

SharePointリストの書式設定(JSON) チートシート

投稿日 2020年02月11

日

はじめに

SharePoint モダンUIを利用している人向け。

JSON記述による、リストの書式設定チートシートを作成しました。

フィールドの値を取得

記述	説明
@currentField	カレントの数値、テキスト、選択肢、日付フィールドを取得
@currentField.title	ユーザの表示名を取得
@currentField.lookupValue	参照列の表示テキストを取得
[\$内部名]	指定した列の値を取得
@now	今日の日付を取得(1日=86,400,000)
@currentWeb	サイトのURLを取得(Onlineのみ)
@me	ログイン中のユーザを取得
@rowIndex	ビューのindex行を取得(0スタート)
@window.innerHeight	ブラウザサイズの高さを取得
@window.innerWidth	ブラウザサイズの幅を取得

参照フィールド

```
{  
  "lookupId": "100",  
  "lookupValue": "North America",  
}
```

ハイパーリンク フィールド

@currentFieldでURLを取得します。

@currentField.descで表示文字を取得します。

人物フィールド

人物フィールドは以下のプロパティを保持している0

```
{  
  "id": "122",  
  "title": "Kalya Tucker",  
  "email": "kaylat@contoso.com",  
  "sip": "kaylat@contoso.com",  
  "picture": "https://contoso.sharepoint.com/kaylat_contoso_com_MThumb.jpg?t=63576928822",  
  "department": "Human Resources",  
  "jobTitle": "HR Manager"  
}
```

場所フィールド

場所フィールドを追加する場合は、ビューから追加する必要がある。

また、Jsonの記述はビューから出来ないなので、リストの設定で設定します。

```
{  
  "Address": {  
    "City": "Knoxville",  
    "CountryOrRegion": "United States",  
    "State": "TN",  
    "Street": "963 Worlds Fair Park Dr"  
  },  
  "Coordinates": {  
    "Latitude": "35.961673736572266",  
    "Longitude": "-83.92420959472656"  
  },  
}
```

```
"DisplayName": "World's Fair Park",
"LocationUri": "https://www.bingapis.com/api/v6/localentities/8346bf26-6da4-104c-6ba5-2334b83f6ac8?setLang=en"
}
```

内部名

内部名は表示されている列の名前とは異なる為、以下の方法で確認する

1. リストの設定を表示
2. 確認したい列のリンクをクリック
3. URLの「&Field=」以降に続く文字をコピー
4. 「%5F」が含まれる場合は、「_」に置き換える

サムネイル

ドキュメントライブラリでは、サムネイルの取得が可能

記述	説明
@thumbnail.small	サムネイル(小)のURLを取得
@thumbnail.medium	サムネイル(中)のURLを取得
@thumbnail.large	サムネイル(大)のURLを取得
@thumbnail.<bouding size>	bouding sizeを超えないサムネイルURLを取得
@thumbnail.<bouding width>x<bouding height>	高さx幅を超えないサムネイルを取得

記述方法

スキーマー

vscodeでコードを張り付けると補完がされて便利です。

列の書式設定.js

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/column-formatting.schema.json"
}
```

ビューの書式設定.js

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/view-formatting.schema.json"
}
```

elmType

要素の種類

要素	説明
div	divタグの挿入
span	spanの挿入
a	リンクの挿入
img	画像の挿入
button	ボタンを追加
svg	svgの挿入
path	pathタグを追加

button要素

button要素には、ボタンがクリックされた時にそのアイテムに対して実行される処理を指定できます。

プロパティは「customRowAction」内に定義する必要があります。

アクション	説明
defaultClick	表示する
share	共有する
delete	削除する
editProps	編集する
executeFlow	フローを実行する～

共有する

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/column-formatting.schema.json",
  "elmType": "button",
```

```

"txtContent": "共有",
"customRowAction": {
  "action": "share"
}
}

```

svg要素

svgはあまり使うことないかなーとも思いましたが、リファレンスにも詳しい説明がなかったのでサンプルを載せておきます。以下は埋め込んだ列に▲を描画するだけのコードです。GitHubのサンプルだとgeneric-svgicon-formatが一番参考になりそうでした。

