

TransAbsentGAS メンテナンス用文書

1. 関数についての説明

下記以外の関数については中身を書き換えることを想定していない。

onFormSubmit(formData)関数

引数`formData`は`FormResponse`クラスの値で、フォーム全体に対する回答が入る。

`itemResponses = formData.response.getItemResponses();`にて、変数`itemResponses`に`ItemResponse[]`クラスの値を取得する（リストになっていることに注意）。この`itemResponse`クラスには、回答内容を始めとし、質問のタイトルなども含まれており余分な情報が多いため、

```
for (let i = 0; i < itemResponses.length; i++) {  
  responseslist.push(itemResponses[i].getResponse()); // 回答をリストに連結  
}
```

変数`responseslist`に情報を整理する。`responseslist`には0から順に回答内容（`string`型）のみが格納されている。例えば、`responseslist[0]`には回答した「名前」の情報が入っているし、`responseslist[1]`には振替/欠席/ドア開けのいずれかの`string`型の値が入っている。

そのため、`responseslist[1]`の値を用いて、分岐処理を作ることができる。

```
if (responseslist[1] === "振替") {  
  }  
// 欠席を選択したとき  
else if (responseslist[1] === "欠席") {  
  }  
// ドア開けを選択したとき  
else if (responseslist[1] === "ドア開け") {  
  }  
// それ以外（未実装）の選択肢を選択したとき  
else {  
  }
```

この文字列比較処理は、フォームの回答選択肢に依存しているため、フォームを改変する際には右辺の文字列を合わせるようにする。また、振替/欠席の処理においては

```
notifyToSlack(messtr, getWebHookURLfromClassName(responseslist[2]), true);
```

`notifyToSlack()`関数の第2引数でWebHookURLを取得するために`getWebHookURLfromClassName()`関数を呼び出す際に`responseslist[2]`を用いるため、フォームの回答選択肢に依存する。そのため、シートのクラス識別名と一致するように選択肢を設定する必要がある。

※最重要注意事項※

Googleフォームの仕様により、スキップされたセクションの回答内容は変数`responseslist`には含まれない。

そのため、セクションと質問の配置の仕方によっては分岐時に使用する`responseslist`のインデックス番号がずれる可能性がある。セクションや質問内容を変更する場合には、次の図のように状況を整理することを推奨する。

ID	振替選択時	欠席選択時	ドア開け選択時		
0	お名前	お名前	お名前		セクション1
1	振替/欠席/ドア開け	振替/欠席/ドア開け	振替/欠席/ドア開け		セクション2
2	現在の受講クラス	現在の受講クラス	自由記述		セクション3
3	受講を希望するクラス	自由記述			セクション4
4	自由記述				

上の例では、`responseslist[0]`及び`responseslist[1]`は全ての選択肢で同じとなるが、自由記述はそれぞれ`responseslist[4]`（振替）、`responseslist[3]`（欠席）、`responseslist[2]`（ドア開け）に格納されることになる。

getWebHookURLfromClassName(className)関数

引数`className`は`string`型の値で、クラス識別名と照らし合わせて検索を行う。

`const sheet = getWebHookSheet();`の処理により、変数`sheet`にWebHookURLのシートを取得する。エラー時は`null`、そうでないときは`Sheet`情報が入る。

`SpreadsheetApp`を用いてスプレッドシートのデータを使用するとき、

1. スプレッドシート全体（`SpreadSheet`クラス）を取得
2. 特定のシート1枚（`Sheet`クラス）を取得
3. セル（`Range`クラス）にアクセス

という手法を取るのが通例である。参照するシートは、`getWebHookSheet()`関数にて取得する。

`classInfo = sheet.getRange(2, 1, sheet.getLastRow() - 1, 2).getValues()`という処理で、`getRange()`にて指定したセルの範囲から値を読み出す。値は2次元リストとして読み出されて`classInfo`に代入される。これは2行目1列目から値が入力されている最終行-1行、2列分読み出しを行う。言い換えると2行目～値が入力されている最終行、AB列の値を読み出すということである。

```
for (let i = 0; i < classInfo.length; i++) {
  if (className === classInfo[i][0]) {
    return classInfo[i][1]; // 正常
  }
}
```

`classInfo[i][0]`はA列の値が入っており、クラス識別名が入っている。`classInfo[i][1]`はB列の値が入っており、WebHookURLが入っている。よって、`className`と一致するクラス識別名のWebHookURLを戻り

値とする処理になっている。

2. 知っておきたいプログラミング知識(JavaScript/GAS)

文字列連結編

`string`型における`+`演算子は、文字列の連結を表している。例えば、次の例を考えてみよう。

```
let a = "Hello,";
let b = "World.";
let c = a + b;
Logger.log(c);
```

ここでは、`a`, `b`, `c`の3つの変数を宣言し、`c`に`a + b`を代入している。ここで`c`に入っている値は`"Hello,"`と`"World."`を連結した`"Hello,World."`である。

このように`string`型においては、`+`演算子を用いることで文字列の連結を行うことができる。

例外対策編

`try`, `catch`文を用いることで例外発生時の処理を決めることができる。`try`, `catch`文は通常次のように書かれる。

```
try {
  // 例外が発生する可能性のある処理（主にライブラリ）
} catch (e) {
  // 例外発生時にどのように対応するか
  Logger.log(e);
}
```

この「例外が発生する可能性のある処理」というのは、自分でプログラミングした処理ではなくライブラリに頼っている処理のことである。今回の場合であれば`SpreadsheetApp.openByUrl(spreadSheetURL)`などの処理が該当する。これは`spreadSheetURL`で指定されたGoogleスプレッドシートを開くライブラリ関数であるが、これはGoogleによって提供されている関数であり、自分でプログラミングしたものではない。ライブラリ関数で発生した例外は`try`, `catch`文で対応するのが一般的であり、ここでは`e`に例外情報を代入させている。`Logger.log(e)`とプログラムに書いておけば、例外情報を出力することができる。

ライブラリ編

Google Apps Scriptにおけるプログラミングでは、Googleが提供する各種Appを用いることが多い。

TransAbsentGASにおいては、`SpreadsheetApp`と`FormApp`を使用している。各クラスの変数に対して、適用できる操作は下記のリファレンスを参照すること。

- SpreadsheetApp
<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet-app>

- Spreadsheetクラス : Googleスプレッドシートファイルにアクセスして変更する
<https://developers.google.com/apps-script/reference/spreadsheet/spreadsheet>
- Sheetクラス : スプレッドシート シートにアクセスして変更する
<https://developers.google.com/apps-script/reference/spreadsheet/sheet>
- Rangeクラス : スプレッドシートの範囲にアクセスして変更する
<https://developers.google.com/apps-script/reference/spreadsheet/range>
- FormApp
<https://developers.google.com/apps-script/reference/forms/form-app>
 - FormResponse : フォーム全体に対する回答
<https://developers.google.com/apps-script/reference/forms/form-response>
 - ItemResponse : フォーム内の 1 つの質問項目に対する回答
<https://developers.google.com/apps-script/reference/forms/item-response>