

(研究ノート)

研究紀要第 75 号

Google Classroom のアシストアプリケーションの作成

山口 直木

An Assist Application for Google Classroom

Naoki YAMAGUCHI

要約

本報告は、本学研究紀要 70 号 (2018 年 9 月 28 日刊行) の「Google Classroom を用いた授業管理の利点と問題点」の中で課題とした Google Classroom の問題点を解決するために開発したアシストアプリケーションに関する報告である。

提案するアプリケーションでは、2 つの課題の解決を図っている。まず 1 つは以前の研究ノートで報告した Google Classroom における生徒リストの並びの問題である。もう 1 つの課題は、今年度新型コロナウイルスの感染拡大に伴う遠隔授業の実施時に生じた学生へのクラスコードの連絡方法である。以上の 2 点の問題を解決することを目的としてアプリケーションの作成を行い、その結果を報告する。

「キーワード」 Google Classroom、Google Apps Script、Web アプリケーション

[Abstract]

This report proposes a solution of Google Classroom, which was the issue in "Advantages and Problems of Class Management Using Google Classroom" in Bulletin No. 70 of Takamatsu University and Takamatsu Junior College (published on September 28, 2018).

The proposed application is trying to solve two problems. The first is the problem of student list alignment in Google Classroom, which was raised in the previous report. Another issue is how to inform the class code of Google Classroom to students. This issue is raised when I used Google Classroom in remote lessons which due to the spread of COVID-19 in the previous year. It is also possible to check the submission status of assignments given on Classroom when rearranging the student list.

[keywords] Google Classroom, Google Apps Script, Web Application

はじめに

Google Classroom¹ (以降、Classroom) は米国 Google 社から提供され、無償、直感的な操作などの特徴があり、世界中で多く利用されている学習管理システム (LSM : Learning Management System) である。本学においても自らの授業に利用している教員もおり、また、遠隔授業の拡大とともにその利用は増えつつある。

受理年月日 2020 年 11 月 17 日 *高松大学経営学部教授

以前の研究ノート²において、Classroom 利用の有効性とその課題を報告した。その中で明らかになった課題の 1 つに受講者リストの並びの問題がある。これは、Classroom が国際標準であるため、日本語での 50 音順の並びが行えないというものである。その結果、Classroom と本学の教務システム上の間で学生リストの並びに差異が生じてしまうというものである。

また、今年度の新型コロナウイルスの感染拡大に伴う遠隔授業の導入時、本学でも多くの教員が Classroom を用いて授業管理を行ったが、その際に新たな課題も明らかになった。Classroom は Google アカウントを有する者が Classroom 上のクラスに対応するクラスコードを入力することによりクラスに参加することが可能となる仕組みであるが、そのクラスコードの学生への通知が難しいというものである。本学の遠隔授業時のクラスコードの通知方法は、教務システム上のシラバスに各講義に対応するクラスコードを記述し、Classroom のクラスに入室を行うようにしたが、シラバスを見ない学生もおり、教務システム上の全ての登録学生が Classroom に登録をすることは、結局対面授業が始まるまで達成されることはなかった。

これら 2 つの課題を解決するために、アシストアプリケーションの開発を行うこととした。そのアプリケーションは、これも Google 社が提供する Google Apps Script³ (GAS) の Web アプリケーションとして作成することとした。GAS とは、Google が提供するクラウド環境で動作するため、サーバ機器など開発環境の整備を必要とせず、Web アプリケーションを開発することが可能である。Web アプリケーションとして開発すれば、URL を通知するだけで利用可能となるため、インストール作業等を必要としない。

特にユーザ登録・認証・管理を全て Google のサービスに依存することが可能であるので、ユーザ管理機能の実装が全く必要なく、そのこともより高速な開発を可能としている。

1. Google Classroom の問題点

Google Classroom は Google 社から無償で提供されている学習管理システムである。世界中で幅広く利用されており、私も担当するほとんどの授業で利用している。Classroom は直感的な使い方をすることができ、スマートフォンなどのモバイルデバイスからでも利用が可能であり、学生にとっても使いやすいサービスであると考えている。その利用には、Google アカウントを有する教員が Classroom のサービスにアクセスし、自分のクラスを作成する。作成したクラスにはそのクラス固有のクラスコードが付与され、それを受講希望の学生に通知し、その学生が Classroom サービス上でクラスコードを入力することにより、クラスへの入室が可能となる。今のところ、教員が強制的に学生をクラスに参加させることは不可能であり、学生のアクションを必ず必要とする。クラスへの参加方法は複数あるが、クラスコードを入力する方法が最もシンプルであると考えているため、この方法を採用している。

クラスに参加した学生と教員は、情報の共有、意見交換などが簡単に行うことができる。

また、Classroom 上で課題を作成し、それを受講生に提示し、受講生からその課題に対する提出物等を受け取ることも簡単に行うことができる。

しかし、世界中で使われることを前提としているため、日本独自の習慣に対応していない。その最も大きなものは、日本では、姓名の 50 音順に学生それぞれに番号を割り振り、それを ID (学籍番号) として利用するということが慣習化しているが、Classroom の場合には、漢字の Unicode のコードの順番でしか受講生のリストのソーティングを行うことができない。そのため、教務システムなどで配布される受講生のリストと Classroom でのリストが異なり、学生を探すことが苦勞することが多々あった。特に課題を提出した学生をチェックするときには大いに苦勞した。

