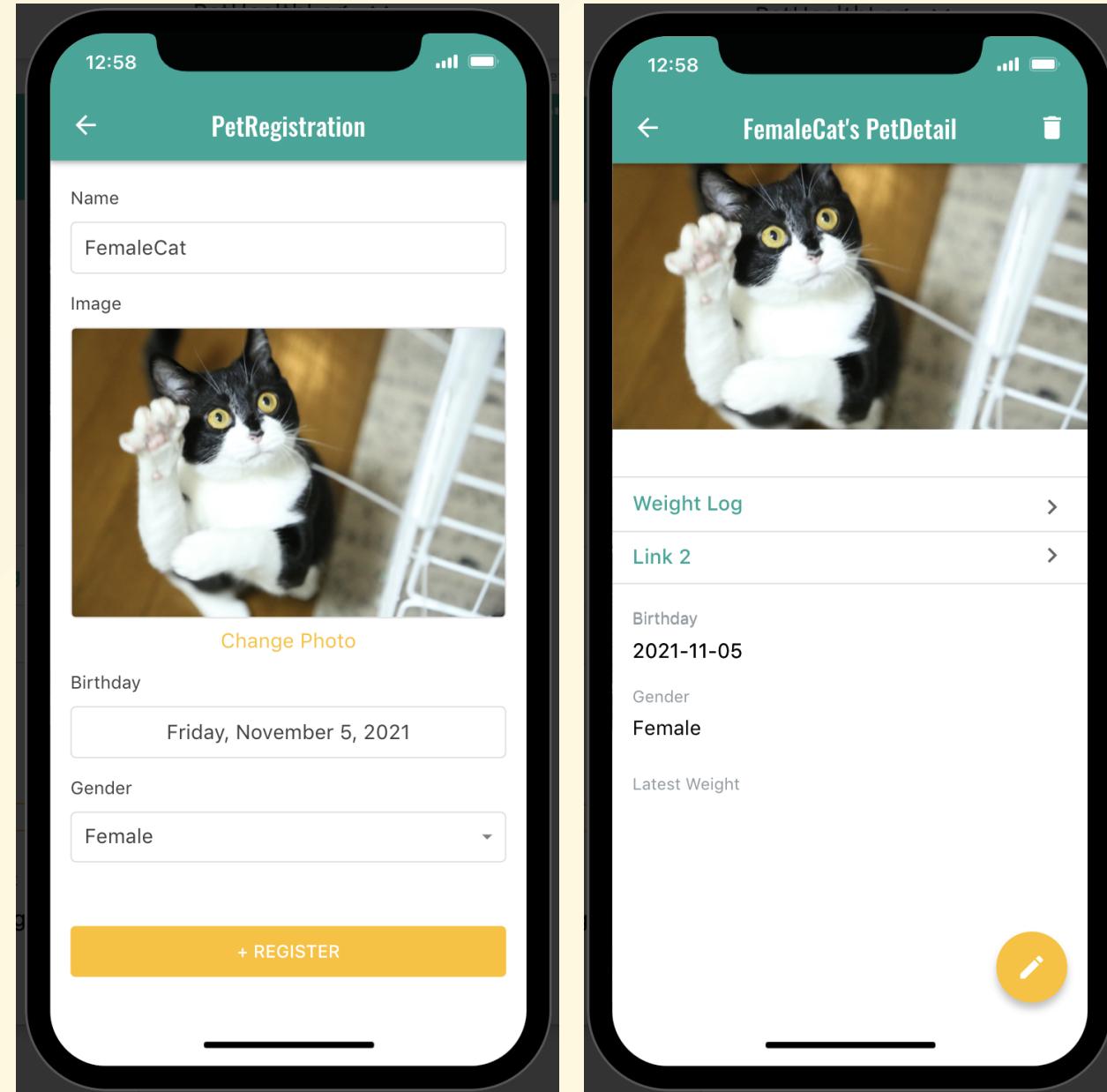
The image displays two side-by-side screens. On the left is a configuration interface for a 'Form Fields' section. It lists four fields: 'Name', 'Image', 'Birthday', and 'Gender'. The 'Gender' field is highlighted with a teal background. Below it, settings for the 'Gender' field include 'Label' (set to 'Gender'), 'Placeholder' (set to 'Select gender...'), a 'Sorting' dropdown (set to 'Select...'), and an error message 'This field is required.' A 'DONE' button is at the bottom right of this panel. At the bottom left is a '+ ADD VISIBLE FIELD' button. On the right is a screenshot of a mobile application titled 'PetRegistration'. The screen shows a form with fields for 'Name' (placeholder 'Enter name...'), 'Image' (button 'Choose Photo'), 'Birthday' (value 'Thursday, May 4th, 2019'), and 'Gender' (placeholder 'Select gender...'). A large orange button at the bottom right is labeled '+ REGISTER'. The top status bar of the app shows 'Adalo' and '12:37 AM'.

- ペット詳細画面に性別の表示欄を追加



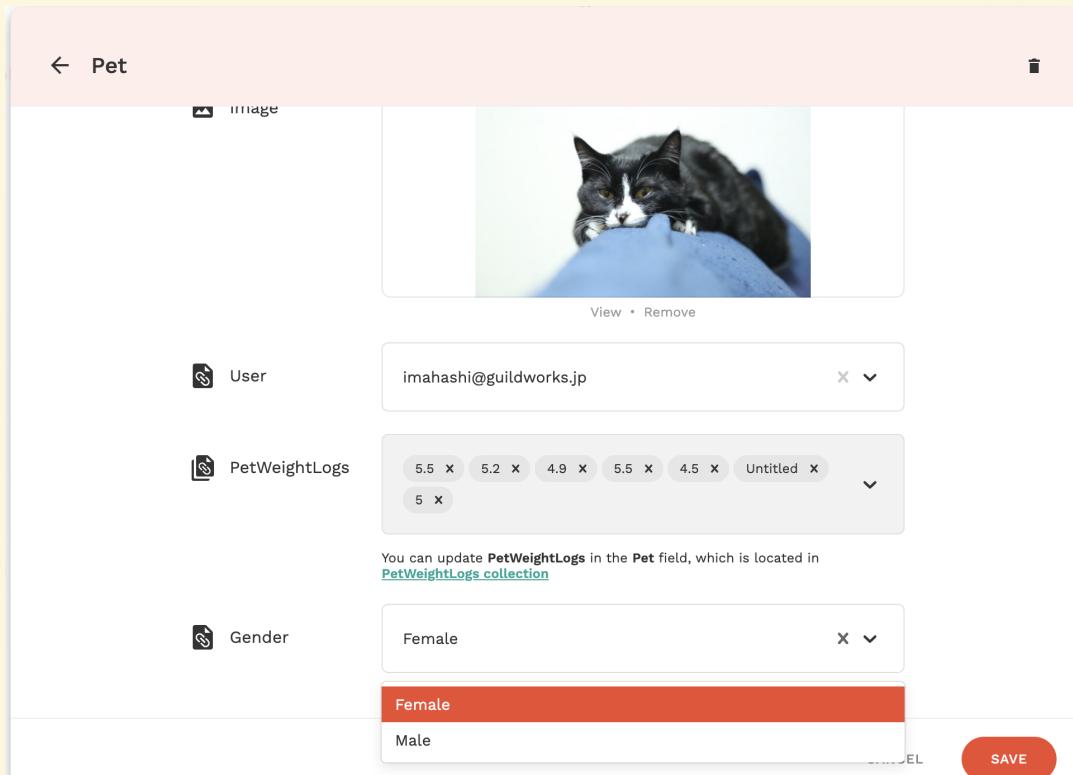
Preview機能で確認しましょう。

- ・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。

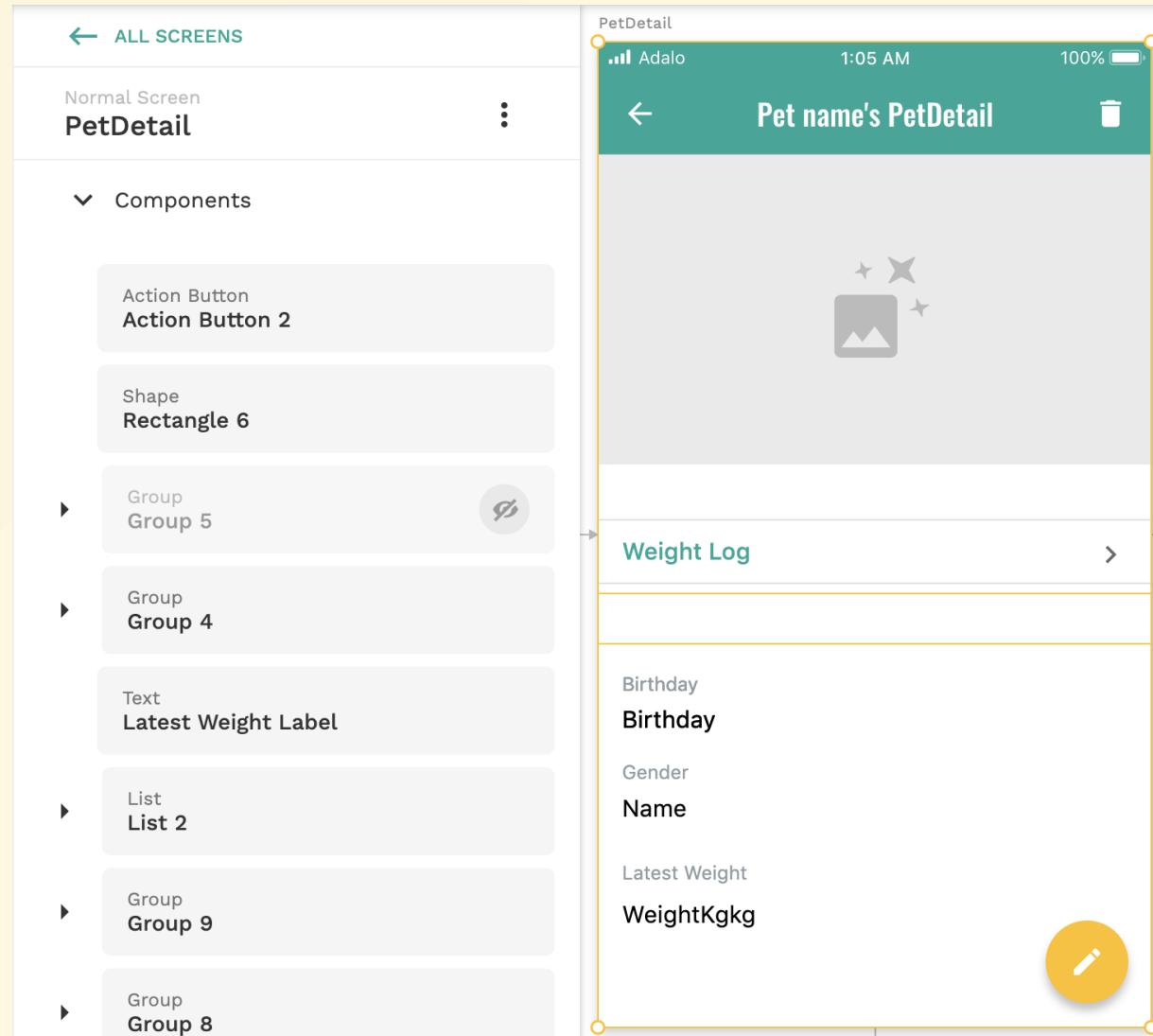
The screenshot shows the MarketPlace interface with the following details:

- Marketplace** logo and title.
- Multiselect Dropdown** icon (represented by a dropdown menu with multiple horizontal bars).
- Description: "Select multiple items from a dynamically-generated list."
- Preview: A visual representation of the component showing a search bar and a list of items with checkboxes, some of which are checked.
- Author: By Conroy Whitney (with a link icon).
- Status: INSTALLED (in a green button).

# コンポーネントの非表示設定

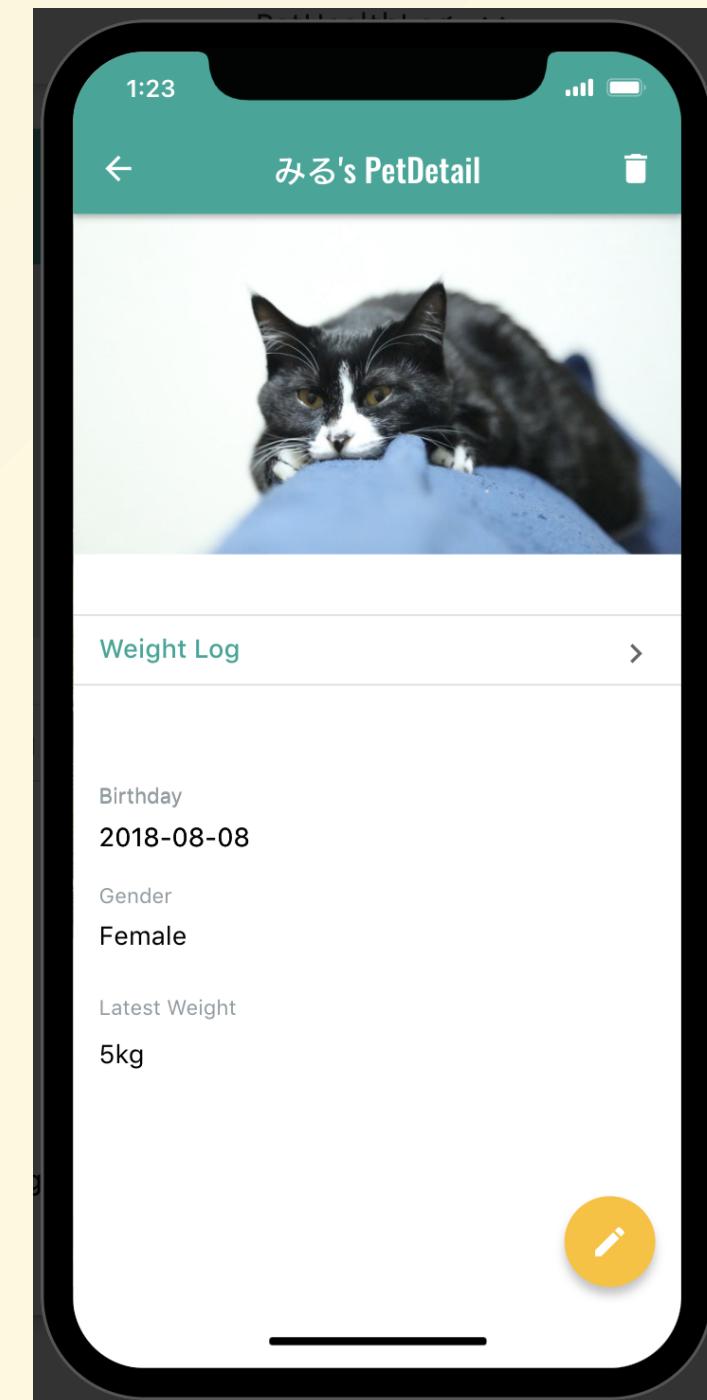
表示しないコンポーネントを削除せずに残しておくことができます。

- ペット詳細画面のコンポーネント一覧で、Link 2を含むGroupにマウスオーバーして、右側の目のアイコンをクリック



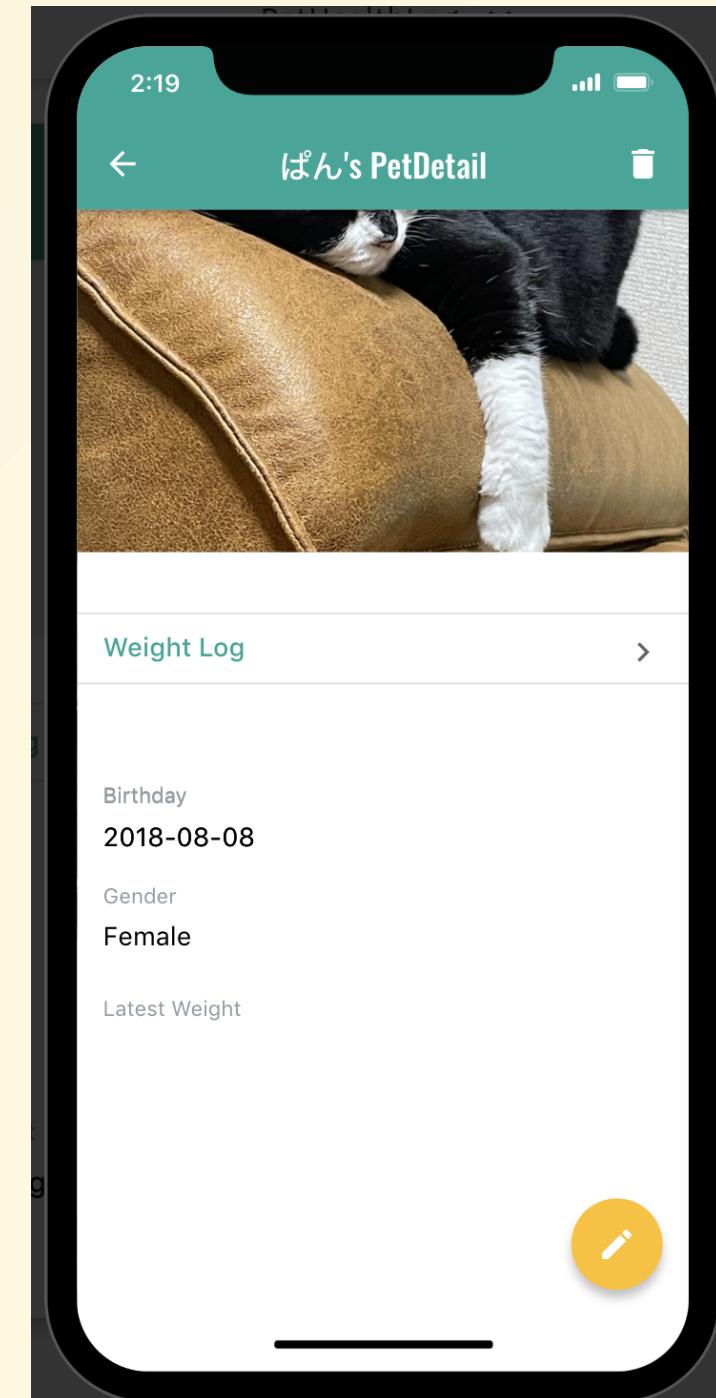
Preview機能で確認すると、  
Link 2が消えています。

もう一度目のアイコンをクリ  
ックすれば、再度表示させら  
れます。

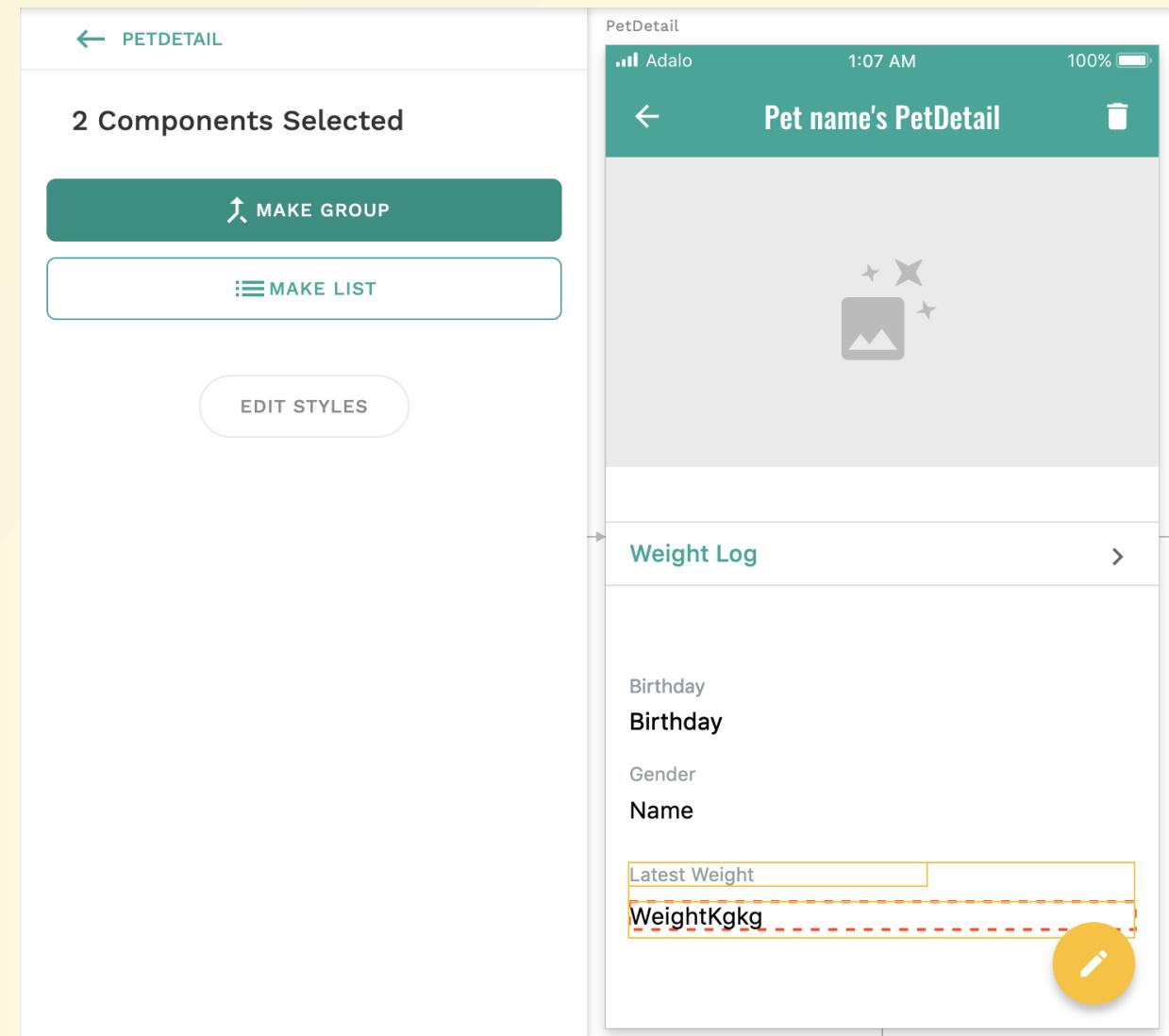


# 条件によってコンポーネントの表示or非表示を切り替える

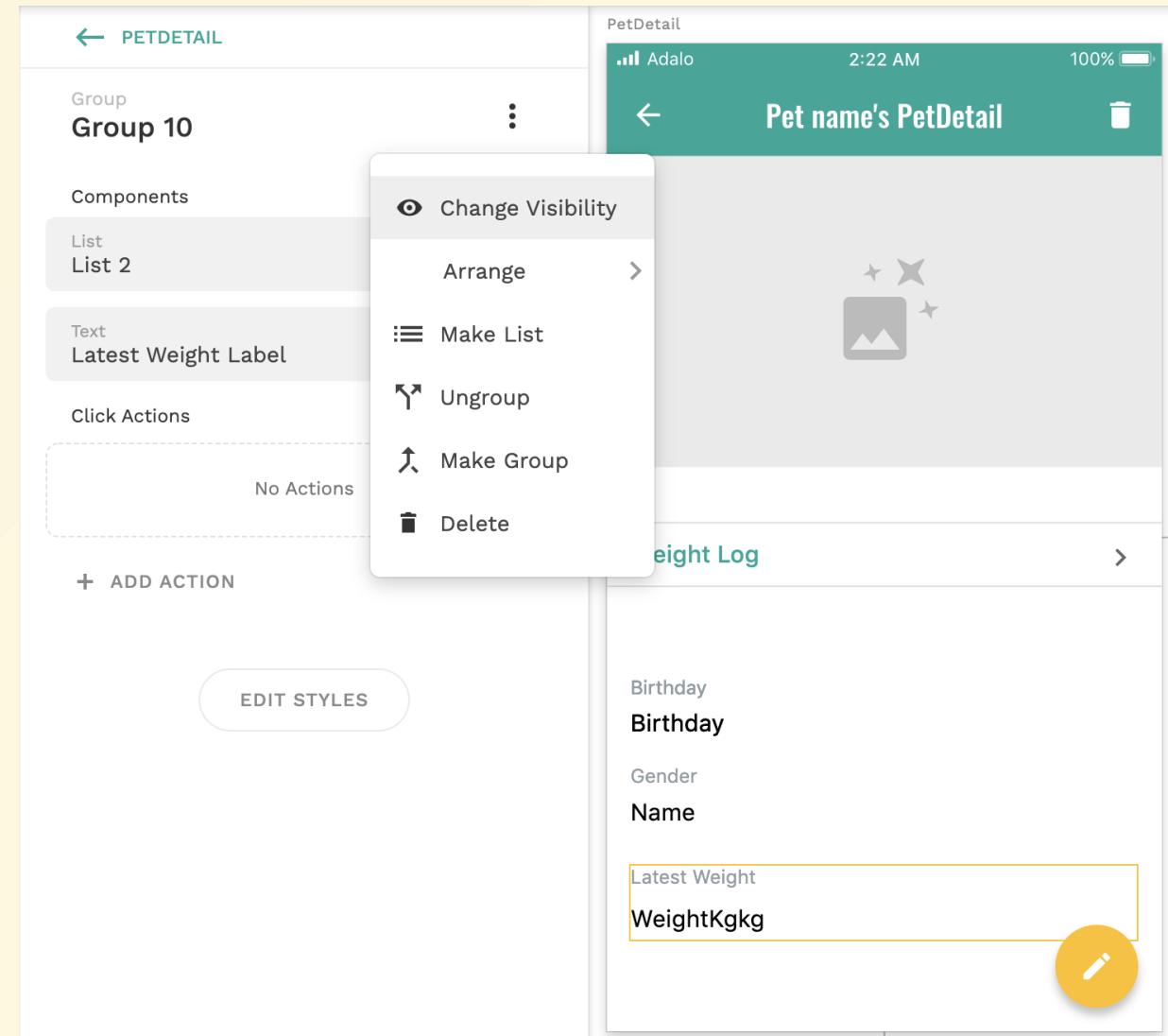
体重が未登録であれば、ペット詳細画面のLatest Weightは非表示にします。



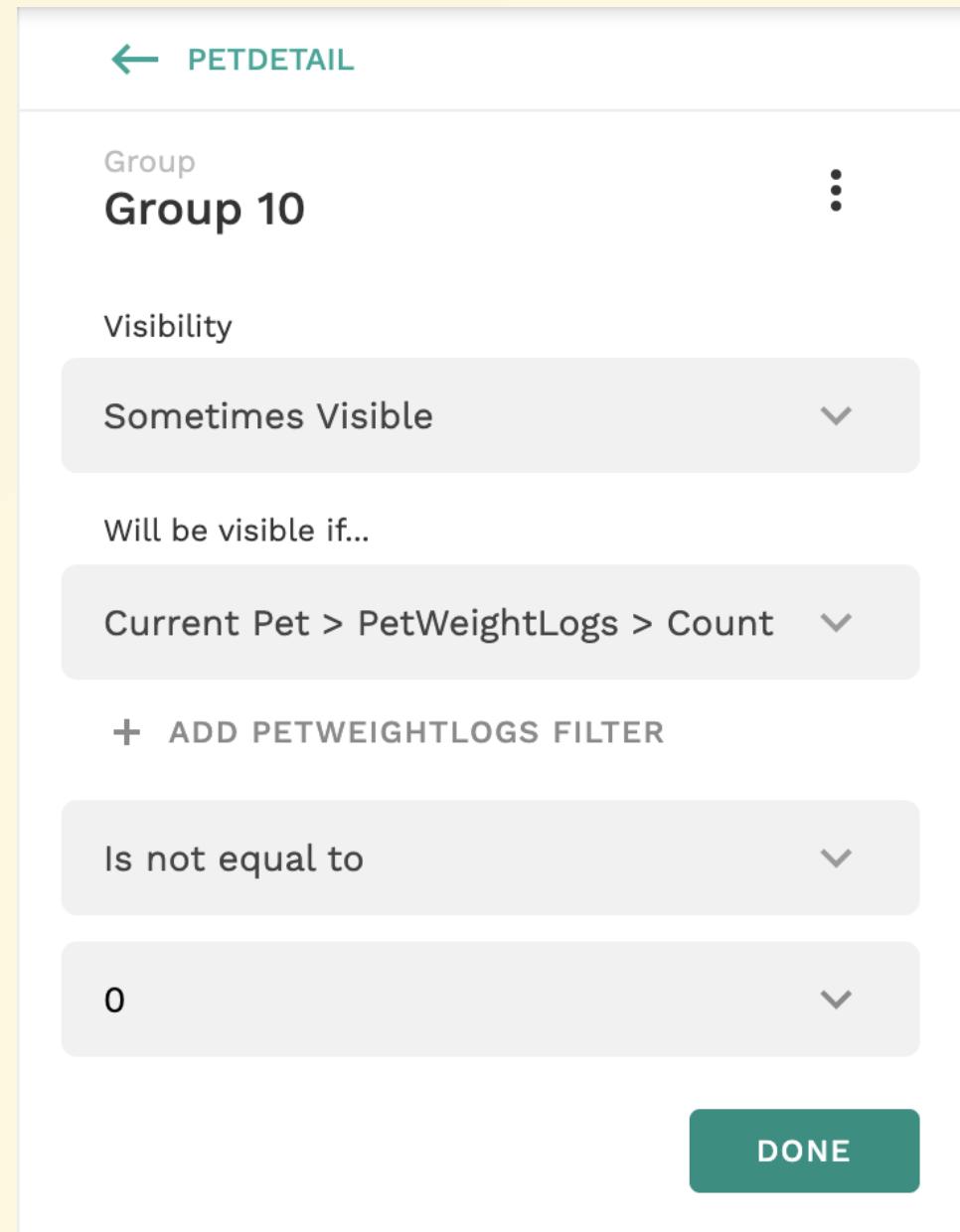
- Latest Weightというラベルとその値を選択し、グループ化します



- Change Visibilityを選択します

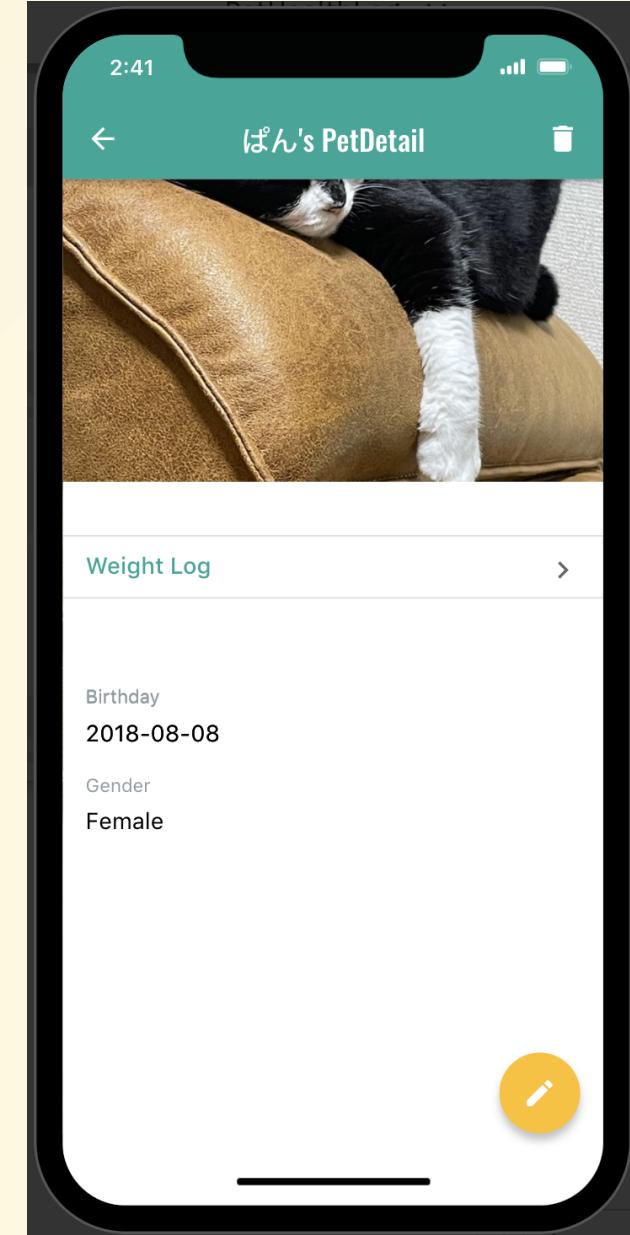


- VisibilityをSometimes Visibleに変更
- Will be visible if...でCurrent Pet > PetWeightLogs > Countを選択
- Is not equal to 0と設定