```

{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/column-formatting.schema.json",
  "elmType": "svg",
  "children": [
    {
      "elmType": "path",
      "attributes": {
        "d": "M12.5,0 L0 25 L25 25 Z"
      }
    }
  ]
}

```

コンテンツテキスト

テキストデータを入力するにはtxtContentを使用する。
リッチテキストだとHTMLがエスケープされて、出力されるので注意。

txtContent.json

```

{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/column-formatting.schema.json",
  "elmType": "div",
  "txtContent": "@currentField"
}

```

forEach

複数値フィールドのに対し、値の数分ループが出来る
以下は参照カラム（複数選択可）のデータに対して、選択した項目ごとに枠線をつける処理

ForEach.json

```

{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/column-formatting.schema.json",
  "elmType": "div",
  "children": [
    {
      "elmType": "div",
      "forEach": "region in @currentField",
      "txtContent": "[$region.lookupValue]",
      "style": {
        "border": "1px #000 solid"
      }
    }
  ]
}

```

children

elmType 内に子要素を作る

```

<div>
  <span> @currentField </span>

```

```
</div>
```

をJSONに置き換えることができる

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/column-formatting.schema.json",
  "elmType": "div",
  "children": [
    {
      "elmType": "span",
      "txtContent": "@currentField"
    }
  ]
}
```

式

txtContent, Style, 属性にはExcel形式の指揮、またか構造構文ツリーを使用できる

Excel形式

<例> if文を使って表示を出し分ける

```
{
  "elmType": "div",
  "txtContent": "=if(@currentField == 'テスト', 'テスト中', ' ')"
}
```

<例> 日付を年月(YYYY/MM)表示にする

```
{
  "elmType": "div",
  "txtContent": "=getYear(@currentField)+'/'+(getMonth(@currentField)+1)"
}
```

※getMonthには+1してあげる

抽象構文ツリーの式

"operator"に処理を指定し"operands"の配列を渡して処理をする

単項演算子

オペランド	説明
toString()	文字列を返す
toLowerCase	小文字に変換する
Number()	数値を返す（数値以外: NaN）
Date()	DateTimeに変換して返す
getDate	指定した日付を返す
getMonth	指定した日付の月を返す
getYear	指定した日付の年を返す
toLocaleString()	ロケールに応じた形式でDateTimeを返す
toLocaleDateString()	ロケールに応じた形式で日付のみ返す
toLocaleTimeString()	ロケールに応じた形式で時間のみ返す
cos	角度のコサインを返す
sin	数値のサインを返す
abs	絶対値を返す
length	配列数を返す
floor	整数で返す
ceiling	切り上げて整数で返す

<例> 日付列のデータから日にちを取得して表示する

単項演算子の場合、operandsに渡す配列は1つ

```
{
  "elmType": "div",
  "txtContent": {
```

```

      "operator": "getDate",
      "operands": ["@currentField"]
    }
  }
}

```

同じ内容を式で書くとこんな感じ(オススメはこっち)

```

{
  "elmType": "div",
  "txtContent": "=getDate(@currentField)"
}

```

二項演算子

オペランド	説明
indexOf	検索でヒットしたindexを返す
join	配列を連結する。第二引数の区切り文字を挿入
pow	指数のべき乗を返す

<例> 配列の区切り文字に"/"を入れる

複数選択可の選択枝列に使ったりします。

```

{
  "elmType": "div",
  "txtContent": {
    "operator": "join",
    "operands": ["@currentField", "/"]
  }
}

```

同じ内容を式で書くとこんな感じ

```

{
  "elmType": "div",
  "txtContent": "=join('@currentField','/')"
}

```

三項演算子

オペランド	説明
substring(string, start, end)	文字列の一部を返す

条件演算子

オペランド	説明
?	["条件式", "正の時", "偽の時"]

<例> 日付が30日の時、trueと表示。それ以外はfalseと表示する

```

{
  "elmType": "div",
  "txtContent": {
    "operator": "?",
    "operands": ["=getDate(@currentField)==30", "true", "false"]
  }
}

```

もちろん、同じ処理を式 (if) でも書けます。

```

{
  "elmType": "div",
  "txtContent": "=if('getDate(@currentField)==30','true','false')"
}