また、今年度新型コロナウイルスの感染拡大に伴い、本学でも遠隔授業を実施したが、その際には新たな問題が発生した。それは、教務システムに登録している受講生と Classroom 登録している受講生が一致せず、だれが Classroom に未登録であるのかを探すことが困難であるというものである。以前は授業中に学生にクラスコードを通知し、クラスに入室を促していたが、遠隔授業のため、その方法はできなかった。そこで、教務システム上のシラバスにクラスコードを記載し、クラスへの入室を促したが、未入室の学生がいなくなることはなかった。この問題は、Classroom 上での生徒のリストを 50 音順に並び替えることで大幅に改善したが、未登録の学生に Classroom のクラスコードを通知することが多いに手間となった。学生がクラスに参加する方法は、招待状を送付する方法もあるが、学生にクラスコードを通知し、Classroom 上でクラスコードを入力する方法が最も簡単であると考え、メールにて自動的に Classroom に未登録の学生にクラスコードを伝達することを問題解決のゴールとした。

2. 提案する Web アプリケーション

提案するアプリケーションは、Google App Script (GAS) を用いて Web アプリケーションとして開発を行った。この章では、最初、Google Apps Script の簡単な説明とその後で提案するアプリケーションの使い方を中心に論じる。そのため、詳しい処理内容については割愛している。

2.1. Google Apps Script

アプリケーションの開発には、GAS を利用した。また、開発環境は GAS の CLI ツールである Clasp⁴を用いた。ここでは、GAS の簡単な説明を中心に行い、Clasp の説明は省略する。

GAS は Google 社が提供するサービスであり、Google 社の様々なサービスを独自に拡張することが可能である。マイクロソフトの Office に付随する VBA とよく似ているが、VBA がローカル環境でのみ動作するのに対し、GAS はクラウド上で動作することを基本として開発されており、Web サーバなどのハードウェアを準備することなく、簡単に Web アプリ

ケーションを作成することが可能である。Web アプリケーションであるため、利用者にはその URL を通知するだけで利用が可能である。すなわち、GAS は開発だけではなく、Web アプリケーションの公開までサービスとして行うことが可能である。

開発したアプリケーションのファイル構成は図 1 のようになる。ここで、server.gs は GAS の Web API を利用するとき、その処理を記述するものである。index.html は、通常の Web アプリケーションと同様にアプリケーションのスタートページである。css.html はページの体裁を整えるスタイルシートを記述するファイルである。通常の Web アプリケーションでは、CSS ファイルの拡張子は css であるが、GAS においては拡張子は gs か html のみなので、このような記述となる。ファイルの内容自体は通常の CSS ファイルと文法的にはほぼ同じである。js.html はスクリプトを記述するファイルであるが、上述と同様に拡張子が通常とは異なる。なお、Javascript のフレームワークとして Vue.js⁵ を用いている。GAS の Web アプリケーションは、フレームワークの設定も可能であり、それらの利用により、より高速な開発を行うことができる。それぞれのファイルの細かな記述方法等は省略する。

これらのファイルの役割はローカルでのファイル（拡張子が html）から、serve.gs に記述した処理を呼び出し、その結果をデータとして戻す、いわゆる API サーバとして GAS が機能していることを表している。戻り値のデータに処理を加え、それを Web ページ上で表示する。また、通常の Web アプリケーションでは、ユーザ登録・認証・管理のための処理が必要となり、その処理を作成することは結構な手間となるが、GAS では、ユーザ管理に伴う処理は、すべて Google アカウント上のサービスとして提供されるため、その処理を行う必要が全くない。その点も開発が容易・高速に行うことが可能である理由の 1 つである。



The image shows a code editor interface. On the left, a file explorer displays a project structure with four files: server.gs, css.html, index.html, and js.html. The main editor area shows the content of server.gs, which is a JavaScript file. The code includes comments and several functions: doGet, include, and getData. The doGet function returns a template, the include function returns HTML content, and the getData function returns a list of courses from a Classroom API.

```
1 // Compiled using ts2gas 3.6.2 (1
2 function doGet(request) {
3     var template = 'index';
4     return HtmlService.createTemp
5 }
6 function include(filename) {
7     return HtmlService.createHtml
8         .getContent();
9 }
10 function getData() {
11     var list = Classroom.Courses.
12     var res = [];
13     for (var i = 0; i < list.leng
14         res.push(list[i]);
15     }
16     return res;
17 }
```

図1 Google Apps Script のファイル構成

2.2. 概要

今回開発した Web アプリケーションのスタートページは、画面左上部のコンボボックスをクリックすると Web アプリケーションにアクセスしたユーザ、Classroom における教師が作成したクラス一覧が表示される。その 1 つを選択すると図 2 のようにそのクラスの参加者一覧が表示される。個人情報保護の観点から学籍番号の下 2 桁のみを表示しているが、学籍番号順になっていることが分かる。リスト上部の順番が違って見えるが、これは上級生が混在しているためである。