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。

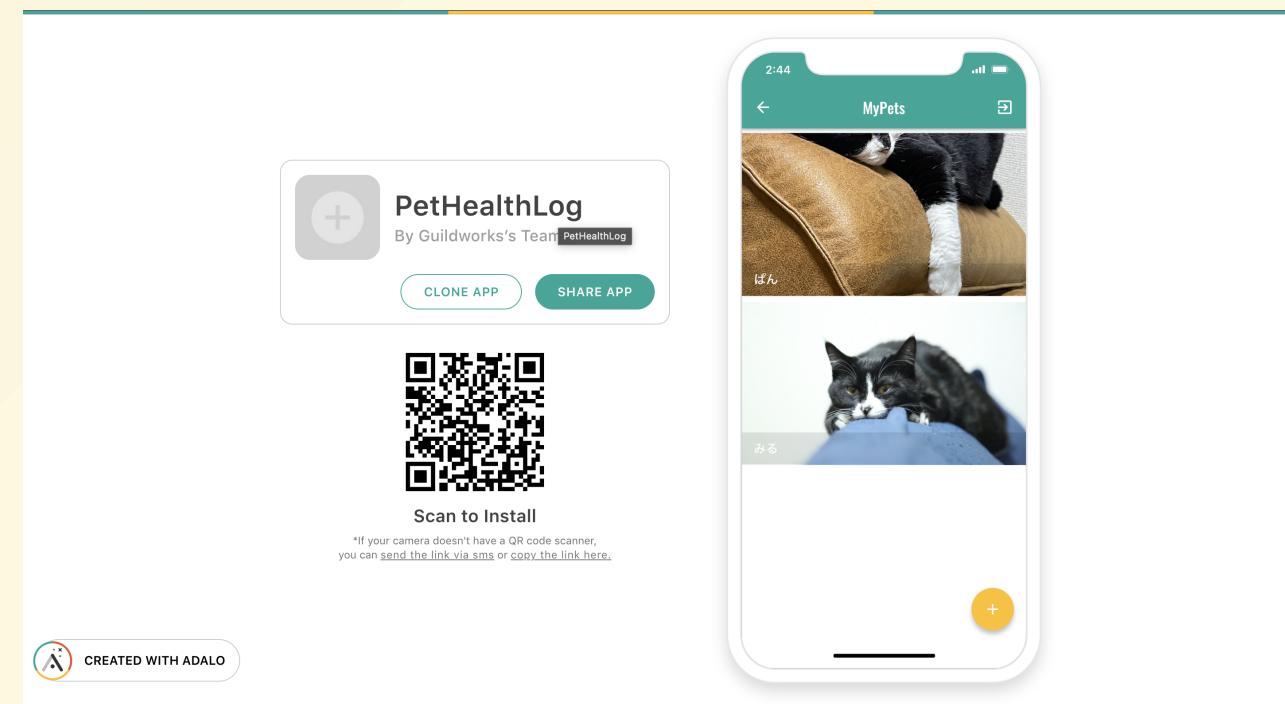
サンプルアプリの改善は一旦ここまでとします。



# クローン用URL

- 以下のURLからここまで  
の作業を反映したアプリをク  
ローンできますので、答  
え合わせに使ってください。

<https://previewer.adalo.com/f1324ea8-ec47-4c22-a3a9-3258044eb754>



# 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。

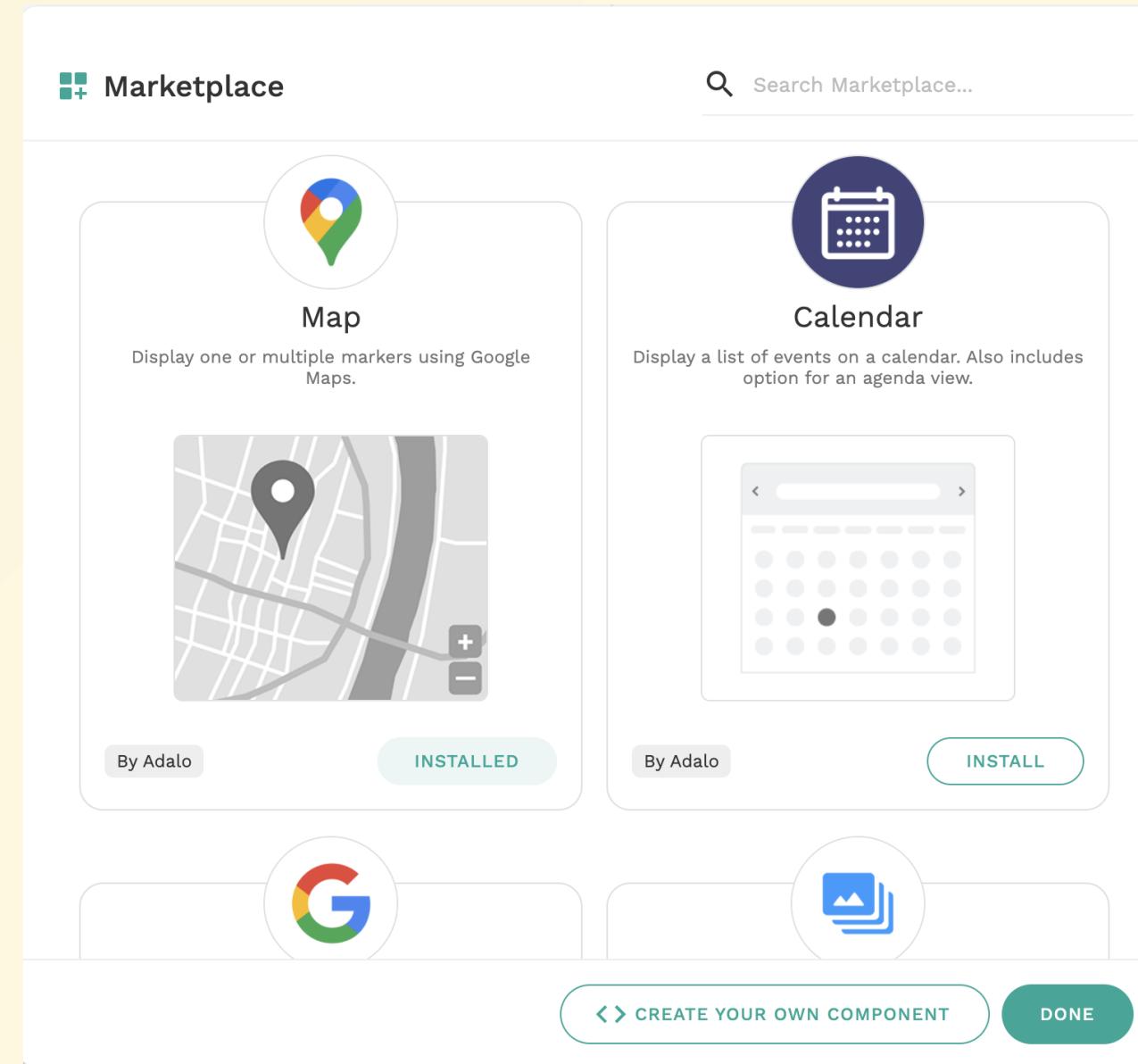
その方法をご紹介します。

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

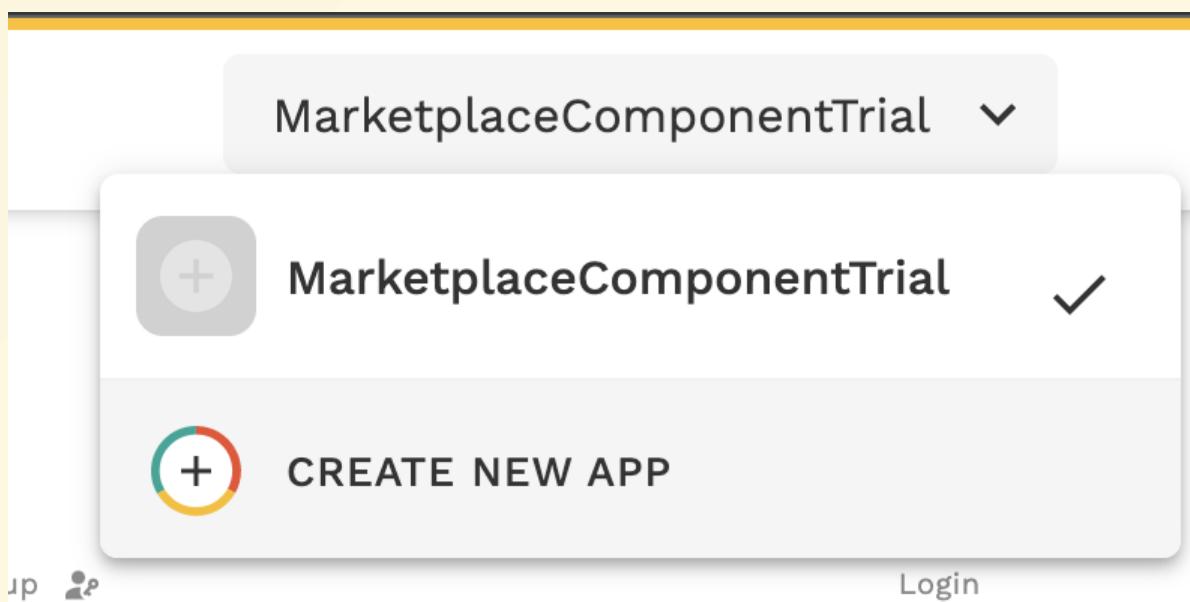
# Marketplaceの外部連携コンポーネント

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。

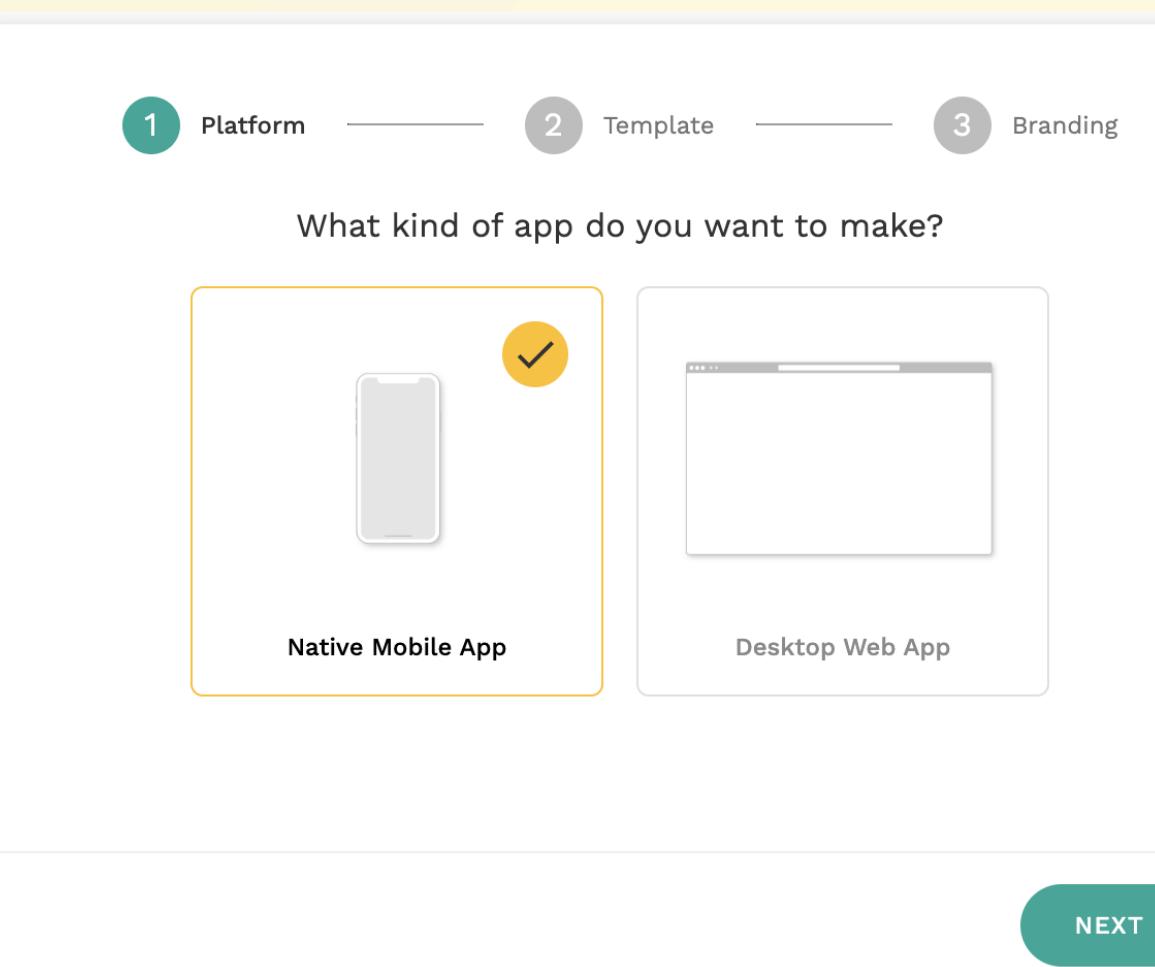


まずはアプリを新規作成しましょう。

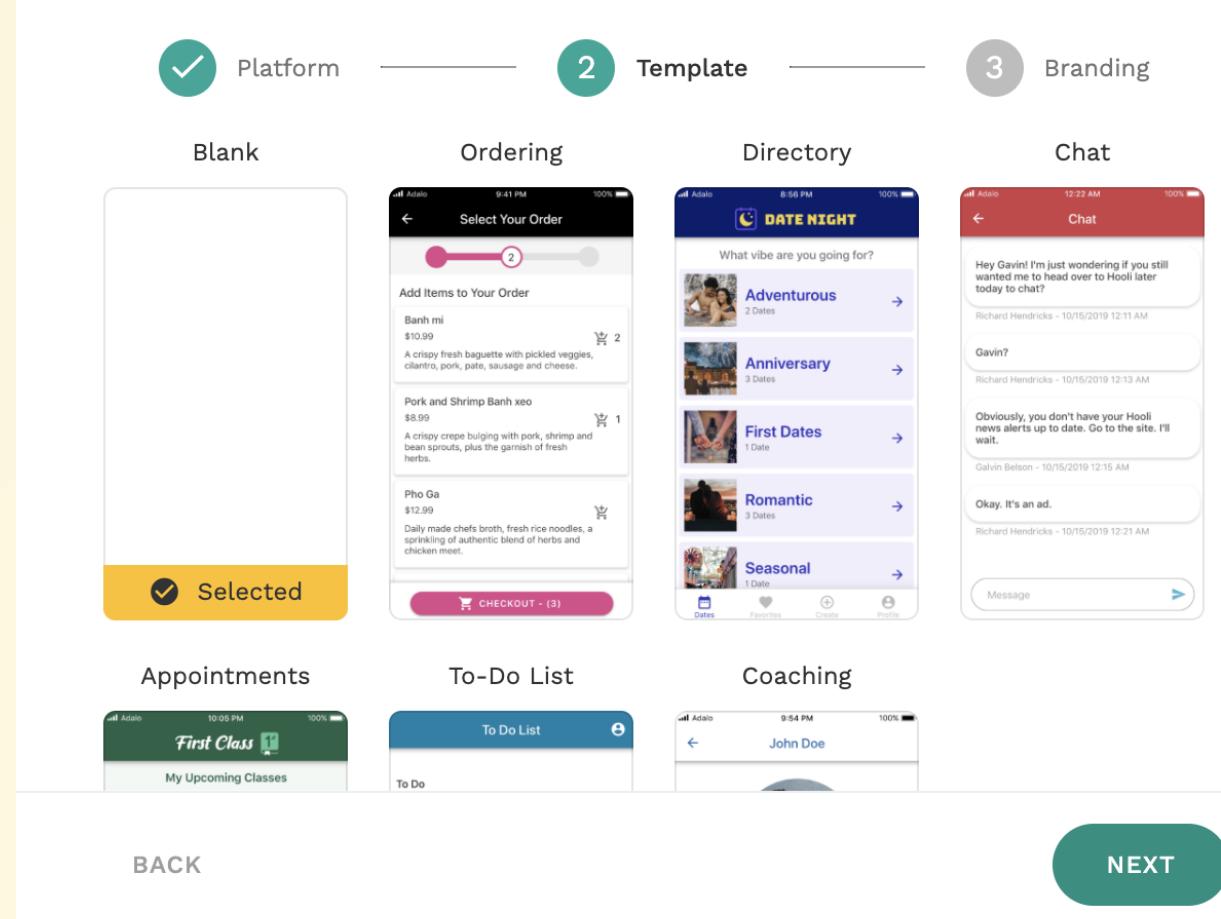
- Adaloにログイン  
<https://app.adalo.com/login>
- CREATE NEW APPをクリック



