

FileMaker Data API を活用して Tableauのデータ分析にチャレンジしよう

株式会社ワークスペース
岡 田 匡

[FileMaker Data API]

FileMaker 17 Data API ガイド

<https://fmhelp.filemaker.com/docs/17/ja/dataapi/index.html>

※ 注意 ※

当資料内に記載の **※FileMakerドメイン※** はお使いのFileMaker Server
環境に適宜置き換えてご利用ください

[準備]

1. FileMaker Server 17 が必要です
Admin Console の コネクタ - FileMaker Data API で「有効」
2. FileMaker カスタムAppの設定が必要です
拡張アクセス権で FileMaker Data APIでのアクセス - FMSのみ(fmrest)「チェック」
3. 有効なSSL証明書が必要です
Admin Console の 構成 - SSL証明書 で「カスタム証明書のインポート」

[FileMaker Data API]

< 扱い方 >

【HTTPメソッド】 : POST・GET・PATCH・DELETE

【URL】 : https://ドメイン に続くURL

【HTTPヘッダ】 : FileMaker Data APIに渡すためのヘッダ情報

【引数】 : FileMaker Data APIに渡すためのデータ

→ ※ 応答 (FileMaker からの戻り値 JSON形式)

FileMaker Data API にログインする

【HTTPメソッド】 : POST

【URL】 : /fmi/data/v1/databases/※ データベース名/sessions

【HTTPヘッダ】 : Content-type : application/json
Authorization: Basic ※ IDとパスワード

【引数】 : 特に何ものなし(注意!! 空の中カッコ = {})

※ データベース名 = "customer"

※ IDおよびパスワード = "api"

base64encode("api:api") でエンコード → YXBpOmFwaQ==

→ ※ 【応答】 アクセストークン、空の応答ボディ、およびエラーコード 0

例)

curl -X POST

https://※FileMakerドメイン※/fmi/data/v1/databases/customer/sessions

-H 'authorization: basic YXBpOmFwaQ=='

-H 'content-type: application/json'

-H 'Content-length: 0'

→ 戻り値の例

```
{
  "response":
    {"token":"db0ee4d19fe9a4a440d834b8887ede374e0785f64b9634a659b"},
  "messages":[
    {"code":"0","message":"OK"}
  ]
}
```

FileMaker Data API からログアウトする

【HTTPメソッド】 : DELETE

【URL】 :

/fmi/data/v1/databases/※ データベース名/sessions/※ セッショントークン

【HTTPヘッダ】 : Content-type : application/json

【引数】 : なし

→ ※ 【応答】 空の応答ボディ、およびエラーコード 0

例)

```
curl
```

```
-X DELETE
```

```
https://workspace.nagoya/fmi/data/v1/databases/customer/sessions/  
db0ee4d19fe9a4a440d834b8887ede374e0785f64b9634a659b
```

→ 戻り値の例

```
{  
  "response": {},  
  "messages": [  
    {"code": "0",  
     "message": "OK"}  
  ]  
}
```

FileMaker Data API でFileMaker Serverの任意の1レコードを取得する

【HTTPメソッド】 : GET

【URL】 : /fmi/data/v1/databases/※ データベース名/layouts/※ レイアウト名/records/※ レコード番号

【HTTPヘッダ】 : Authorization: Bearer ※ セッショントークン

【引数】 : なし

→ ※ 【応答】 JSON形式のレコードデータおよびエラー 0

例)

curl

https://※FileMakerドメイン※/fmi/data/v1/databases/customer/layouts/customer/records/10

-X GET

-H "Authorization: Bearer 507f2bd724f858415412dba0fa97fa967a2b5f379610238f6f"

→ 戻り値の例

```
{
  "response":
    {"data":
      [
        {"fieldData":
          {"姓_漢字": "大西",
           "名_漢字": "靖",
           ... (略) ...
          "recordId": 10
         },
        "portalData":
          {
          },
          "recordId": "10",
          "modId": "3"
        }
      ]
    },
  "messages":
    [
      {"code": "0", "message": "OK"}
    ]
}
```

[Tableau から FileMaker Serverに接続する]

https://※FileMakerドメイン※/fmi/data/v1/tableau/fm_connector.html

【ソースデータベース名】

【ソースレイアウト名】

【アカウント名】

【パスワード】

todoufuken に 接続してみる

https://※FileMakerドメイン※/fmi/data/v1/tableau/fm_connector.html

【ソースデータベース名】 todoufuken

【ソースレイアウト名】 todoufuken

【アカウント名】 api

【パスワード】 api

1. 緯度・経度 を地理的役割に変更
2. 経度 = 列 / 緯度 = 行 にドラッグ&ドロップ
3. 緯度・経度をディメンジョンに変換
4. 都道府県名を「マーク」の「ラベル」にドラッグ
5. 都道府県庁所在地を「マーク」の「ラベル」にドラッグ

