

makrj.shがあまりにもよかった理由

makrj.shが自分の中ではかなり良くて、昔でいう「外人4コマ」みたいにガッツポーズしてるくらいなのだけけど、検索しても反響が全くなく一体どういうことなのか、私は別の世界線に存在しているのだろうか

— div_jp (@div_jp) [2016年9月21日](#)

とツイートしただけで何も示しがないのは示しがないので、どういうところが良いのか、また私がどのように使っているのかを事例をあげつつ需要を無視して紹介していこうと思います。

これから使ってみたいなと興味がある方の知識の足しになってもらい、POSIX原理主義もシェルスクリプトも、少しでも広まってくれればいいなと思います。

このドキュメントの内容全ては、あくまでも私がこうして使っていますがどうですか？という**提案**なのでこれが絶対だ、というわけではありません。ご了承ください。

makrj.shやJSONPath-value形式の元ネタはここから勝手にいただきました。

<http://qiita.com/richmikan@github/items/0dc3330163c86b249bcd>

完全に二番煎じなので元ネタの方が理解できた方はこの先は読む必要はないのでブラウザを閉じてください。

製作してくれた作者様に感謝。

ザックリ言うとmakrj.shはJSONPath-value形式で保存されたデータをJSON形式に変換するスクリプトです。

JSONPath-value形式=データベース？

JSONPath-value形式の一般的な使われ方は、データベースだと私は思います。ウェブ関係でデータベースといえばSQLというのが主流で、データをゴチャゴチャするというのが最初に思いつきますが、まずはよく使われるお小遣い帳を例にとってJSONPath-value形式の紹介をしてみたいと思います。

表計算形式

お小遣い帳 (../DATA/okodukai_sheet.csv)

| No | 日 | 科目 | 用途 | 入金 | 出金 |
|----|----|-----|---------|----|-------|
| 1 | 10 | 食費 | オーソンの弁当 | | 550 |
| 2 | 24 | 光熱費 | 水道代 | | 900 |
| 3 | 25 | 光熱費 | ガス代 | | 1000 |
| 4 | 25 | 家賃 | 1月分 | | 70000 |
| 5 | 27 | 光熱費 | 電気代 | | 11200 |
| 6 | 31 | 通信費 | プロバイダ | | 8000 |

| | | | | | |
|---|----|-----|---------|--------|------|
| 7 | 31 | 給与 | 1月分 | 250000 | |
| 8 | 31 | 通信費 | NTT基本料金 | | 1700 |

一般的なお小遣い帳は細かい差はあるでしょうが、大体この表のようなイメージだと思います。これをノートだったりエクセルだったり記入しているはずです。そしてこれをJSONPath-value形式に変換するとこうなります。

JSONPath-value形式

JSONPath-value形式 (../DATA/okodukai_value形式.txt)

```
$$.1.日 10
$.1.科目 食費
$.1.用途 オーソン弁当
$.1.出金 550
$.2.日 24
$.2.科目 光熱費
$.2.用途 水道代
$.2.出金 900
$.3.日 25
$.3.科目 光熱費
$.3.用途 ガス代
$.3.出金 1000
$.4.日 25
$.4.科目 家賃
$.4.用途 1月分
$.4.出金 70000
$.5.日 27
$.5.科目 光熱費
$.5.用途 電気代
$.5.出金 11200
$.6.日 31
$.6.科目 通信費
$.6.用途 プロバイダ
$.6.出金 8000
$.7.日 31
$.7.科目 給与
$.7.用途 1月分
$.7.入金 250000
$.8.日 31
$.8.科目 通信費
$.8.用途 NTT基本料金
$.8.出金 1700
```

一文字目の「\$....」はわからないけれども、よく見るとちゃんと金額も数字がありますし、通信費だったり家賃だったりとしっかり書いてある様子ですね。

JSONPath-value形式は、一定の規則に従って並び替えられたものになるので、内容が増えたり減ったりするものではありません。

ではどのように並び替えられたのかというと、公式のページ(英語) <http://goessner.net/articles/JsonPath/> に書いてあるので読んでみてください。

英語ができないので、google翻訳こんにゃくを使って分かったことは、

- XML文書からデータを抽出するために考えられたらしい。
- \$はルートオブジェクトを指しているらしい。
- .はカレントオブジェクトを指しているらしい。
- []は繰り返し載せたい場合に使う?みたいだ。よくわからない。
- *はワイルドカードといって、すべての名前を表しているらしい。
- JSONPath列と値列の間は半角空白1つ空けるルールで、例えば1行目の「\$.1.日 10」というのは、「\$.1.日」がJSONPath列で、必ず半角の空白を挟んだとなりの「10」が値部という構造になっているようだ。