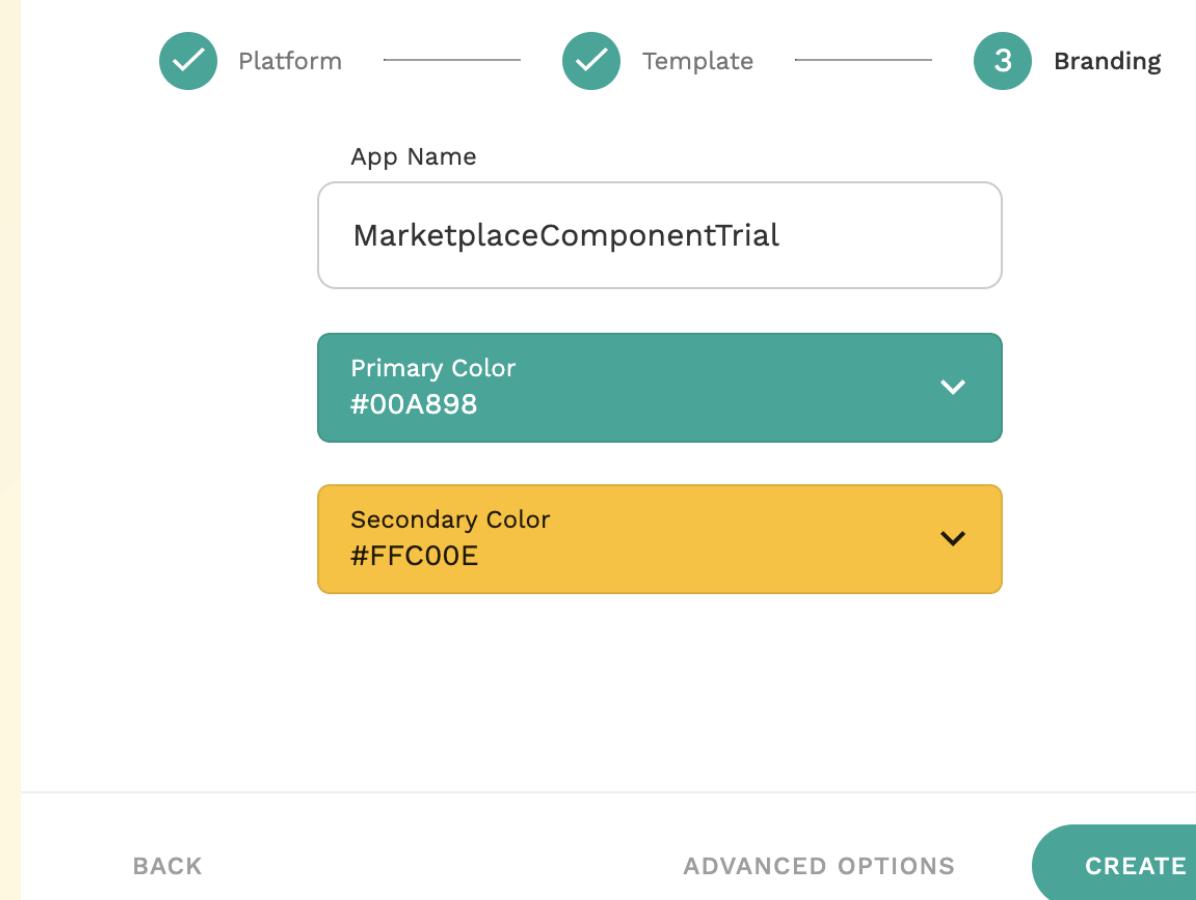
- PlatformはNative Mobile Appを選択



- TemplateはBlankを選択

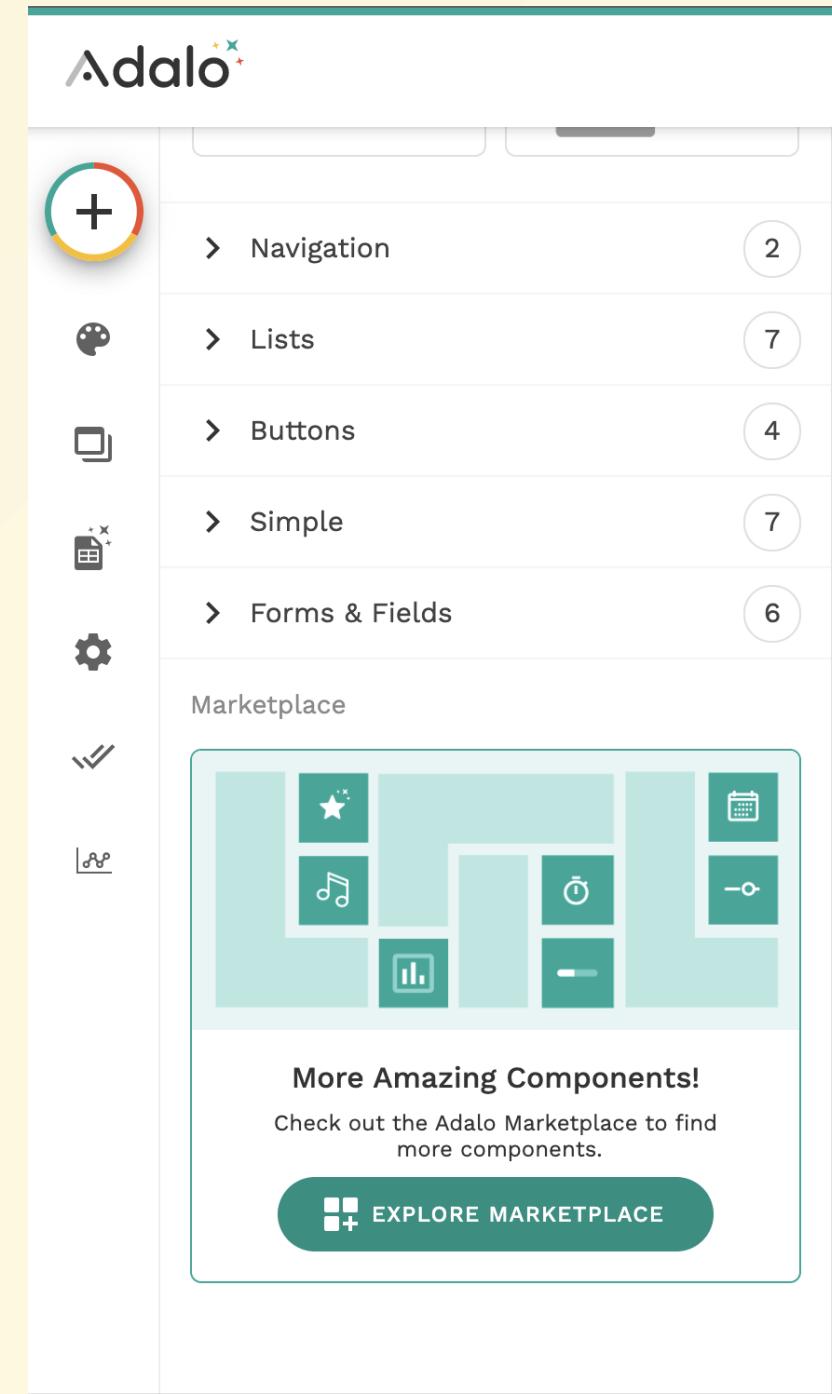


- App Nameには MarketplaceComponentTrial を入力
- Colorは自由に設定してください

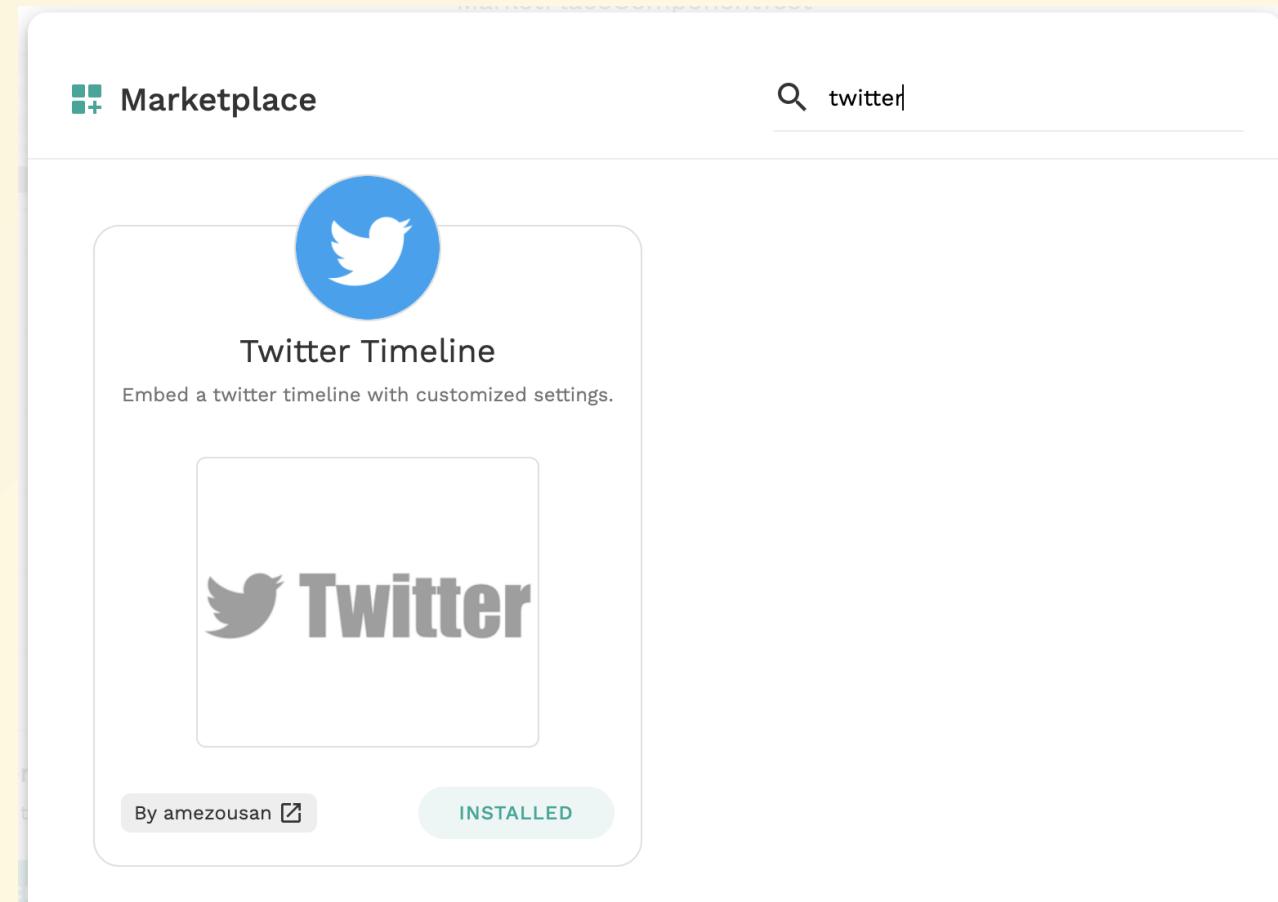


# Twitter Timeline コンポーネント

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

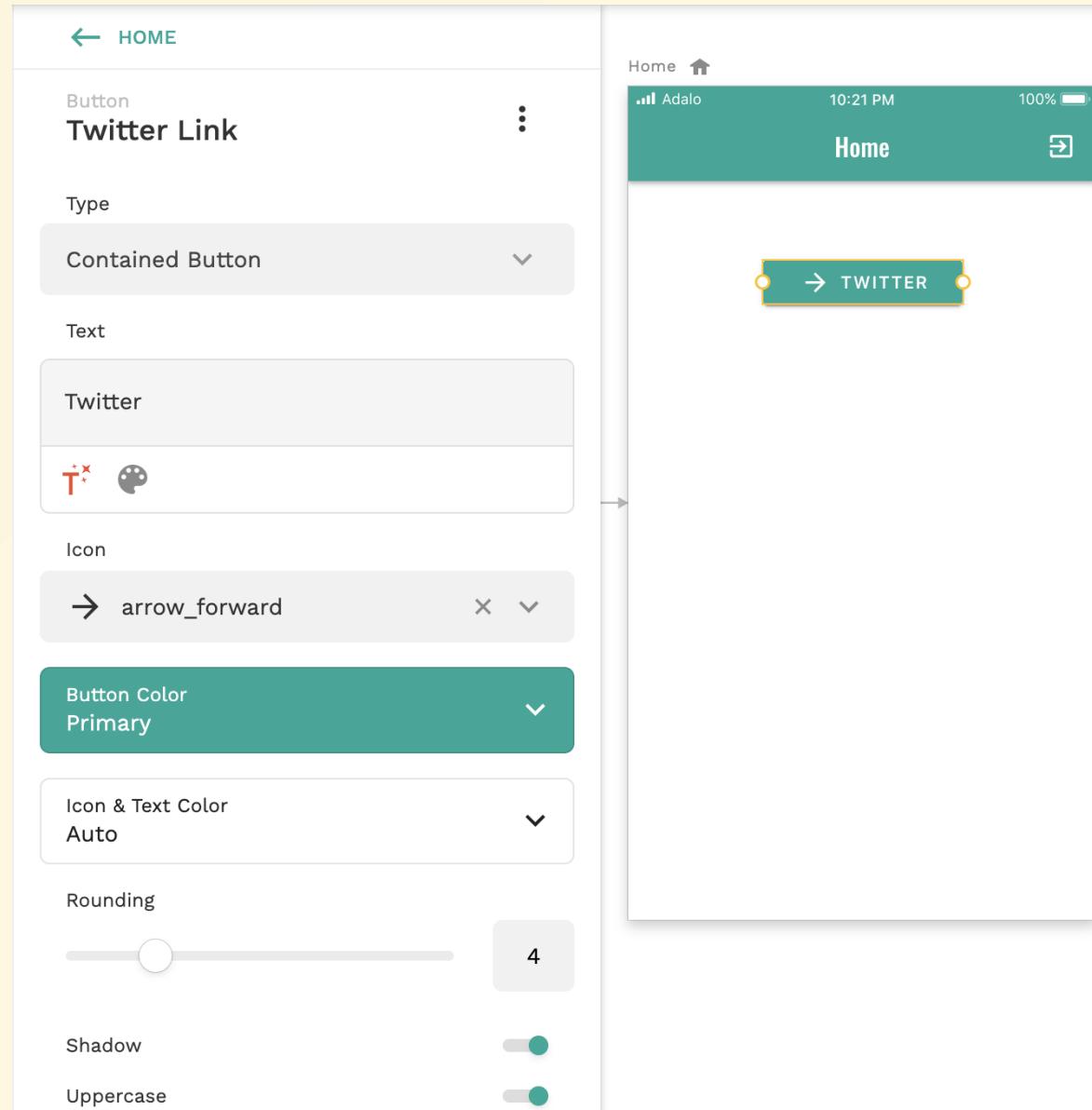


- Twitter TimelineコンポーネントをINSTALL

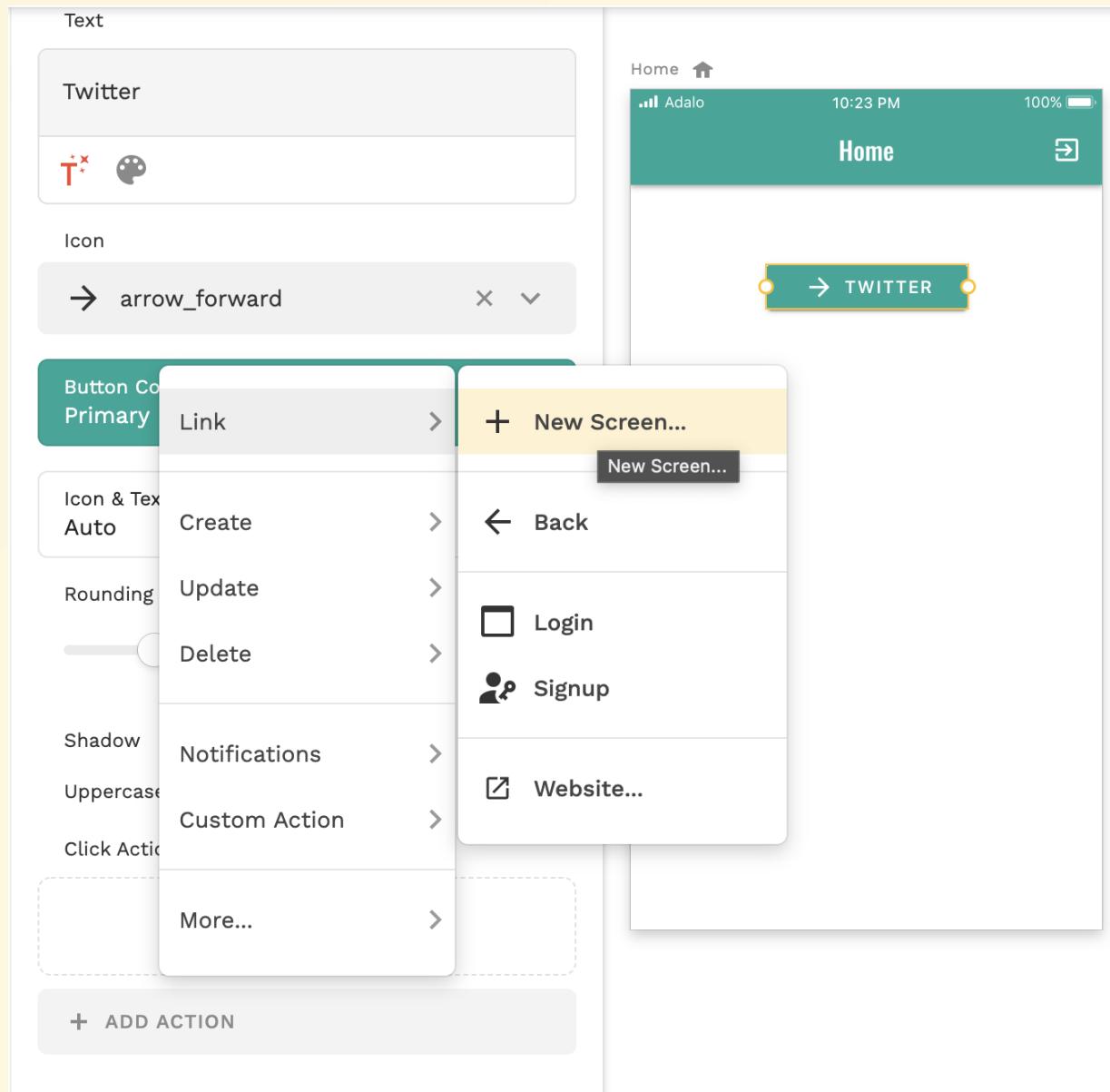


# Twitter Timelineコンポーネントを配置します。

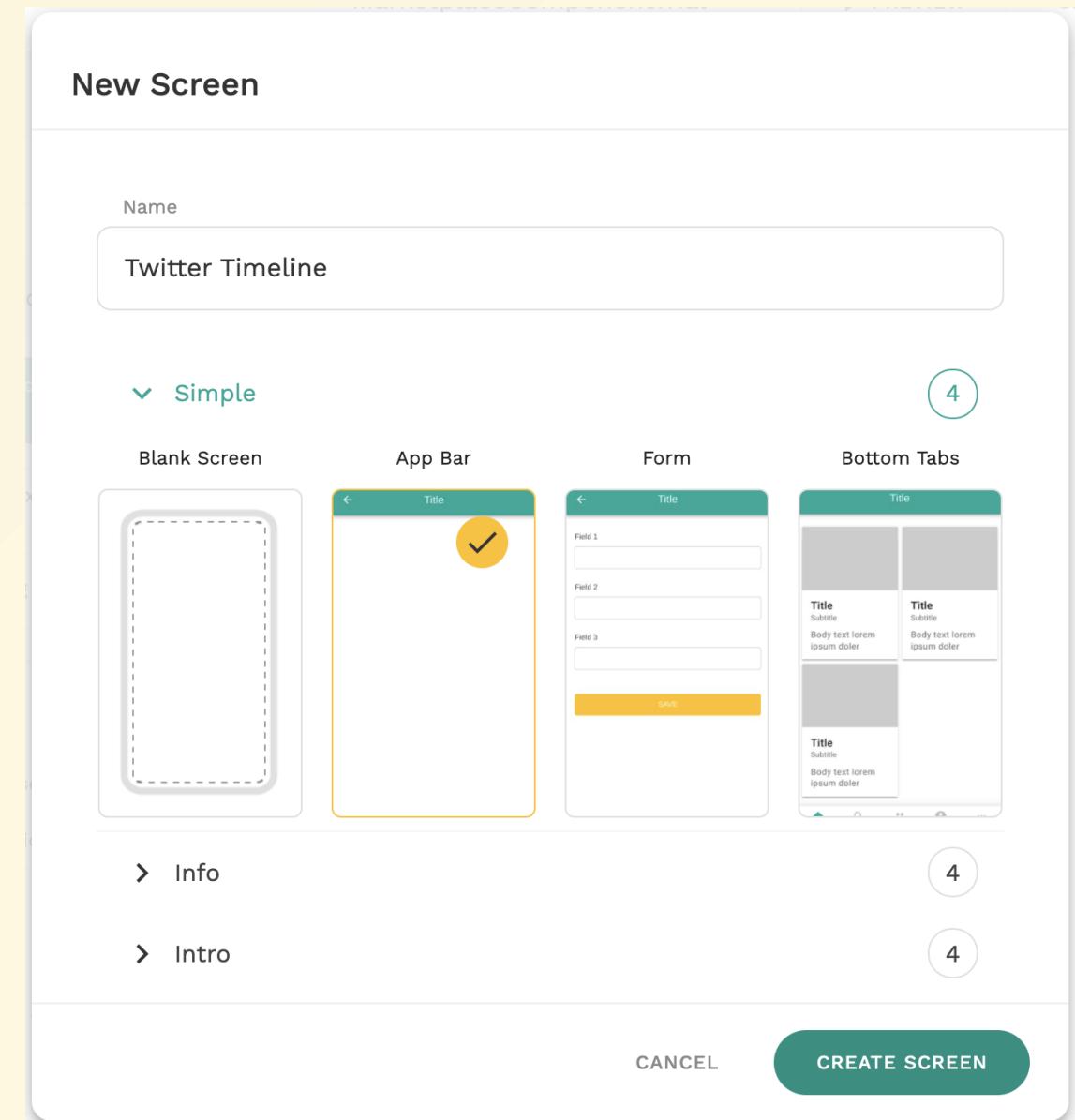
- Home ScreenにTwitter用画面へのLinkボタンを追加



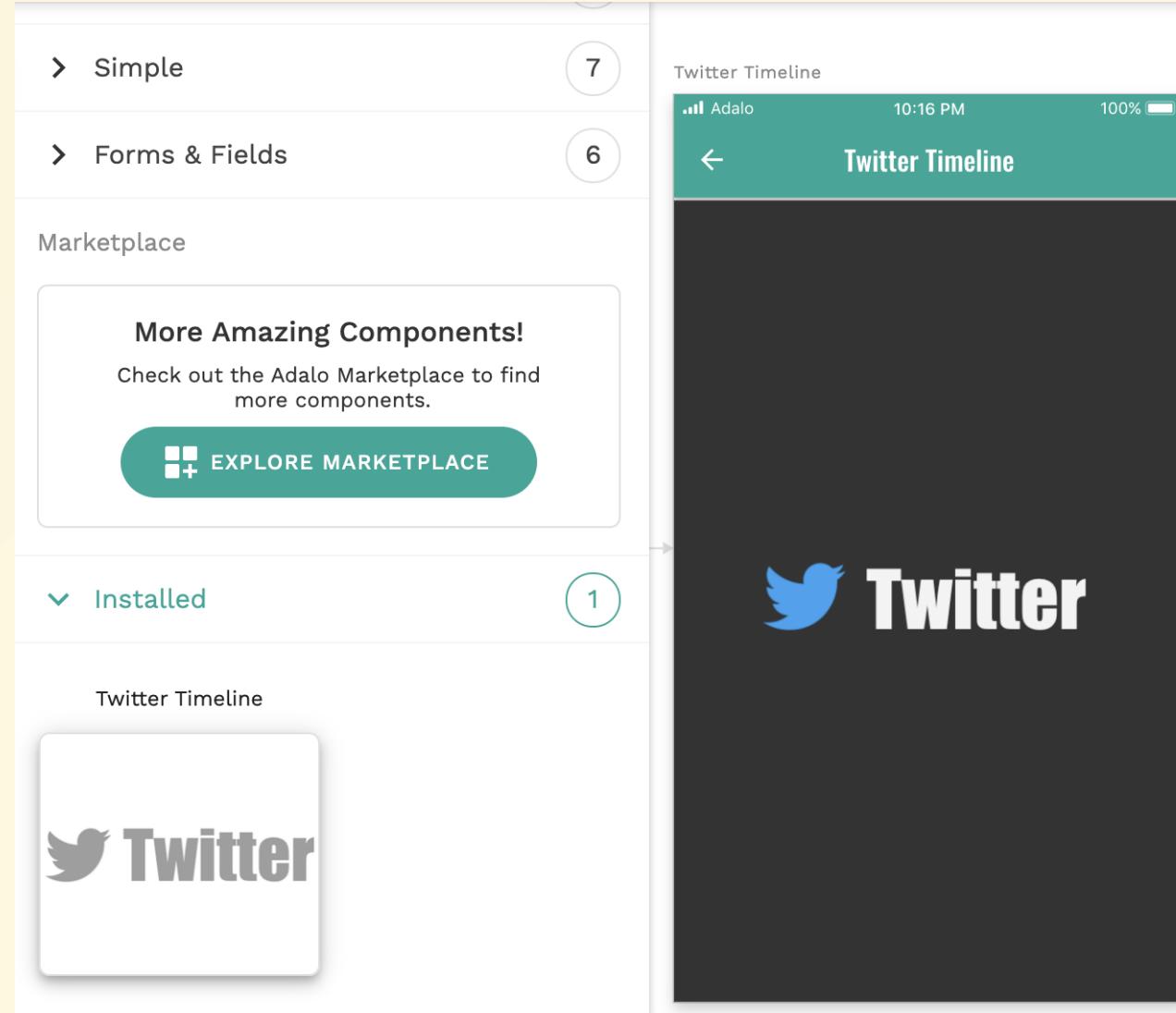
- ADD ACTIONからNew ScreenへのLinkを追加



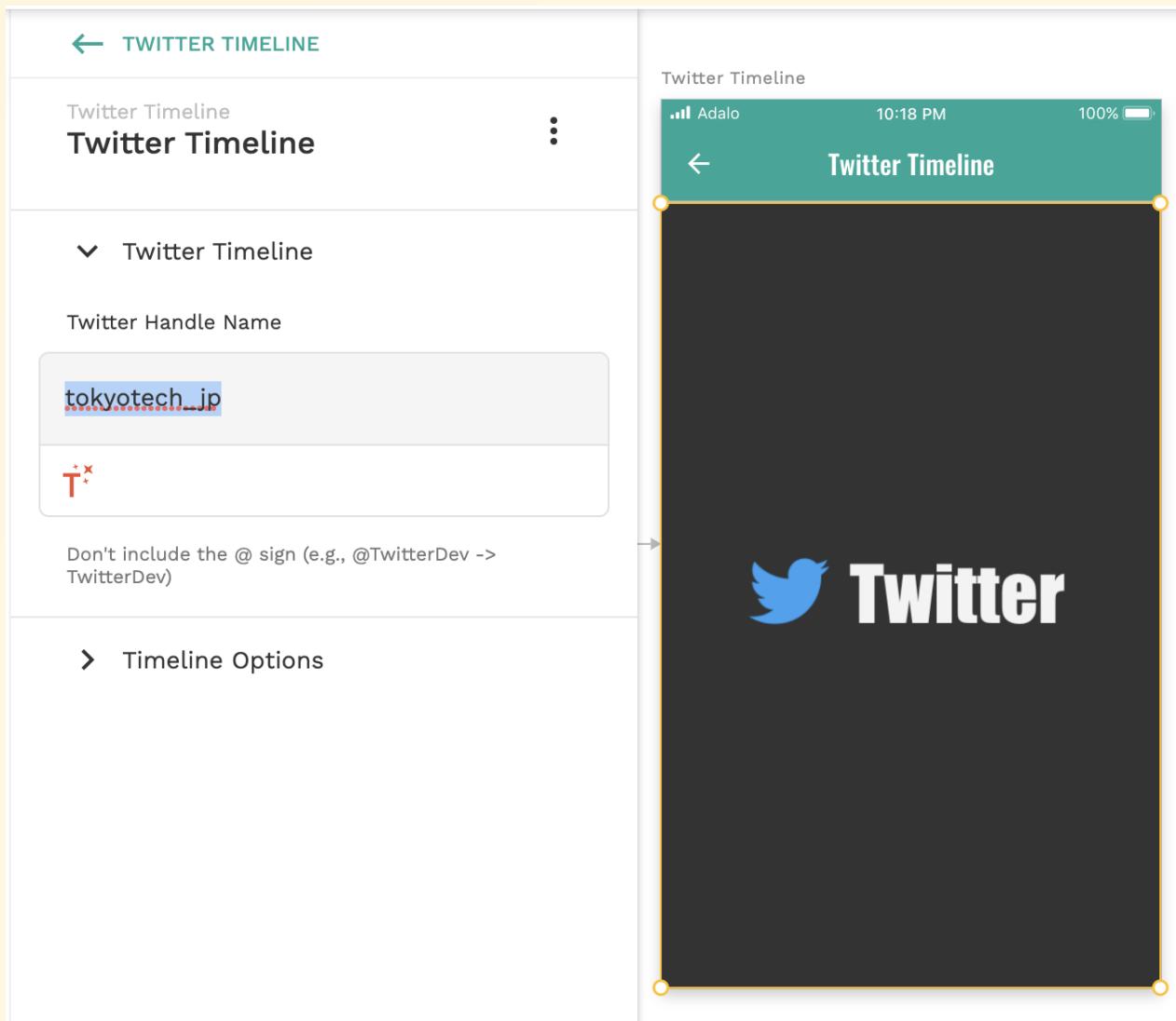
- Templateで"App Bar"を選択し、"Twitter Timeline" 画面を作成



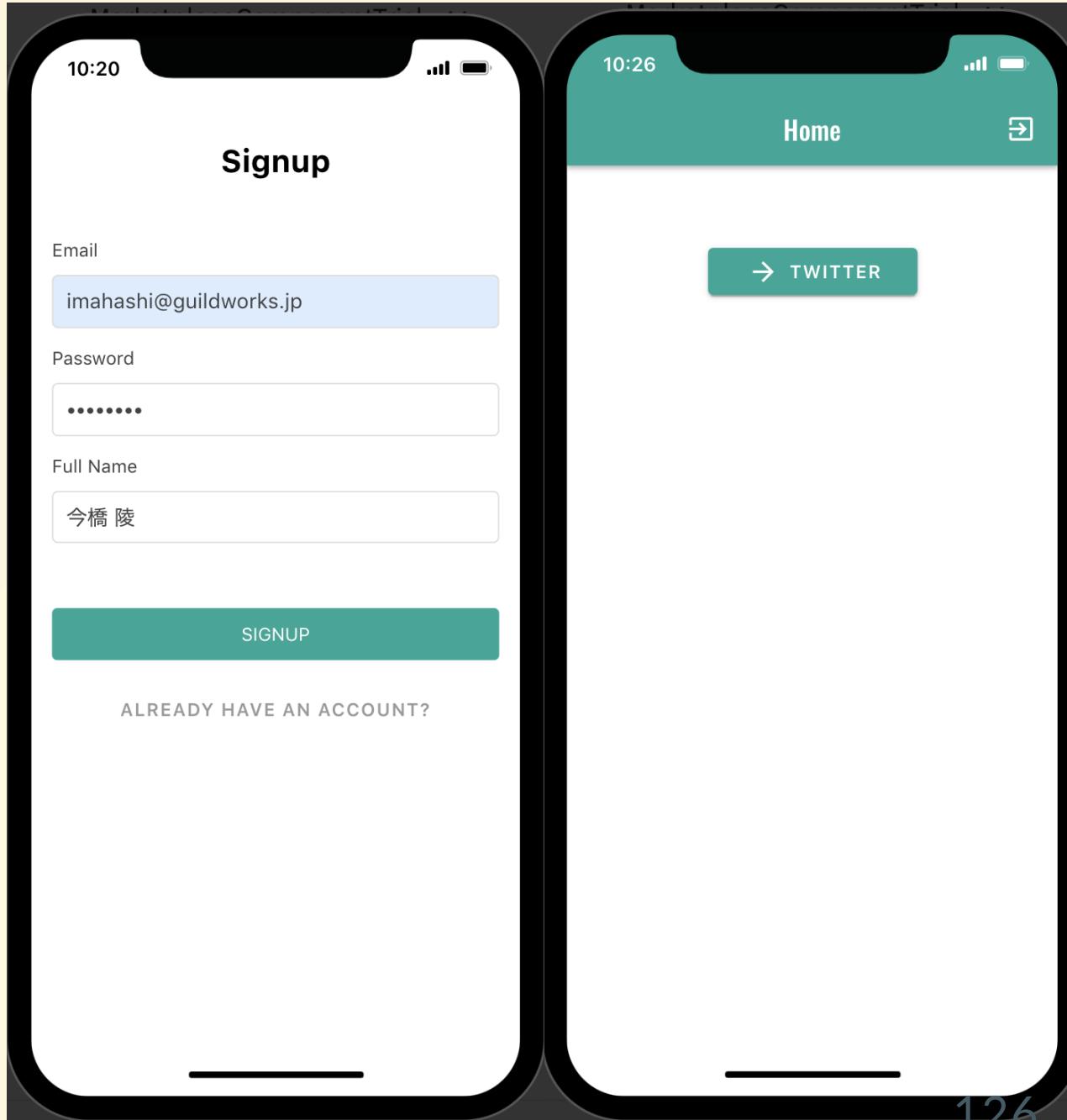
- Twitter Timelineコンポーネントを配置



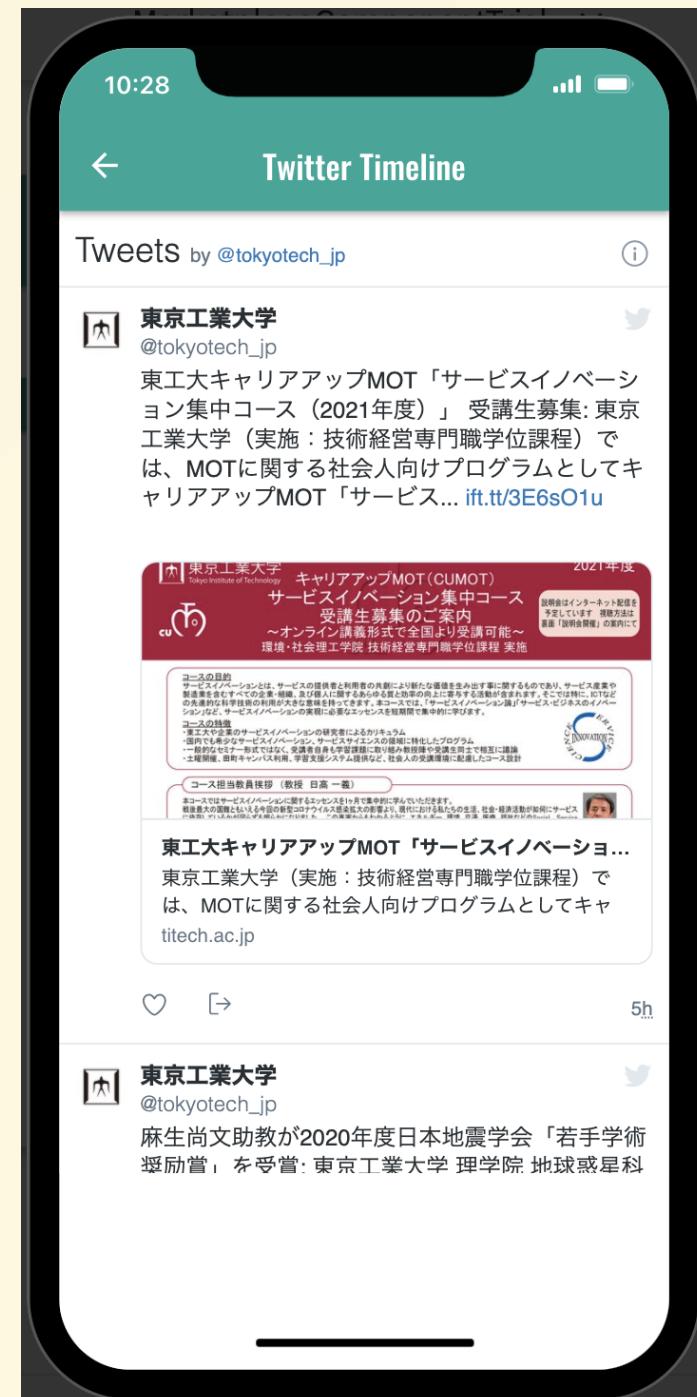
- Twitter Handle Nameに "tokyotech\_jp" と入力
  - 好きなTwitterアカウントのHandle Nameに変更してもOKです



- Preview機能でSignupして、Twitterボタンをクリック



- 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

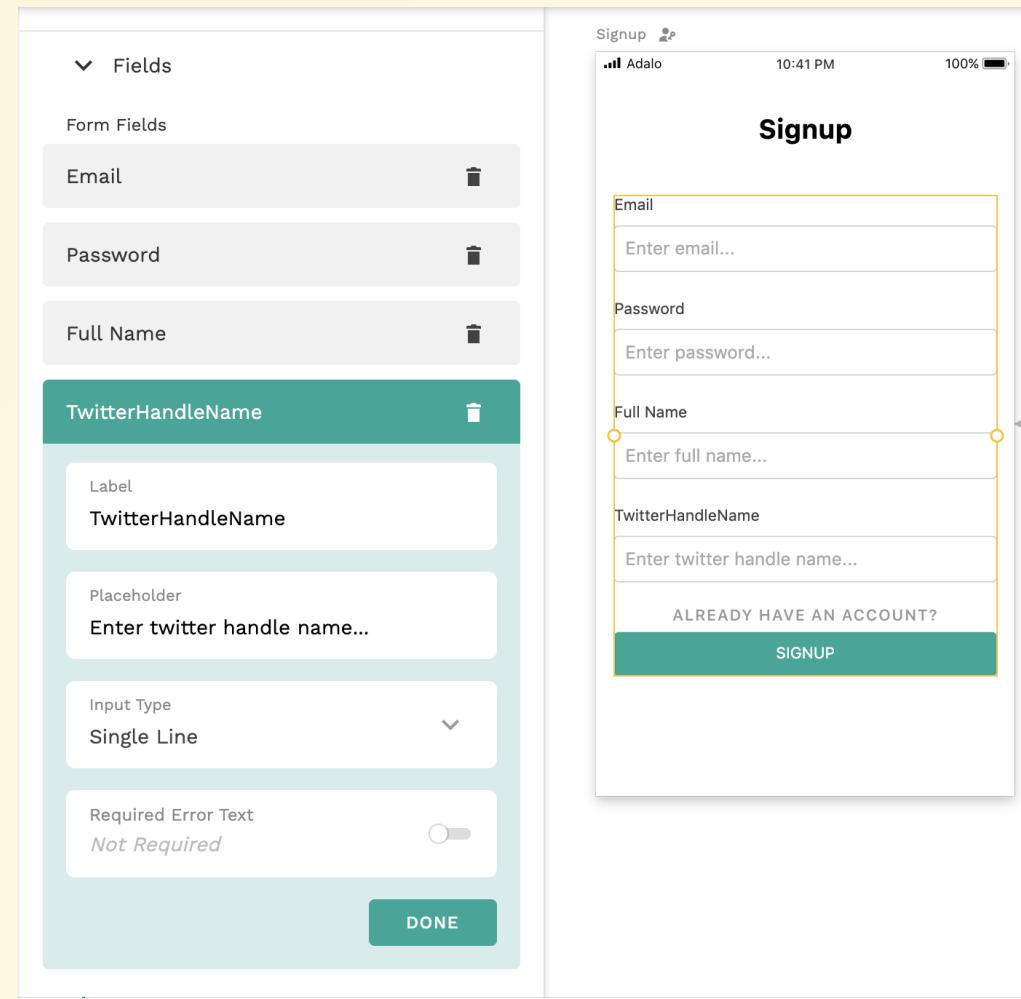


ログインしたユーザー自身のTwitterアカウントの投稿が一覧表示されるように修正しましょう。

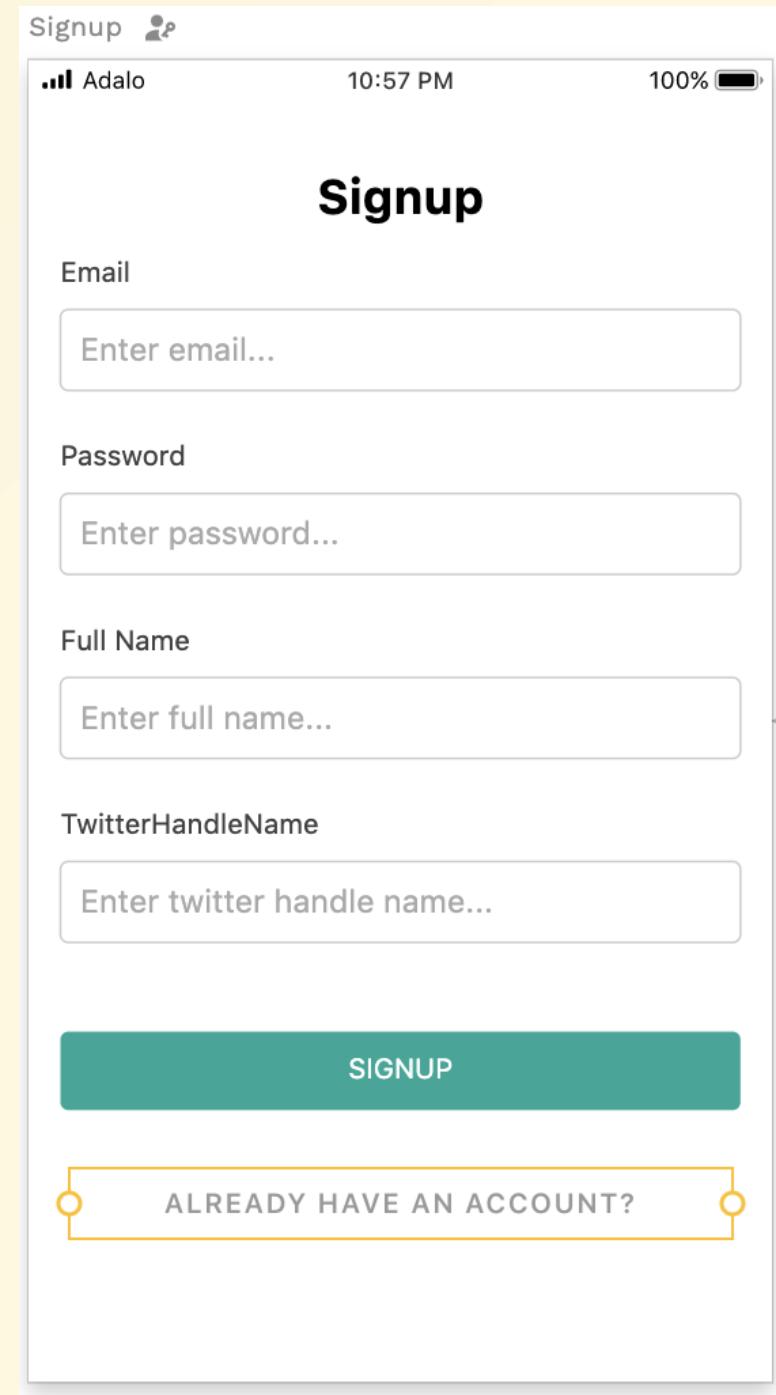
- Users CollectionにTwitterHandleName Propertyを追加
  - TypeはTextを選択

The screenshot shows the 'Database Collections' interface in Airtable. On the left, there's a sidebar with various icons: a plus sign in a circle, a paint palette, a square, a document with a star, a gear, a checkmark, and a refresh symbol. The main area shows a 'Users' collection with one record. The 'New Property' dialog is open at the bottom, titled 'New Property'. It has a 'Type' dropdown set to 'Text' and a 'Name' input field containing 'TwitterHandleName'. At the bottom right of the dialog are 'CANCEL' and 'SAVE' buttons. A red button labeled '+ ADD PROPERTY' is located at the bottom center of the screen.

- Signup画面のFormをクリック
- Fields > ADD VISIBLE FIELD > TwitterHandleName
- Not Requiredに変更

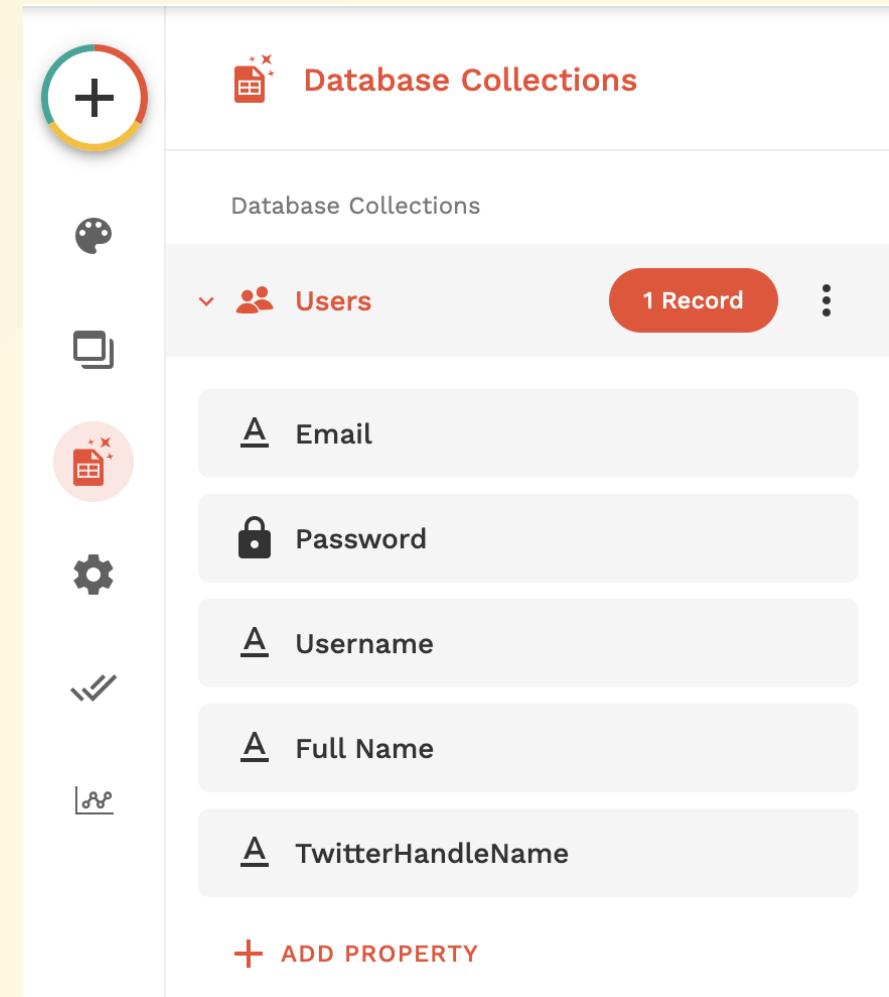


- "ALREADY HAVE AN ACCOUNT?"というリンクがフォームに重なってしまっていたので、下に移動



登録済のユーザーに  
TwitterHandleNameを設定し  
ておきましょう

- Users Collection の "1 Record" ボタンをクリック



- 登録済のユーザーのレコードをクリック
- TwitterHandleNameを入力  
(自分のアカウントでも、好きなアカウントでもOK)

Jser

A Email imahashi@guildworks.jp

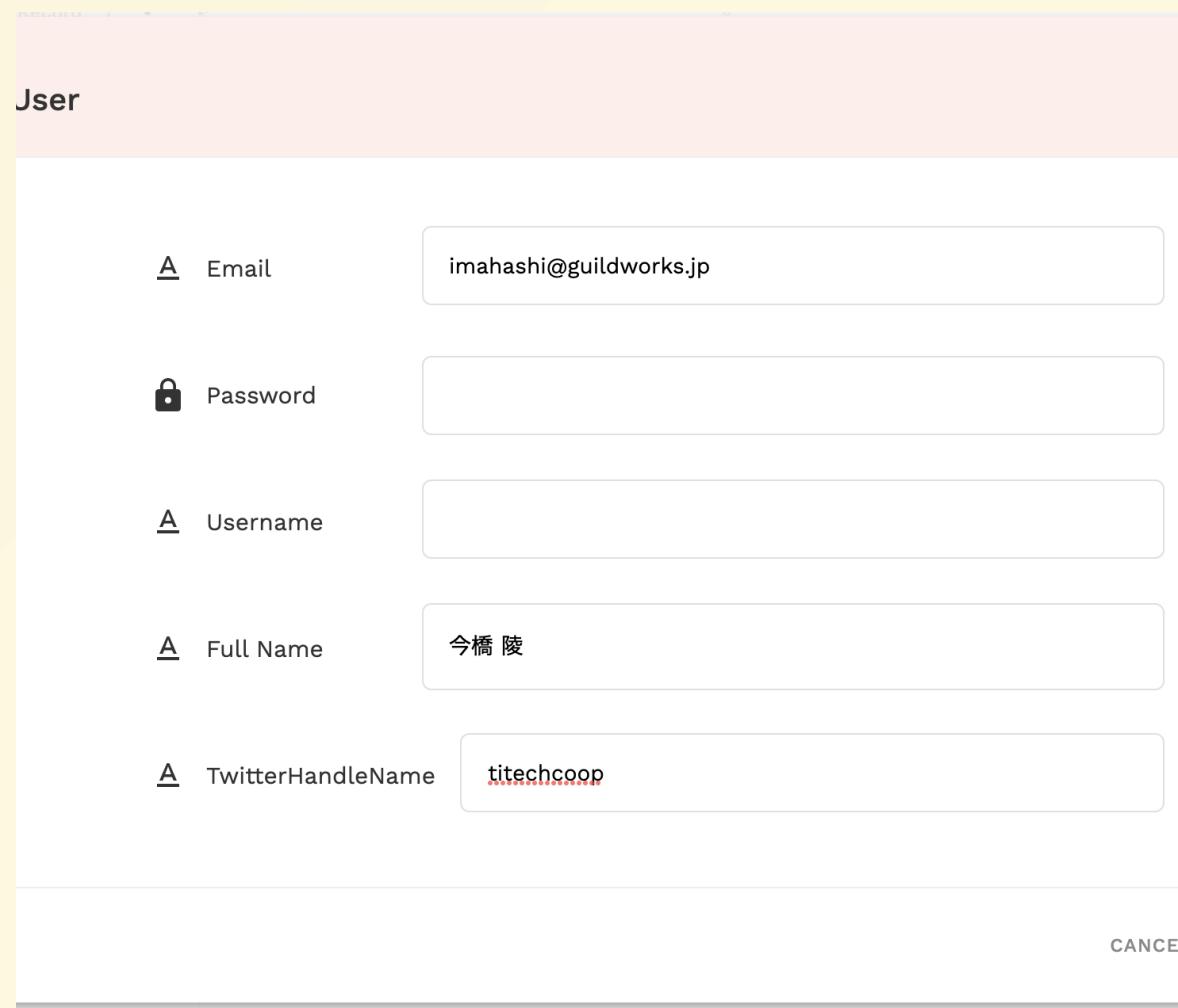
A Password

A Username

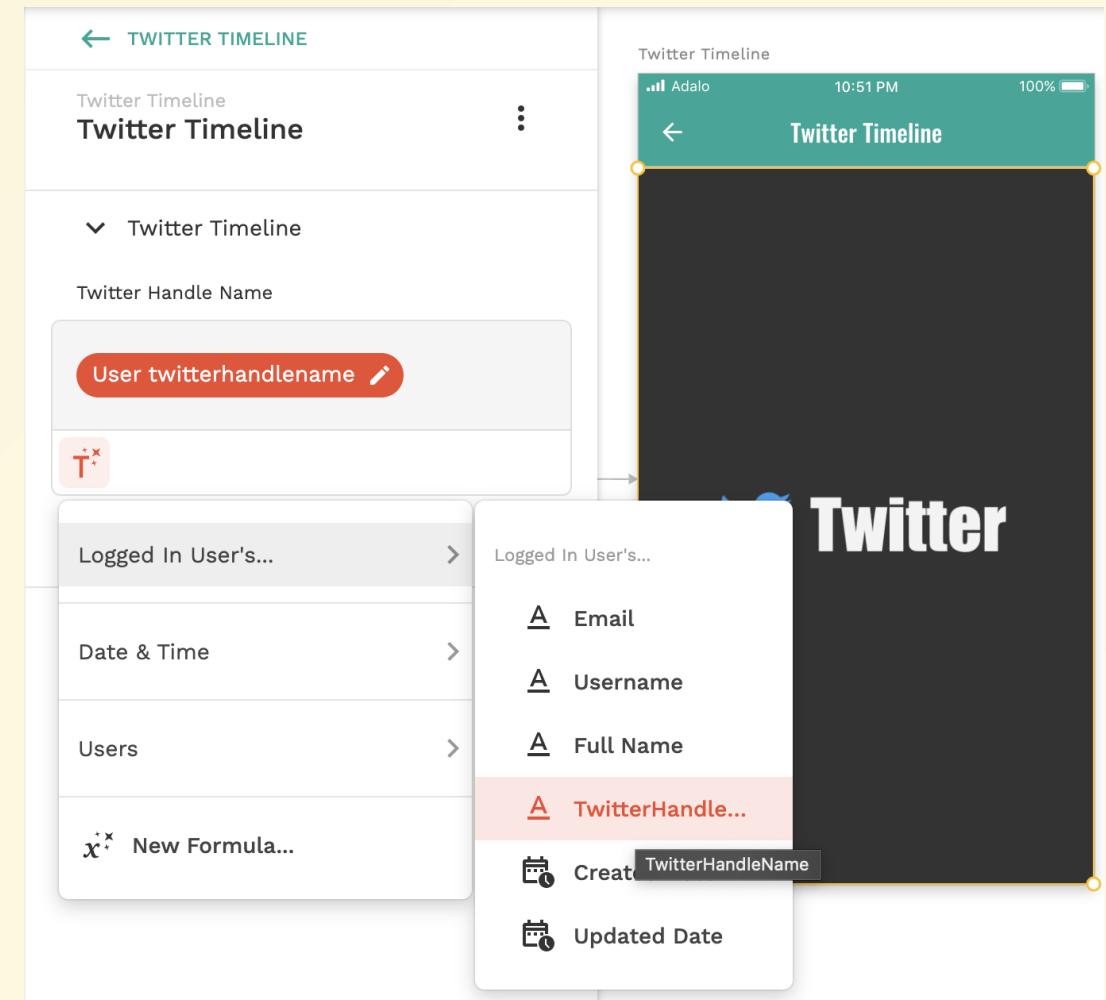
A Full Name 今橋 陵

A TwitterHandleName titechcoop

CANCEL S



- Twitter TimelineコンポーネントのTwitter Handle Nameを"Logged In User's TwitterHandleName"に変更

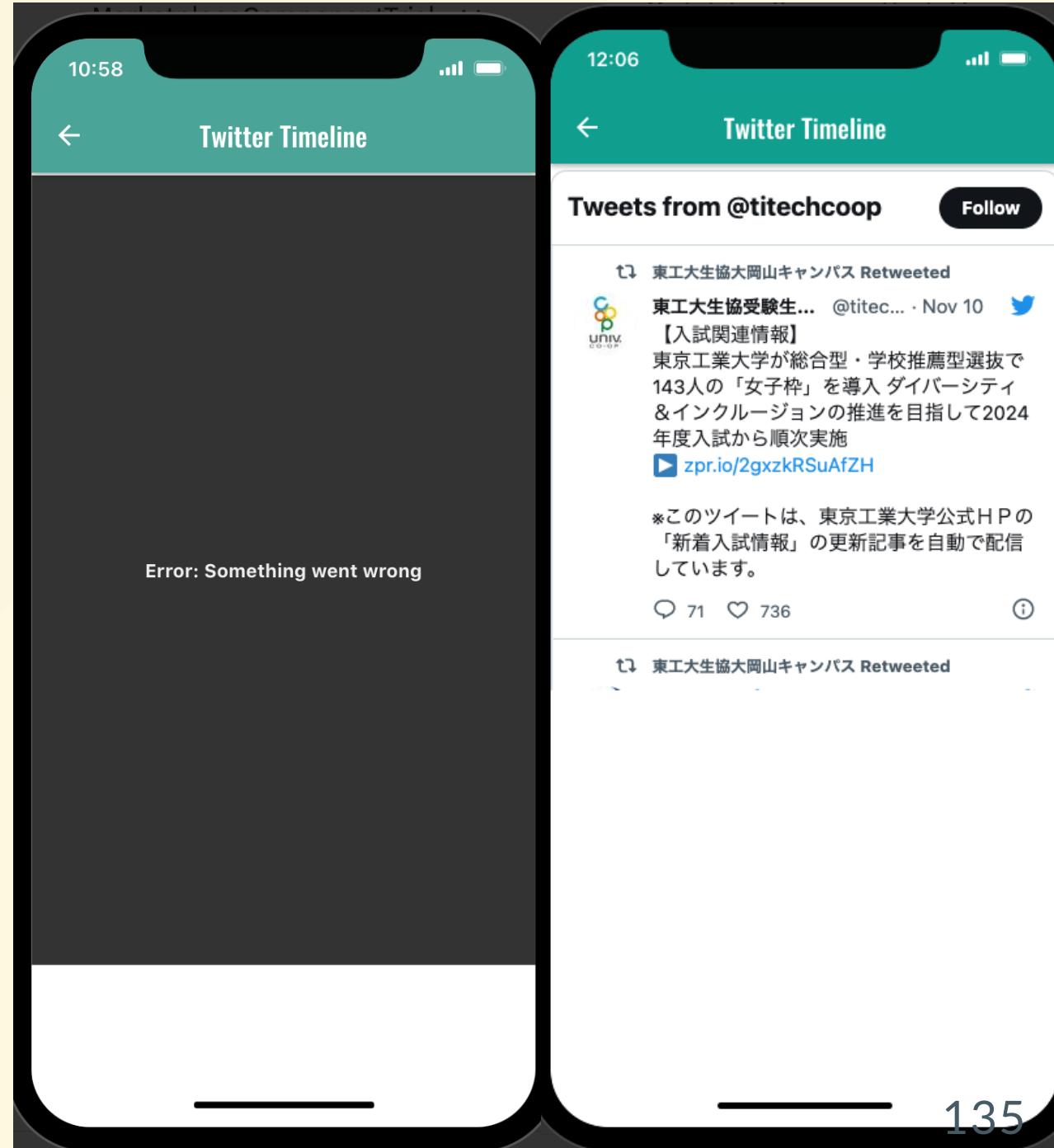


理論上はこれでログインユーザー自身が設定したTwitterアカウントの投稿一覧が表示されるはずです。

Preview機能で確認してみましょう。

最初はエラーが起きると思います。

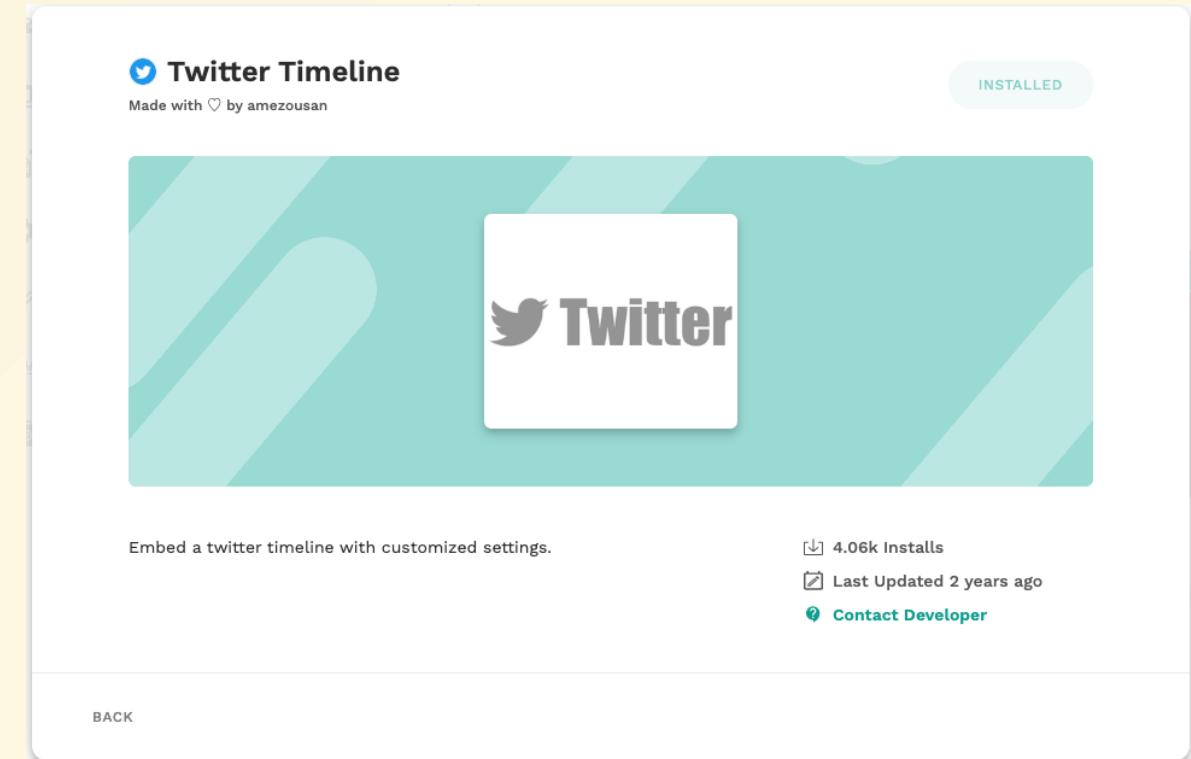
一度戻るボタンでホーム画面に戻ってから再度タイムラインを表示すると、設定したTwitterアカウントの投稿一覧が表示されます。



Twitter Timelineは有志の方が無償公開してくれているコンポーネントです。

誰かが作ってくれているものが、いつも自分の思い通りに動くとは限りません。

「不具合があったりもするんだな」くらいは覚えておいてもらえると嬉しいです。



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, and DOCUMENTATION. Below that is a secondary navigation bar with links for VOTE, BREEDS, IMAGES/SEARCH, FAVOURITES, and UPLOAD. A large green button labeled "LOVE IT" and a red button labeled "NOPE IT" are visible. The main content area features a video player displaying a cat's paw reaching for a laser pointer. Below the video is a caption that reads "The Cat API has delivered Billions of requests, to over 30K developers". On the left side, there's a section titled "Because everyday is a Caturday." with a subtext about it being a public service API for cats. Logos for various technologies are displayed at the bottom left: React, jQuery, Vue.js, Node.js, and Angular.