marathon に接続してみる

https://※FileMakerドメイン※/fmi/data/v1/tableau/fm_connector.html

【ソースデータベース名】 marathon

【ソースレイアウト名】 marathon

【アカウント名】 api

【パスワード】 api

1. 距離を列にドラッグ
2. 距離をディメンジョンに変更
3. 走行ペースを行にドラッグして「詳細」を「時間」、「メジャー」を「最小値」、「連続」に変更
4. 12Lap / 14Lap もそれぞれドラッグして3種類の折れ線グラフを作成する

railway に接続してみる

https://※FileMakerドメイン※/fmi/data/v1/tableau/fm_connector.html

【ソースデータベース名】 railway

【ソースレイアウト名】 railway

【アカウント名】 api

【パスワード】 api

1. 緯度・経度 を地理的役割に変更
2. 経度 = 列 / 緯度 = 行 にドラッグ&ドロップ
3. 緯度・経度をディメンジョンに変換
4. 地図を拡大して鉄道網を確認
5. 鉄道会社名を「マーク」の色にドラッグ(全てのメンバーを追加)
6. 右の凡例から「ゆりかもめ」を選択 - 保持のみを選択
(左上のフィルターが「ゆりかもめ」となっている)
7. 地図を拡大
8. わかりにくいので「駅名」をラベルにドラッグ
9. まだわかりにくいので「鉄道会社名」の色の編集 - ゆりかもめ - 赤を選択
10. 少しわかりやすくなったが、地図が単色なのでわかりやすく地図を変更
11. MapBox を使う(前で説明)
12. メニューのマップ - バックグラウンドマップ - マップサービス
13. MapBoxを追加

Tableau で表示するカスタムAppを作ろう

材料：

- ・ 県庁所在地.xls
- ・ 市区町村別人口.xls
- ・ 都道府県別インターネットの利用状況及びインターネットの端末別利用状況(平成29年).xlsx

- FileMaker でインポートをしてカスタムAppを作る
- 都道府県名でリレーションを作成

Tableau で表示するカスタムAppを作ろう

https://※FileMakerドメイン※/fmi/data/v1/tableau/fm_connector.html

【ソースデータベース名】 research

【ソースレイアウト名】 research

【アカウント名】 api

【パスワード】 api

1. 緯度・経度 を地理的役割に変更
2. 経度 = 列 / 緯度 = 行 にドラッグ&ドロップ
3. 緯度・経度をディメンジョンに変換
4. 都道府県名・市区名をラベルにドラッグ
5. 各都道府県別の世帯数を見て見ましょう - 世帯数を色にドラッグ
6. 色の編集で「オレンジ - 白 - 青の分化」に変更して「ステップドカラー」を適用しましょう

7. 都道府県別の男性の数・女性の数と世帯数の比率を見て見ましょう
 - a. 世帯数のワークシートを複製します
 - b. 男性のワークシートを作ります
 - c. 女性ののワークシートを作ります別々のワークシートでは比較しづらいので ...
8. 新しいダッシュボードを追加ボタンをクリックしてここに3つのワークシートをドラッグします

9. 続いて、都道府県と携帯・パソコン・スマートフォン・タブレットの利用率について見てみましょう
※ サイズごと - サイズの変化は「範囲ごと」で「●」が綺麗に収まるサイズ

[Tableau 画面と用語の説明]

< 増分更新を有効にする >

抽出全体を更新する代わりに、前回データを抽出して以降の新しい行のみを追加するように抽出を構成するときに利用する。

日々の販売データなど

< ライブ接続と抽出 >

ほぼリアルタイムのデータを提供する際に用いる。抽出はデータのスナップショット。

< 地理的役割 >

マップビューでデータをプロットするのに役立つタイプ。

< ディメンジョンとメジャー >

ディメンジョンは名前・日付・地理的データなどの定性的値が含まれる。

メジャーは測定可能な数値と定量的な値が含まれる。メジャーは集計ができる。

緑のメジャーおよびディメンジョンは「連続」をあらわす。

青のメジャーおよびディメンジョンは「不連続」をあらわす。

<ディメンジョンとメジャー>

ディメンジョンは名前・日付・地理的データなどの定性的値が含まれる。
メジャーは測定可能な数値と定量的な値が含まれる。メジャーは集計ができる。

緑のメジャーおよびディメンジョンは「連続」をあらわす。
青のメジャーおよびディメンジョンは「不連続」をあらわす。

<Tableau で扱う型>

テキスト：テキスト
日付：日付
日付・時刻：タイムスタンプ
数値：数字
その他、ブール値・地理的値などもある。

<Tableau Prep>

データの結合・形式変換・クリーニングの手間が簡素化される製品。

<ブレンド>

データソースの表をもうひとつのデータソースのデータ列で補完する組み合わせ法。
FileMaker のリレーションに似ている

<ダッシュボード>

分析をするためのUI。