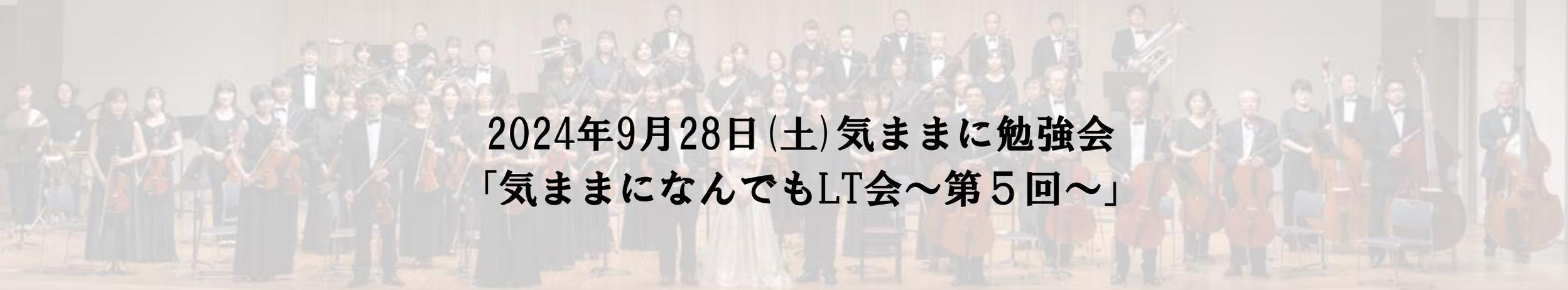


Power Automateで 月の最終営業日にする 仕事を自動化したい

2024年9月28日(土) 気ままに勉強会
「気ままになんでもLT会～第5回～」



自己紹介

- ・ ハンドルネーム : yukio
- ・ X (旧Twitter) : [@yukio 365](https://twitter.com/yukio365)
- ・ ブログ : <https://yukio-ms365.hatenablog.com/>



- ・ 山口県出身・在住、1990年生まれ
- ・ 県内の非IT企業勤務中の2022年3月頃にPower Platformに出会い、夢中で勉強した結果、社内で幾つものアプリ/フローを運用する市民開発者に
→2024年9月～Power Platform関連のお仕事にフルリモ転職
- ・ 趣味はコントラバス演奏 (オーケストラ、吹奏楽団) 今はお休み中だがいつか復帰したい。

【チョットデキル】

Power Appsキャンバスアプリ
Power Automate(クラウドフロー)
Power Automate for desktop

【勉強中】

Power Appsモデル駆動型アプリ
Dataverse
Power BI

今回のお話、ブログにまとめております

[https://yukio-
ms365.hatenablog.com/entry/2024/05/06/115149](https://yukio-ms365.hatenablog.com/entry/2024/05/06/115149)



前職時代のお話

- ・ 山口県に本社があって、東京に営業所がある会社で市民開発者をしていました。
- ・ 東京に出張した際、東京営業所の社員から「**月の最終営業日に特定のメールを送る仕事があるが、それを忘れがち**」という話を聞きました。
- ・ 山口本社の人向けのアプリばかり作っていたのに、本社の人にはPower Platform になに一つ興味を示さない中、東京の人が一番最初に興味を示してくれて嬉しくなりました。
- ・ ちょうど自分も月末営業日にしたい仕事があったので、検証作業をした結果、フローを完成させることができました。
- ・ 今回はそのフローを紹介します。



月の最終日を算出するのは簡単

The screenshot shows a workflow editor with three steps:

- Step 1: 手動でフローをトリガーします** (Manual trigger). It includes a field for "対象の日" (Target date) with a placeholder "日付を入力または選択してください (YYYY-MM-DD)".
- Step 2: 時間への追加 1か月足す** (Add time 1 month). It has fields for "基準時間" (Reference time) set to "対象の日", "間隔" (Interval) set to "1", and "時間単位" (Time unit) set to "ヶ月".
- Step 3: 時間からの減算 dayOfMonthを引く** (Subtract time day of month). It has fields for "基準時間" (Reference time) set to "算出時間", "間隔" (Interval) set to "dayOfMonth(...)", and "時間単位" (Time unit) set to "日".