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 header bar with navigation links: PRICING, ABOUT, and DOCUMENTATION. Below the header, the main title "The Dog API - Dogs as a Service" is displayed, followed by the tagline "Because everyday is a Dog Day." A short description below states: "A public service API all about Dogs, free to use when making your fancy new App, Website or Service. [Get your license now.](#)" To the left of the main content area, there are icons for React, jQuery, Vue.js, Node.js, and Angular. The central part of the page features a large image of a white dog sitting on a green lawn. Above 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. At the bottom of the page, a blue footer bar contains the text: "The Dog API has delivered Billions of requests, to over 30K developers".

APIを使用する際には、APIの提供者からAPIキーを発行してもらう必要がある場合が多いです。

The Cat APIでもAPIキーが必要になりますので、発行してもらいましょう。

- 下にスクロールしてPricing の欄の "SIGNUP FOR FREE" をクリック

## Pricing

Free



\$0  
No hidden costs

Free for you to use on your non-monetized app

[SIGNUP FOR FREE](#)

- 10,000 requests / month
- Free Code Samples
- [Community forums](#)
- SSL Encryption

- E-mail、App Description、type of projectを入力し、"SIGNUP" をクリック

## Signup for an API key

It's completely free, email is only be used to send you an API Key & your stats, and you can use the API as much as you like. No Spam Ever.

E-mail

 your-email-address@example.com

App Description

 Sample App

What type of project will you use the API for?

A personal project

A project for school/college/university

A business project

Opt-in to my newsletter about other APIs i'm building?

SIGNUP

[Privacy Policy](#) | [Terms & Conditions](#)

- メールで送られてきたAPIキーを確認(後で使います)

The screenshot shows an email client interface with a single message. The message is from 'Welcome to The Cat API!' (represented by a cat icon) to 'aden@thatapiguy.com'. The email was sent on November 25, 2021, at 17:01 (1 day ago). The subject of the email is 'The Cat API - API Key'. The body of the email contains the following text:

Thanks for signing up, welcome to the API!

Your API key:

xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Use it as the 'x-api-key' header when making any request to the API, or by adding as a query string parameter e.g. 'api\_key=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx' [More details on authentication.](#)

[API Documentation](#) | [Postman Collection](#) | [Public Forum](#)

Join our new API Slack Group - [That API Slack Group](#)

All the best, Aden & That API Team.

p.s. We build APIs for companies & good causes. If you're interested find out more at <https://thatapicompany.com>

At the bottom of the email, there is a small note: 'Big love!'

次に、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(先程のメールにも"API Documentation"というリンクが記載されています)

<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。

Exampleと同じように、ボタンを押すと画像が切り替わるようになります。

#### Quickstart - Get a random Kitty

Using the language / framework of your choice:

1. Load <https://api.thecatapi.com/v1/images/search>
2. Get the first Array object of the JSON response
3. Load it's .url
4. Enjoy the kitty 😺

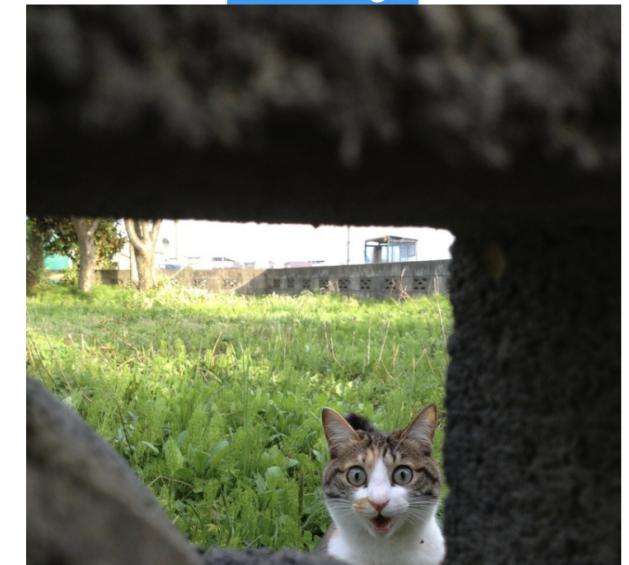
Bonus: Add a button to repeat the request for a new kitty!

#### Example

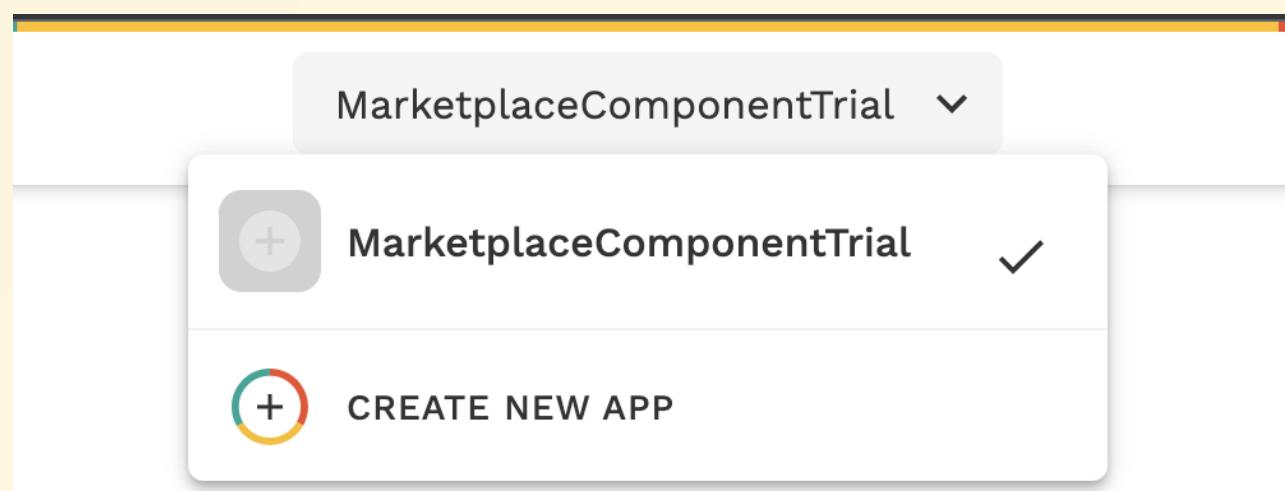
Uses <https://api.thecatapi.com/v1/images/search> to show one random image every time "Another" is clicked.

Example Source Code: VueJS

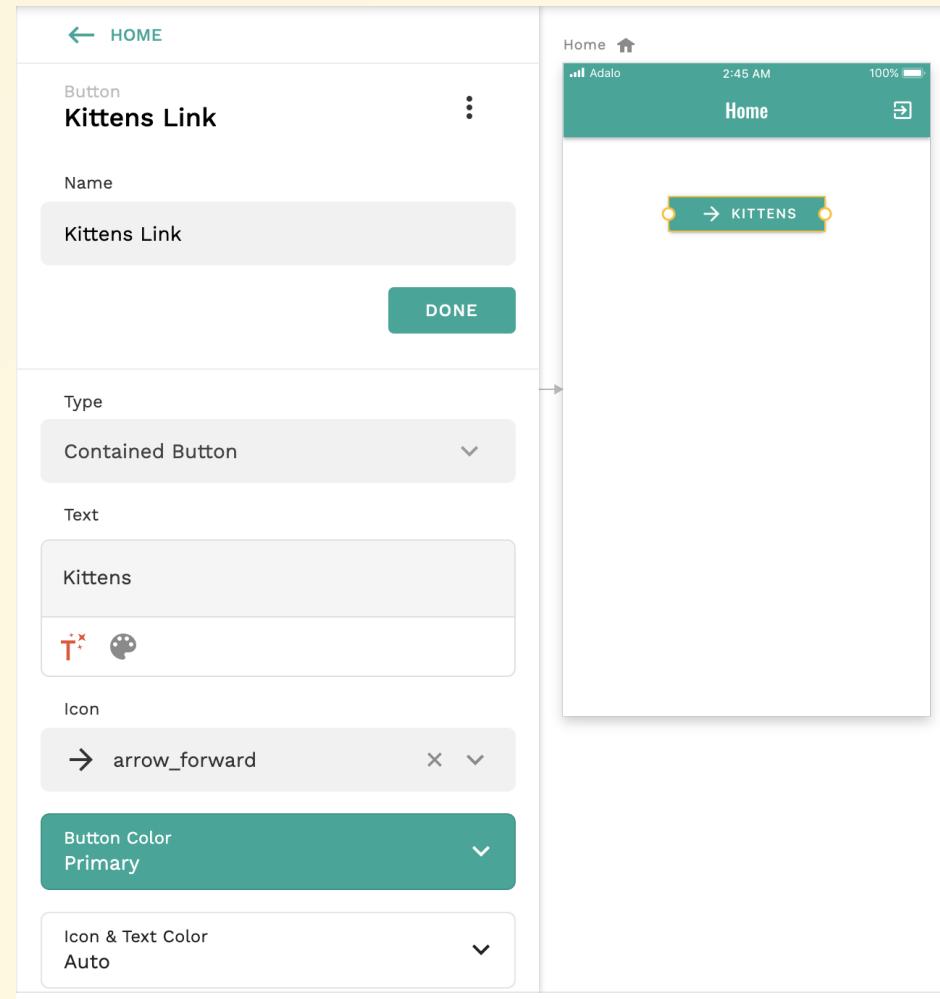
ANOTHER ↗



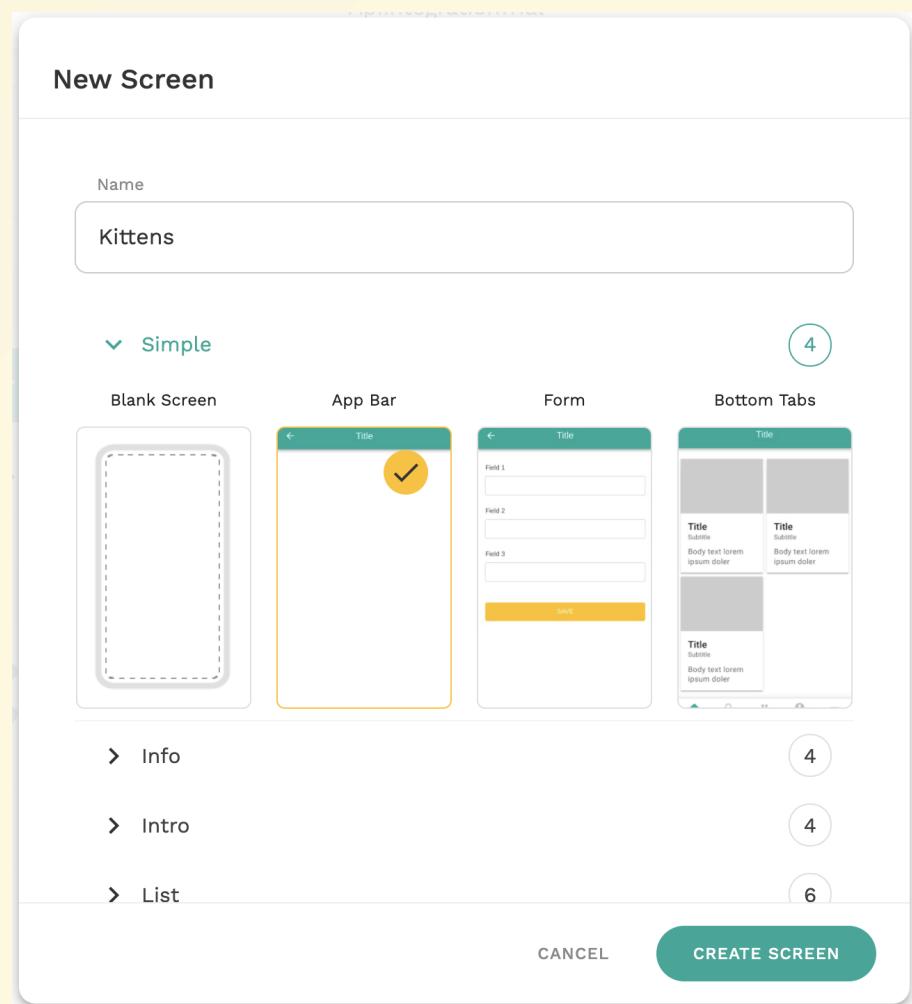
- Adaloの管理画面でCREATE NEW APP
- 設定は以下の通り
  - Platform: Native Mobile App
  - Template: Blank
  - App Name: ApilntegrationTrial



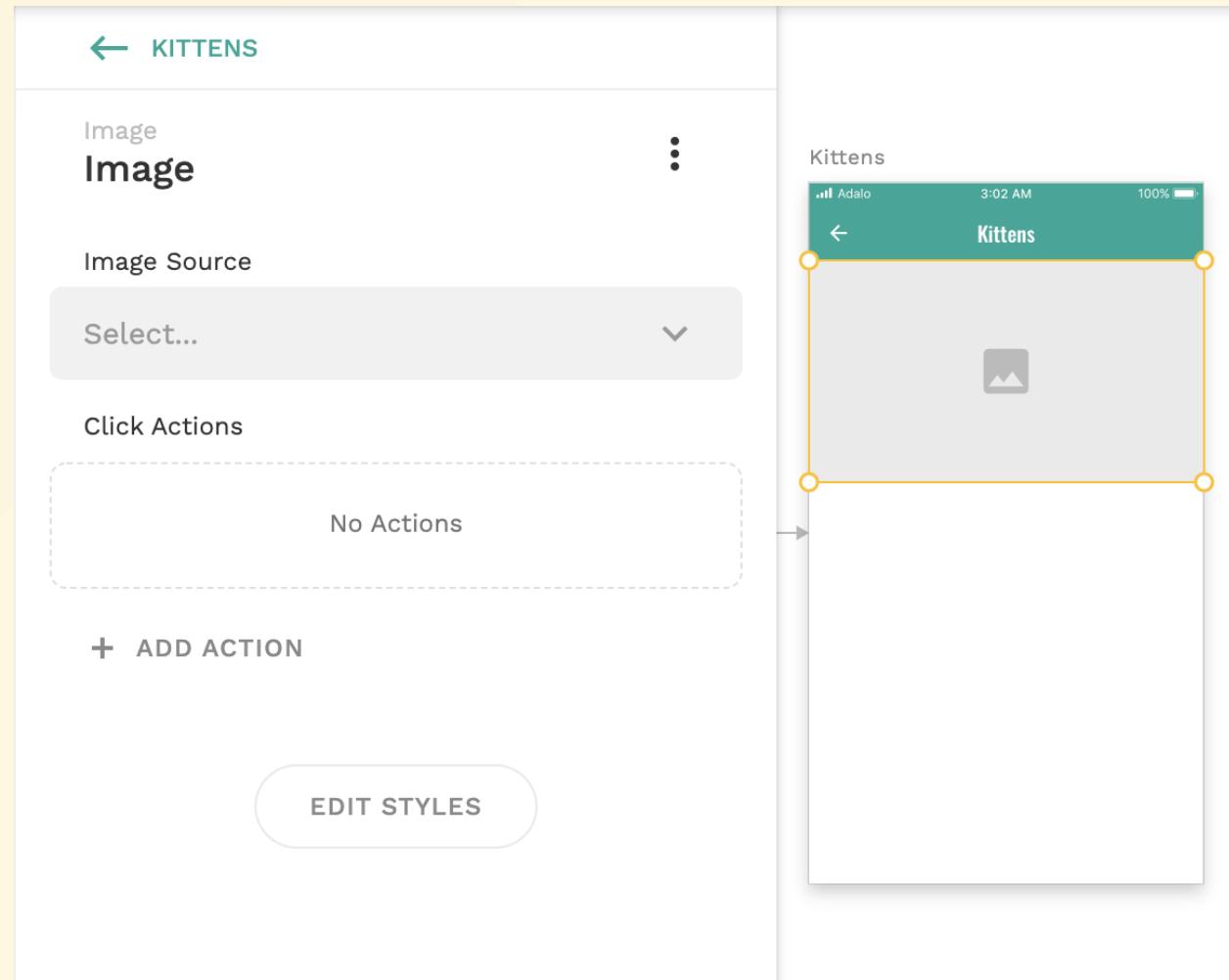
- Home画面に子猫画像表示画面へのリンクボタンを追加
- ADD ACTIONからNew ScreenへのLinkを設定



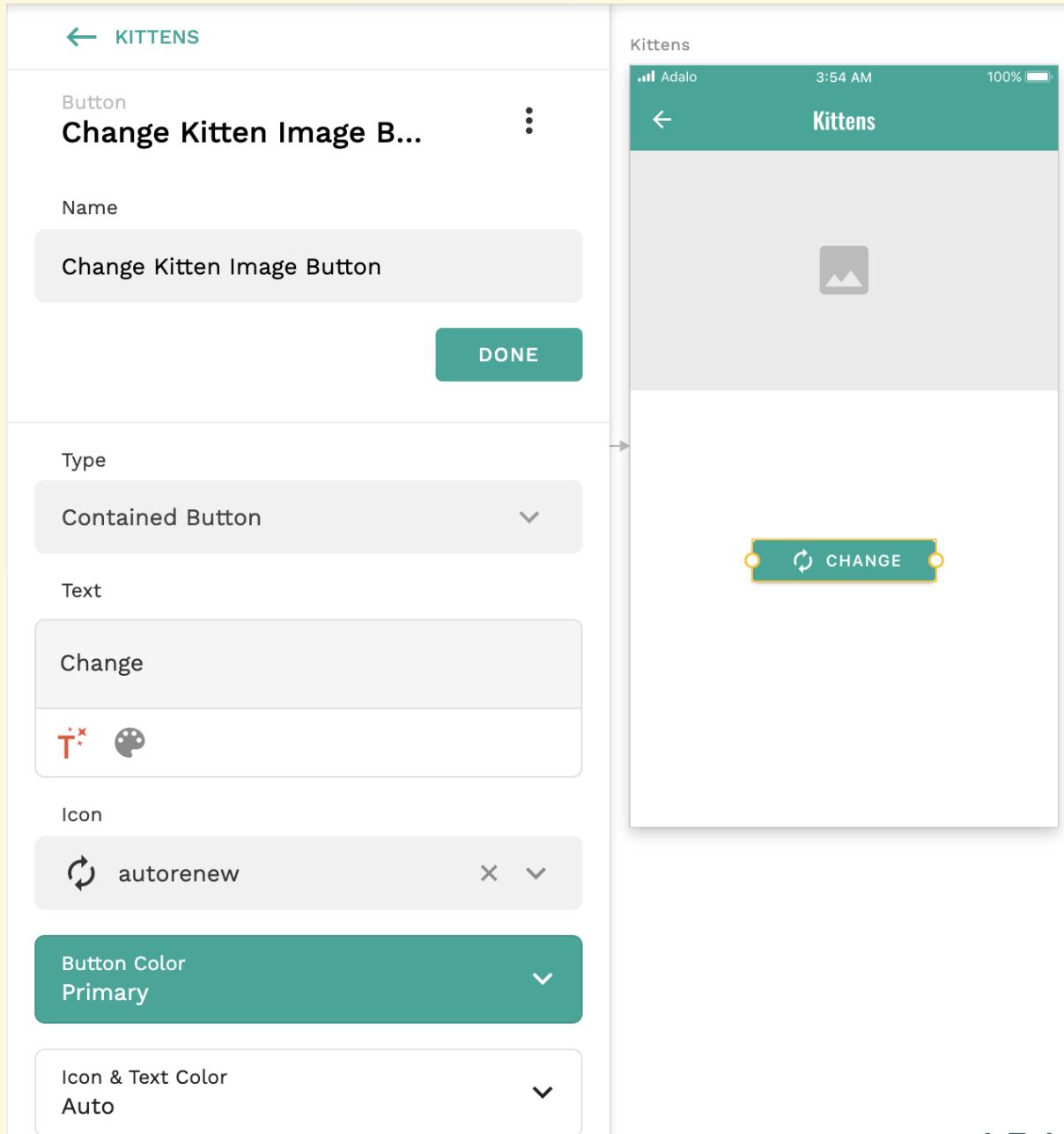
- TemplateにApp Barを選択し、Kittens画面を作成



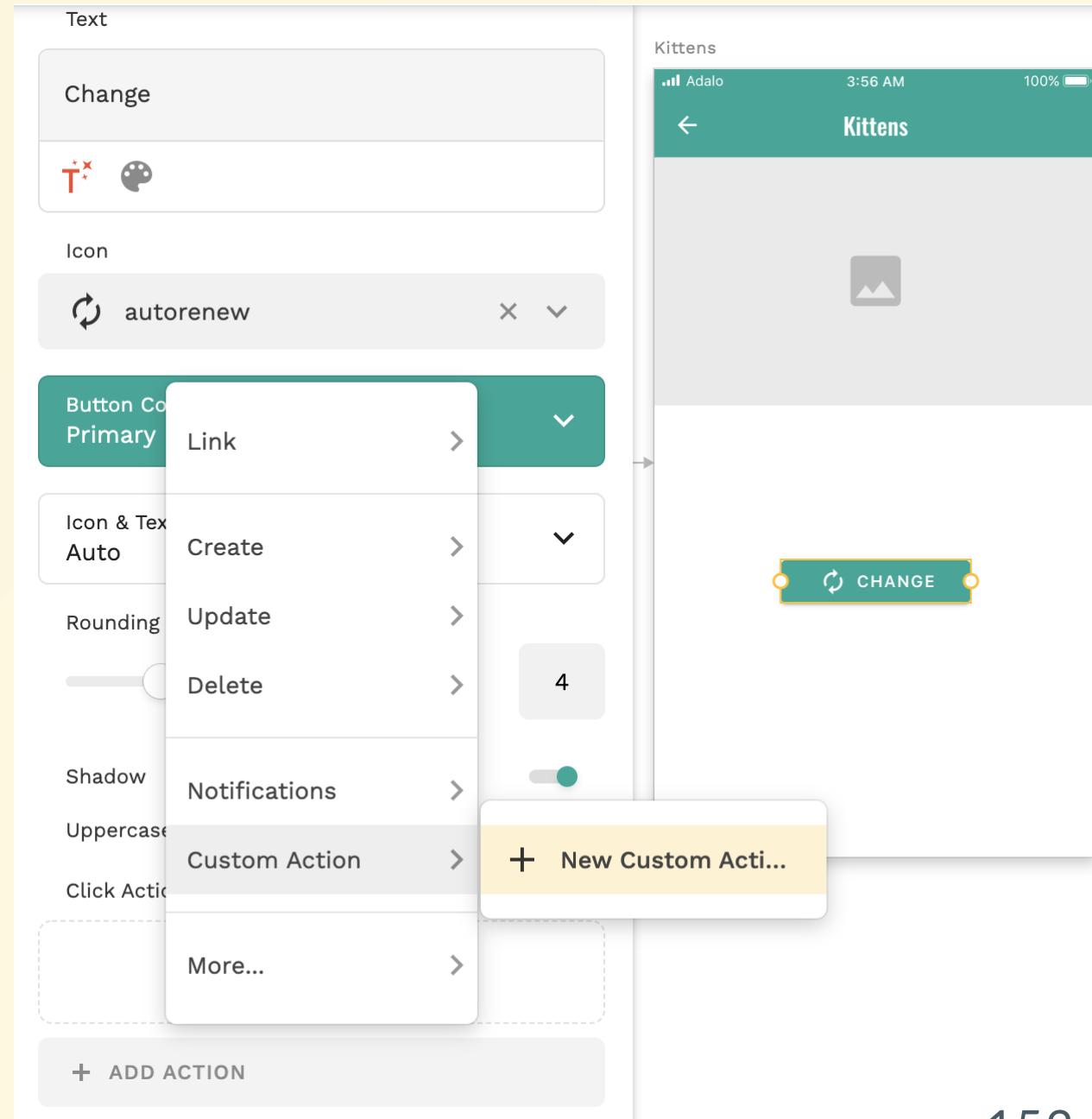
- Imageコンポーネントを画面上に配置
- コンポーネントの設定はそのままにしておく(後で設定します)



- 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

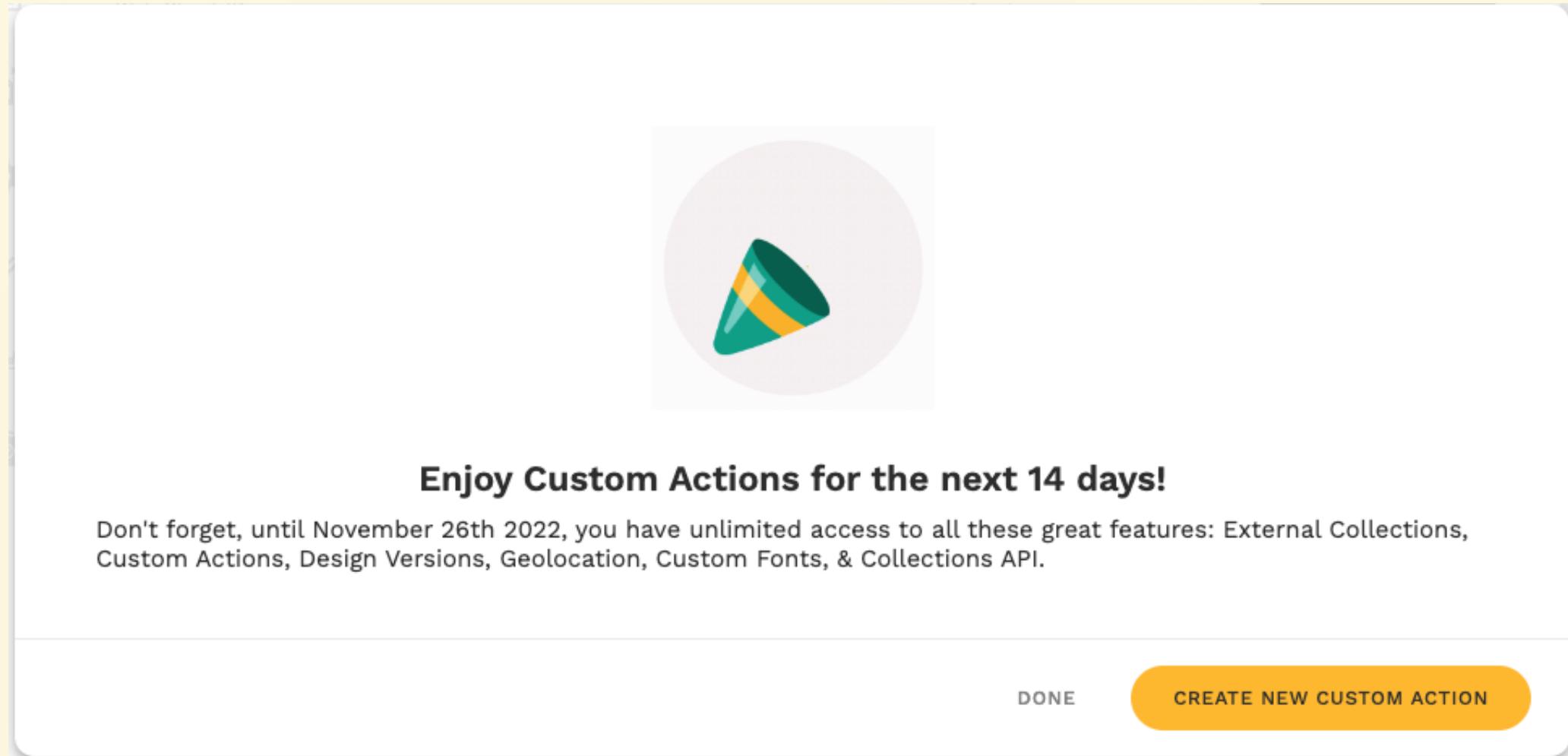
The screenshot shows a promotional banner for a 14-day free trial. At the top, it says "Unlock Premium Features for 14 Days with a Free Trial". Below this, a sub-headline reads "Begin your 14-Day Free trial today & experience premium features including Custom Fonts, External Collections, Custom Actions, Versions, Geolocation & more!". The main content area is divided into six cards, each representing a premium feature:

- Geolocation** (NEW): Utilize this popular feature to access your users' locations to customize their experience.
- Design Versions** (NEW): Make changes, save designs, and restore versions with this feature.
- Collections API**: Make your Collection data accessible through your app's very own custom API.
- Custom Actions**: Trigger unique actions to extend the functionality available in your app.
- Google Fonts**: Access over 1000 custom fonts perfect for customizing the look & feel of your app.
- External Collections**: Connect & use the data you already have inside your Adalo-made app.

At the bottom right of the banner, there are two buttons: "NOT RIGHT NOW" and a red "START FREE TRIAL" button.

トライアルが開始しました。

- "CREATE NEW CUSTOM ACTION" をクリック



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

The screenshot shows a user interface for creating a new API endpoint. It consists of three horizontal tabs at the top: 'Name & Type' (highlighted in yellow), 'API Request', and 'Outputs'. Below the tabs, there are two input fields: 'Name' containing 'GetRandomKitten' and 'Type' containing 'Create'. At the bottom right are 'CANCEL' and 'NEXT' buttons, with 'NEXT' being highlighted in yellow.

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

Name  
GetRandomKitten

Type  
Create

CANCEL NEXT

次に、送信するAPI Requestを設定していきます。

The screenshot shows a user interface for configuring an API request. The process is divided into three steps:

- 1 Name & Type (Completed)
- 2 API Request (Current Step)
- 3 Outputs

Under the API Request step, the following fields are visible:

- API Base URL:** https://api.example.com/v1/action
- Method:** Select a Method (dropdown menu)
- Headers & Queries:** No Auth Parameters
- Body:** Enter JSON here
- Inputs:** No Inputs (with a red 'ADD ITEM' button)

At the bottom right of the screen, there are two buttons: "BACK" and a large yellow "RUN TEST REQUEST" button.

以下のURLにアクセスして、  
使用するAPIのドキュメントか  
ら設定項目を確認します。

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

- API Base URLは  
<https://api.thecatapi.com/v1/images/search>
- MethodはGET
- Headerに x-api-key という  
NameでAPI keyを設定

## GET /images/search

Search & Iterate through all public images. <https://api.thecatapi.com/v1/images/search>

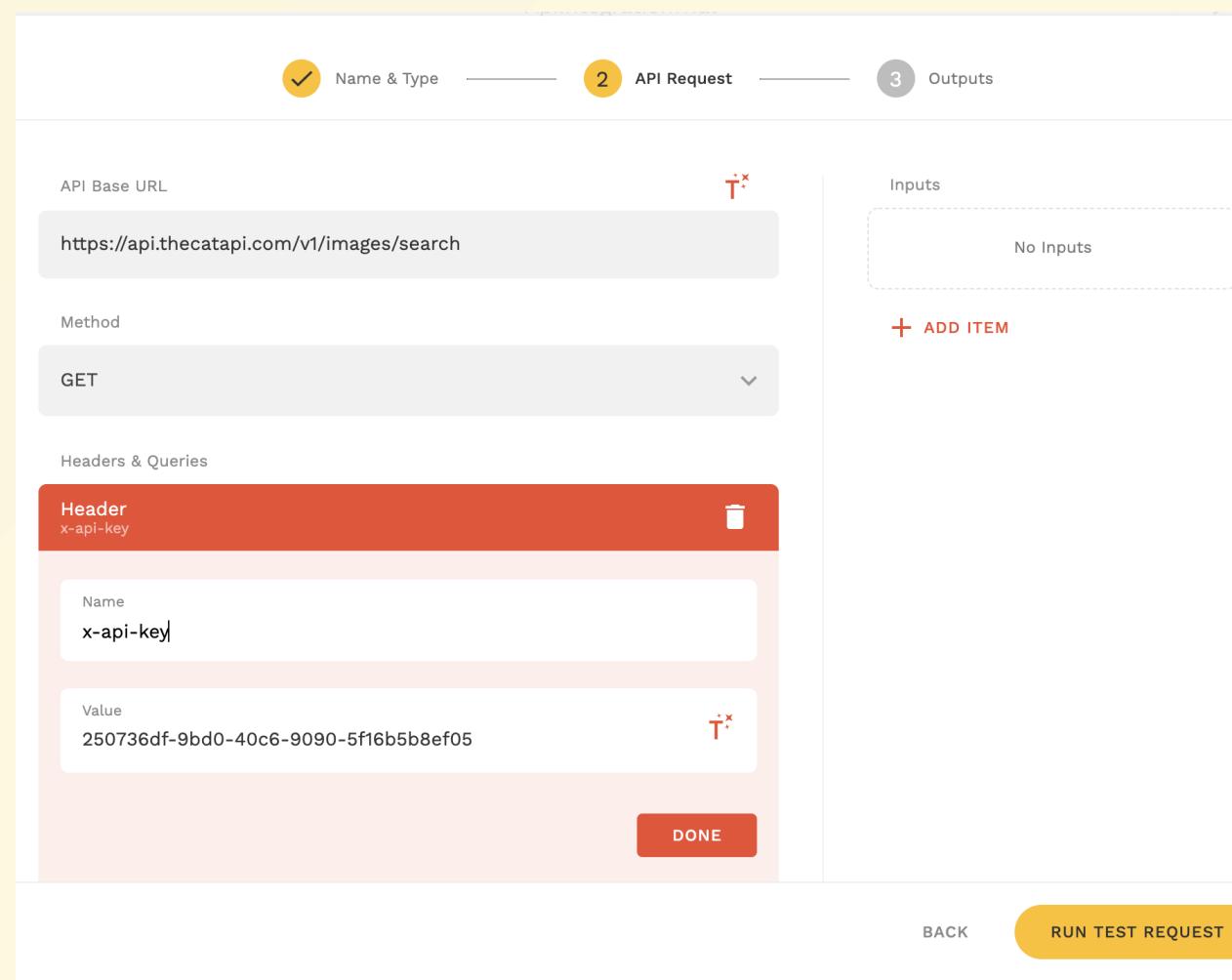
### Authorization

#### ▼ apiKey - x-api-key

Just use the API Key emailed to you when you signed up.

