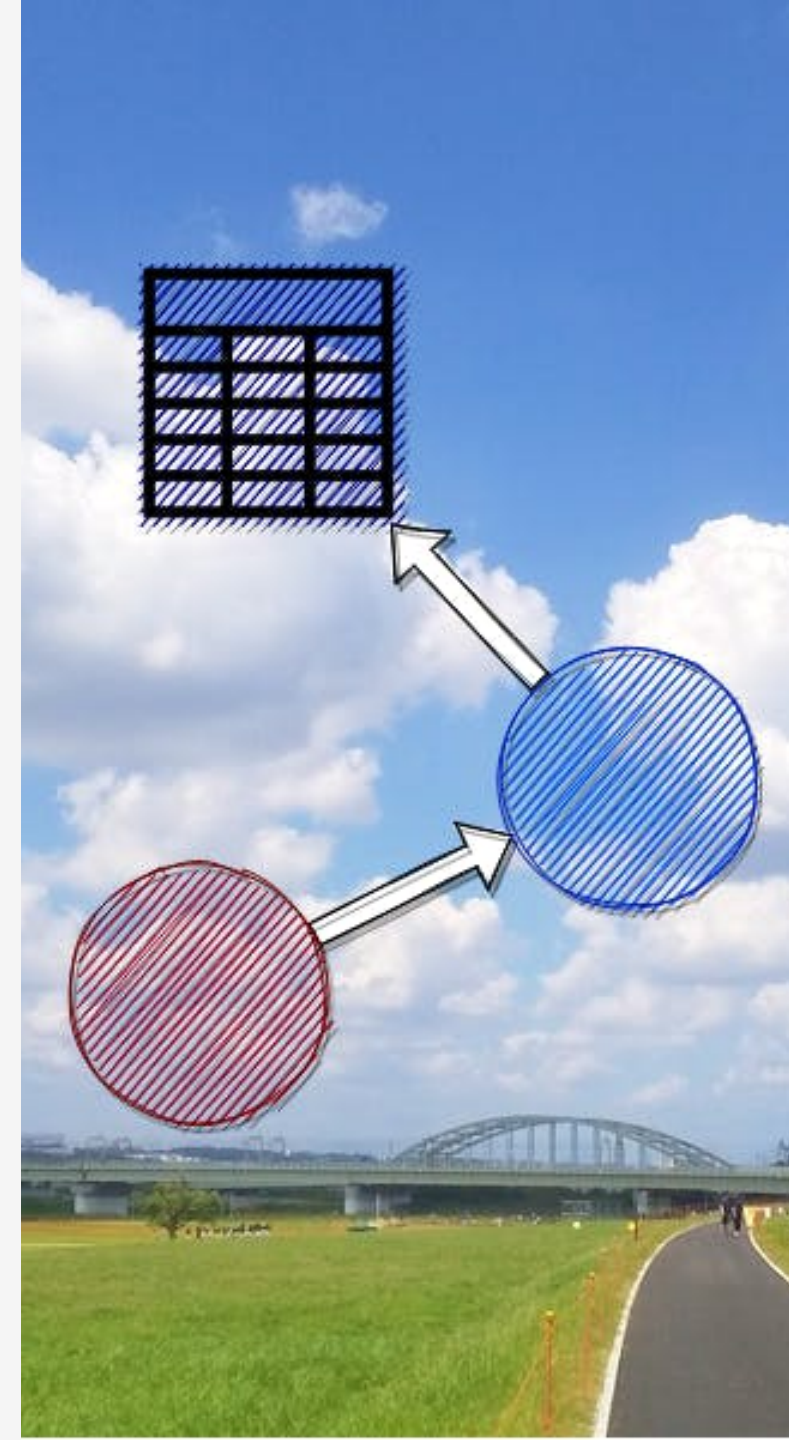


AppSheet をヘッドレス CMS 風にする

テーブルの行を API で取得する
ような場合



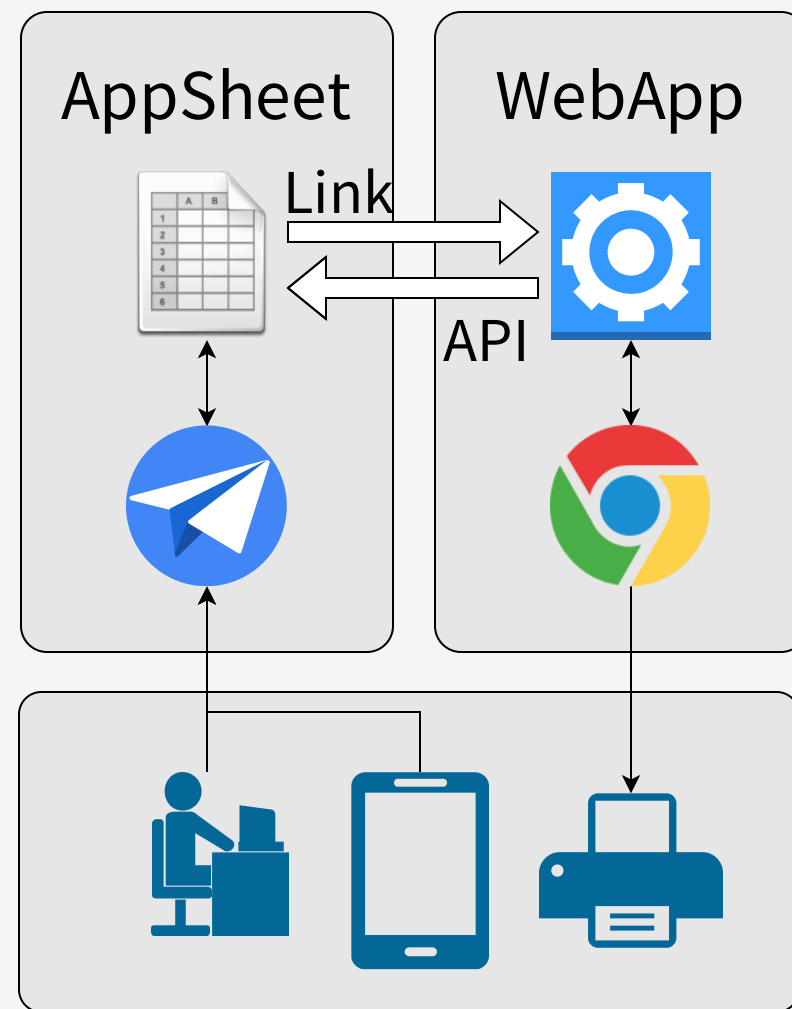
背景

差し込み印刷的 + α なアプリを AppSheet 上で作ることになる。

AppSheet でも 印刷できるとのことだが、今回は「簡易な画面遷移でプレビュー画面を表示したい」ので少し要件にあわない。

背景

最終的には印刷(プレビュー)用ページは別途ウェブアプリを作成し、AppSheet の Behavior から外部ページを開く(クエリーパラメーター経由で行の ID を渡す)ことで対応となった。