例

9月28日



(1か月足す)



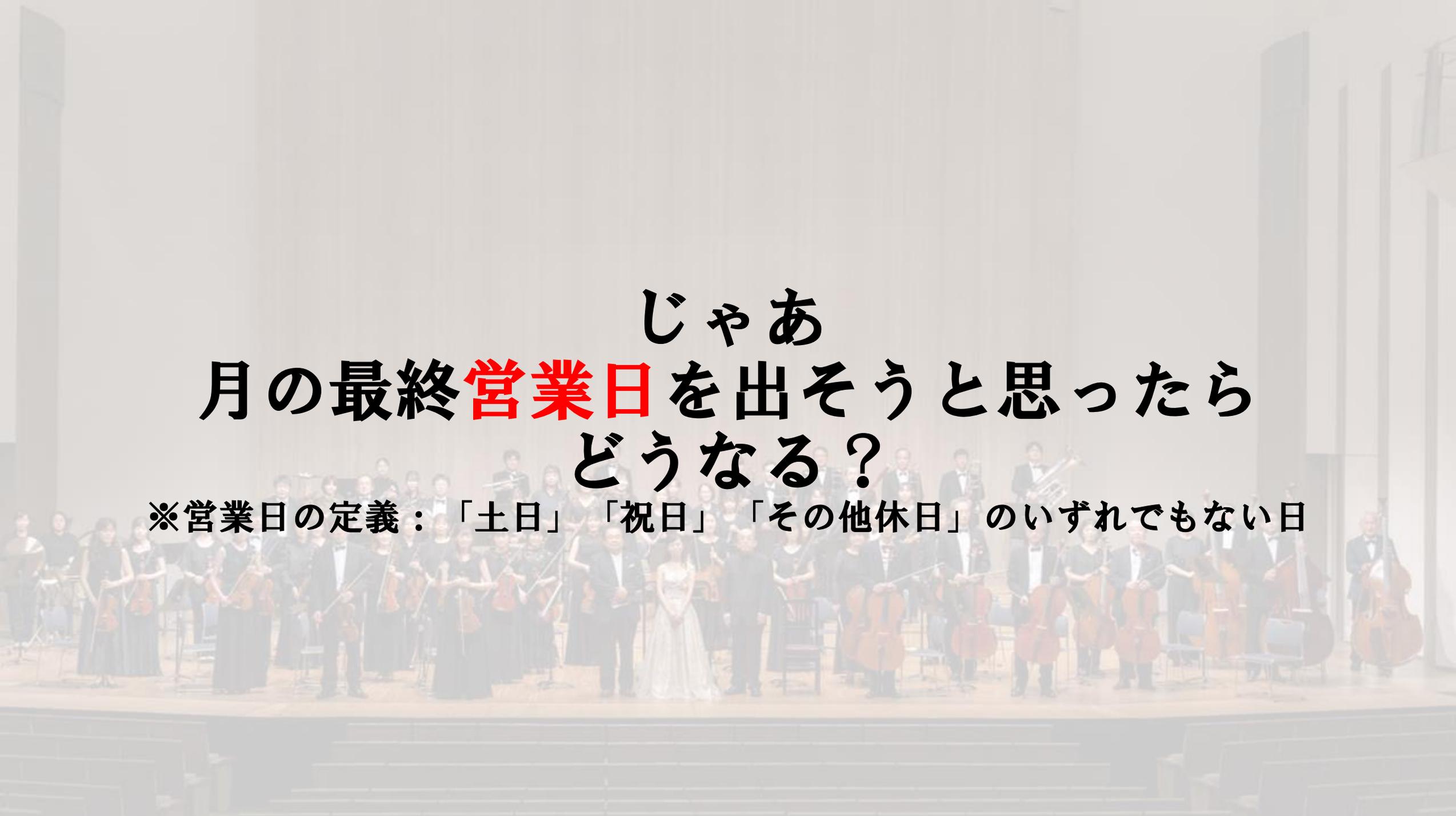
10月28日



(日数(28日)を引く)