- Name: x-api-key
- Location: header

- 確認した結果を踏まえてAPI Requestを設定
  - API Base URLは  
<https://api.thecatapi.com/v1/images/search>
  - MethodはGET
  - Headerに x-api-key というNameでAPI keyを設定
- "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(Magic Text Output Properties)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

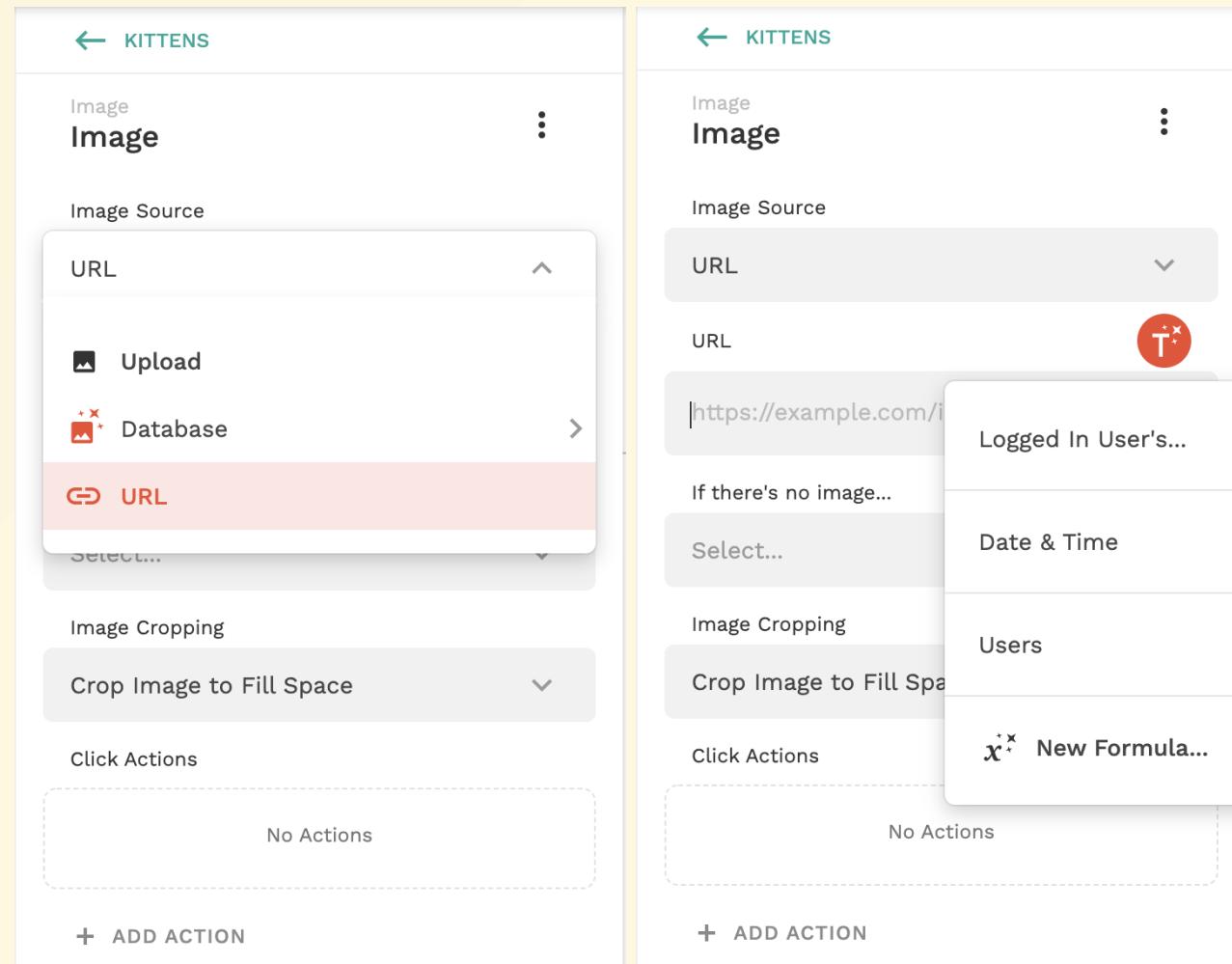
The screenshot shows a user interface for creating a custom action. At the top, there are three tabs: "Name & Type" (checked), "API Request" (checked), and "Outputs" (step 3). Below the tabs, a message "Test Successful!" is displayed. Under "URL Called", the URL <https://api.thecatapi.com/v1/images/search> is shown. A note below says "Magic Text Output Properties - Learn how to use these in subsequent actions [here!](#)". There are four listed properties: "id" (value: CzXQVvf23), "url" (value: <https://cdn2.thecatapi.com/images/CzXQVvf23.jpg>), "width" (value: 976), and "height" (value: 549). Each property has a delete icon to its right. At the bottom left, there is a red "+ ADD ITEM" button, and at the very bottom, a "SHOW FULL RESPONSE" link.

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。

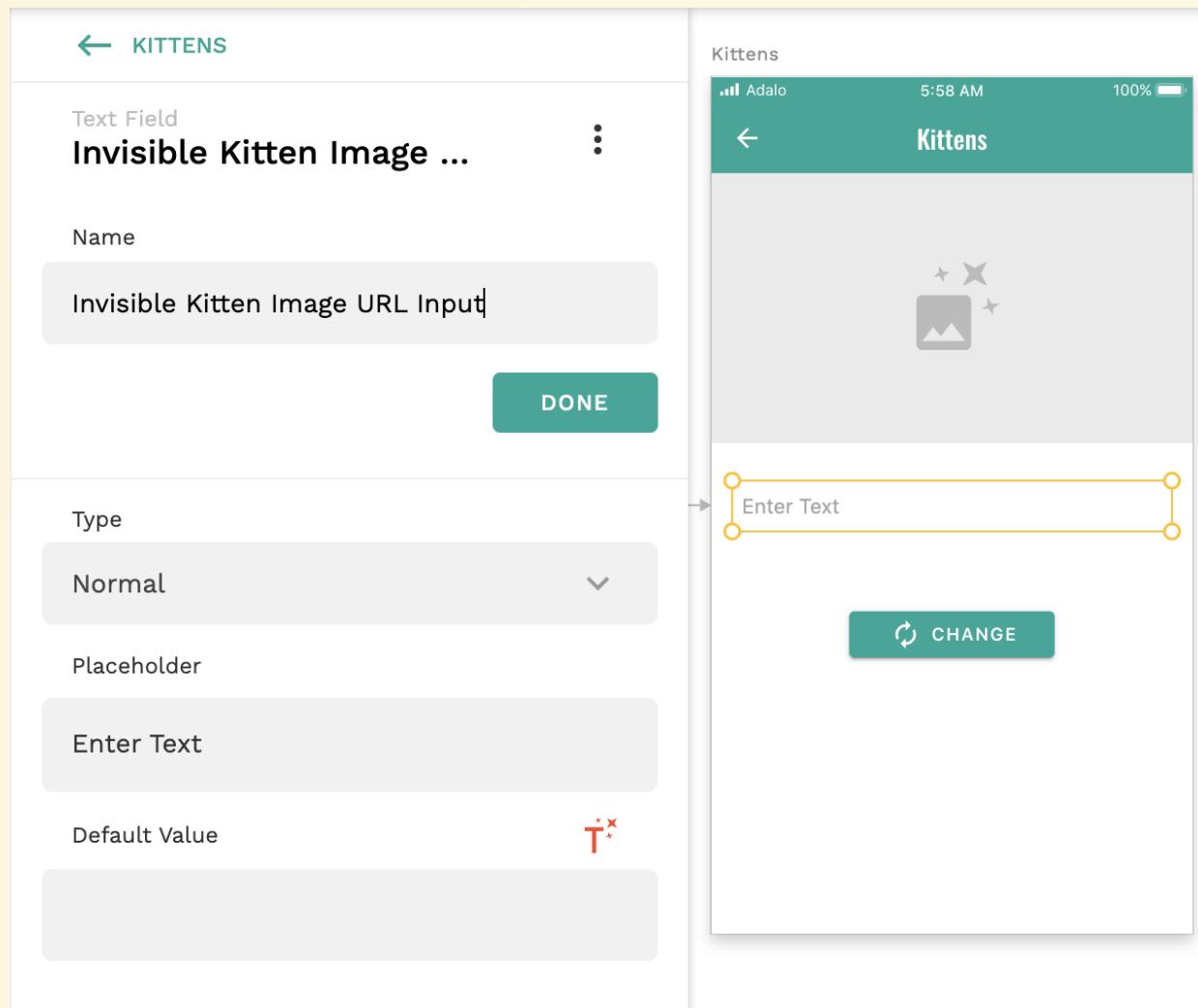
そのままでは、選択肢の中にAPIから取得したデータは出てきません。

A url

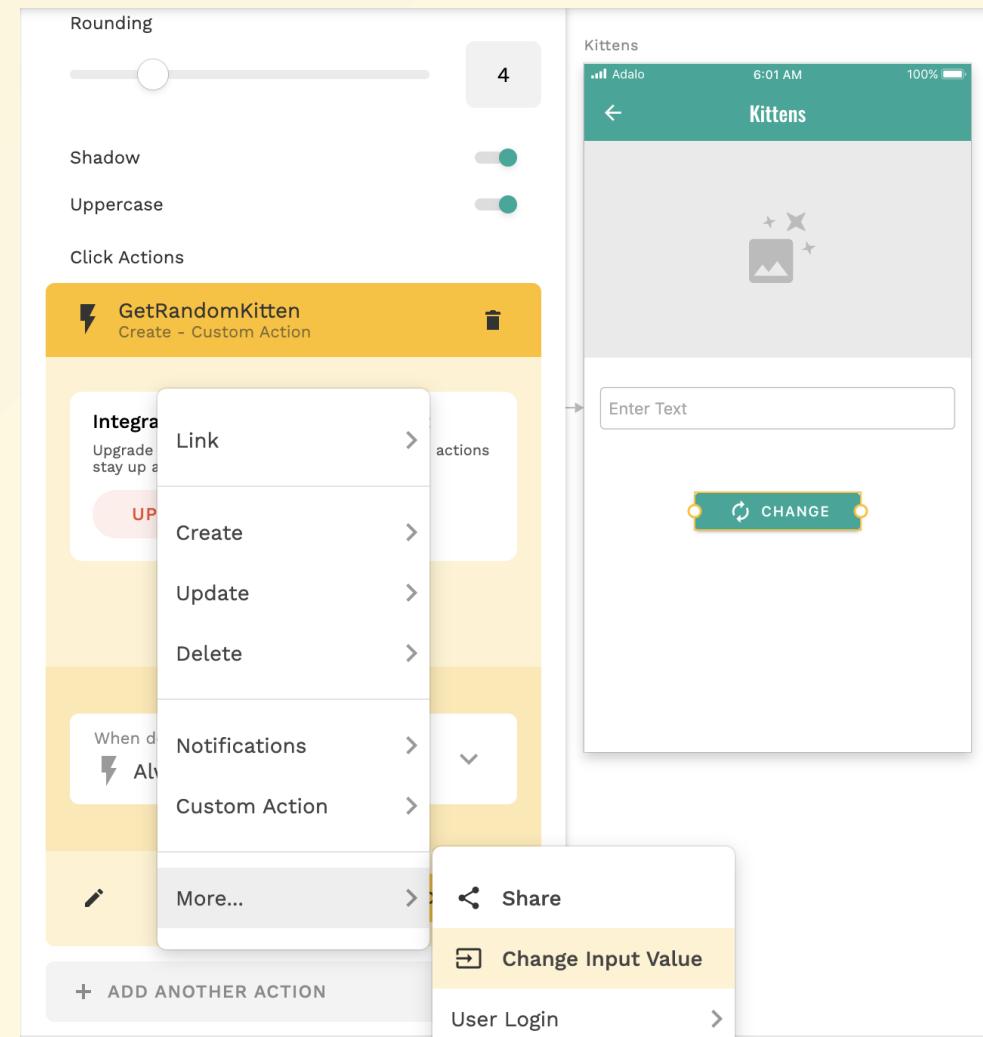
<https://cdn2.thecatapi.com/images/6k7.jpg>



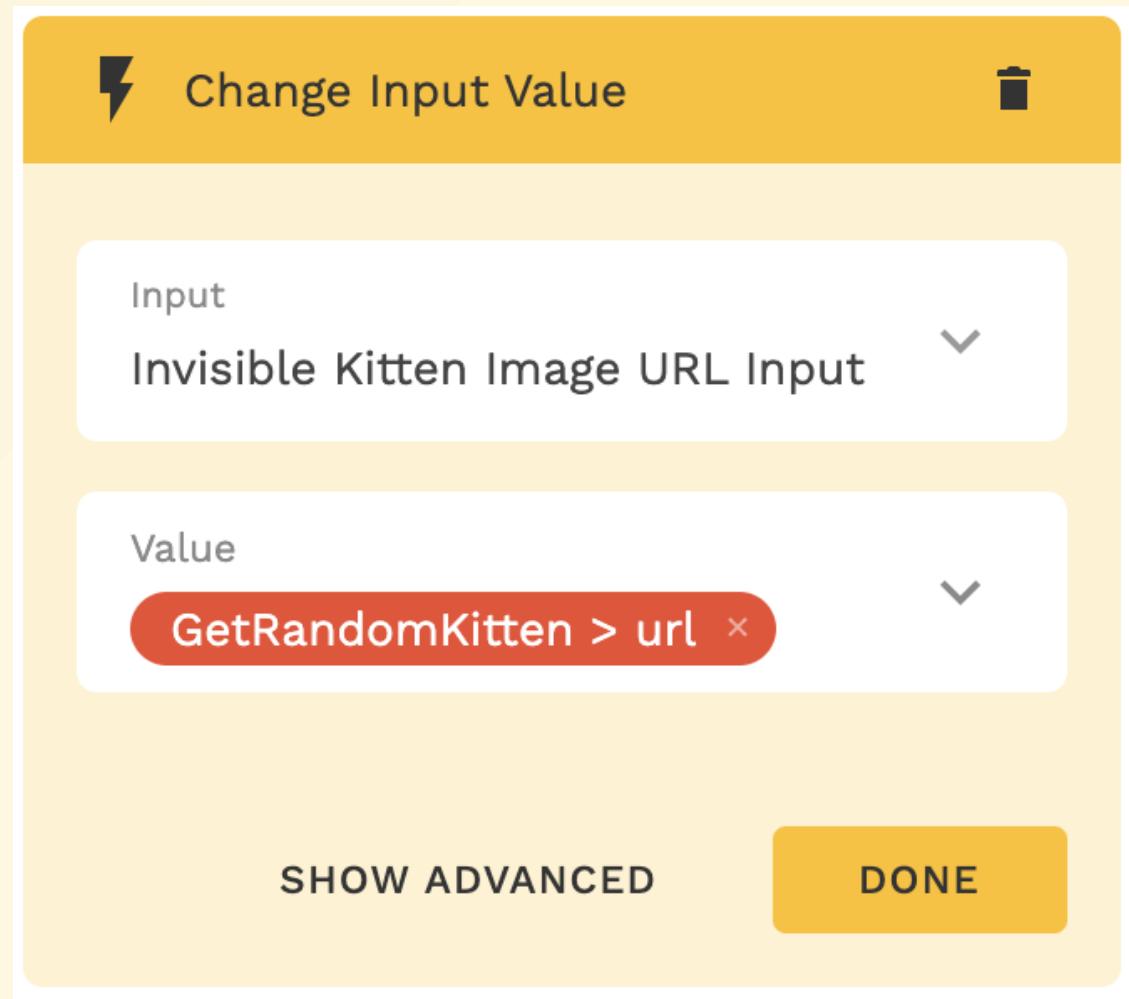
- TextInputコンポーネントを画面上に追加
- Nameを "Invisible Kitten Image URL Input" に変更



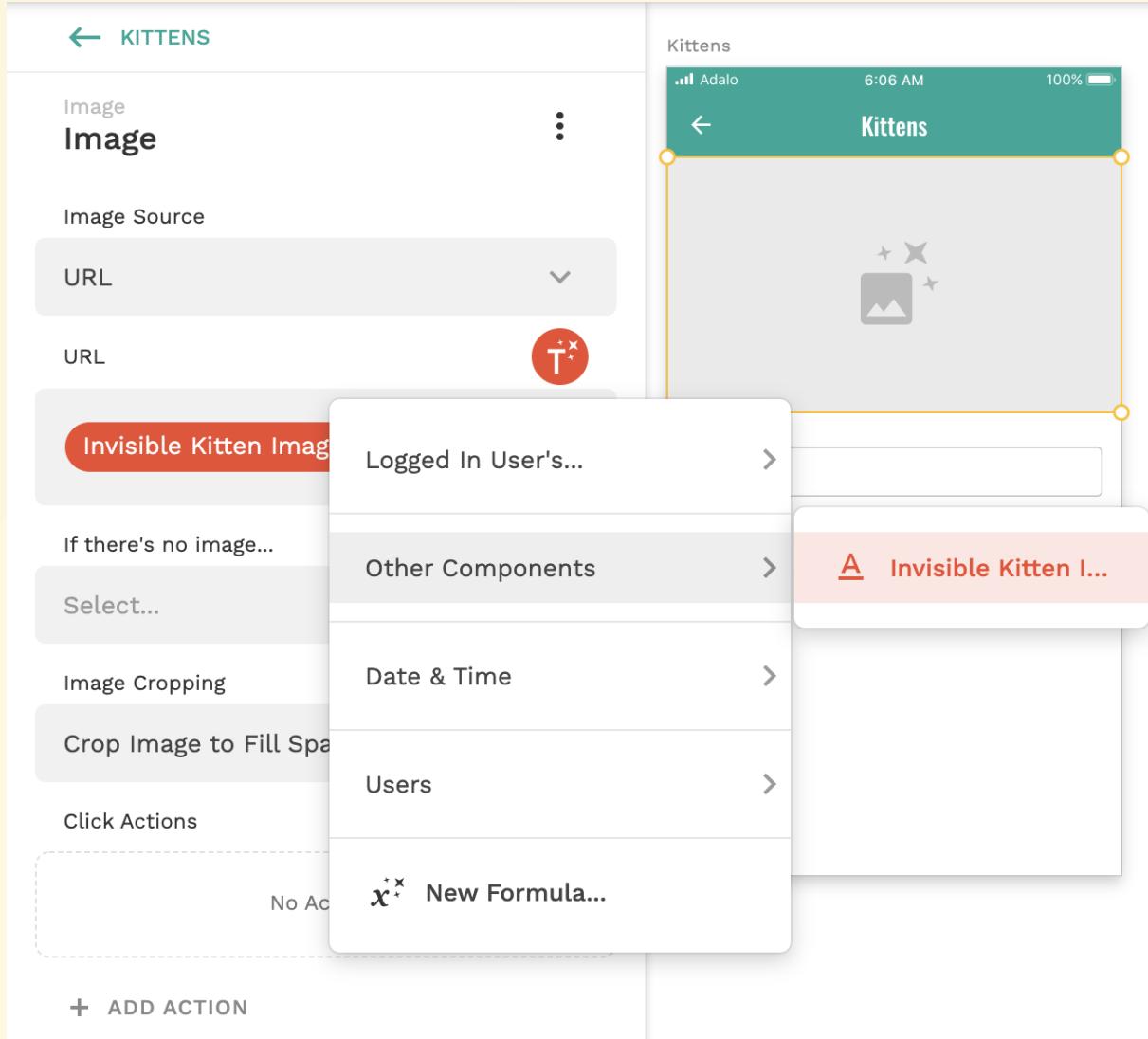
- "Change Kitten Image Button" をクリック
- ADD ANOTHER ACTION から Change Input Value を選択



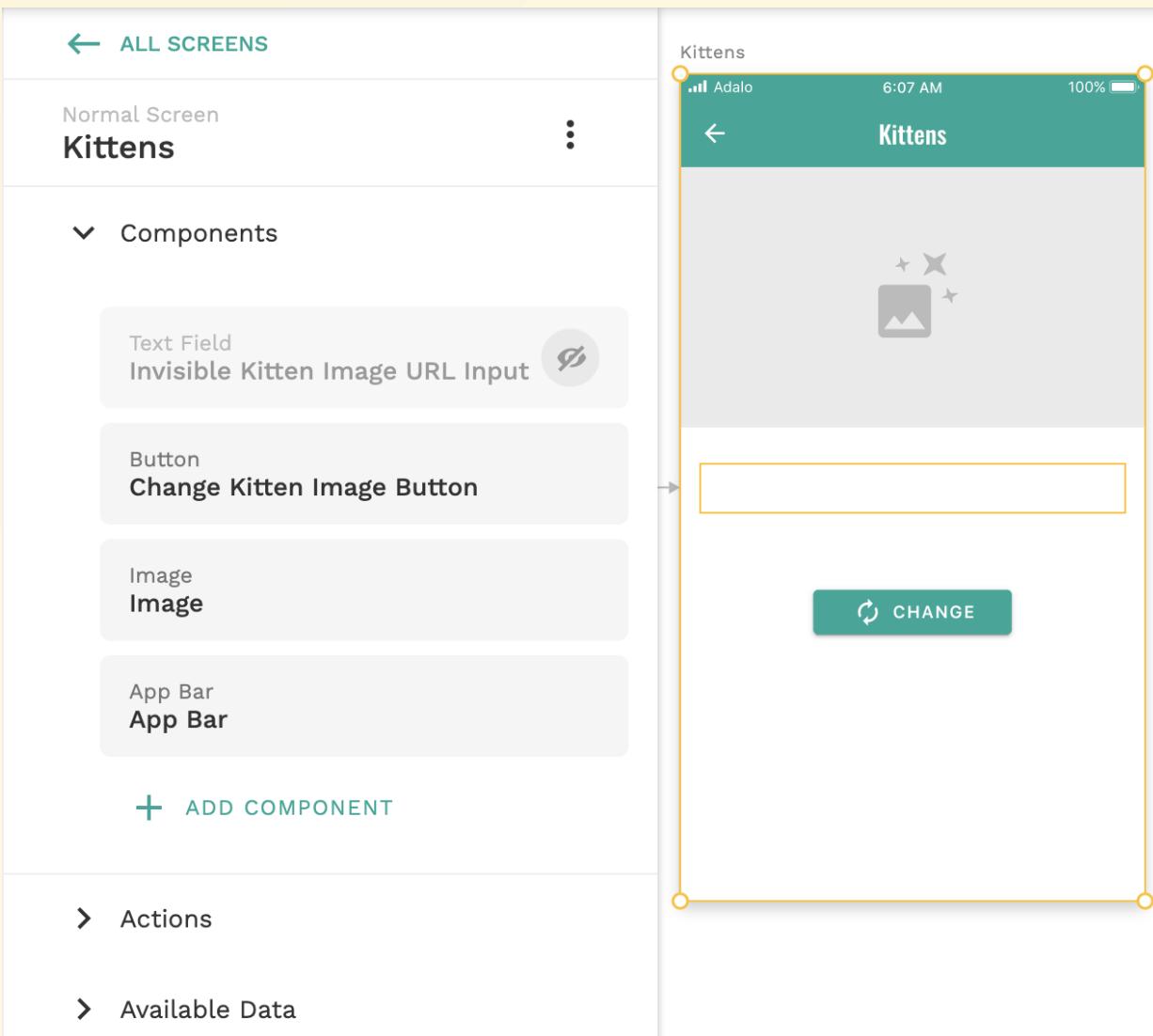
- Inputに "Invisible Kitten Image URL Input" を設定
- Valueに "GetRandomKitten > url" を設定
- "DONE" をクリック



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

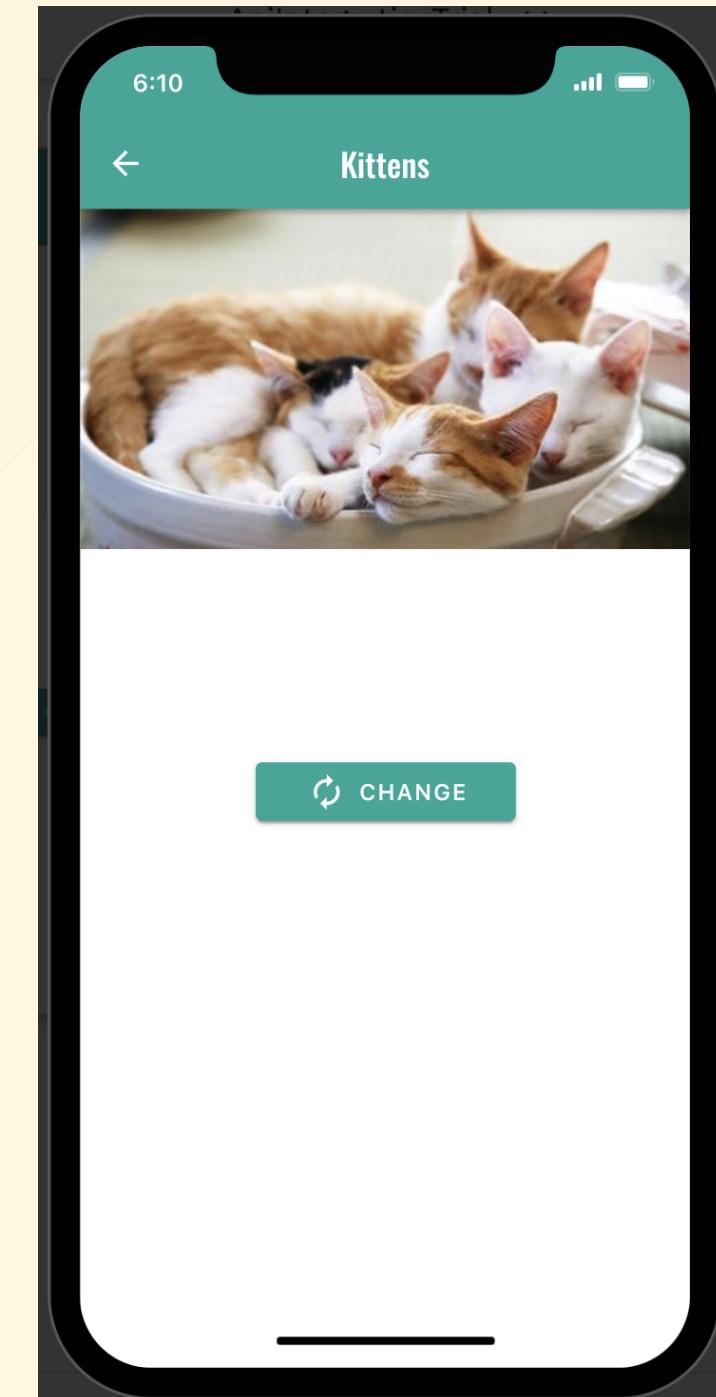


- 画面名 "Kittens" をクリック
- "Invisible Kitten Image URL Input" の右側の目のアイコンをクリックして非表示にする



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。

>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。

14日間のフリートライアルは、Development Phaseの前に終了します。

Custom Actionを多用する可能性がある場合、Development Phaseの作業開始前に新しくAdaloのアカウントを作成し、フリートライアルを新たに開始することをおすすめします。

## 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。

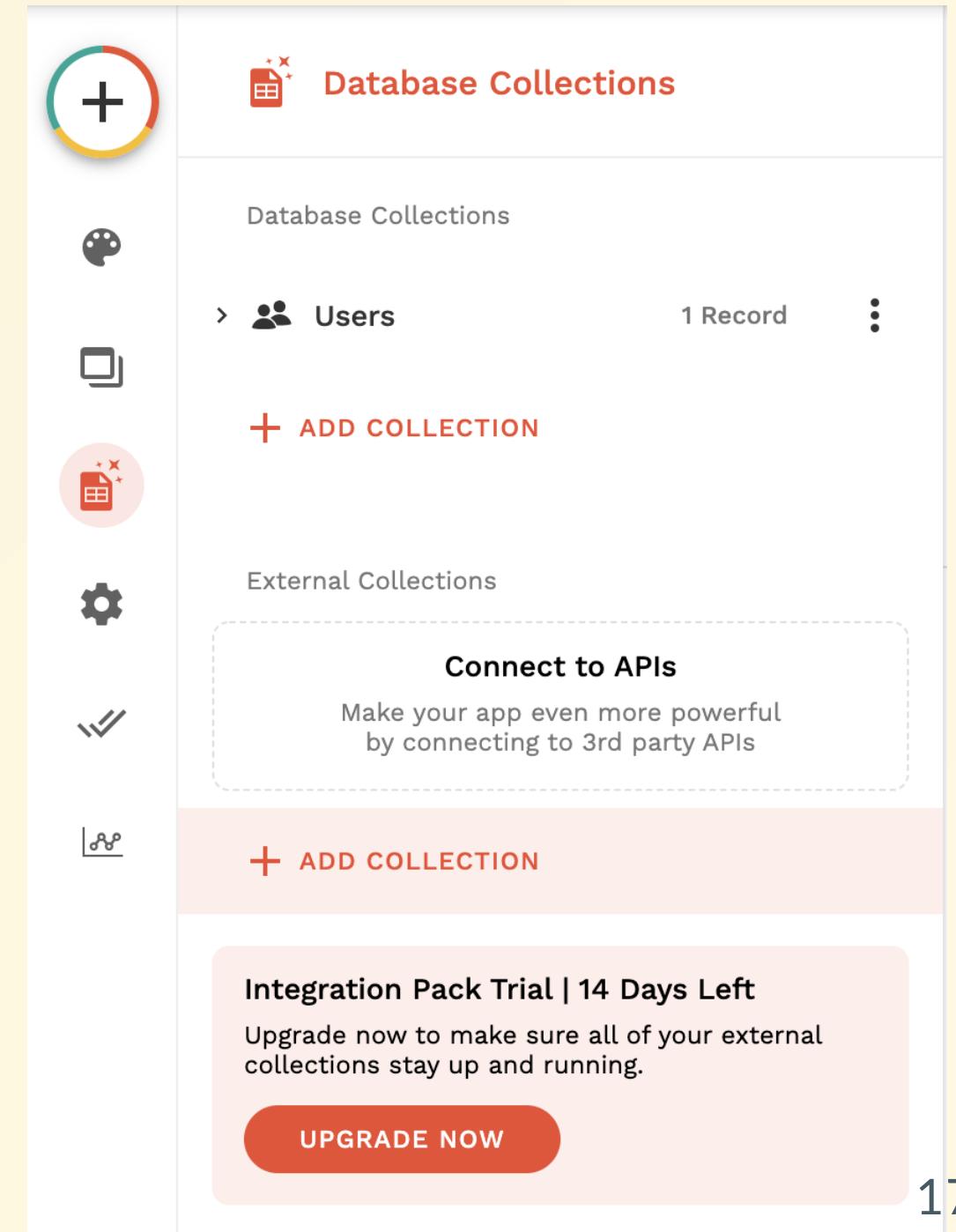
このAPIを使って、猫の品種の一覧を取得&表示しましょう。

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

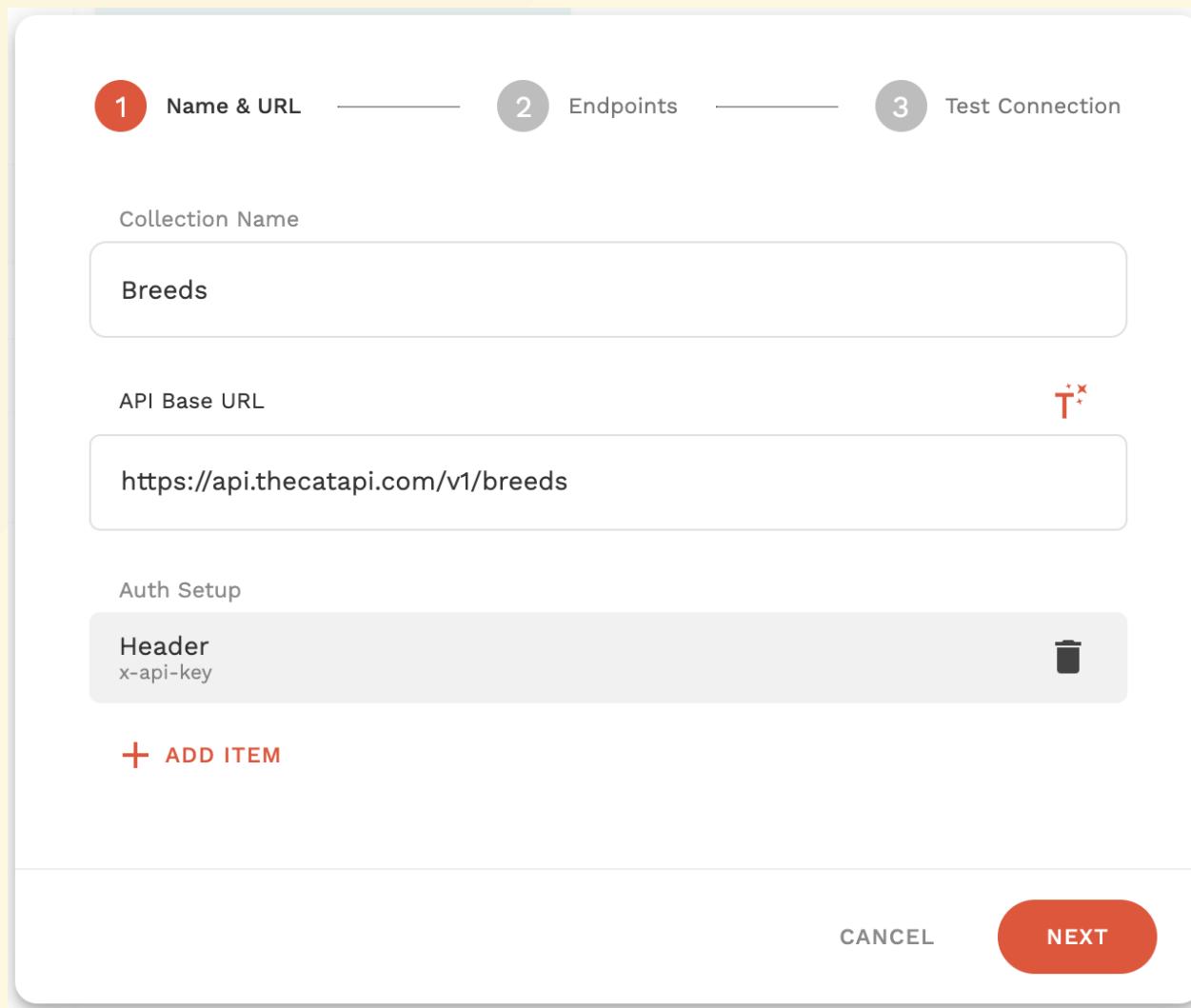
## ⌚ Send a Test Request

The screenshot shows the Postman interface for sending a test request. At the top, there is an orange 'Send' button, a 'GET' method indicator, and the URL `https://api.thecatapi.com/v1/breeds`. Below the URL, there are tabs for 'Settings', 'Headers [1]', 'Query [3]', and 'Code Generation'. The 'Headers [1]' tab is selected. Under 'Headers', there is a 'Add Header' button and a field containing the header `x-api-key`. To the right of the header field is a colon followed by the placeholder `{$$.env.x-api-key}`.

