

**Programming Boot Camp**

# **Adaloでのデータ操作と外部連携**

**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
- サンプルアプリを完成させよう
- 外部サービス連携
- 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用しました。(演習の中でデータベースを使った人もいるかもしれません)
- 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構築し、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。
- その後は、Adaloだけでは実現できないことを、Adaloと外部のサービスを連携させることで実現する方法をいくつかご紹介します。

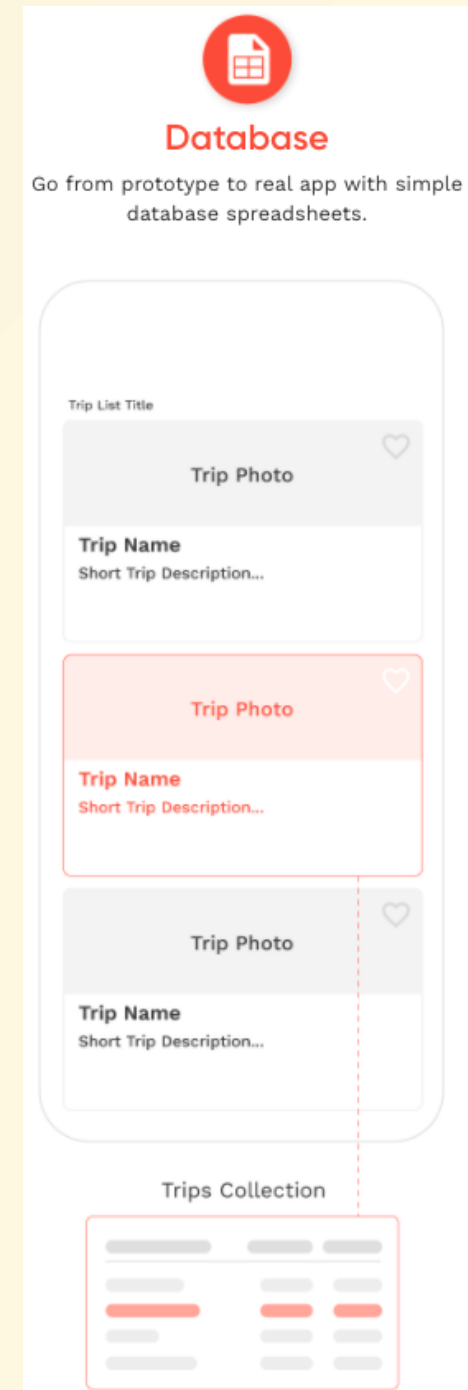
# データベースについて学ぼう

まず、これから学んでいく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、読込(表示)、更新、削除が行われる。
- 例: Chatアプリの場合

Database Collections		
>	 Users	3 Records
>	 Messages	7 Records
>	 Conversations	3 Records
>	 Read Status	22 Records



もう少し詳しいデータベースの説明は、以下のようになります。

- データベースとは、電子的にアクセスされる情報の整理された集合です。
- データベースを使ってユーザーはコンピュータに保存されている情報を作成(CREATE)、読み取り(READ)、更新(UPDATE)、削除(DELETE)することができます。これらの機能を総称してCRUDと呼びます。
- 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。(実際に、表計算ソフトをデータベースとして使うことも可能です)

# Adaloのデータベースの基本



このアイコンで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 displays a database application interface. On the left is a sidebar with a list of collections: 'Database Collections', 'Users' (3 Records), 'Email', 'Password', 'Username', 'Full Name', 'Profile Picture', 'Conversations (Creator)s', 'Messages (Sender)s', 'Read Statuses', 'Conversations (Members)s', and 'Friends'. The 'Users' collection is selected. The main area shows a table titled 'Users' with 3 records. The table has columns for Email, Password, Username, Full Name, and Profile Picture. The 'Password' column is marked as hidden. A '+ ADD USER' button is in the top right, and a 'DONE' button is in the bottom right.