9月30日 (最終日)



じゃあ
月の最終**営業日**を出そうと思ったら
どうなる？

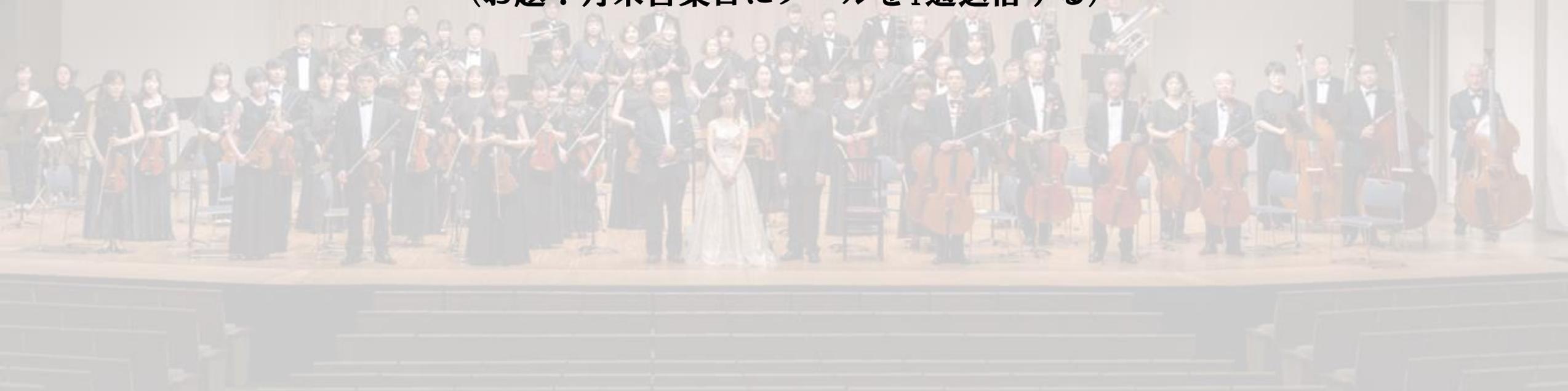
※営業日の定義：「土日」「祝日」「その他休日」のいずれでもない日

結構大変でした…



では1個ずつ見ていきます

(お題：月末営業日にメールを1通送信する)



(1) トリガー → タイムゾーン変換

(SNSでみた情報)

トリガーを**平日だけ**にすると後のアクションが簡略化できそう

繰り返し

* 間隔: 1

* 頻度: 日

プレビュー
毎日の 9:30 に実行する

詳細オプションを表示する

現在の時刻

この手順では、追加情報が必要ありません。後続の手順で出力を使用できるようになります。

+

タイムゾーンの変換

* 基準時間: 現在の時刻 ×

書式設定文字列: yyyy/MM/dd ×

* 変換元のタイムゾーン: (UTC) 協定世界時

* 変換先のタイムゾーン: (UTC+09:00) 大阪、札幌、東京

(2) 変数の初期化

{x} 変数を初期化する 対象日 ? ...

*名前

*種類

値



{x} 変数を初期化する 曜日番号 (日曜日0・・・土曜日6) ? ...

*名前

*種類

値

{x} 変数を初期化する 祝日 ? ...

*名前

*種類

値



{x} 変数を初期化する 営業日判定用変数 (整数) ? ...

