

# Google Apps Scriptを使った施設予約システムの開発

## Development of facility reservation system using Google Apps Script

真壁 友

MAKABE Tomo

キーワード：Google Apps Script, プログラミング, システム開発

Keywords：Google Apps Script, Programming, Agile development

### 要旨

2020年4月から新型コロナウイルス感染症の影響で大学の授業が遠隔授業となった。その後、学生の学内施設の利用を段階的に進めていった。その過程で施設予約のシステムを、急遽開発する必要が生じた。そこで簡易な開発が可能な開発環境としてGoogle Apps Scriptを用い、大学施設の予約システムを作成した。本報告はその内部の仕組みについて解説する。

### 1. はじめに

2020年4月に新型コロナウイルス感染防止のために、長岡造形大学では校舎の閉鎖を行なった。授業はzoomなどを使い遠隔で行なった。しかし、全ての授業、演習を行うことは困難であり、実習や演習を行う授業は学内の施設を利用することとなった。授業以外でも学生が作品制作を行う都合上、学内の施設を学生に開放する必要があった。そこで、学生が学内で密にならないように管理しながら施設を利用してもらう必要が生じた。そのために、オンラインで施設予約を行えるシステムが必要となった。予約管理をメールなどで行うには、人手による処理が必要となり運用が困難である。また、長岡造形大学の学内で使用しているシステムPaletteでは小回りのきく運用が難しい。そこで急遽、学内の施設利用を行えるように施設の予約システムを構築する必要があった。本報告では、そのシステムの概要とコードについての解説を行う。

### 2. 開発環境

開発にはGoogle Apps Scriptを用いた。予約システムであるのでログイン処理が必須となる。学外者や他学生による不正な予約やアクセスを防止するために身元をはっきりとさせたログインが必要となる。新規のシステムを運用すると、それに伴うIDとパスワードの発行が必要となる。その情報の周知までを考えると膨大な手間を要するので、それは避けたい。そこですでに活用しているGoogleのシステムを利用することとした。GoogleドライブやGメールなどをすでに運用しているので、学生と教職員はアカウントを持っている。これを利用することでIDやパス

ワード発行の手間をなくした。また、ログイン処理などはGoogleの管理に任せ、その開発を省くことができるのも大きな理由である。またGoogle App ScriptはGoogleカレンダーやGoogleスプレッドシートとも連携して情報の更新、管理が比較的行いやすいという面もある。

Google Apps Scriptの開発画面を図1に示す。この画面はGoogleドライブの「新規」ボタンから「その他」→「Google Apps Script」と選ぶことで表示される。「コード.gs」はデフォルトで作成されるファイルになる。

### 3. 必要となるページとコード

システムに必要なのは「予約状況を見るためのページ」図2と「予約を入力するページ」図4の2つである。

ユーザーは予約状況を確認し、空いていれば予約ページに移動して予約入力を行う。予約できる施設はスプレッドシート(図3)に入れたものがポップアップメニューに表示される。2021年9月現在では図3に記載されている施設だけを管理している。このスプレッドシートに定員を設定することにより、定員までの予約入力が可能となっている。

#### 3.1 予約状況ページ

予約状況を見るための画面は図2を参照。このページのindex.htmlとコード.gsはgitHubを参照していただきたい。

index.html

[https://github.com/mkbtm/NID\\_FacilityReservation/blob/main/yoyakujoukyou\\_index.html](https://github.com/mkbtm/NID_FacilityReservation/blob/main/yoyakujoukyou_index.html)

コード.gs

[https://github.com/mkbtm/NID\\_FacilityReservation/blob/main/yoyakujoukyou\\_code.gs](https://github.com/mkbtm/NID_FacilityReservation/blob/main/yoyakujoukyou_code.gs)

コード1はindex.htmlでこのページの基本デザインを決めている。「コード.gs」はGoogle Apps Scriptの処理、スプレッドシートやカレンダーの情報を扱う処理をしている。またユーザの操作に伴う処理はindex.htmlのJavaScriptで行なっている。このJavaScriptからコード.gsの関数を呼び出し処理を行なっている。



図1 Google Apps Script 開発画面

### 3.2 予約入力画面

予約入力画面(図3)も処理の基本としては確認画面と同じである。index.htmlとコード.gsはgitHubのリンクを参照していただきたい。

index.html

[https://github.com/mkbtm/NID\\_FacilityReservation/blob/main/yoyakunyuryoku\\_index.html](https://github.com/mkbtm/NID_FacilityReservation/blob/main/yoyakunyuryoku_index.html)