	Email	Password	Username	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	

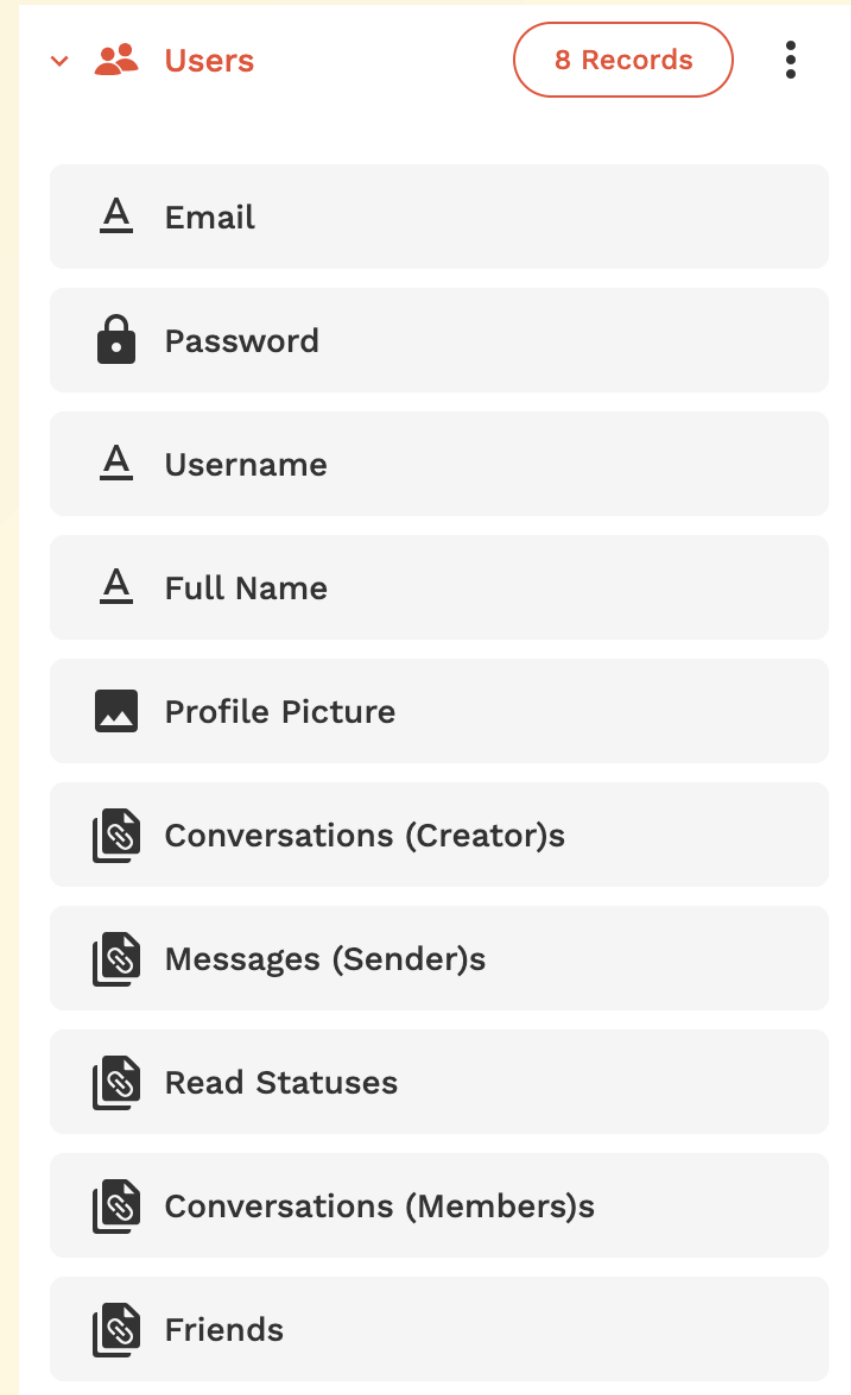


- Collectionは、データベースで扱う様々なデータをデータの種類ごとに分割し、整理するためのものです。(類似の言葉として、「テーブル」があります)
- アプリの中でユーザーが登録、表示、更新、削除といった操作を行うものがCollectionとなります。(「名詞」として表現できるものがCollectionになることが多いと言われます)
- デフォルトでは、UsersがCollectionとして用意されており、それ以外は開発するアプリに合わせて追加していきます。

※ どのようなCollectionを用意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オススメです)

## Propertyとは

- Propertyは、コレクションに含まれる(可能性のある)データが持つ項目です。
- 例えばUsers Collectionでは、Eメール、パスワード、名前、プロフィール写真などのPropertyを持つことが考えられます。



- Propertyとして扱うデータがどのようなものを定義するため、様々なTypeが用意されています。Propertyの追加時にどれか一つを選択します。
  - Text
  - Number
  - True/False
  - Date/Time
  - Date
  - Image
  - File
  - Relationship

## Relationshipとは

- 1つのレコード(データセット)に対して多数のプロパティを保存する代わりに、Relationshipと呼ばれる複数のコレクションを関連づけるための特別なプロパティを設定します。これにより、Collectionを人間が理解しやすい形で分割することができます。
- AdaloのRelationshipでは、コレクション間でリンクされるレコードの数に応じて、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