```

二項算術演算子

二つの配列が必要なオペランド

オペランド	説明
+	

-	
/	
*	
<	
>	
<=	
>=	

<例> 名前の後ろに「さん」と連結する

オペランドを使った場合

```
{
  "elmType": "div",
  "txtContent": {
    "operator": "+",
    "operands": ["@currentField.title", "さん"]
  }
}
```

式を使った場合

```
{
  "elmType": "div",
  "txtContent": "= @currentField.title + 'さん'"
}
```

オペランドの入れ子

オペランドは入れ子になりがちですが、とても読みづらいです。

以下の公式サンプルを例に分解して説明します。

```
{
  "operator": "?",
  "operands": [
    {
      "operator": ">",
      "operands": [
        "@currentField",
        "40"
      ]
    },
    "100%",
    {
      "operator": "+",
      "operands": [
        {
          "operator": "toString()",
          "operands": [
            {
              "operator": "*",
              "operands": [
                "@currentField",
                2.5
              ]
            }
          ]
        },
        "%"
      ]
    }
  ]
}
```

上記のコードを分解すると3階層になっていることが分かります。

第一階層

第一階層には条件演算子が使われています。
ポイントは(1)条件式と、(2)偽の時に書かれている入れ子です。
operandsは配列を指定するので[]で囲います。配列内でオペランドを入れ子にしたいときは{}で囲んで、その中に式を追加します。

```
{
  "operator": "?",
  "operands": [ { "(1)条件式" }, "100%", { "(2)偽の時" } ]
}
```

第 2 階層

第一階層で{}に囲われていたところが第 2 階層です。

(1)条件式

式で表すと、@currentField > 40 です

```
{
  "operator": ">",
  "operands": [
    "@currentField",
    "40"
  ]
}
```

(2)偽の時

偽の時はさらに入れ子になっていますが、
式で表すとtoString(@currentField * 2.5) + "%"ということです。

```
{
  "operator": "+",
  "operands": [
    {
      "operator": "toString()",
      "operands": [
        {
          "operator": "*",
          "operands": [
            "@currentField",
            2.5
          ]
        }
      ]
    },
    "%"
  ]
}
```

複数フィールド

以下の演算子は複数データを持つフィールドに対して処理が出来る

オペランド	説明
length([配列])	配列の要素数を返す
join([配列] , [区切り文字])	配列を連結する
loopIndex([イテレータ])	forEachで指定したイテレータ変数を引数に入れると、インデックスを取得できる

style(CSSによる装飾)

styleプロパティを使用して、cssを記述できます

```
style.json
{
  "elmType": "div",
  "txtContent": "@currentField",
  "style": {
```

```
"border":"1px #000 solid"
}
}
```

枠線 / 背景色

プロパティ	説明
background-color	背景色の指定
fill	svgファイルの色指定
background-image	背景に画像を指定 ※urlを指定しても反映されない不具合あり？
border	枠線のスタイルを指定
border-color	枠線の色を指定
border-bottom	枠線下部のスタイルを指定
border-style	枠線の線の形状を指定
border-radius	角の丸みを指定
border-top-left-radius	左上の丸みを指定
border-top-right-radius	右上の丸みを指定
border-bottom-right-radius	右下の丸みを指定
border-bottom-left-radius	左下の丸みを指定
border-bottom-color	枠線下部の色を指定
border-bottom-style	枠線下部の線の形状を指定
border-bottom-width	枠線の太さを指定
border-left-XX	枠線左側のスタイルを指定
border-top-XX	枠線上部のスタイルを指定
outline	アウトラインのスタイルを指定
outline-color	アウトラインの色を指定
outline-style	アウトラインの線の形状を指定
outline-width	アウトラインの線の太さを指定

ボックス

プロパティ	説明
display	インラインとかブロックを指定する
box-shadow	影をつける
box-sizing	要素のサイズに枠線を含むかの指定が出来る
box-decoration-break	複数の行、段、ページに渡る場合に描画する方法を指定する
overflow	要素がはみ出たときの制御を指定
overflow-x	要素が左右にはみ出したときの制御を指定
overflow-y	要素が上下にはみ出したときの制御を指定
overflow-style	対応ブラウザ無し
rotation	要素を回転させる
rotation-point	回転する中心座標を決める ※対応ブラウザなし