*名前

*種類

値

なぜ1にするのか？は後ほど解説

(3) 一旦「月の最終日」を求める

- ・ 土日祝はこの時点では考慮しない

時間への追加 1か月先

* 基準時間 変換後の時間 ×

* 間隔 1

* 時間単位 ケ月

+

時間からの減算 dayOfMonth(対象日)を引いて今月の最終日を求める

* 基準時間 算出時間 ×

* 間隔 dayOfMonth(...) ×

* 時間単位 日

(4) さっき求めた月の最終日情報を変数に格納

{x} 変数の設定 対象日

*名前 Day

*値 `formatDateTim...`

{x} 変数の設定 対象日の曜日番号

*名前 WeekDay

*値 `dayOfWeek(...)`

`formatDateTime (`

時間からの減算 `dayOfMonth(対象日)`を引いて今月の最...

算出時間

`'yyyy/MM/dd'`

)

`dayOfWeek (`

時間からの減算 `dayOfMonth(対象日)`を引いて今月の最...

算出時間

)

(5) 祝日判定 (初回)

Outlookの「日本の祝日」カレンダーを使う
その日が祝日であれば何らかの配列
(JSON)、祝日じゃなかったら空の配列が
取得できる

イベントのカレンダー ビューの取得 (V3)

*カレンダー ID: 日本の休日

*開始時刻: Day × T00:00:00

*終了時刻: Day × T01:00:00

詳細オプションを表示する

JSON の解析 対象日のカレンダービューのJSON解析(初回)