- 1つのレコードが、別のコレクションにある複数のレコードと関係を持つことを意味します。
- 表示している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。
- 例えば、主催者がイベントに対して1人だけいる場合の、主催者とイベントのRelationshipは1対多です。

A **User** can have multiple **Events**

A **Event** belongs to one **User**

A **User** can only have one **Event**

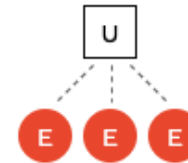
A **Event** can have multiple **Users**

## 多対多

- 両方のコレクションのレコードが、もう一方のコレクションの複数のレコードにリンクできることを意味します。
- 例えば、参加者は複数のイベントに参加できるし、イベントには複数の参加者がいるという場合の、参加者とイベントのRelationshipは多対多です。

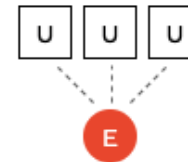
### How are Users & Events related?

U = Users E = Events



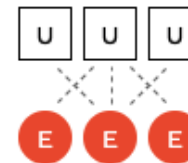
A **User** can have multiple **Events**

A **Event** belongs to one **User**



A **User** can only have one **Event**

A **Event** can have multiple **Users**



A **Event** can have multiple **Users**













A **User** can have multiple **Events**

CANCEL

DONE

## Recordとは

- Recordは、コレクション内へ情報を保存する際に1セットとなるデータの組み合わせです。
- Usersの例では、1人のユーザーがコレクション内の1つのレコードとみなされます。

Users					
<div><div>+ ADD USER</div><div></div></div>					
<input type="checkbox"/>	 Email	 Password	 Username	 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	
<div>DONE</div>					

- Recordの登録は基本的にアプリの画面上のフォームからできるようにしますが、レコードビューの右上にある「+Add xxxx」ボタンから追加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。



# 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・構築しましょう。

## 前回のレクチャーで作成したアプリのクローン用URL

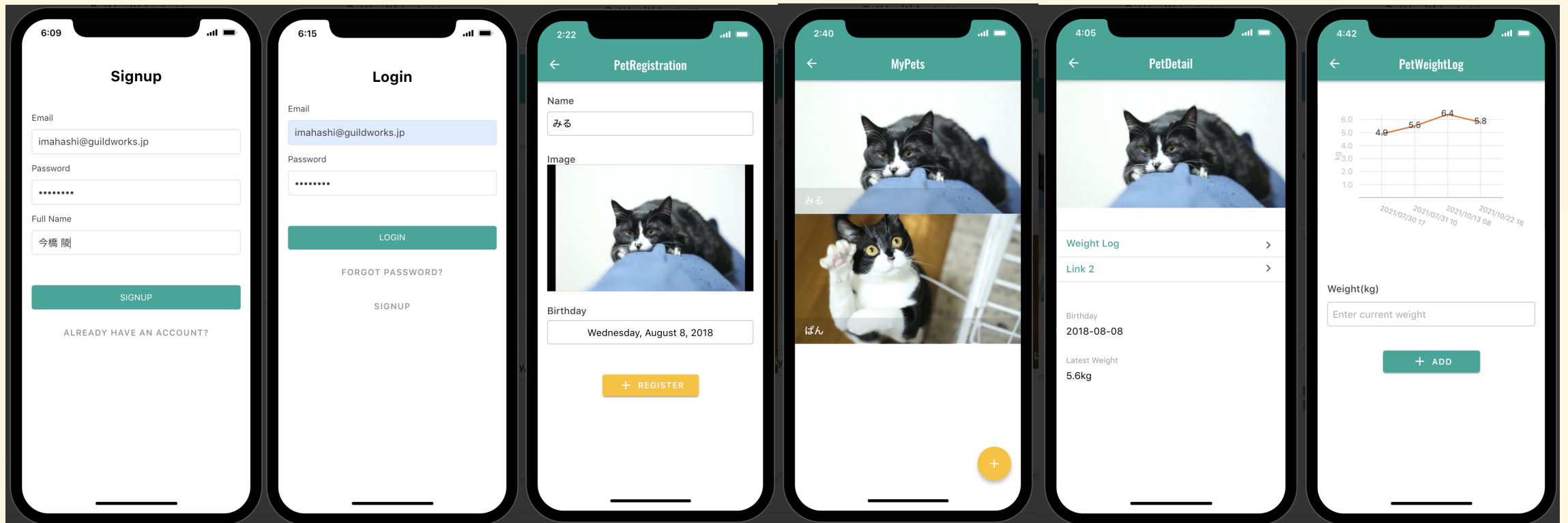
- 以下のURLからアプリをクローンしてください。それを使ってここからのレクチャーを進めます。