プロパティ	説明
opacity	不透明度を指定
cursor	要素上にマウスポインタがある時のカーソルの表示を指定

余白

外側の余白

プロパティ
margin
margin-bottom
margin-left
margin-right
margin-top

内側の余白

プロパティ
padding
padding-bottom

padding-left
padding-right'
padding-top

サイズ

プロパティ	説明
height	高さを指定
width	幅を指定
max-height	要素の最大の高さを指定
max-width	要素の最大の幅を指定
min-height	要素の最小の高さを指定
min-width	要素の最小の幅を指定
clip	ボックスをクリッピングする

フォント

プロパティ	説明
font	フォントのスタイルを指定
font-family	フォントファミリーを指定
font-size	文字の大きさを指定
font-style	フォントの斜体などを指定
font-variant	先頭の文字を大きくする(アルファベットのみに対応)
font-weight	フォントの太さを指定
font-size-adjust	小文字の高さの比率を指定
font-stretch	フォントの横幅を指定する

テキストの装飾

プロパティ	説明
color	フォントの色を指定
direction	記述方式の指定
unicode-bidi	Unicode文字の記述方式を上書き
letter-spacing	字間スペースの指定
line-height	行間スペースの指定
text-align	ブロック要素または表セルボックスの水平方向の配置
text-decoration	テキストの装飾線を指定
text-indent	字下げの指定
text-transform	英数字の大文字/小文字表示を指定
white-space	改行/スペース/タブの扱いを指定する
word-spacing	単語の間隔を指定する
text-align-last	最後の行の配置を指定
text-justify	text-align: justifyが指定されたときに両端揃えの種類を指定する
text-shadow	テキストに影を指定
word-break	overflowを使う
word-wrap	overflowを使う
text-wrap	※対応ブラウザ無し
text-outline	※対応ブラウザ無し
punctuation-trim	※対応ブラウザ無し
hanging-punctuation	※対応ブラウザ無し

Flex

親コンテナに指定するプロパティ

プロパティ	説明
flex-direction	並び順を指定
flex-wrap	折り返しを指定
flex-flow	並び順&折り返しを指定
align-items	アイテムの垂直方向位置を指定
justify-content	アイテムの水平方向位置を指定

子アイテムに指定するプロパティ

プロパティ	説明
-------	----

flex-grow	アイテムのサイズ割合を指定
flex-shrink	アイテムの縮小率を指定
flex	アイテムの拡張&縮小率を指定

配置

プロパティ	説明
position	ボックスの基準位置を指定
bottom	下からの位置
top	上からの位置
right	右からの位置
left	左からの位置

float

position:staticのみ適用可能

プロパティ	説明
float	左寄せ/右寄せの指定
clear	回り込みを禁止

Box

Boxは廃止されている為、使うならflexに統一した方がよさそう

プロパティ
box-align
box-direction
box-flex
box-flex-group
box-lines
box-ordinal-group
box-orient
box-pack