*コンテンツ: value ×

*スキーマ

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "subject": {
        "type": "string"
      },
      "start": {
        "type": "string"
      }
    }
  }
}
```

サンプルから生成

変数の設定 対象日は祝日か?

*名前: NationalHoliday

*値: 本文 ×

JSONを解析

配列を変数に入れる

参考：JSON解析

手動でフローをトリガーします

トリガーの日付 日付を入力または選択してください (YYYY-MM-DD)

+ 入力の追加

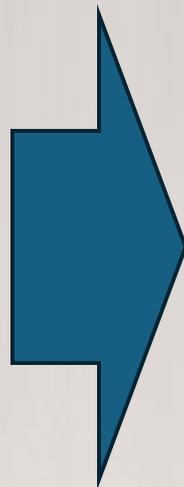
イベントのカレンダー ビューの取得 (V3)

*カレンダー ID 日本▼

*開始時刻 トリガーの日付 × T00:00:00

*終了時刻 トリガーの日付 × T01:00:00

詳細オプションを表示する ▼



JSON の解析

コンテンツ

*スキーマ

サンプルから生成



サンプル JSON バイロードの挿入

完了

これだけのフロー作って走らせて、祝日のJSONをコピー

ここにJSONを貼り付ける

(6) Do Until

Do until 土日祝かどうかの判定ループ

{x} Flag x 次の値に等しい 0

詳細設定モードで編集

制限の変更

{x} 変数の設定 いったんFlag変数リセット

*名前 Flag

*値 0

条件 曜日判定 (0なら日曜日、6なら土曜日)

条件 祝日判定 (emptyがtrue = 祝日ではない)

条件 祝日ではないが一般的な休日があるのでそれを判定する

条件 1以上ならもう1回ループしないといけなので日数を1日前にする

ここには「Do Untilを抜ける条件」を入れる。

このDo Untilは必ず1回以上通ってほしいので、変数「Flag」は初期化の時点で1にしておいた

{x} 変数を初期化する 営業日判定用変数 (整数)

*名前 Flag

*種類 整数

値 1

(6-1) 土日かどうかの判定

条件 曜日判定 (0なら日曜日、6なら土曜日) ...

`f_x createAr...` × 次の値を含む `{x} WeekDay` ×

+ 追加

✓ はいの場合

`{x}` 変数の値を増やす 土日なので+1

*名前	Flag
値	1

✗ いいえの場合

アクションの追加

`createArray(0,6)`

createArrayについては

- ・ 以前、私のSNS投稿に対して、Hiroさんがブログにまとめて下さいました。
- ・ <https://mofumofupower.hatenablog.com/entry/2022/08/02/155107>



(6-2) 祝日かどうかの判定

条件 祝日判定 (emptyがtrue=祝日ではない) ...

`empty(...)` × 次の値に等しい `true` ×

+ 追加

✓ はいの場合

アクションの追加

✗ いいえの場合

{x} 変数の値を増やす 祝日なので+1

*名前 Flag

値 1

empty (
 変数
 {x} NationalHoliday)

(6-3) 祝日ではないが一般的な休日の判定

※ブログ未掲載。追記予定

The screenshot shows a Power Automate flow with the following components:

- Condition:** 条件 祝日ではないが一般的な休日があるのでそれを判定する (Condition: It is not a holiday but there are general holidays, so judge it).
- Array:** 1229と1230 (1229 and 1230).
- Logic:** createArray (create array) followed by formatDateTime (format date and time).
- True Path (はいの場合):** Contains an action '変数の値を増やす' (Increase variable value) with variable name 'Flag' and value '1'.
- False Path (いいえの場合):** Empty path with 'アクションの追加' (Add action) button.

Red arrows point from the 'createArray' and 'formatDateTime' actions in the logic to their respective code snippets in the red boxes below.

`createArray('12/29', '12/30')`
※他の日付入れてもOK

`formatDateTime(`

Snippet of the formatDateTime action configuration:

- 変数 (Variable): {x} Day

`, 'MM/dd')`

(7-4) Flagの値に応じて処理を分ける

条件 1以上ならもう1回ループしないといけないので日数を1日前にする ...

{x} Flag x	次の値以上	1
------------	-------	---

+ 追加

「土日」「祝日」「一般的な休日」のどれか1つでも当てはまると、変数「Flag」は1以上の値になるので、1日前で同じ判定を繰り返す。0であればその日が最終営業日となるので、Do Untilを抜ける

Do until 土日祝かどうかの判定ループ ...

{x} Flag x	次の値に等しい	0
------------	---------	---

[詳細設定モードで編集](#)

[制限の変更](#)

(7-4) 最終営業日でなかった場合

✓ はいの場合

× いいえの場合

🕒 時間からの減算 Day

*基準時間 {x} Day x

*間隔 1

*時間単位 日

アクションの追加

1日減算

{x} 変数の設定 Day

{x} 変数の設定 WeekDay

📅 イベントのカレンダービューの取得 (V3) 2

{/} JSON の解析 対象日のカレンダービューのJSON解析(ループ内)

{x} 変数の設定 NationalHoliday

1日前に対して(4) (5)と同じことをして、次のDo Untilの準備

(8) 今日是最終営業日か？

条件 こうして求まった月末営業日と、トリガーの日付はイコールか？ ...

formatD... x 次の値に等しい formatD... x

+ 追加 v

formatDateTime (

タイムゾーンの変換

変換後の時間

, 'yyyy/MM/dd')

formatDateTime (

変数

{x} Day

, 'yyyy/MM/dd')

最終営業日が入ってる

(8-1) 最終営業日だったらメールを送る

✓ はいの場合

メールの送信 (V2)

*宛先

*件名

*本文

Font 12 **B** *I* U

今日は月の最終営業日です。

〇〇の件よろしくお願いいたします。

詳細オプションを表示する ▾

✗ いいえの場合

アクションの追加

今後の課題と、試してみたいこと

- ・ 「毎日」にしていたトリガーを「平日だけ」にしてもっと簡略化する
- ・ 「祝日判定はGoogleカレンダーの方が安定する」という話をどこかで見たので、GoogleカレンダーVer.も作ってみたい
- ・ 「最終営業日のn営業日前」みたいなことも実装できないか試してみる



ご清聴ありがとうございました