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



- Collection Name: Breeds
- Base URL:  
<https://api.thecatapi.com/v1/breeds>
- Auth Setup
  - Header x-api-key: 発行したAPI Key



1 Name & URL    2 Endpoints    3 Test Connection

Collection Name  
Breeds

API Base URL  
https://api.thecatapi.com/v1/breeds

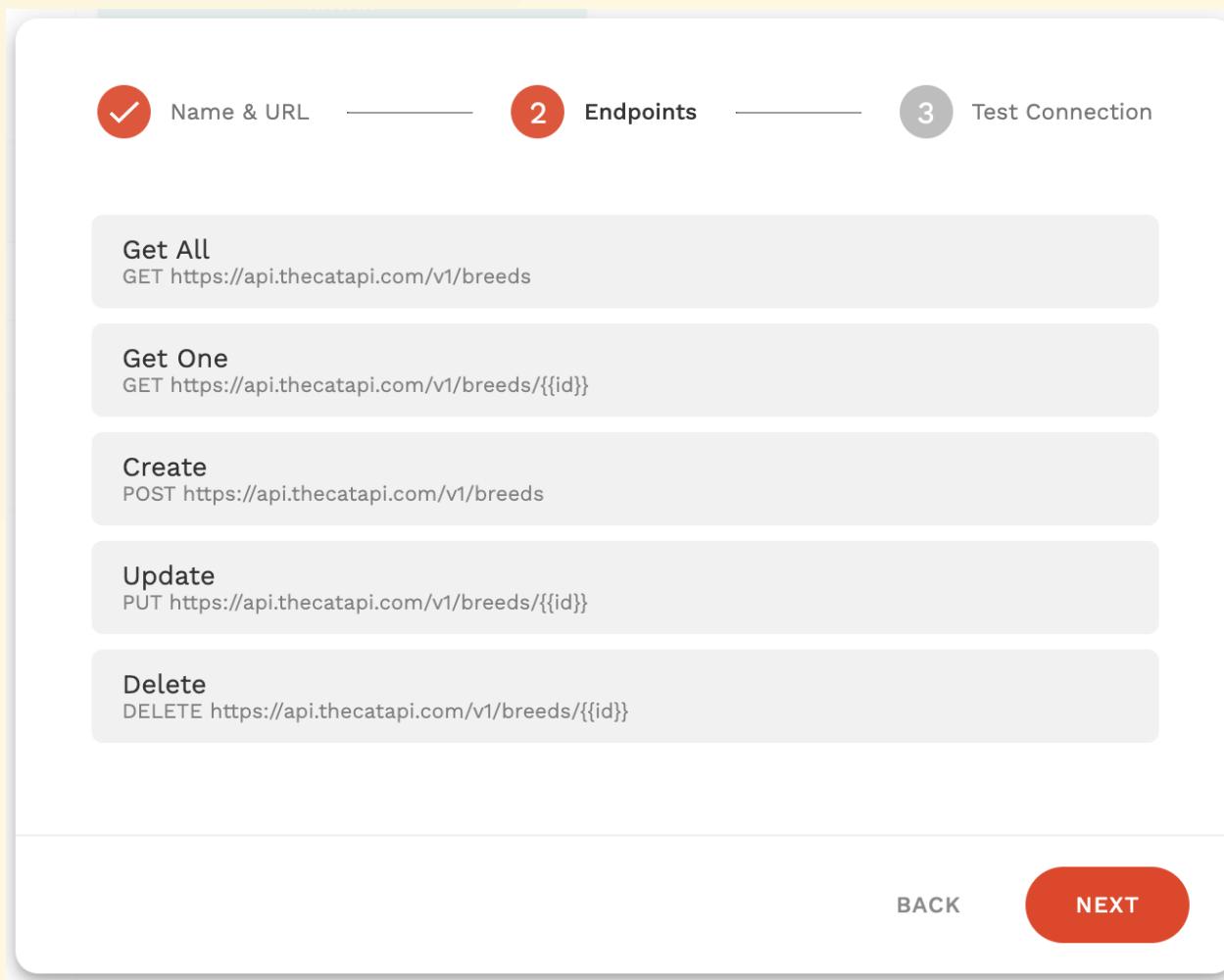
Auth Setup  
Header  
x-api-key

+ ADD ITEM

CANCEL    NEXT

Adaloでは、APIでアクセスするリソース(この例ではbreeds)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、"CREATE COLLECTION" をクリック

The screenshot shows a user interface for creating a collection in MongoDB. At the top, there are three status indicators: "Name & URL" (green checkmark), "Endpoints" (green checkmark), and "Test Connection" (red circle with the number 3). Below these, the text "Test Successful!" is displayed, followed by the message "Here's the full response we got back:". A JSON document is shown, representing the response from the test connection. The JSON is a single object with a key-value pair where the value is an array containing one element, which is another object representing the Abyssinian cat breed. The JSON is truncated at the bottom.

```
[{"id": "abys", "name": "Abyssinian", "weight": {"imperial": "7 - 10", "metric": "3 - 5"}, "cfa_url": "http://cfa.org/Breeds/BreedsAB/Abyssinian.aspx", "vetstreet_url": "http://www.vetstreet.com/cats/abyssinian", "vcahospitals_url": "https://vcahospitals.com/know-your-pet/cat-breeds/temperament", "temperament": "Active, Energetic, Independent, Intelligent, Gentle", "origin": "Egypt", "country_codes": "EG", "country_code": "EG", "description": "The Abyssinian is easy to care for, and a joy to have in your home.", "life_span": "14 - 15", "indoor": 0, "lap": 1, "alt_names": "", "adaptability": 5, "affection_level": 5, "child_friendly": 3, "dog_friendly": 4, "energy_level": 5, "grooming": 1, "health_issues": 2, "intelligence": 5, "shedding_level": 2, "social_needs": 5, "stranger_friendly": 5, "vocalisation": 1, "experimental": 0}
```

BACK CREATE 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。

External Collections

▼  Breeds 

A id

A weight > imperial

A weight > metric

A name

A temperament

A origin

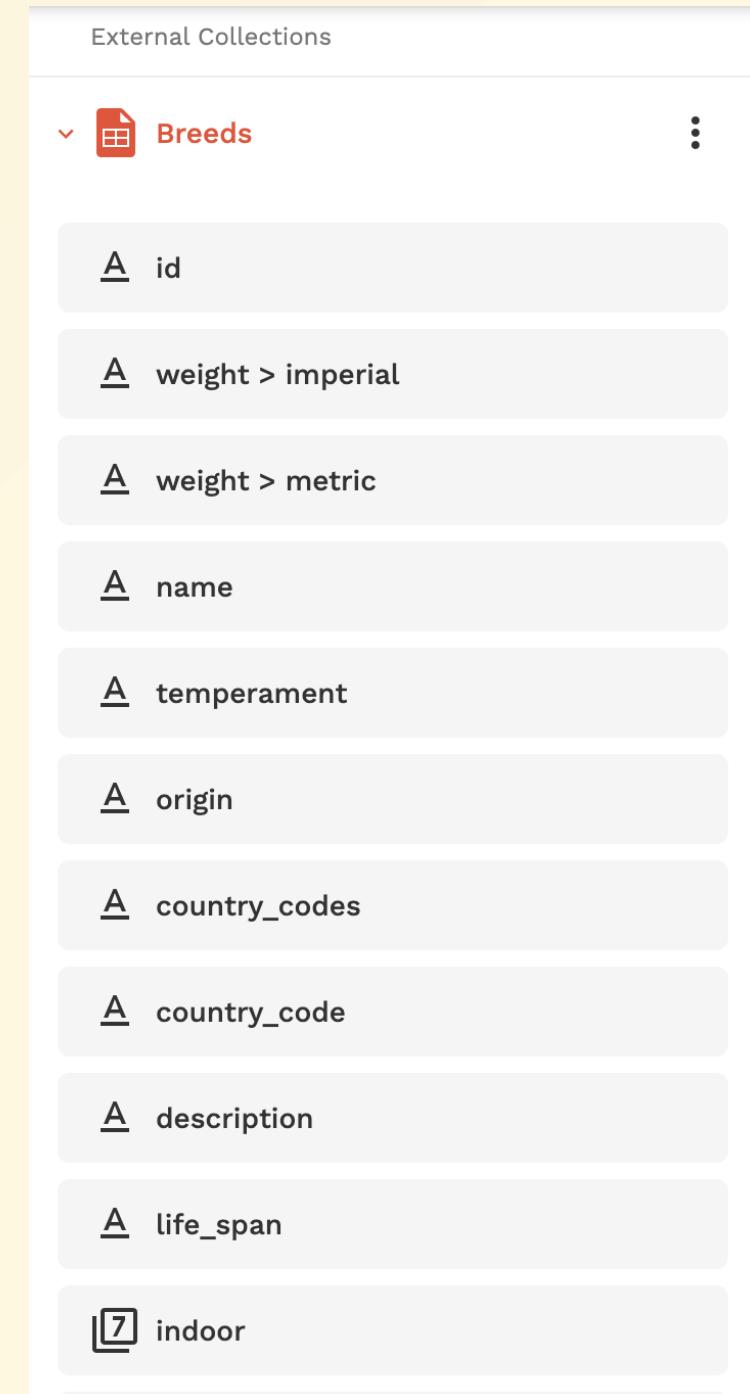
A country\_codes

A country\_code

A description

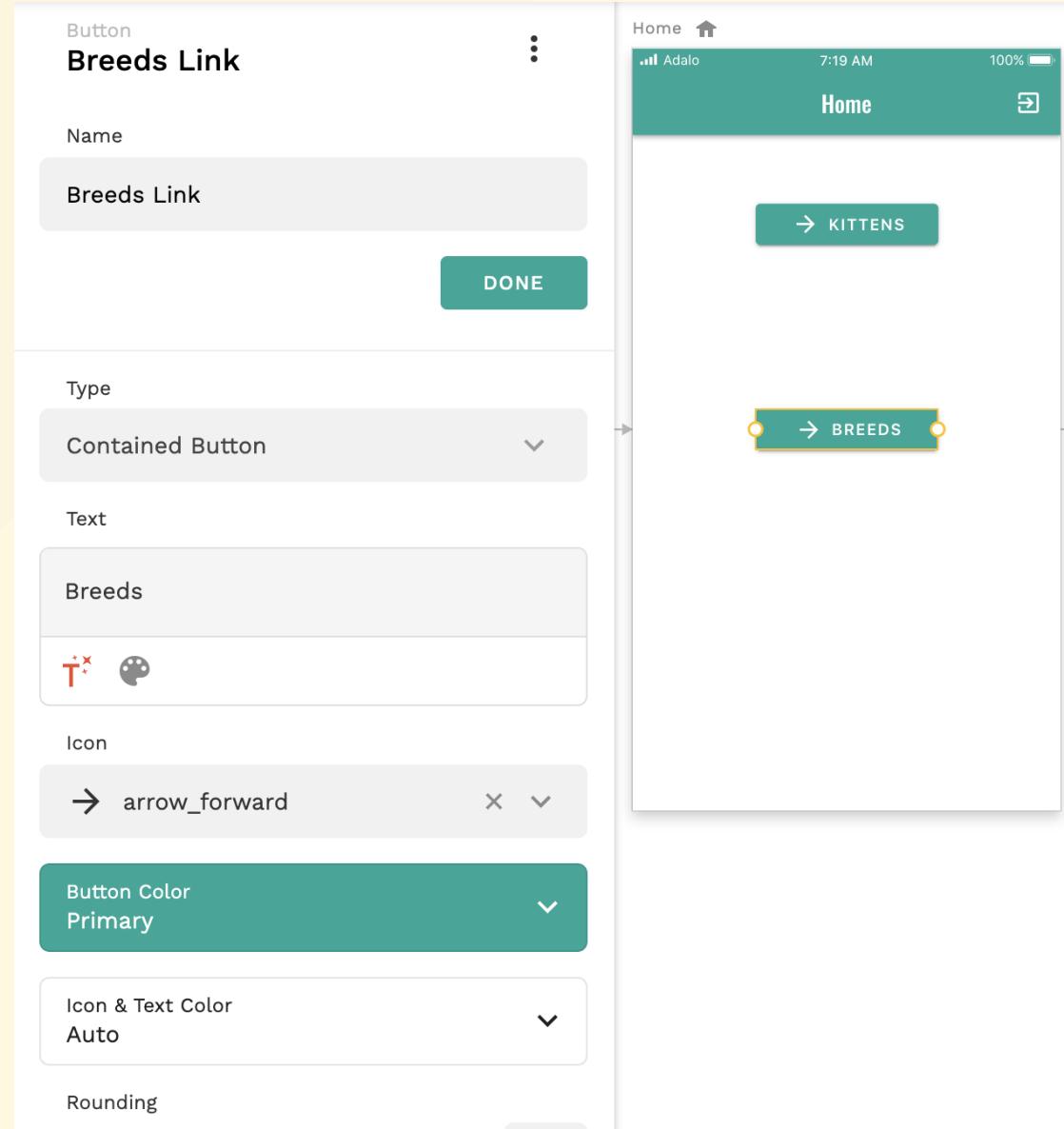
A life\_span

 7 indoor

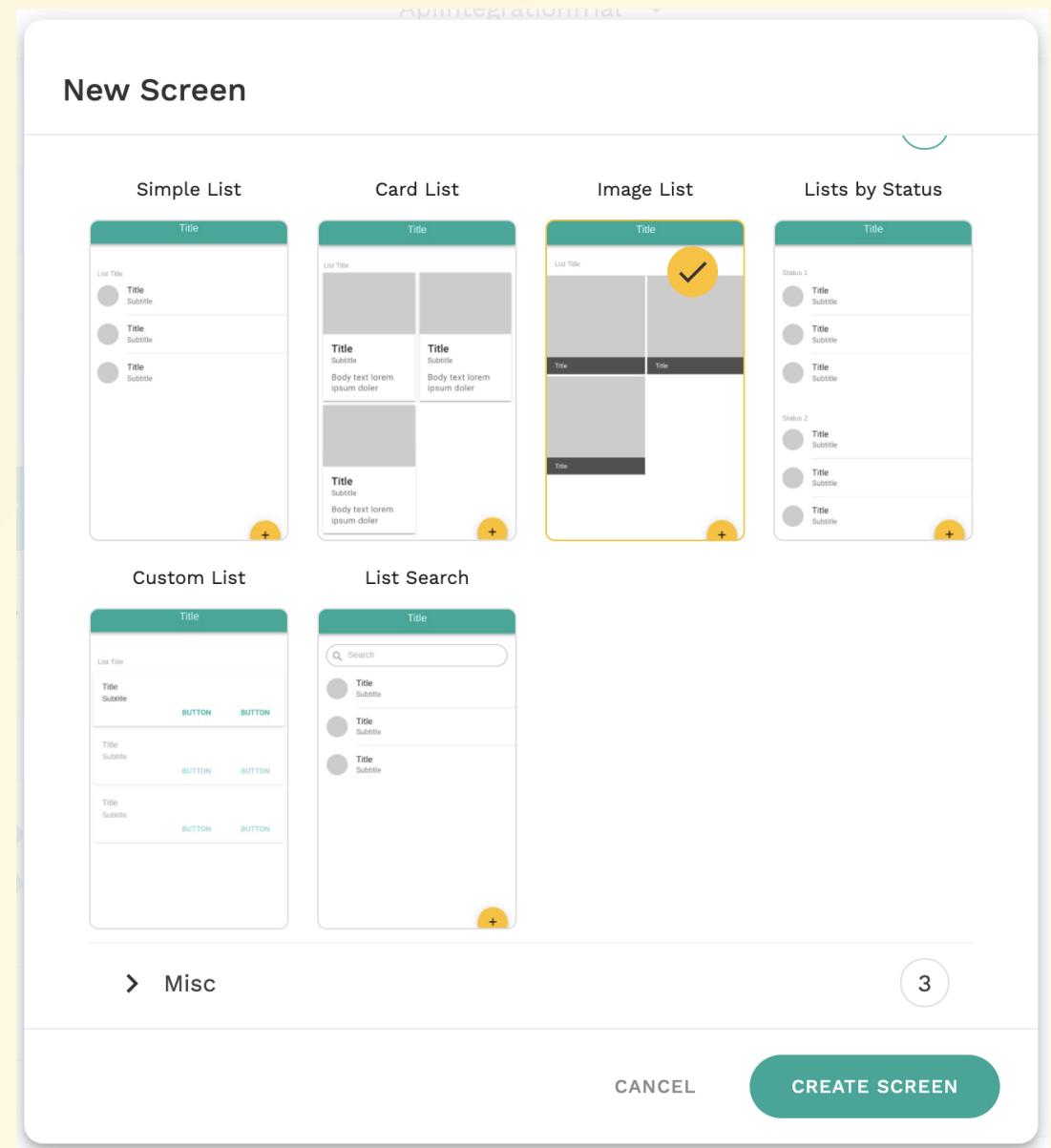


取得したデータを一覧表示しましょう。

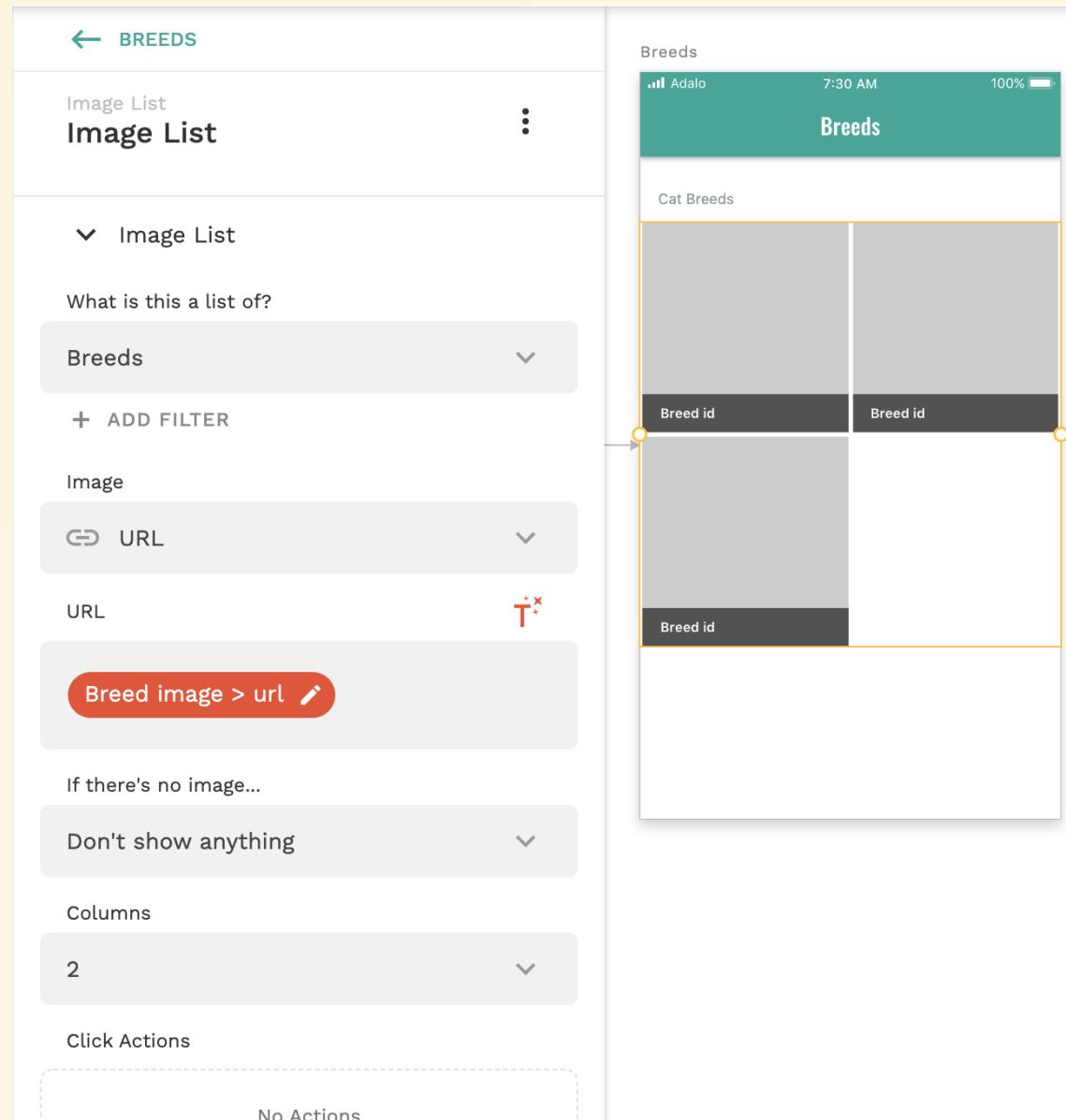
- Home画面に "Breeds Link" ボタンを追加
- ADD ACTIONからNEW SCREENへのLINKを追加



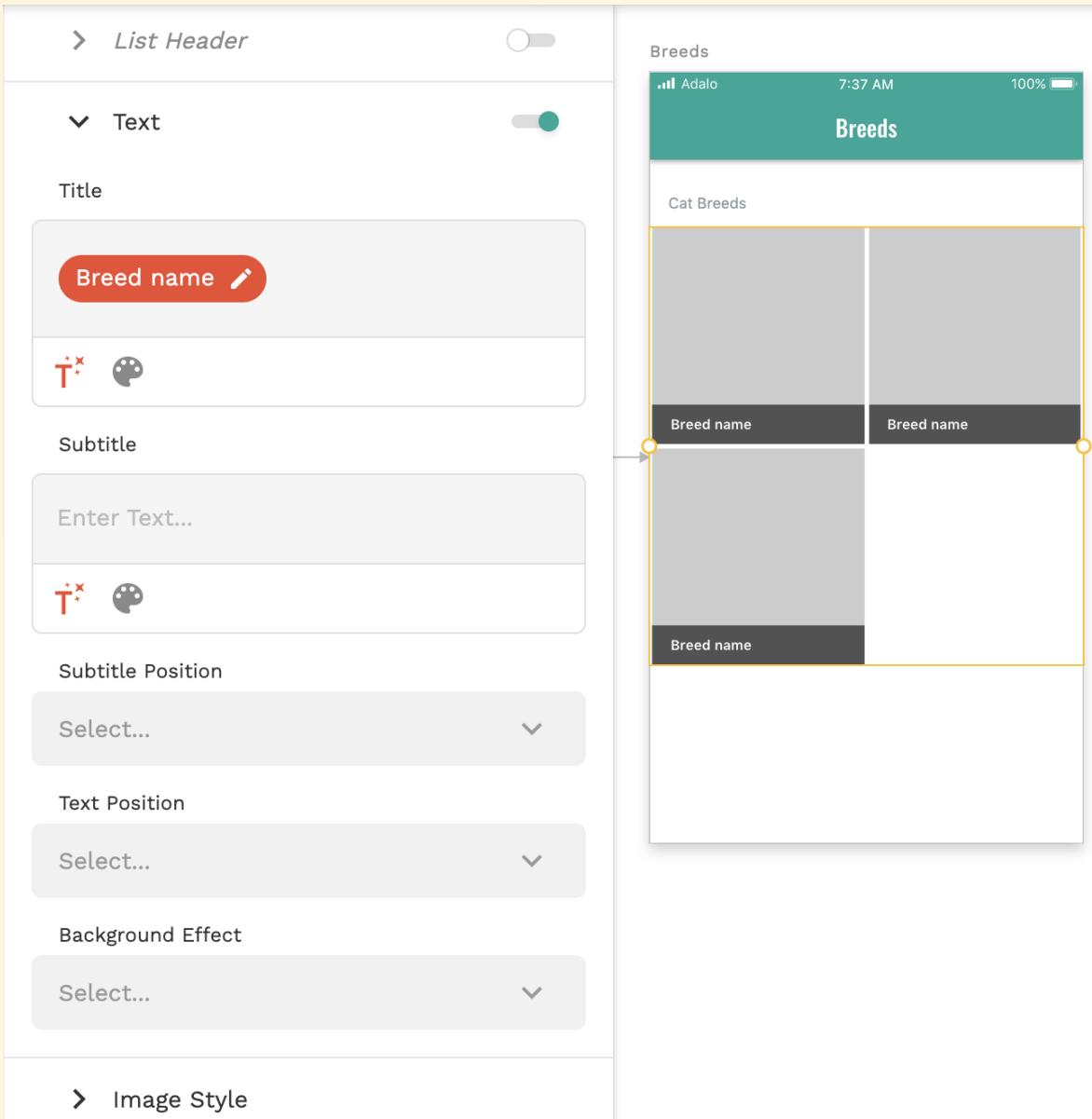
- Nameに "Breeds"と入力
- TemplateでImage Listを選択
- CREATE SCREEN をクリック



- List TitleのTextを"Cat Breeds"に変更
- Image ListをBreeds Collectionのリストとして設定
- ImageのURLに "image > url" を設定
- "If there's no image..." に "Don't show anything" を設定
- 右下の+ボタンは不要なので削除

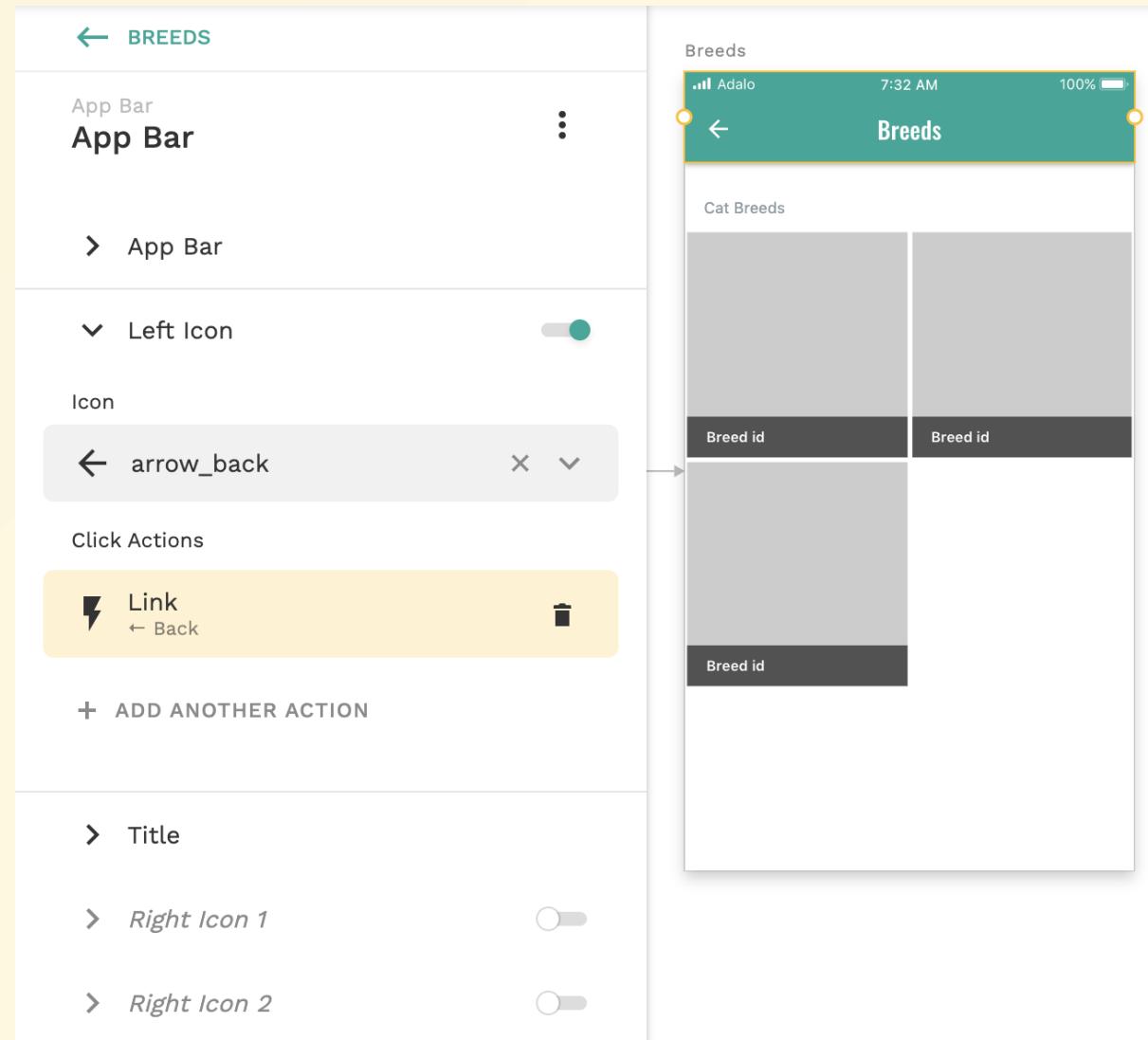


- Textを "name" に変更

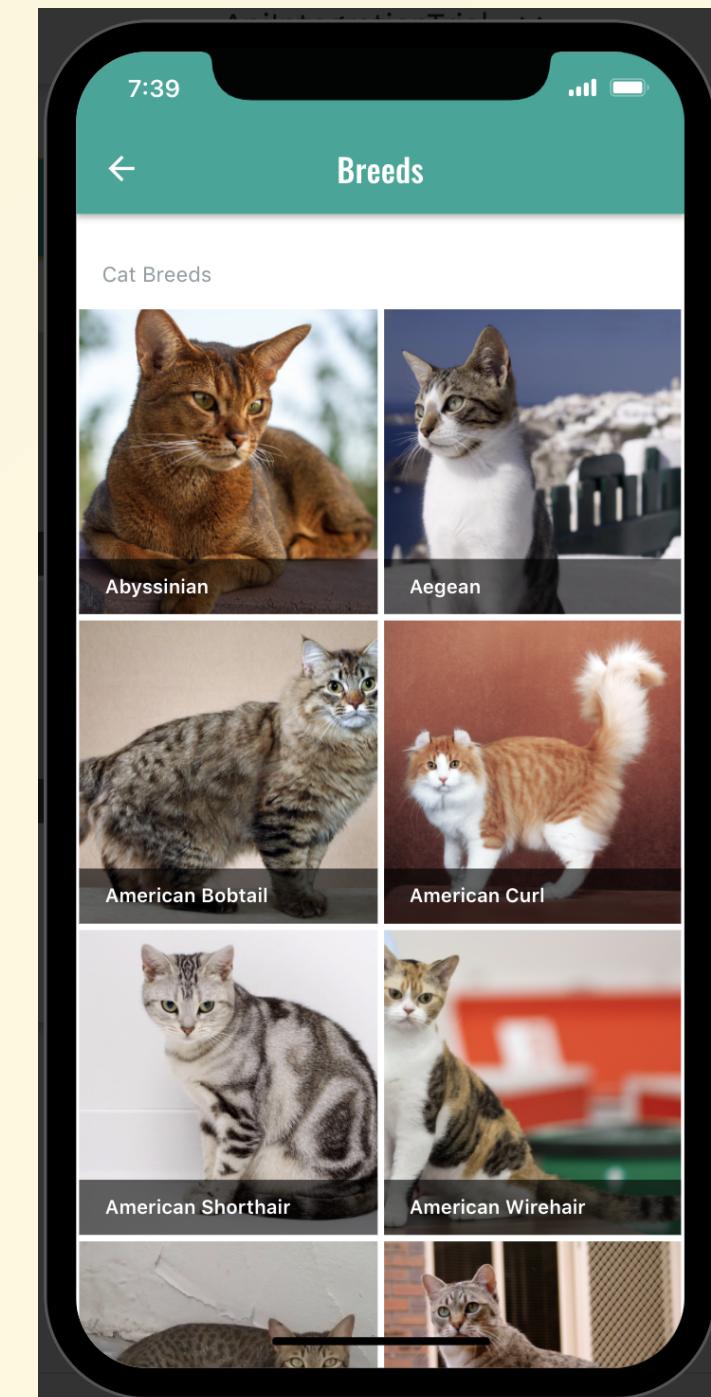


戻るアイコンが非表示になつ  
ているので、表示します。

- AppBarをクリック
- Left IconのトグルをONに変更



Preview機能で確認すると、猫の品種の一覧が表示されました。

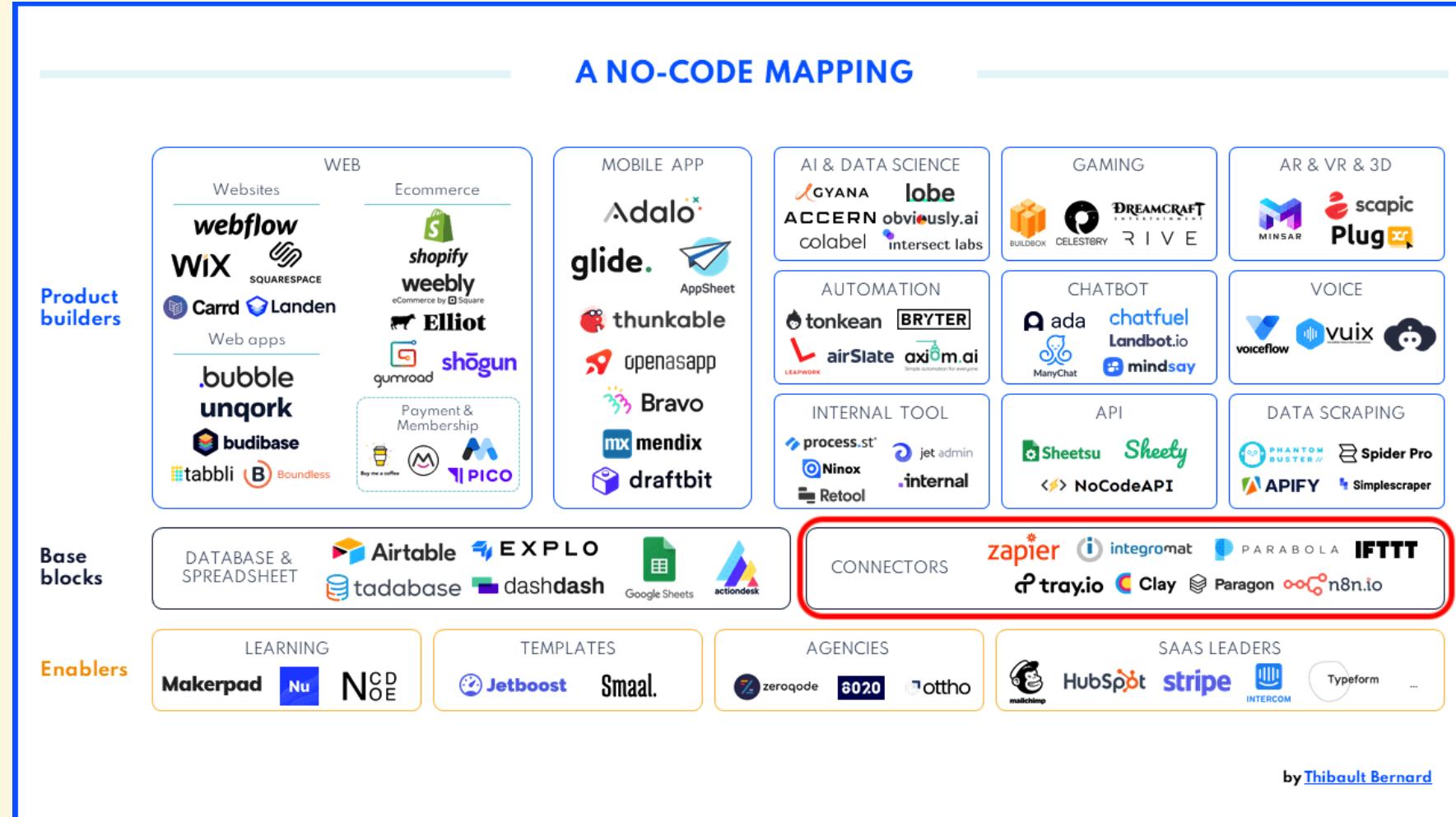


## 連携サービス

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Zapierというサービスを紹介します。



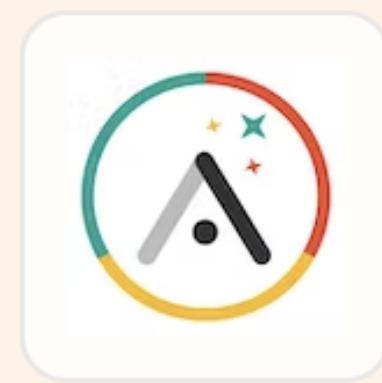
# Zapier

- Zapierを使えば、案内に従って操作することで、簡単に外部サービスを連携させられます。試してみましょう。

The screenshot shows the Zapier website with a specific integration highlighted. At the top, there's a navigation bar with links for Product, Solutions, Resources & Support, Pricing, Contact Sales, Log in, and a prominent orange "Sign up" button. Below the navigation, a teal sidebar on the left features the Adalo logo and the text "Adalo Integrations App Builder". The main content area has tabs for "Integrations" (which is selected) and "Help". A section titled "Do more with Adalo integrations" explains that Zapier lets you connect Adalo with thousands of popular apps to automate work. It includes a large orange button labeled "Connect Adalo to 5,000+ apps". Below this, a horizontal line of logos for various companies (Slack, Meta, Shopify, Dropbox, Asana) indicates their trust in Zapier. At the bottom, there's a search bar.