コード.gs

[https://github.com/mkbtm/NID\\_FacilityReservation/blob/main/yoyakunyuryoku\\_code.gs](https://github.com/mkbtm/NID_FacilityReservation/blob/main/yoyakunyuryoku_code.gs)



図2 予約確認画面



図4 予約入力画面

施設名	定員
演習スタジオA	1
演習スタジオB	1
演習スタジオC	1
演習スタジオD	1
演習スタジオE	1
演習スタジオF	1
演習スタジオG	1
演習スタジオH	1
演習スタジオI	1
演習スタジオJ	1
演習スタジオK	1
演習スタジオL	1
演習スタジオM	1
演習スタジオN	1
演習スタジオO	1
演習スタジオP	1
演習スタジオQ	1
演習スタジオR	1
演習スタジオS	1
演習スタジオT	1
演習スタジオU	1
演習スタジオV	1
演習スタジオW	1
演習スタジオX	1
演習スタジオY	1
演習スタジオZ	1

図3 施設名と定員のスプレッドシート

#### 4. 処理手順

処理の中で情報を html → JavaScript → Google Apps Script と受け渡しをしている。また、画面表示を行う場合は逆の流れで情報が受け渡される。

確認画面の施設名のポップアップメニューを例に取り説明する。

index.html が読み込まれると

`<body onload="pageInitialize () ;">` で JavaScript の `pageInitialize ()` が実行される。この中で、次の文で Google Apps Script を呼び出している。

```
appList = google.script.run.withSuccessHandler (prepareMenu)
    .getFacilitiesList () ;
```

コード `.gs` の `getFacilitiesList ()` を呼び出し、処理が終了すると戻り値を持って `prepareMenu` を実行する。

コード `.gs` の `getFacilitiesList` は SSID で指定されたスプレッドシートの値を読み取る。読み取った値は変数 `returnArray` にその値を収納して index.html の JavaScript に処理を返す。変数 `appList` に配列の形式で Google Apps Script から返された値が入っている。以下のような処理で、html のポップアップメニューの内容を更新している。

```
for (var i=1;i<appList.length ;i++) {
    select.options[i-1] = new Option (appList[i][0]) ;
    select.options[i-1].value = appList[i][0];
}
```

このような一連の処理で、スプレッドシートやカレンダーの情報を web ページ上に表示している。本来であればページを表示する前にこのような処理を行ってからページの表示処理を行うのが望ましい。今回は開発期間が短いため、そのような処理を省略した。

なお、今回掲載したコードには css 部分は紙面の都合上省略している。また見た目のデザイン (スタイル) については「やりたい学生が出てきたらやらせよう」という姿勢で、必要最低限でやめている。

#### 5. まとめ

短期間で開発したシステムであるが使用者と開発者が近い立場であり、必要な機能に絞って開発が可能であった。このようなシステムを作る場合、通常は「仕様書」を作成してから取り掛かる。この仕様書を作るのが開発メーカーの SE や営業担当者等である。顧客の要望を聞きながら仕様書を作成するが、顧客も必要な機能がわかっていないので、過大な要求を盛り込む。結果として開発期間、開発コストが増え、使いにくいものができ上がる。

コードを理解する人材が増え、簡単な処理を行うコードであれば組織内で作成できるような未来がこれからは必要であろう。デジタルネイティブとはそういうことであると考える、教育に取り組みたい。

#### 参考文献

「Google Classroom API を Google Apps Script で使ってみる」  
[https://qiita.com/yanosen\\_jp/items/f22bf36e5766395fed0e](https://qiita.com/yanosen_jp/items/f22bf36e5766395fed0e)  
(2021/10/4 最終アクセス)

詳解！ Google Apps Script 完全入門 ～ Google Apps & G Suite の最新プログラミングガイド  
高橋宣成 (著) 秀和システム

Google Apps Script 公式ドキュメントサイト  
<https://developers.google.com/apps-script/reference/document>  
(2021/10/4 最終アクセス)

「Google Apps Script で社内システムを作ってみた (1)」  
<https://www.symmetric.co.jp/blog/archives/1087>  
(2021/10/4 最終アクセス)

#### 付録

本稿で使用しているプログラムソースコードは長岡造形大学リポジトリ (<https://nagaoka-id.repo.nii.ac.jp>) で付録として公開している。

(長岡造形大学リポジトリ / 長岡造形大学研究紀要第 19 号 / 付録 / Google スクリプトを使った施設予約システムの開発【コード】)