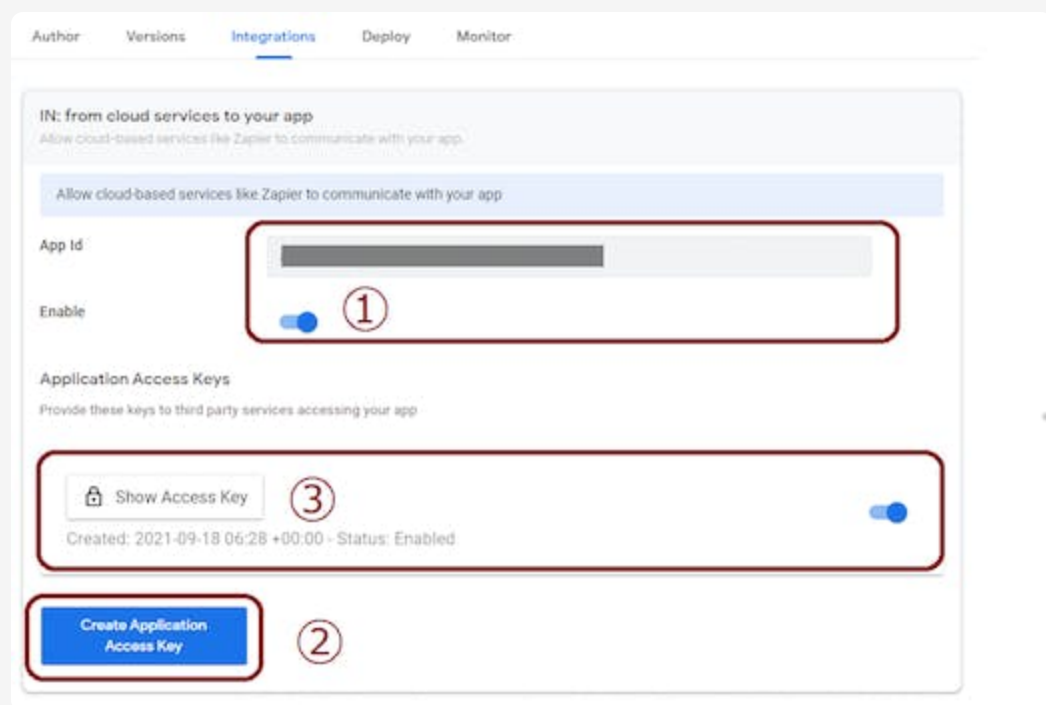


外部アプリからの連携

AppSheet API の利用

APIの有効化

AppSheet は外部からの連携用に [API](#) が用意されているので、[アプリケーションの Access Key を作成](#)すれば利用可能になる。



API でデータ(行)を取得

テーブルから行を取得する場合は、以下のような **POST** を実行する。

URL

```
https://api.appsheet.com/api/v2/  
  {URL エンコードされた AppId}/  
    tables/{URL エンコードされたテーブル名}/  
      Action?ApplicationAccessKey={Access Key}
```

DATA

```
{"Action": "Find", "Properties": {}, Rows: []}
```

curlでのサンプル

```
curl \  
  -X POST \  
  -H "Content-Type: application/json" \  
  -d '{"Action": "Find", "Properties": {}, Rows: []}' \  
  'https://api.appsheet.com/api/v2/.... '
```

AppSheet API の注意点

日本語固有のものからシステムの的なものなど

主な注意点

AppSheet API は比較的簡単に実行できるのだが、実際の利用では以下のような注意点があった。

- テーブルやカラム名等が日本語になっている
- データ(行)を選択する Selector の組み立てが難しい
- API 実行の利用制限がある(ように見える)
- 画像への直接アクセスはできない

テーブルやカラム名等が日本語になっている

AppSheet でアプリを作成する手順的に(スプレッドシートの内容からデータのカラムを構成するなど)、日本語のテーブル名やカラム名が利用される傾向は高くなる。

TypeScript の型定義などでは日本語も使えるようだが、トラブルを避けるために受信したデータを英語カラム名へマッピングすることになった。

API レスポンスの例:

```
[
  {
    "_RowNumber": "2",
    "id": "d9c9210f",
    "created": "09/16/2021 08:27:59",
    "updated": "09/18/2021 15:40:15",
    "タイトル": "テスト 1",
    "本文": "本文のテスト 1。",
    "og:title": "",
    "og:description": "",
    "og:image": "テスト_Images/d9c9210f.og-image.987023.jpg"
  }
]
```

データ(行)を選択する Selector 式の組み立てが難しい

[AppSheet の Selector](#) を組み立てるようなライブラリーは(ざっと探した限りでは)なさそうだったので、独自に組み立てることになる。

しかし、エスケープの方法などの具体的な手順が見当たらなかったので、独自に処理するなら注意が必要になる。

今回は、危ない橋はわたらずに「全件受信してから必要な部分だけ取りだす」方法で対応となった。

API 実行の利用制限がある(ように見える)

[Reading Records from a Table](#) の API Response HTTP Status Codes に記載はないが、**429 Too Many Requests** も返ってくる。

過去数時間で数回しか API を実行していないときに返ってくることもあり(直後に再実行するとほぼ成功する)、「なにをやったら **429** になるのか？」が現状では不明。

現状ではエラーメッセージを表示することで対応しているが、別途対応が必要になる可能性もある。

画像への直接アクセスはできない

API レスポンス内での `Image` 型は、ファイル名が `テスト_Images/d9c9210f.og-image.987023.jpg` のよう格納される(データファイルからの相対パス)。

よって、単純に Rest API を利用しただけでは画像を表示できない。

これについてはアクセスの手順があるので次章に記述。

画像を表示

外部アクセス用URLの生成

gettablefileurl の利用

AppSheet ではアプリ外部から画像ファイルなどへアクセスする方法が提供されている。

- [Displaying Images and Documents | AppSheet Help Center](#) の「External URL for Image Files」

事前に `CONTEXT('AppName')` がわかっているならば、仮想カラムを用意しなくとも URL を取得できる(ただし、公式な保証はない)。

```
https://www.appsheet.com/template/gettablefileurl
?appName={URL エンコードされた AppName}
&tableName={URL エンコードされたテーブル名}
&fileName={URL エンコードされたファイル名}
```


おわりに

おわりに

画像の扱いなどで少し戸惑うことがありつつも、「AppSheet でのデータ管理を軸に入出力を補佐するアプリを作成する」という構成は比較的容易に行えた。

また、ヘッドレス CMS の主な使い方である [「静的サイトの生成」](#)にも利用可能であった。

ただし、ブログなどのメディアを更新する場合は「コンテンツのエディターや公開管理機能なども必要となってくる」ので、専用サービスを置き換えるものではないと思われる(おそらく補完的な利用が適している)。