といったところでしょうか。

例えば表形式(お小遣い帳)一行目の

| No | 日 | 科目 | 用途 | 入金 | 出金 |
|----|----|----|---------|----|-----|
| 1 | 10 | 食費 | オーソンの弁当 | | 550 |

このデータを表すにはエクセルなら(A2:F2)だけれども、JSONPath-value形式の場合では「\$.1.*」と書くことができ、

また、全ての出金データが知りたい場合は「\$.*.出金」と表すことができる、ということらしい。もしエクセルなら(F2:F9)ですね。

こうやって同じ情報を、違う方法で表現することができるんだ、ということがわかって頂けたかと思います。

つまり、エクセルのような行と列で構成されていたデータが、JSONPath-value形式では1セルを1行単位で表すことができるということなんです、素晴らしい!

私はこのことを知り、データベースはSQLやエクセルだけのものじゃねーぞと言われたような気がして、衝撃が走り、カルチャーショックを受けました。こういう方法もあるんだと。常識が否定された感じでしたね。皆さんはいかがでしょう?

JSON形式

そして冒頭のツイートに戻りますが、なぜ私がガッツポーズをしたのかというと、makrj.shというスクリプトが登場したからです。

makrj.shはJSONPath-value形式をJSON形式に変換するスクリプトで、JSON形式というのは、JSON (JavaScript Object Notation)の略で、これもデータ交換フォーマットで、やはり簡易のデータベースとして利用されています。

具体的にはこういう形式になります。お小遣い帳のデータを使って変換しました。

JSON形式(../DATA/okodukai_json.json)

```
{
  "1": {
    "出金": 550,
```

```

        "用途": "オーソン弁当",
        "科目": "食費",
        "日": 10
    },
    "2": {
        "出金": 900,
        "用途": "水道代",
        "科目": "光熱費",
        "日": 24
    },
    "3": {
        "出金": 1000,
        "用途": "ガス代",
        "科目": "光熱費",
        "日": 25
    },
    "4": {
        "出金": 70000,
        "用途": "1月分",
        "科目": "家賃",
        "日": 25
    },
    "5": {
        "出金": 11200,
        "用途": "電気代",
        "科目": "光熱費",
        "日": 27
    },
    "6": {
        "出金": 8000,
        "用途": "プロバイダ",
        "科目": "通信費",
        "日": 31
    },
    "7": {
        "入金": 250000,
        "用途": "1月分",
        "科目": "給与",
        "日": 31
    },
    "8": {
        "出金": 1700,
        "用途": "NTT基本料金",
        "科目": "通信費",
        "日": 31
    }
}

```

JSON形式もメモ帳で手入力できるのですがこれがまあ 面倒くさいし、必ず入力ミスします。本当に面倒くさい。

そこへmakrj.shが発表され、一発で変換できるということがわかり、ガッツポーズをしたというのが事の顛末になります。

makrj.shの登場により、表計算形式からJSONPath-value形式、そしてJSON形式と、必要な時に自由自在に行き来することができるようになったのです。

以上、makrj.shがあまりにもよかった理由でした。

JSONPath-value形式用例

私はJSONPath-value形式がデータベースとしても使えることが分かったので、いろいろなデータを、個人的なデータベースとして利用しています。

ここからは、私が利用している事例を載せていきたいと思います。あくまでも一例です。これが正しいというわけではありません。

本

```
$.4863542097.タイトル Windows/Mac/UNIXすべてで20年動くプログラムはどう書くべきか一度書けばどこでも、ずっと使えるプログラムを待ち望んでいた人々へ贈る
$.4863542097.出版社 シーアンドアール研究所
$.4863542097.ページ数 336
$.4774173444.タイトル シェルプログラミング実用テクニック
$.4774173444.出版社 技術評論社
$.4774173444.ページ数 416
```

ISBN番号をキーにして、タイトル、出版社、ページ数と必要なデータを置いています。もし他にデータが必要になれば、その都度増やしていけばいいでしょう。

住所録

```
$.1.姓 山田
$.1.名 太郎
$.1.郵便番号 529-0505
$.1.住所1 滋賀県長浜市余呉町小原
$.1.住所2 *-*-*
$.1.電話番号 045-**-****
$.2.姓 田中
$.2.名 一郎
$.2.郵便番号 915-0242
$.2.住所1 福井県越前市栗田部町
$.2.住所2 *-*-*
$.2.電話番号 078-**-****
```

これも同様です、住所1と住所2で分けていますが、わかりやすいから分けているだけで、1行にしても問題ありません。

例えばマイナンバーとかを、新たに付け足したくなったら、
「\$.1.マイナンバー ***-**-****」と行を追加すればいいだけです。

[index](#)