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

TODO: 演習の前に一度テンプレートのアプリを例にデータベース設計の手順を紹介して、一度流れを理解してもらう

# 演習

画面を見て必要そうなデータをリストアップし、どのようなCollectionに分けられるかや、各Collectionに必要なPropertyとそのTypeを考えましょう。



- 次のスライド以降に答えはありますが、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オススメします。
- 最初の必要そうなデータのリストアップはテキストファイルに書き起こしてみてください
- それ以降は、テキストファイルに書き起こしながら考えても良いですし、実際にAdaloのデータベースで設定しながら考えても良いです。

## 演習の答え

まずは、必要そうなデータをリストアップ。

- 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

どのようなCollectionに分類できるかを考える。

- Users
- Pets
- PetWeightLogs

※ ペットの体重は1匹のペットに対して登録日時別で複数登録されるため、Collectionを分けます(ペットとペットの体重が1対多の関係になります)

Propertyを追記する(一旦Relationshipは除く)。  
TODO: 画面に収める

- Users
  - Email
  - Password
  - FullName
- Pets
  - Name
  - Image
  - Birthday
- PetWeightLogs



Relationshipを追記する。  
TODO: 画面に収める

- Users
  - Email
  - Password
  - FullName
  - Pet(1対多)
- Pets
  - Name
  - Image
  - Birthday

# データベース操作

設計・構築したデータベースを使って、サンプルアプリでデータを操作できるようにしましょう。

## データの作成(CREATE)

TODO: 作成済みのフォームで実際にデータを登録できるようにしよう

## データの表示(READ)

TODO: 作成済みの画面に実際に登録したデータを表示させよう

## データの更新(UPDATE)

TODO: 登録したデータを更新する画面を作ろう

## データの削除(DELETE)

TODO: データの削除ができるボタンを作ろう

# 参考: 実際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->データベース作成->それらの連携という流れで作成を進めました
  - どのような画面かがイメージできていないと、それに必要なデータが分からず、データベースの設計も難しいと考えたため、このようなレクチャーの流れにしています
- 今後Adaloを使ってアプリ開発をする際は、必要なデータが分かっているのであれば、先にデータベースの設計・構築を行うことをオススメします。それによって、UIの作成も簡単に行うことができます。
  - とはいえ、実際の開発は一方的な流れではなく、試行錯誤してデータベースとUIを行ったり来たりすることになります。


# サンプルアプリを完成させよう



## TODO


- 通知(ペットの登録が0匹の時に登録を促すメッセージを表示させる)
- バリデーション(必須チェックくらい?)
- Tips






# TODO: 通知(ペット0匹通知)の紹介

Title  
ペットを登録しましょう 


Body  
まだ User full name  さんのペットは登録されていないようです。   
右下のボタンを押してペットを登録してください。


Screen  
Home 

When does this happen?  
 Sometimes... 

This action will only happen if...  
Logged In User > Pets > Count 

+ ADD PETS FILTER

Is equal to 

0 

## TODO: バリデーションの紹介

- 必須チェック
- 他はテキストとVisibilityを駆使すること。デフォルトのフォームでやれないのはとても面倒。。

<https://note.com/uramot/n/n035bc528aefc>

## TODO: Tipsの紹介

- Auto Refreshオプション
  - 最新のn件だけ表示
- TODO: 追加

## クローン用URL

- 以下のURLから完成したアプリをクローンできるので、答え合わせに使ってください  
TODO: URL貼り付け

# 外部サービス連携

## TODO

- Zapier紹介
  - Slack連携?
  - Google Spreadsheet or Firestoreとの連携でレコード上限50件を回避させる？(せっかくAdaloのデータベースを学んだのに...というところはある)
  - Twitter or Instagram連携?

# 演習

どれでも好きなものにチャレンジしてみてください。

- 他のテンプレートアプリの画面を見ながらデータベース設計を試みる
- サンプルアプリに画面を追加し、必要なCollectionやPropertyをデータベースに追加する
- 好きなアプリを新しく作成する

できたらSlackでURLを共有して、みなさんに見てもらいましょう

## 演習結果の発表

(時間があれば)

演習で作った画面を紹介してみませんか？

# まとめ

-



**以上です！**

**お疲れさまでした！**