グリッドレイアウト

プロパティ
grid-columns
grid-rows

※ グリッドトラックを指定するプロパティがサポートされていない為、使いどころがよくわからなかった…。

表示

プロパティ	説明
visibility	要素を表示/非表示にする

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/column-formatting.schema.json",
  "elmType": "button",
  "txtContent": "Publish to Teams",
  "customRowAction": {
    "action": "executeFlow",
    "actionParams": "={\"id\": \"021ec4e0-8f85-46b3-a557-a4456960a4eb\",
    \"headerText\": \"Publish ' + [$Title] + '\", \"runFlowButtonText\": \"Publish now\"}"
  },
  "style": {
    "background-color": "#008082",
    "color": "white",
    "border-radius": "10px",
    "visibility": "=if([$PublishedToTeams] == 'Yes','hidden','visible')"
  }
}
```

重なり順

プロパティ	説明
z-index	深度を設定

段組み

複数行テキスト(リッチ)ではうまく動かないかも

プロパティ
column-count
column-fill
column-gap
column-rule
column-rule-color
column-rule-style
column-rule-width
column-span
column-width
columns

テーブル

tableタグはサポートしていないけど

Displayで"table"を指定して合えば使える

プロパティ	説明
border-collapse	境界線をくっつける
border-spacing	枠線の間隔を指定
caption-side	配置位置を指定※topとbottom以外はcss2.1で削除されているので注意
empty-cells	内容を持たない表を描画するかしないか
table-layout	横幅の指定
vertical-align	垂直方向の配置を指定 ※ インライン要素にも適用可

属性(attributes)

attributes要素を追加すると以下の属性を設定できます。

属性	説明
href	リンクの示すハイパーリンク
rel	リンク先のソースの属性
src	埋め込むファイルのパス
class	クラス名を指定
target	リンクの表示方法を指定
title	要素の補足(ツールチップで表示される
iconName	UI Fabricのアイコンを追加する
d	svgデータの情報を指定する

WAI-ARIA

属性	
aria	
role	

roleについてはこちらの記事がとても分かりやすくまとまっていました。
<https://qiita.com/nezurika/items/eac689a97895a27b8791>

UI Fabric

色を統一したり、アイコンをつけるときに便利

アイコン

Fabric Iconから欲しいアイコンを探す。
アイコンをつける場合には"iconName"を使用する

```
iconName.json
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/column-formatting.schema.json",
```

```

"elmType": "span",
"txtContent": "テスト",
"attributes": {
  "iconName": "Mail"
}
}

```

カラー

Theme Slotsで欲しい色を探す。
色を指定するには、"class"を使用する。

class	説明
ms-fontColor-themePrimary	フォントの色を指定
ms-bgColor-themePrimary	背景の色を指定
ms-borderColor-themePrimary	枠線の色を指定

```

class.json
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/column-formatting.schema.json",
  "elmType": "div",
  "txtContent": "テスト",
  "attributes": {
    "class": "ms-bgColor-red"
  }
}

```

ホバー時の色

"--hover"を末尾につけることで、ホバー時の色を指定できます

class	説明
ms-fontColor-themePrimary--hover	ホバー時のフォントの色を指定
ms-bgColor-themePrimary--hover	ホバー時の背景の色を指定
ms-borderColor-themePrimary--hover	ホバー時の枠線の色を指定

```

class.json
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/column-formatting.schema.json",
  "elmType": "div",
  "txtContent": "テスト",
  "attributes": {
    "class": "ms-bgColor-red--hover"
  }
}

```

フォントサイズ

class	説明
ms-fontSize-s	文字サイズ小
ms-fontSize-m	文字サイズ中
ms-fontSize-l	文字サイズ大
ms-fontSize-su	文字サイズ特大

条件付き演算子(if)

Excel形式で記述が出来る

※「=」は「==」にする必要がある

例) 値が70の時、「just」それ以外の時は空白を表示する

```

sample.json
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/column-formatting.schema.json",
  "elmType": "div",
  "txtContent": "=if(@currentField == 70,'just', '')"
}

```

}

ビューの書式設定

レイアウトの指定

記述	説明
additionalRowClass	列の書式設定は残る
rowFormatter	列を完全に上書きする
tileProps	タイルレイアウトを指定する

ビューに対する設定

記述	説明
hideSelection	行を選択する機能を無効(=true)
hideColumnHeader	ビュー内の列見出しを非表示(=true)

タイルレイアウト

タイルレイアウト(tileProps)のプロパティ。その他のレイアウトはビューの書式設定を参照。
tilePropsには書式を指定する特定のプロパティが存在する。

記述	説明
height	高さを指定
width	幅を指定
formatter	rowFormatter(列の書式設定)と同じ記述が出来る
hideSelection	行を選択する機能を無効

tileProps.jeson

```
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/view-formatting.schema.json",
  "tileProps": {
    "height": "250",
    "width": "200",
    "hideColumnHeader": true,
    "hideSelection": false,
    "formatter": {
      "elmType": "div",
      "txtContent": "[$Title]",
      "style": {
        "border": "1px #000 solid"
      }
    }
  }
}
```