Adaloにはメール送信機能がありません。

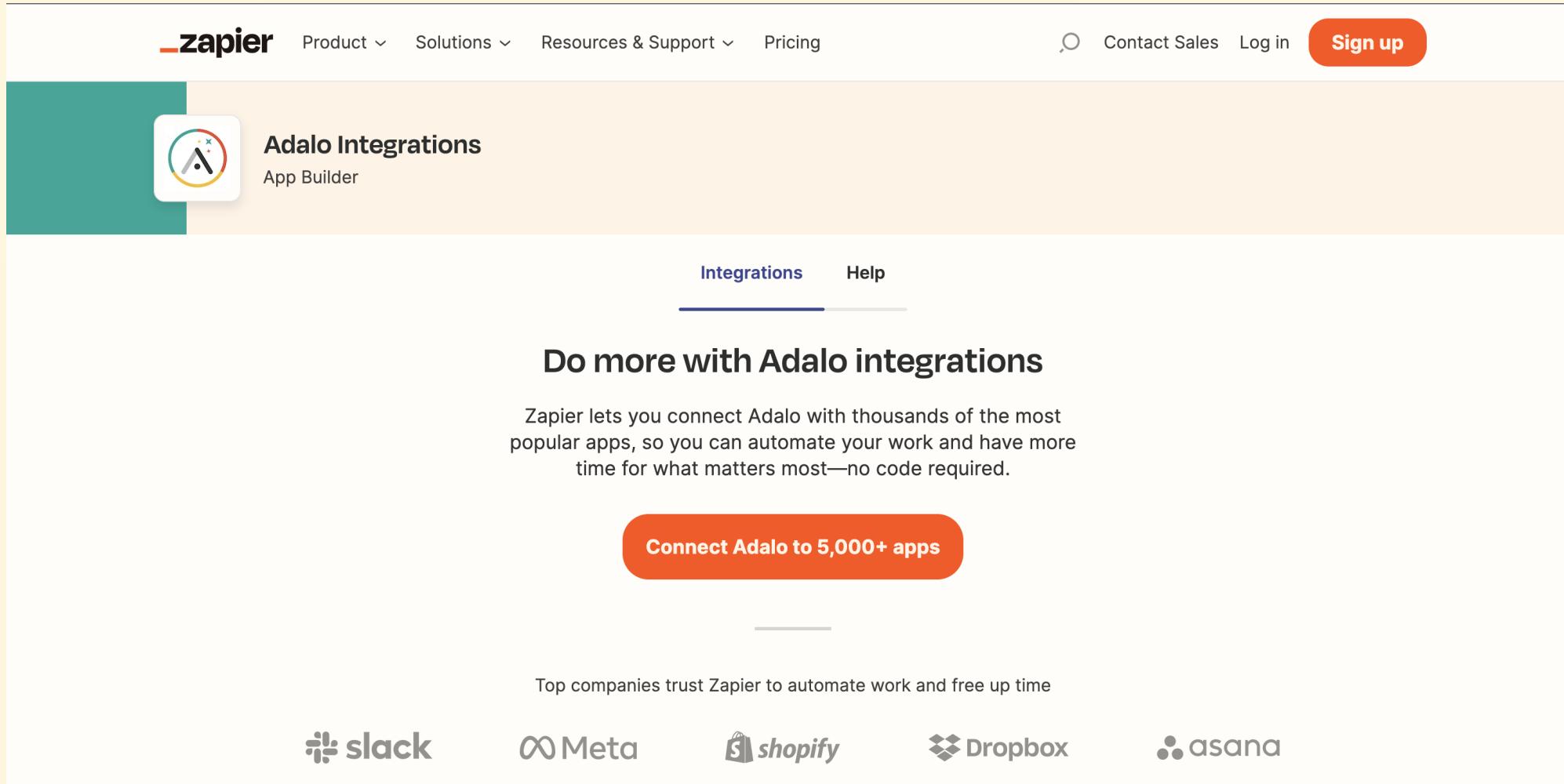
今回は、Zapierを使って、  
AdaloとGmailを連携させることで、アプリへSignUpした人に自動でWelcomeメールが送信されるようにしていきます。



- 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、  
<https://zapier.com/apps/adalo/integrations> にアクセスして、"Connect Adalo to 5,000+ apps" をクリック



The screenshot shows the Zapier website with the Adalo Integrations page open. The top navigation bar includes links for Product, Solutions, Resources & Support, Pricing, Contact Sales, Log in, and a prominent orange "Sign up" button. On the left, there's a teal sidebar featuring the Adalo logo and the text "Adalo Integrations App Builder". The main content area has a yellow gradient background. It features a large orange button at the top with the text "Do more with Adalo integrations" and "Connect Adalo to 5,000+ apps". Below this, a section titled "Top companies trust Zapier to automate work and free up time" lists logos for Slack, Meta, Shopify, Dropbox, and Asana.

zapier Product Solutions Resources & Support Pricing Contact Sales Log in Sign up

Adalo Integrations  
App Builder

Integrations Help

Do more with Adalo integrations

Zapier lets you connect Adalo with thousands of the most popular apps, so you can automate your work and have more time for what matters most—no code required.

Connect Adalo to 5,000+ apps

Top companies trust Zapier to automate work and free up time

slack Meta Shopify Dropbox asana

- 好きな方法でSign upしてください

The screenshot shows the Zapier sign-up page. At the top left is the Zapier logo. Top right features links for "Contact Sales" and "Log in". Below the header is a large promotional message: "Join millions worldwide who automate their work using Zapier." To the right of this message is a sign-up form. The form includes three social login buttons: "Sign up with Google" (blue), "Sign up with Facebook" (dark blue), and "Sign up with Microsoft" (black). Below these is a horizontal line with the text "OR". Underneath "OR" are fields for "Work Email (required)" and "First Name (required) / Last Name (required)". Each of these fields has a corresponding input box: "Work Email", "First Name", and "Last Name". At the bottom of the form is a large orange button labeled "Get Started Free". A small note at the bottom states: "By signing up, you agree to Zapier's [terms of service](#) and [privacy policy](#)".

- ロールと従業員の数を入力して、"Continue"をクリック

Hey Ryo, help us customize your experience by  
telling us a bit about yourself

Select the role that best describes you (required)

IT

Data Science

Customer Support Ops

Sales Ops

Project Management

HR & Recruiting Ops

Marketing & Marketing Ops

Engineering

Business Owner

Other

How many employees does your company have? (required)

1-49

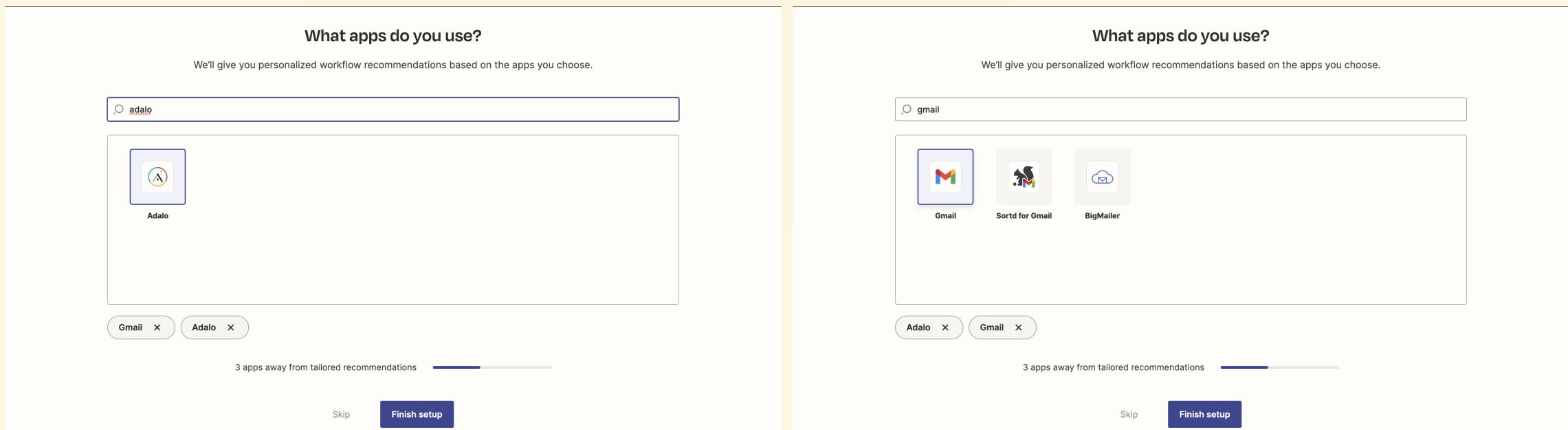
50-249

250-999

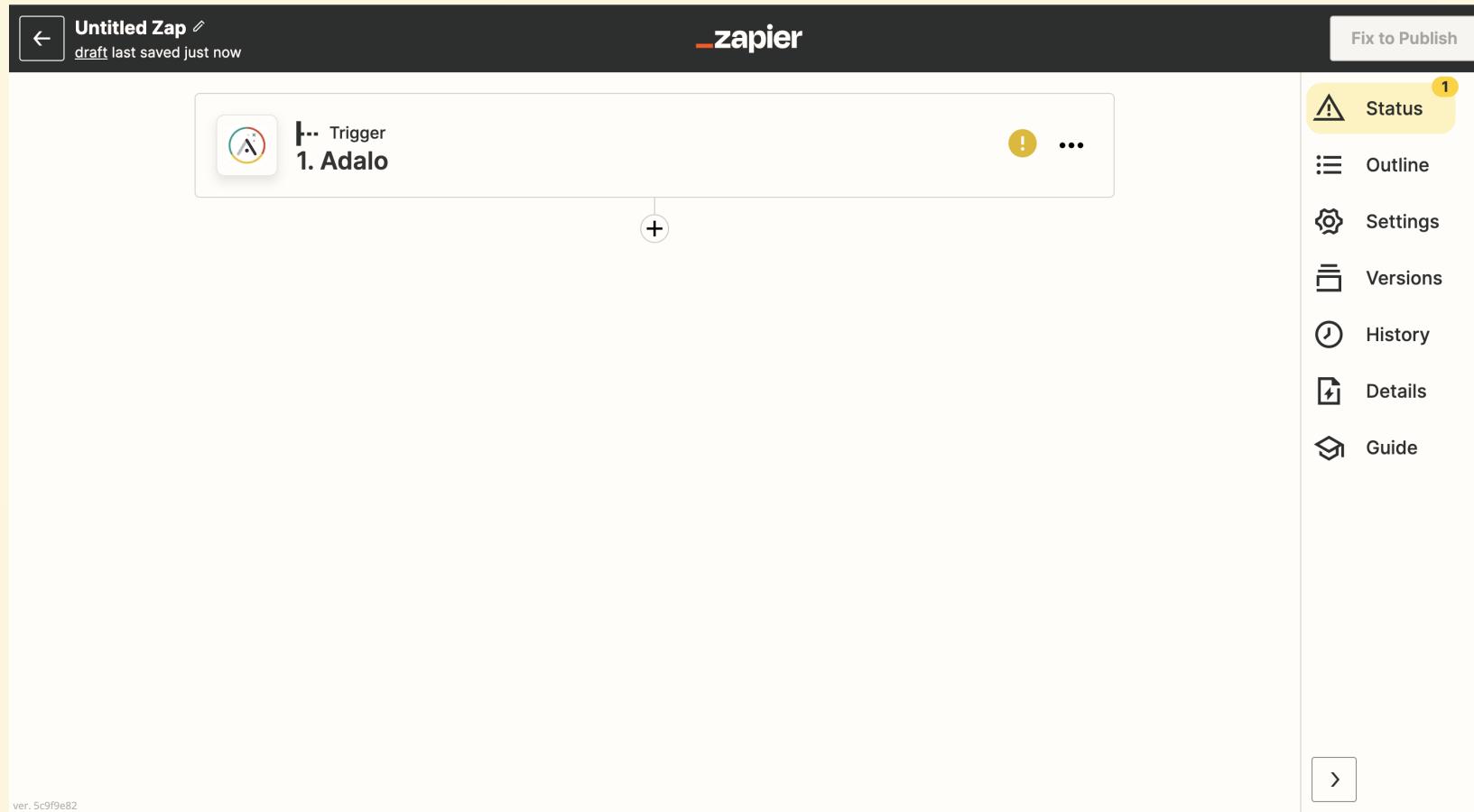
1,000+

Continue

- 使用するアプリにAdaloとGmailを追加して、"Finish setup"をクリック

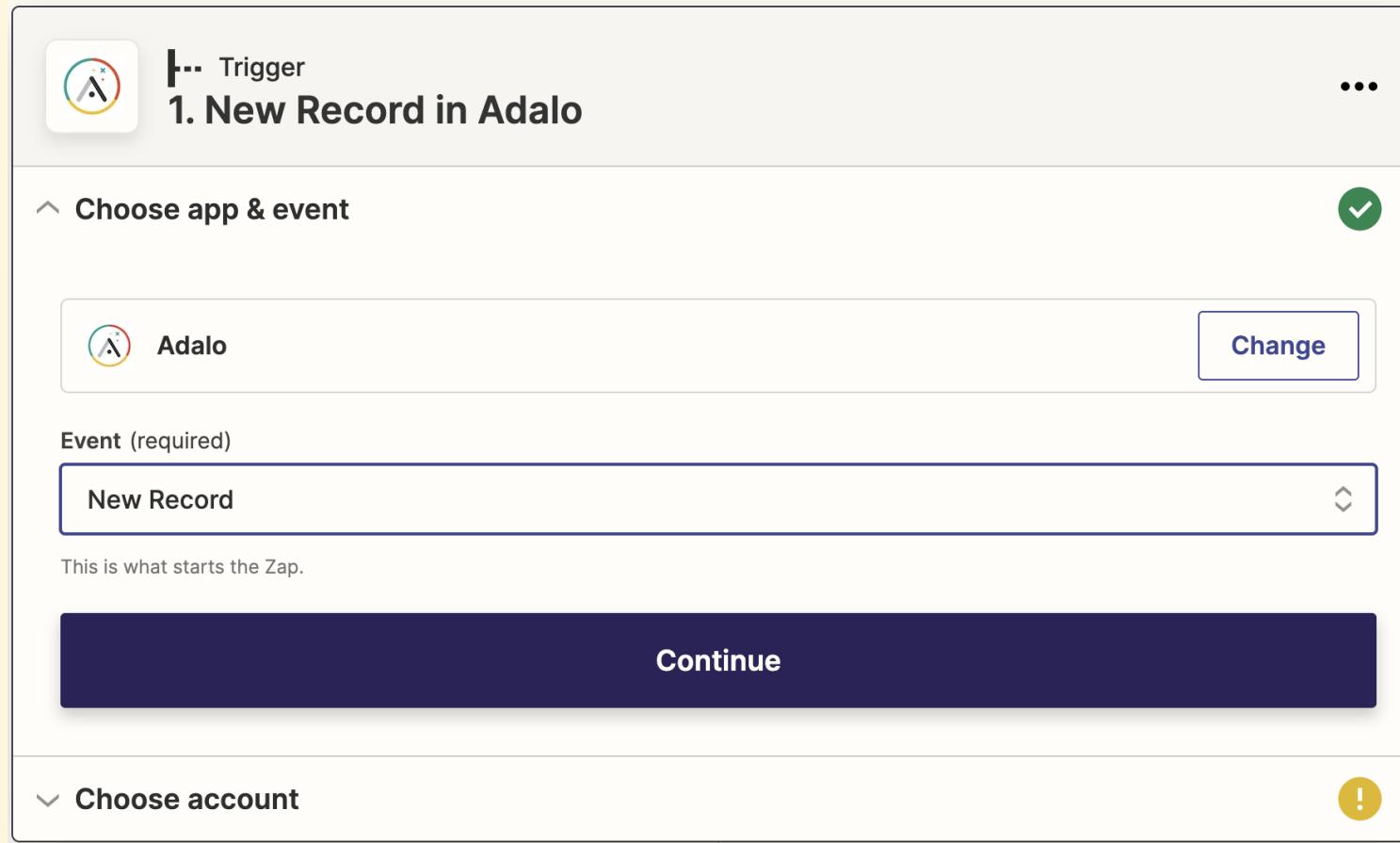


Zapと呼ばれる、サービス連携設定の編集画面が表示されます。

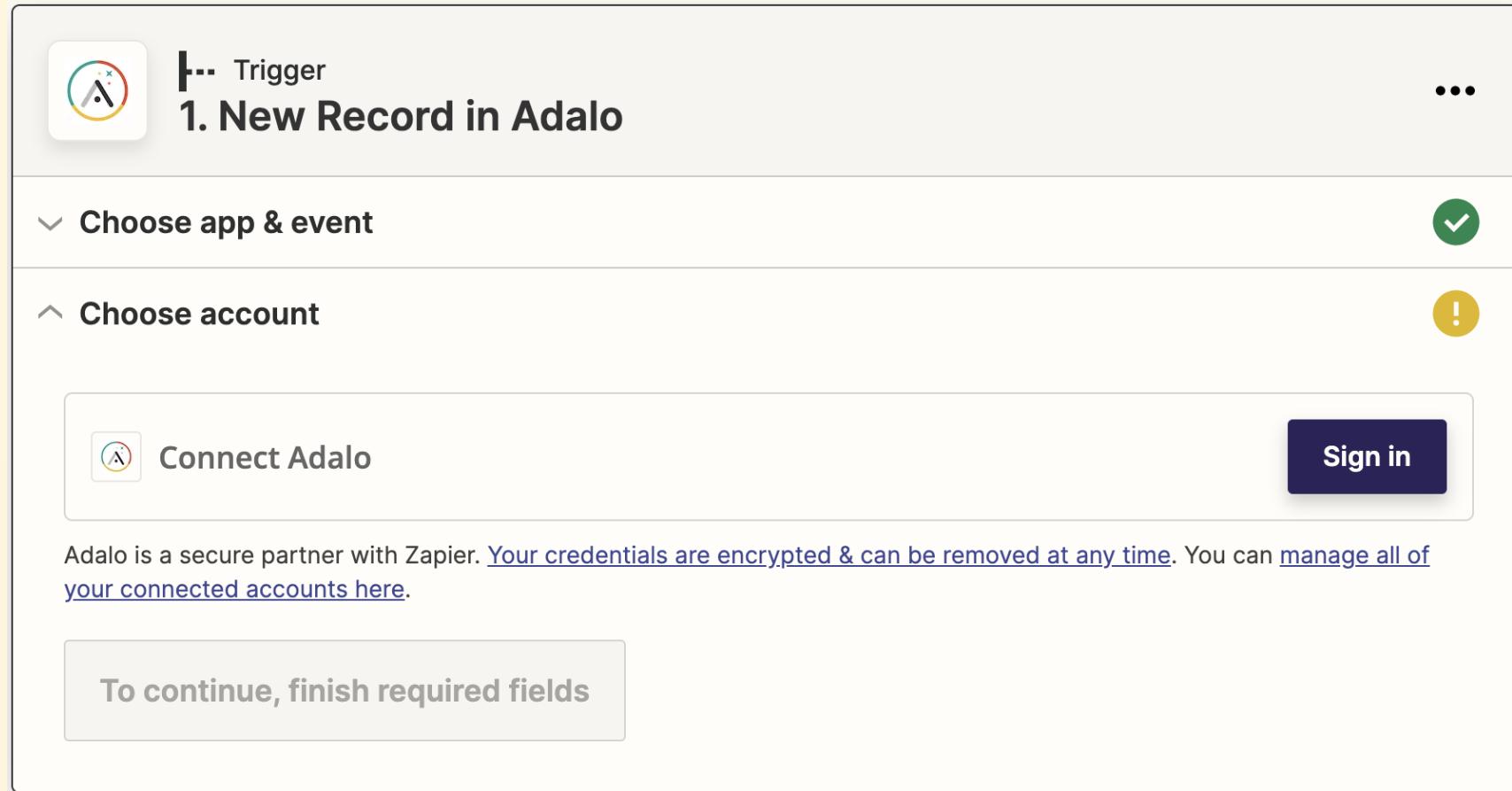


ver. 5c9f9e82

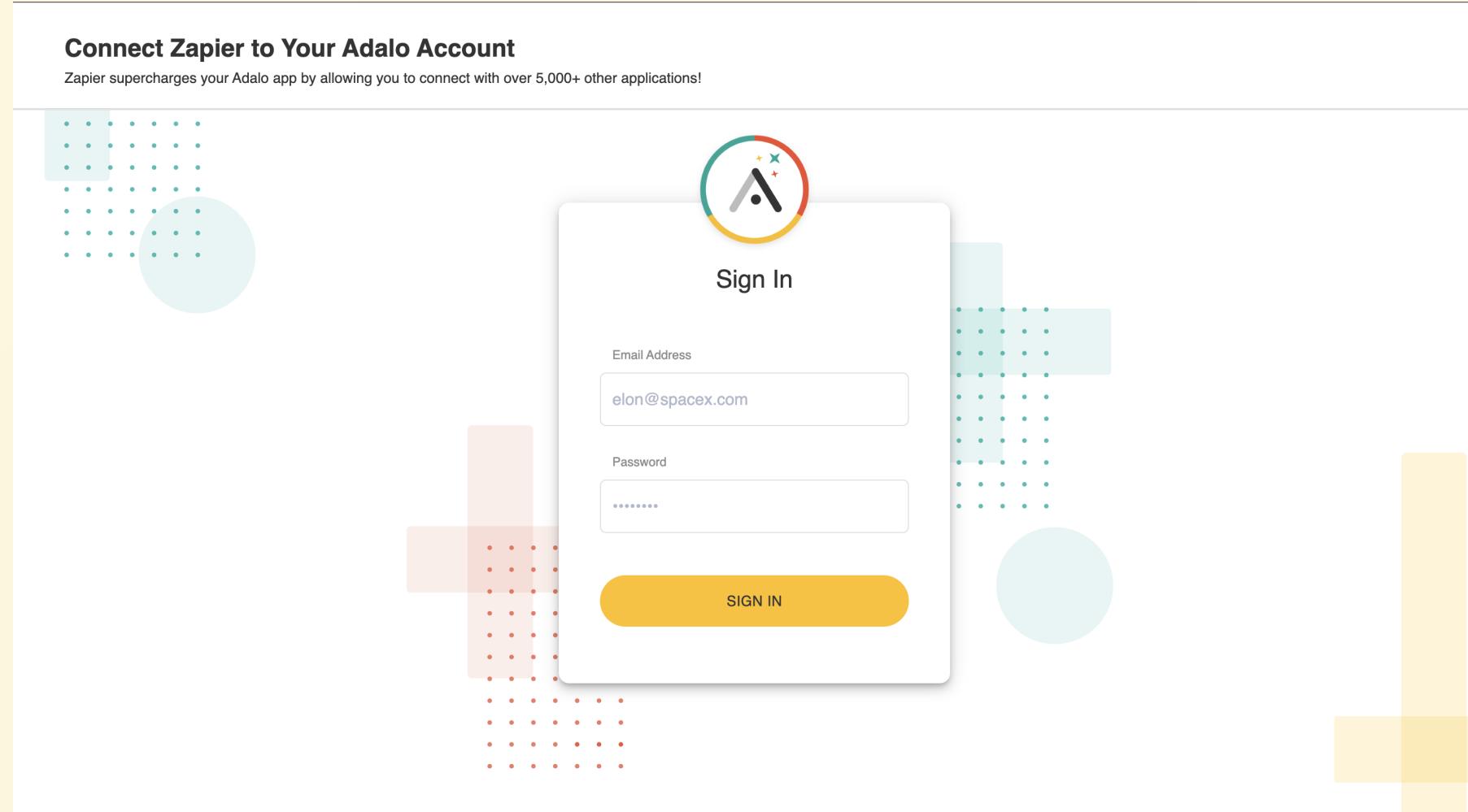
- Trigger("1. Adalo"と書かれている長方形)をクリック
- Eventに"New Record"を設定
- "Continue"をクリック



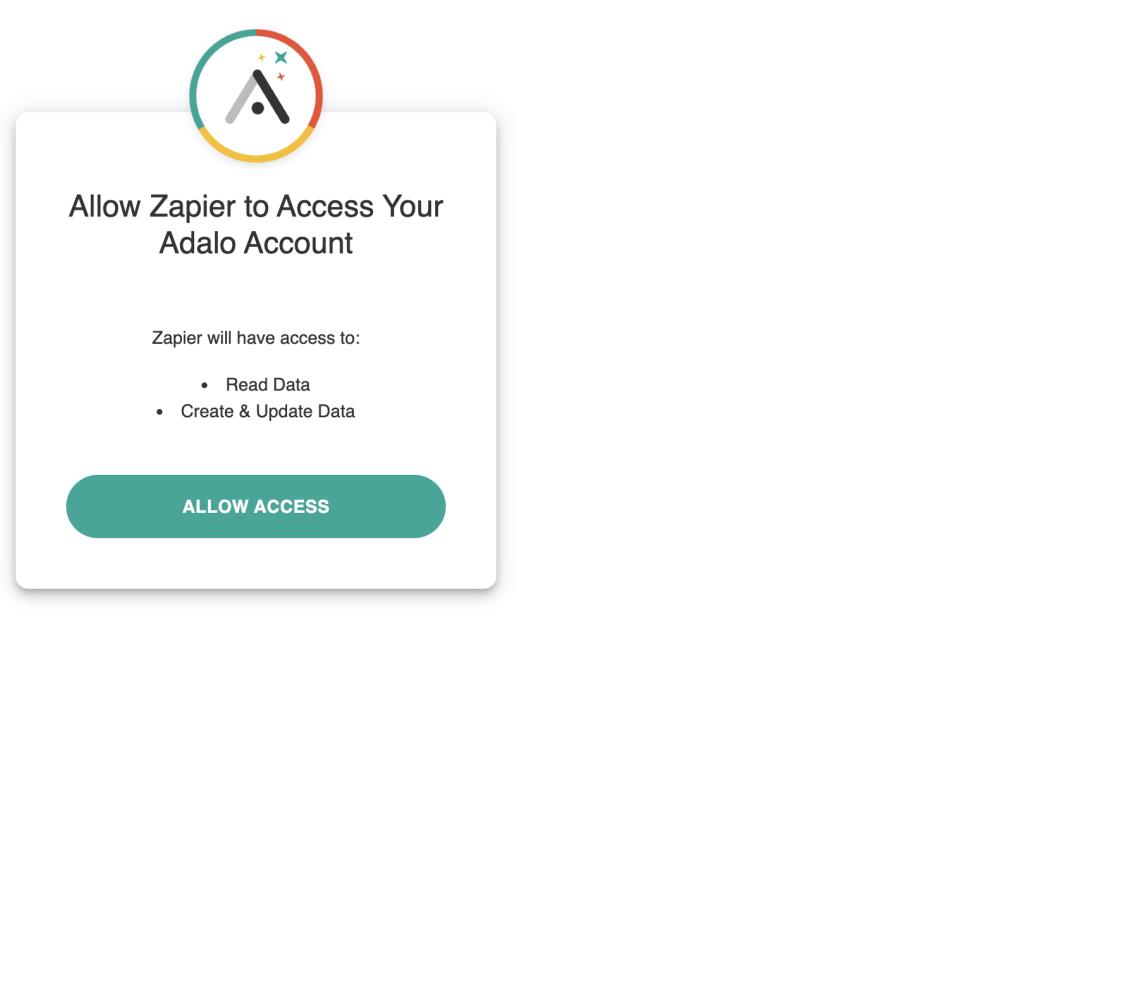
- "Sign in" をクリック



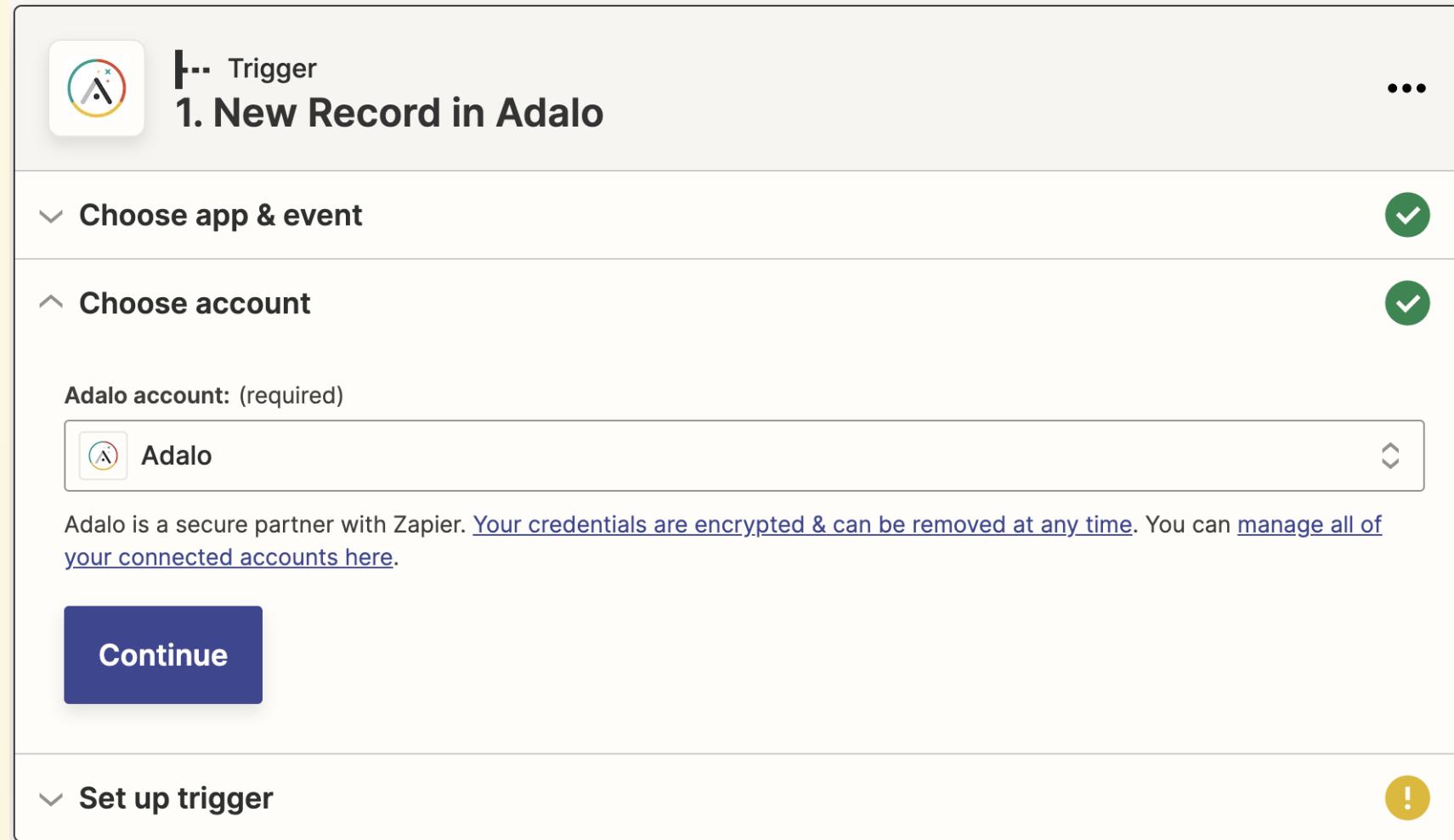
- AdaloのアカウントのEmail AddressとPasswordを入力して、"SIGN IN"をクリック



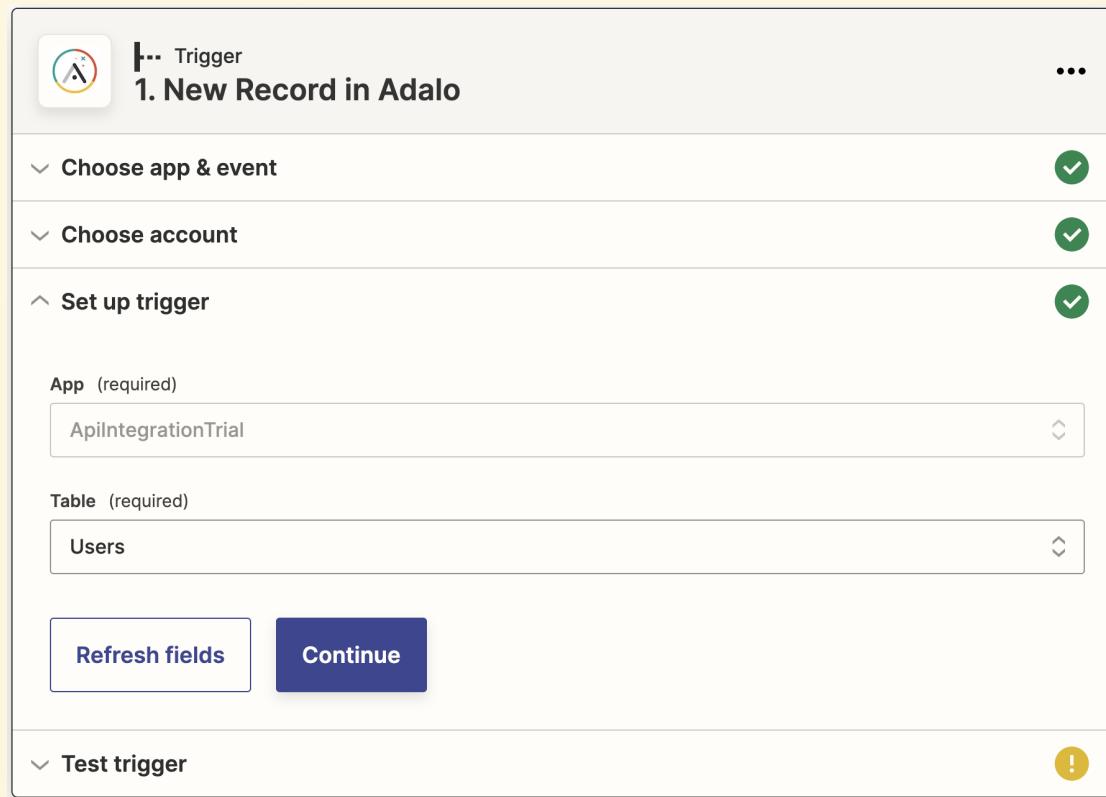
- "ALLOW ACCESS"をクリック



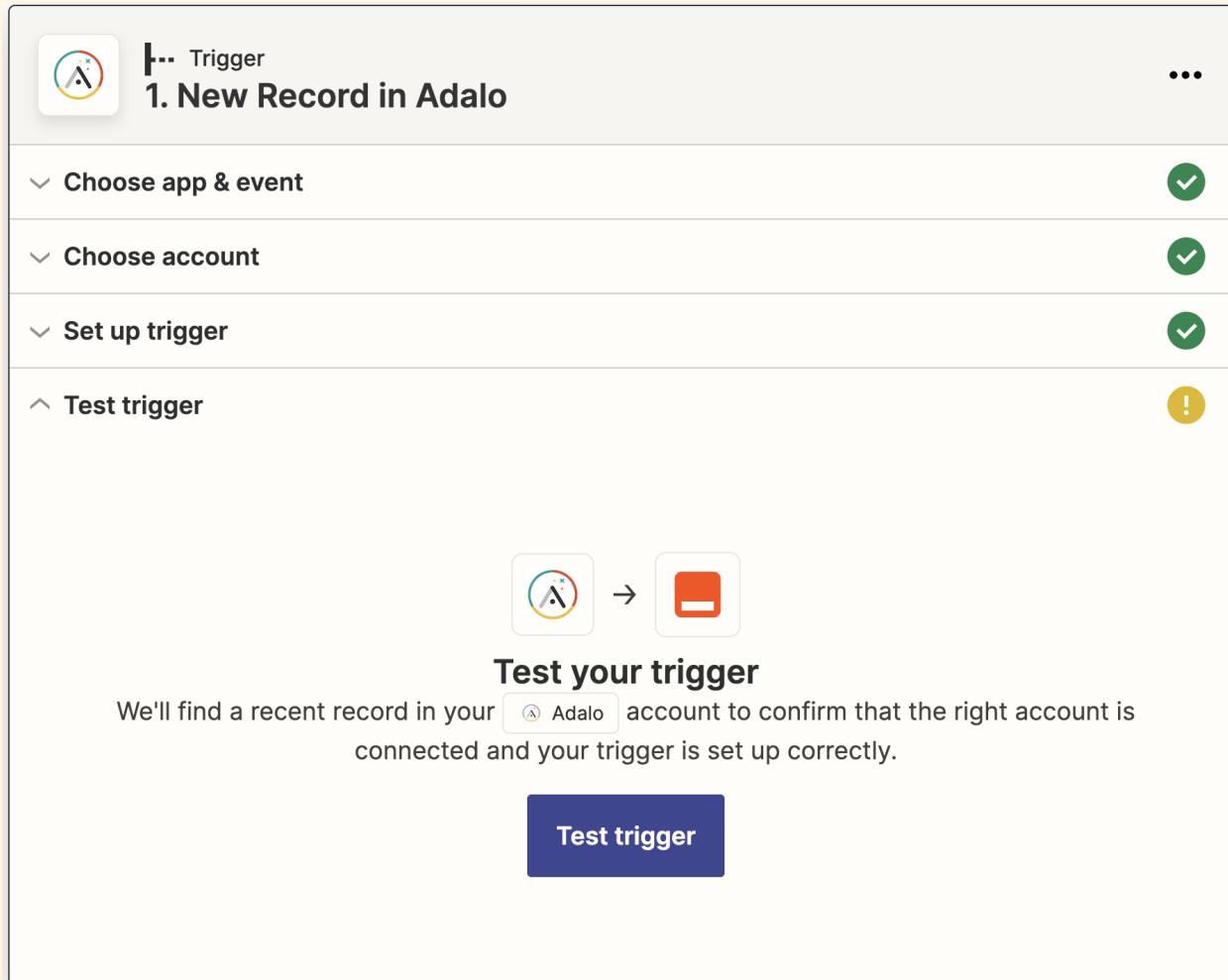
- Adalo accountが設定されたのを確認して、"Continue"をクリック



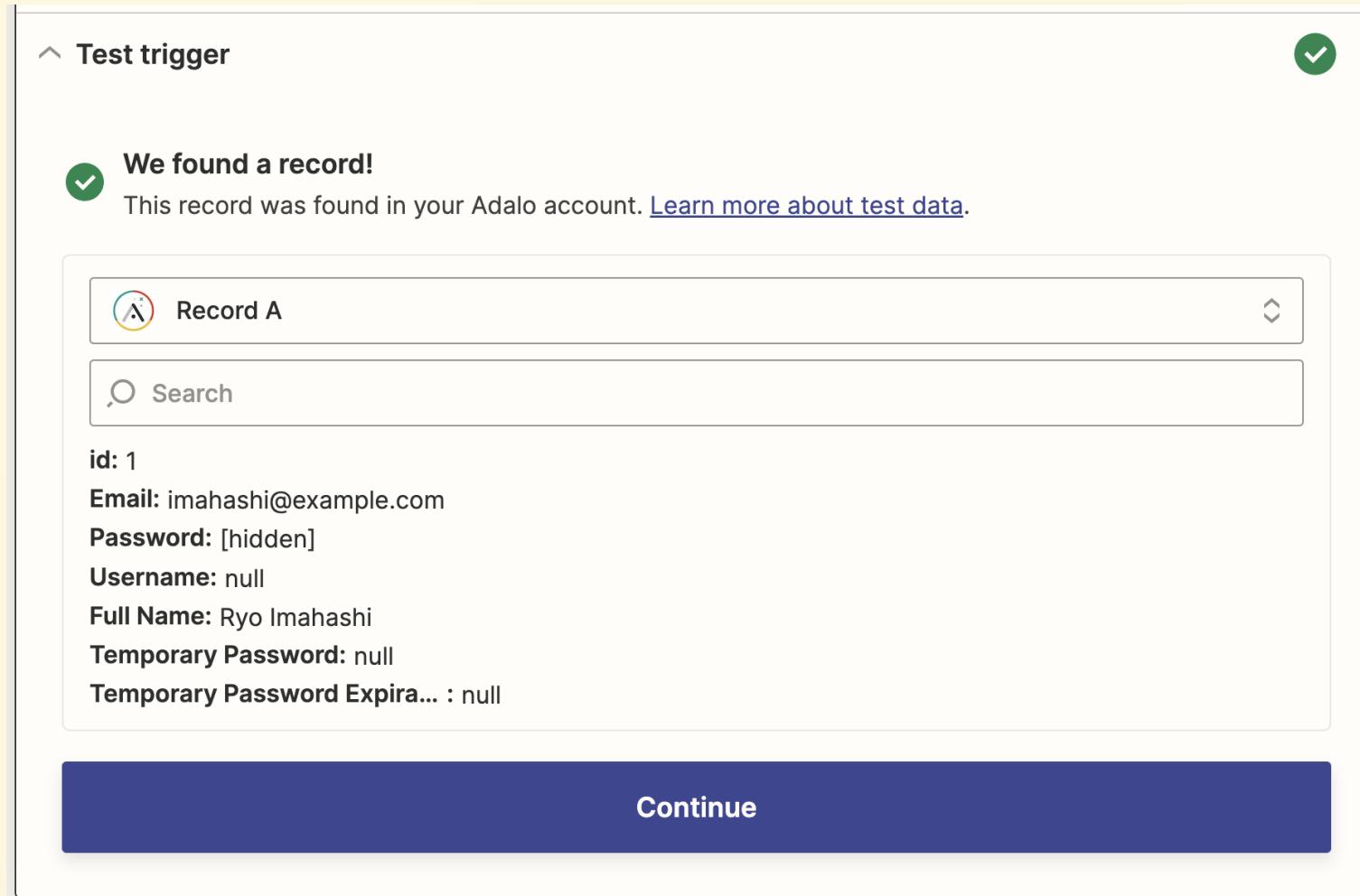
- Appに"ApilntegrationTrial"を設定
- Tableに"Users"を設定
- "Continue"をクリック



- "Test trigger"をクリック



- レコードが見つかったら、"Continue"をクリック



- Actionを実行するAppにGmailを選択

The screenshot shows the Zapier interface for selecting an action. At the top, it says "2. Action" and defines an action as "An action is an event a Zap performs after it starts". There are "Learn more" and "..." buttons.

**App event**: Do something in an app

**Built-in tools**: Try one of our [20+ built-in tools](#)

**Available Apps:**

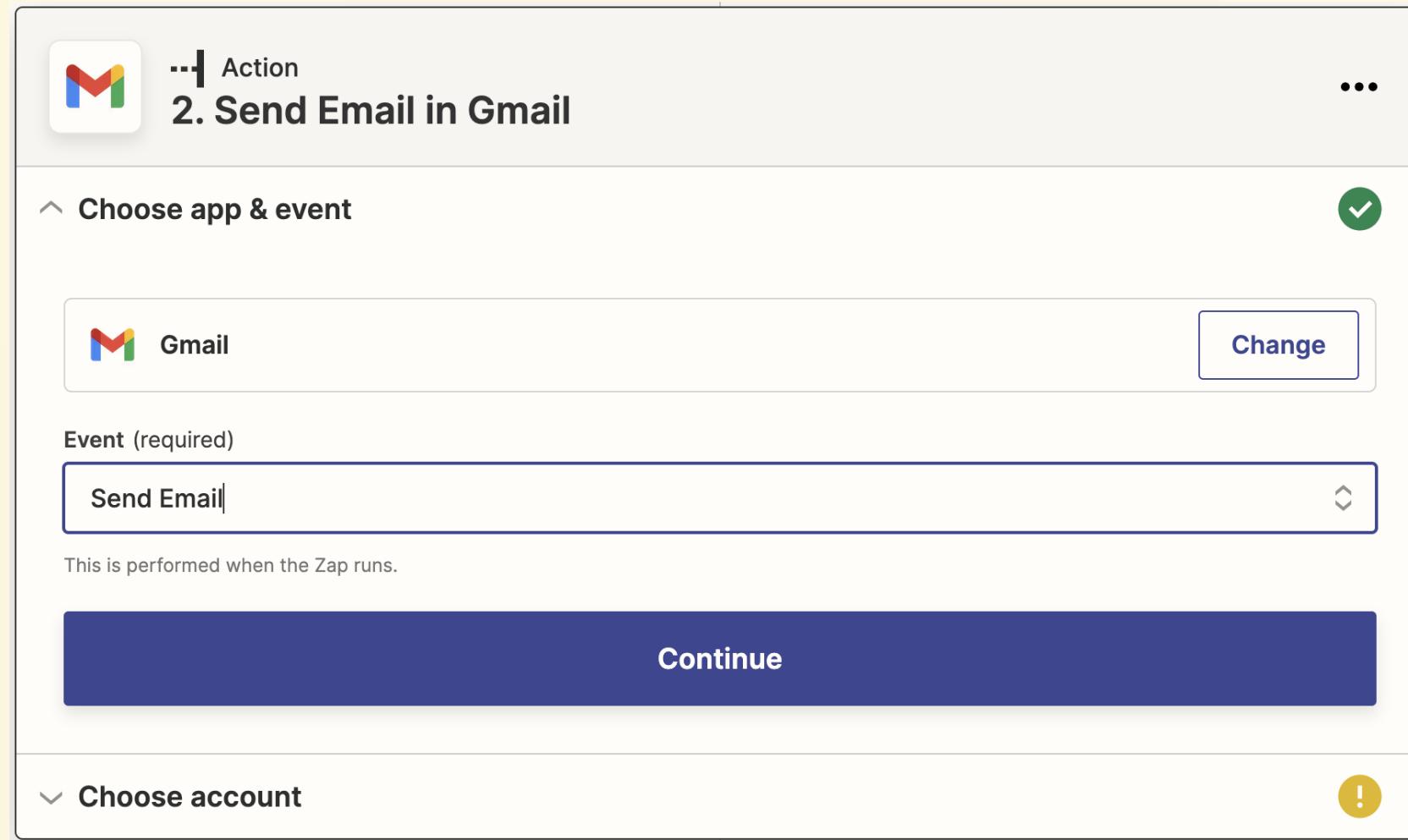
- Adalo
- Gmail
- Google Sheets
- Filter by Zapier
- Slack
- Google Calendar
- Webhooks by Zapier **Premium**
- Formatter by Zapier
- Email by Zapier
- Mailchimp

...and over 5,200+ more

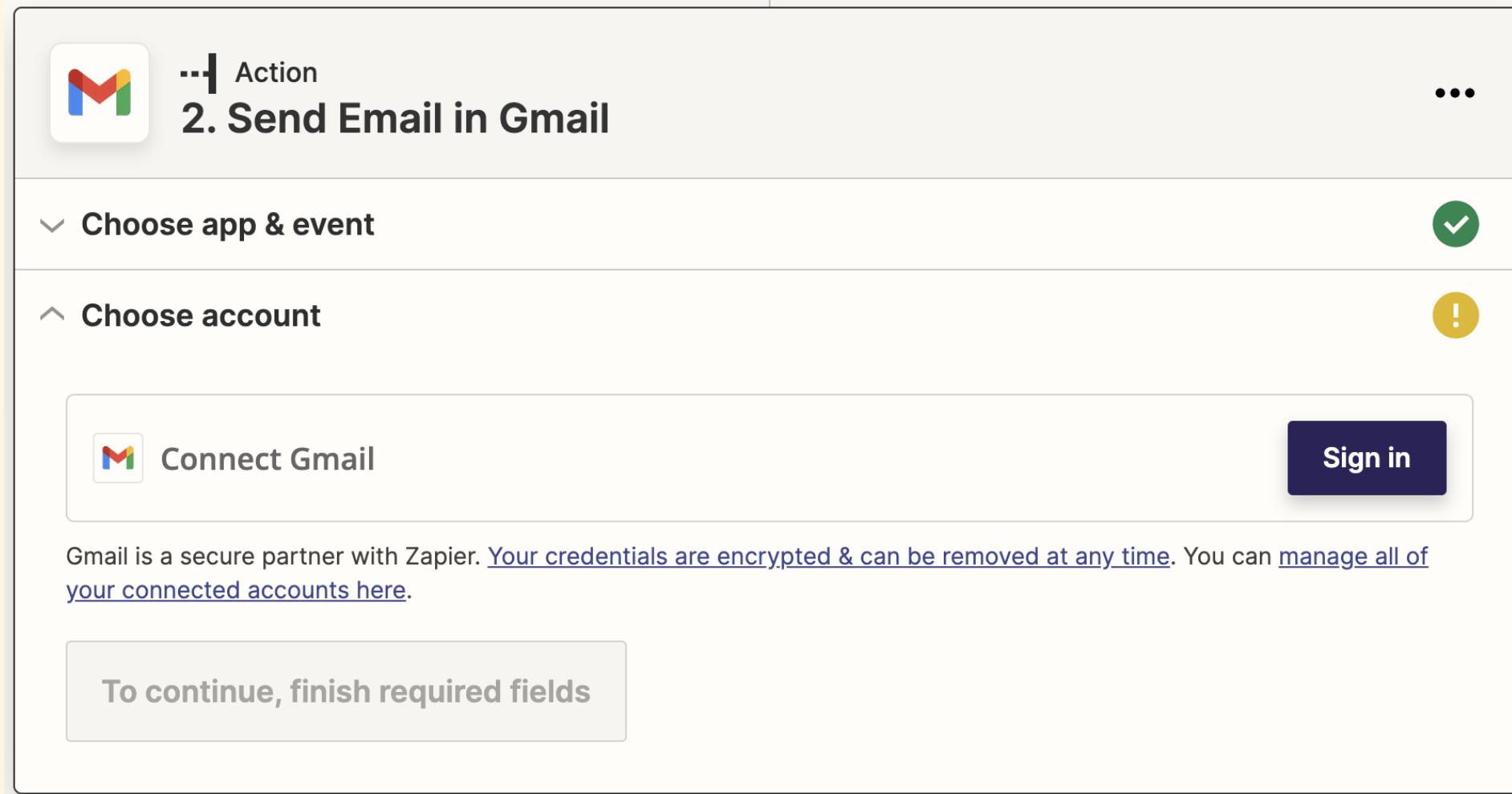
**Built-in tools:**

- Path**: Build different steps for different rules
- Delay**: Pause actions for a certain amount of time
- Filter**: Only proceed when a condition is met
- Format**: Change how incoming data is formatted

- Eventに"Send Email"を設定
- "Continue"をクリック



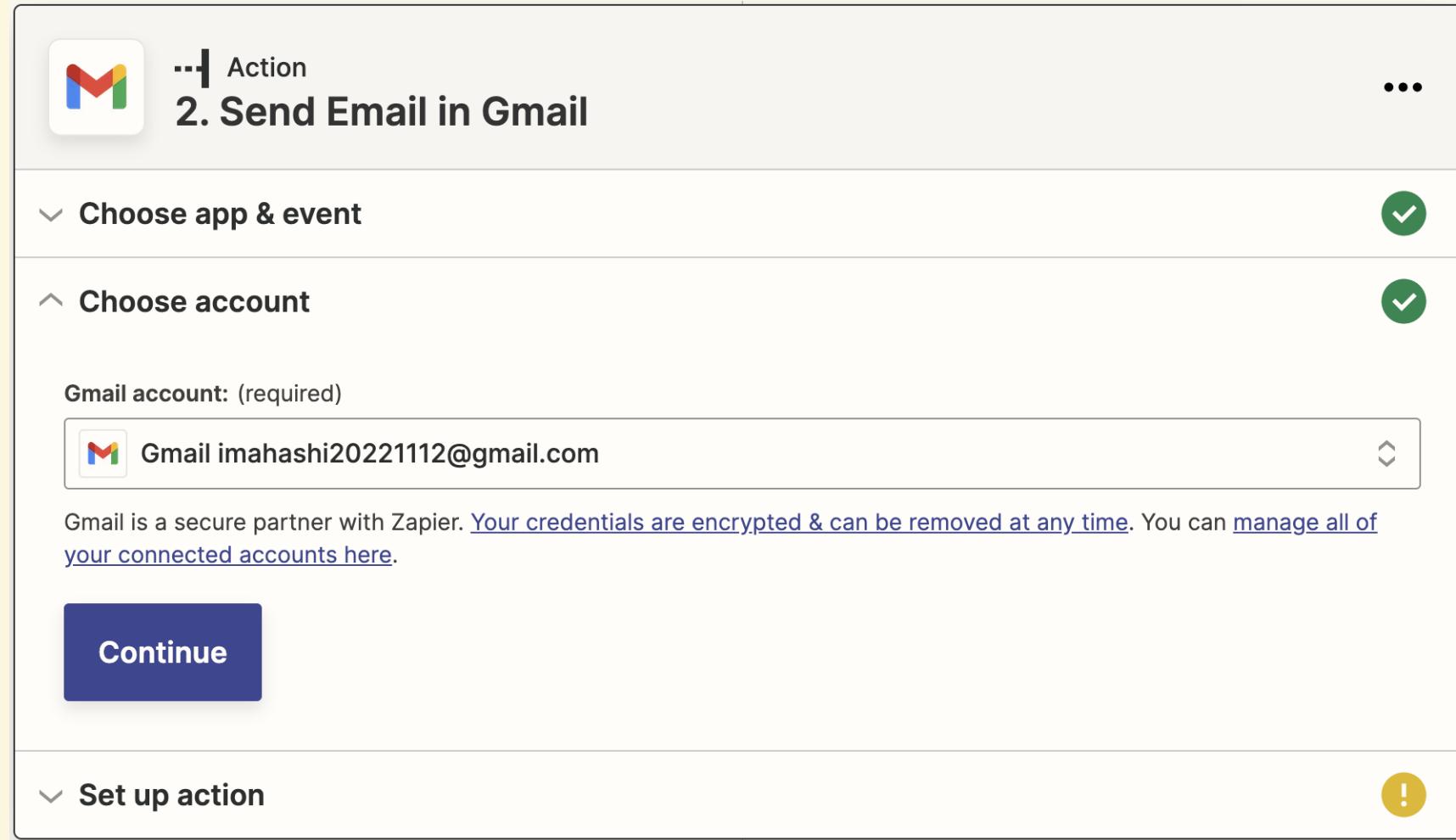
- "Sign in" をクリック



- Welcomeメールの送信元にするアカウントを選択



- Gmail accountが設定されたことを確認して、"Continue"をクリック



- ToにEmailを設定
- Fromに自分のGmailアドレスを設定
- From Nameに自分のアプリの名前を設定
- Subject, Bodyを自由に入力
- "Continue"をクリック

To  
 X

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... [more](#)

Cc  
  
Who should be cc'd on this email? Multiple email addresses can be entered either individually, or as a comma separat... [more](#)

Bcc  
  
Who should be bcc'd on this email? Multiple email addresses can be entered either individually, or as a comma... [more](#)

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

From Name

Reply To  
  
Specify a single reply address other than your own.

Subject (required)

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

Body (required)  
  
Hello, 1. Full Name: Ryo Imahashi.  
 Thank you for signing up my app.  
 I hope you enjoy it.

- 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 action"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、"Test action"はクリックしない(メール送信が失敗するため)
- "Publish Zap"をクリック

The image shows two side-by-side screenshots of the Zapier 'Test action' step for a 'Send Email to Gmail' zap.

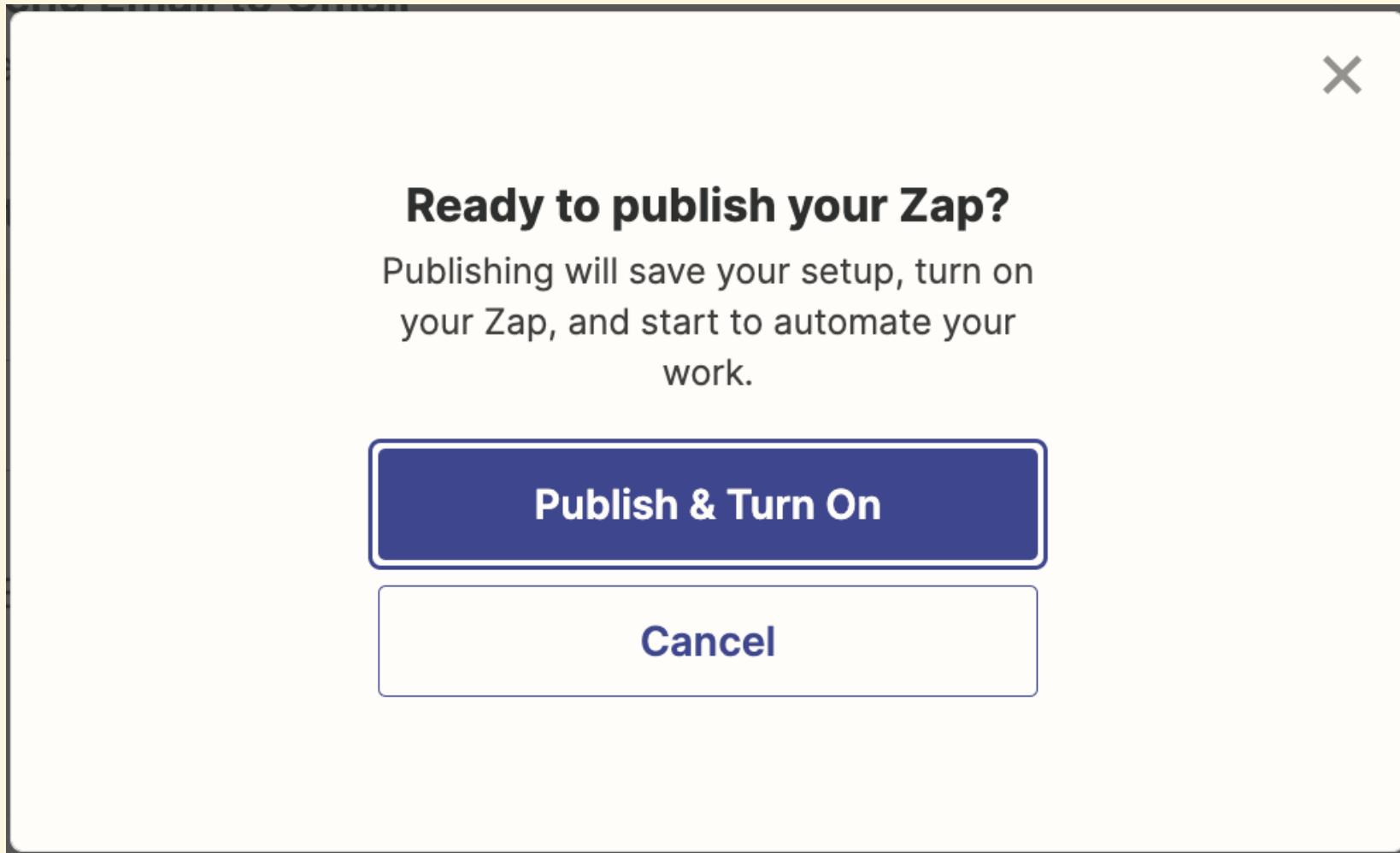
**Screenshot 1 (Left):** This screenshot shows the initial configuration of the action. It includes:

- A 'Send Email to Gmail' icon.
- The text: "Send Send Email to Gmail".
- The note: "To test Gmail, we need to create a new send email. This is what will be created:"
- A search bar with placeholder text: "Search item data..."
- Email content preview:
  - body: Hello, Ryo Imahashi.  
Thank you for signing up my app.  
I hope you enjoy it.
  - body\_type: plain
  - from: imahashi20221112@gmail.com
  - from\_name: Imahashi's App
  - subject: Welcome to my app
  - to:
    - 1: imahashi@example.com
- An 'Empty fields:' section listing: Cc: empty (optional), Bcc: empty (optional), Reply To: empty (optional), Signature: empty (optional), Label/Mailbox: empty (optional), and Attachments: empty (optional).

**Screenshot 2 (Right):** This screenshot shows the result of the test action. It includes:

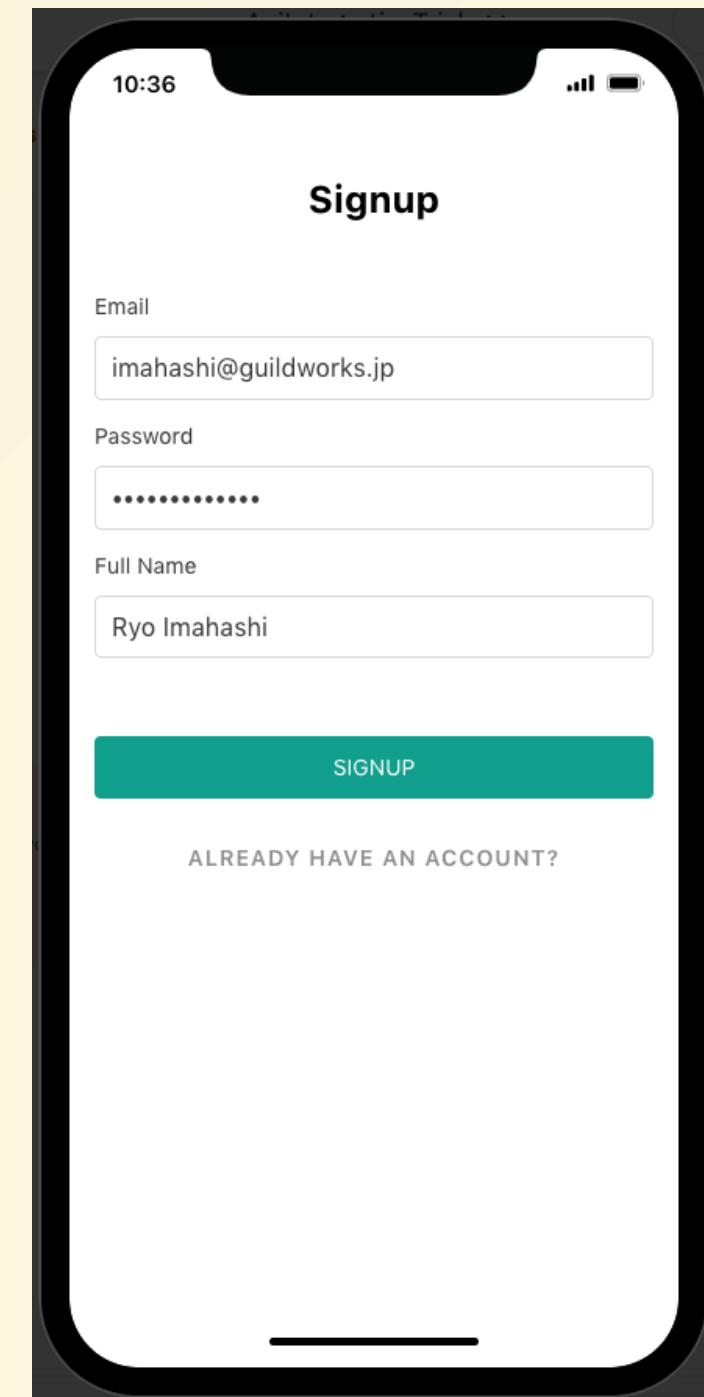
- A 'Send Email to Gmail' icon.
- The text: "Send Send Email to Gmail".
- The note: "We'll use this as a sample for setting up the rest of your Zap."
- A message: "A Send Email was sent to Gmail about 2 minutes ago".
- A search bar with placeholder text: "Search item data..."
- Email details:
  - id: 1846c01f2d6ddf3b
  - threadId: 1846c01f2d6ddf3b
  - labelIds:
    - 1: SENT
- Buttons at the bottom: "Retest action" and "Publish Zap".

- "Publish & Turn On"をクリック



AdaloのApIntegrationTrial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



Signupしても、すぐにはメールが届きません。  
14日間はZapierのProfessional Planのフリートライアル期間のため、  
2分間隔でZapが実行されます。それ以後はFree Planとなり、15分間  
隔でZapが実行されます。(参考: <https://zapier.com/app/pricing>)

The screenshot shows the Zapier pricing page with three plan options:

- Free**: For individuals who need the basics of automation. \$0 USD, free forever. Includes 100 tasks /mo. Current plan button.
- Starter**: For individuals with simple, multi-step automation needs. \$19.99 USD /month, billed annually. Includes 750 tasks /mo. Upgrade button.
- Professional**: For individuals that need the full power of Zapier's automation platform. \$49 USD /month, billed annually. Includes 2K tasks /mo. Upgrade button.

Below the plans, there are sections for "Free plan features" and "Starter plan features". A callout box highlights the update frequency for the Professional plan.

Plan	Tasks per Month	Price
Free	100	\$0 USD
Starter	750	\$19.99 USD /month, billed annually
Professional	2K	\$49 USD /month, billed annually

**Free plan features**

- ⚡ 5 Zaps
- ⌚ 15 min update time

**Starter plan features**

- ⚡ 20 Zaps
- ⌚ 15 min update time

**How often Zapier checks for new data to start your Zap.**

- ⌚ 2 min update time

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"をクリック

The screenshot shows the Zapier interface. On the left is a sidebar with navigation links: Dashboard, **Zaps** (which is selected and highlighted in orange), Transfers, My Apps, Zap History, Explore, Get Help, Free Plan, Tasks (1 / 1,000), and Zaps (Unlimited). Below the sidebar, a message says "Monthly usage resets in 29 days". The main area is titled "Zaps" and "Home". It features a search bar, a dropdown menu for "Private folders", and a table with columns: Name, Owner, Warnings, Last edit, and Running. A single zap is listed: "Name your Zap" by "20221112 I.", created 18m ago, with "No warnings", "Last edit 18m ago", and "Running" status (green button). To the right of the zap is a context menu with options: View, Details, History, **Run** (which is highlighted in blue), Rename, Share, Change ownership, Duplicate, and Delete.

Signupから2分経つか、手動でZapを実行した後に、SignupしたメールアドレスでWelcomeメールを受信していることが確認できます。

Welcome to my app 外部 受信トレイ ×

 **Imahashi's App** <imahashi20221112@gmail.com>  
To 自分 ▾

Hello, Ryo Imahashi.  
Thank you for signing up my app.  
I hope you enjoy it.a

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。

今後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。

参考: [Popular ways to use Adalo workflows](#)

Popular ways to use Adalo workflows

Send Gmail emails for new Adalo records  
by Zapier • Adalo + Gmail Try it ...

Create Google Sheet spreadsheet rows for new Adalo records  
by Zapier • Adalo + Google Sheets Try it ...

Add new Adalo records to Airtable  
by Zapier • Adalo + Airtable Try it ...

Send Twilio texts for new Adalo records  
by Zapier • Adalo + Twilio Try it ...

Send Slack channel messages for new Adalo records  
by Zapier • Adalo + Slack Try it ...

Add Mailchimp subscribers from new records in Adalo  
by Zapier • Adalo + Mailchimp Try it ...

Add Webflow CMS items to new Adalo records  
by Zapier • Adalo + Webflow Try it ...

Add Adalo records for new Google Sheets spreadsheet rows  
by Zapier • Google Sheets + Adalo Try it ...

Load more

Adaloでの外部連携についての紹介は以上です。

# Adaloでのチーム開発のやり方

演習に入る前に、DevelopmentPhaseでのチーム開発に向けて、Adaloでの共同作業の方法をお伝えしておきます。

Adaloにはチームメンバーを共同編集者として招待する機能がありますが、これは有料プランでないと使えません。

The screenshot shows the Adalo app interface. On the left is a sidebar with various icons: a plus sign in a circle, a gear, a list, a square, a document, a gear, a checkmark, a line, and a refresh symbol. The main area shows the 'Settings' screen with sections for 'App Settings' and 'Display Settings'. Below these is a collapsed section for 'App Access'. A modal window is overlaid on the screen, titled 'Upgrade to Add Editors!'. The modal contains the text 'Switch to a paid plan to start collaborating on your app design.' At the bottom right of the modal is a large green button labeled 'UPGRADE'. At the bottom left of the modal is a smaller button labeled 'BACK'.

Settings

App Settings

Display Settings

App Access

Ryo Imahashi  
imahashi+20221105@guildworks.jp

Add Team Member

Invite New Team Members...

Upgrade to Add Editors!

Switch to a paid plan to start collaborating on your app design.

BACK

UPGRADE

そのため、チームの全員が同じAdaloアカウントでログインして、共同編集をしていきましょう。

- 自分たちのアプリの開発をAdaloで始める時には、チーム用のGoogleアカウントを1つ作成して、そのアカウントでAdaloに登録してください  
[https://accounts.google.com/signup/v2/webcreateaccount?  
continue=https%3A%2F%2Faccounts.google.com%2FManageAcco  
unt%3Fnc%3D1&dsh=S50453738%3A1637917137418951&biz=f  
alse&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)
- 全員がそのメールアドレスとパスワードを使ってログインすることで、共同編集ができます

# Adaloでの同時編集の注意点

- 他の人が行った編集は自分の画面にリアルタイムで反映されません。反映させるためにはリロードが必要です。
- 同じ画面を同時編集すると、先に行われた編集が後から行われた編集に上書きされて、最後に編集した人の画面の状態になります。
  - 編集したコンポーネントは別でも、先に行われた編集は取り消されます。
- Actionも、同じ画面に同時に追加すると、後の方だけが保存されます。
- Databaseも、同時編集した場合は最後に編集した人の状態で上書きされます。

- 別画面の同時編集なら、どちらも保存されるので大丈夫です。編集する時は、どの画面かをチームメンバーに共有しましょう。みんなで同じPCを見ながら一緒に編集するか、画面ごとに担当者を決めるのがおすすめです。
- 編集する画面を切り替える前には、画面をリロードして最新の状態を反映しましょう(そうすることで、他の人が編集した画面を自分が古い状態に戻してしまうことを防げます)
- 他の人の担当画面をうっかり触ってしまわないように、Canvas上で画面同士の距離を離してから開発するのもおすすめです。
- Databaseは、誰か1人担当者を決めてその人が更新するようにしましょう。

Adaloでのチーム開発のやり方については以上です。

DevelopmentPhaseでチーム開発を始める際に、実際にチームメンバー同士で同時編集をしながら、上記の注意点について確認することをおすすめします。(作業内容が消えてしまうのは悲しいので😢)

# 演習

1. 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ください。
  - 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)
2. (演習1で時間が余れば)アプリを1つ自由に開発してください。

\* (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ったら、SlackでURLを共有して、みなさん  
に見てもらいましょう。

## 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

- ・ 今回のレクチャーでは、データベースを設計し、アプリからそのデータベースに対してCRUD操作を行えるようにしました。
- ・ また、アプリの改善をしながら、Adaloの機能をいくつか紹介しました。
- ・ 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！