学籍番号	氏名
1111019	山田 太郎
1111072	田中 花子
1111057	佐藤 健一
1111001	鈴木 一郎
1111002	高橋 美咲
1111003	中村 誠二
1111007	小林 真由
1111008	渡辺 拓也
1111010	松本 直樹
1111011	大野 悠介
1111012	斎藤 光太郎

図2 生徒の一覧

学籍番号	氏名	課題の提出状況
1111019	山田 太郎	未提出
1111072	田中 花子	提出
1111057	佐藤 健一	未提出
1111001	鈴木 一郎	未提出
1111002	高橋 美咲	提出
1111003	中村 誠二	提出
1111007	小林 真由	未提出
1111008	渡辺 拓也	未提出
1111010	松本 直樹	未提出
1111011	大野 悠介	提出
1111012	斎藤 光太郎	提出

図3 「課題の提出状況」のページ

受講生の並び換えは、Classroom の API からクラスへの参加者（生徒）のデータを取得

し、そのデータの中の **Email** アドレスを用いてソーティングを行い、そのソートされたリストを表示している。

次に、図 2 の上部の「課題の提出状況」をクリックしたときのページ遷移の結果を図 3 に示す。

表の中で「提出」とあるのが **Classroom** を通じて課題を出題し、その課題を提出済みの学生である。

最後に、図 2 の「未登録の学生」をクリックしたときのページ遷移を説明する。ボタンをクリックすると、ボタンのラベルが「ファイルを選択」に変更され、そのボタンをもう一度クリックするとポップアップ画面が立ち上がり、ファイル選択の画面へと遷移する。

この時、本学の教務システムから取得した講義の登録者のファイルを選択すると、図 4 の画面になる。ページの下部に **Classroom** に未登録の学生一覧、今回は 1 名のみが掲載されていることが分かる。ここで、「全員にメールを出す」をクリックすると **Gmail** の下書きフォルダに **bcc** 欄に学生のメールアドレスを列挙したものが作成され、メールアドレス横の「メールを出す」をクリックすると、宛先欄にそのメールアドレスが入力された下書きメールが作成される。



名前	ステータス	アクション
山田 太郎	未登録	メールを出す
田中 花子	未登録	メールを出す
佐藤 一郎	未登録	メールを出す
鈴木 美咲	未登録	メールを出す
高橋 健太	未登録	メールを出す
北村 真由	未登録	メールを出す
中村 大地	未登録	メールを出す
小林 悠太	未登録	メールを出す
渡辺 未来	未登録	メールを出す
森田 翔太	未登録	メールを出す
伊藤 莉子	未登録	メールを出す
水野 隼斗	未登録	メールを出す
山本 結衣	未登録	メールを出す
末廣 悠希	未登録	メールを出す
未登録学生	メールアドレス	全員にメールを出す
山田太郎	stg.takamatsu-u.ac.jp	メールを出す

図 4 未登録の学生

GAS の 1 つの特徴として、1 つのサービスから様々なサービスを利用することも可能であり、今回は、**Gmail** のサービスを利用して課題解決を図った。このように **GAS** を使うことにより、**Google** 社が提供するサービス群を組み合わせることでより便利がサービスを開発することも可能である。

3. 考察・問題点

提案するアプリケーションは提示した図からもわかるように **CSS** 等でのページの体裁を

一切行っていない。そのため、見づらい部分も多く改善が必要である。また、教務システム上の受講者リストを自動的に取得することが不可能であるため、いちいち、教務システムから受講者リストをダウンロードし、そのファイルを読み込ませる必要がある。その処理は、パソコン操作に不慣れな教員には難しいと考えられる。そのため、教務システムとの連携は大きな課題であると感じている。

以上のような理由から、すべての教員に Web アプリケーションの URL を公開せず、数人の教員のみ公開している。また、不具合や誤作動が多々あるということも公開しない理由としては大きい。

提案する Web アプリケーションには、いろいろな課題はあるが、GAS を用いて Classroom を拡張する有効性を示すことはできたと考えている。

おわりに

本報告は、Google Classroom の 2 つの課題、受講生リストの 50 音順による並び替えと未登録学生へのメールによるクラスコードの通知の解決を計るための Web アプリケーションの概要を説明したものである。Web アプリケーションは、Google Apps Script を用いてより高速にかつ容易に作成することができたと思っている。現在のところ、多くの問題点があり、だれでも気軽に使えるような代物ではないが、パソコンの操作にある程度自信のあるものであれば、利用できるものと考えている。このことから、Classroom の拡張の有効性が確かめられたと自画自賛している。

今後の課題としては、すべての人が気軽につかるようなユーザインターフェースの改良とバグの修正にあると思っている。特に教務システムとの連携は非常に大きいものである。

参考文献

1. Google Classroom
<https://edu.google.com/intl/ja/products/classroom>
2. Google Classroom を用いた授業管理の利点と問題点
高松大学・高松短期大学研究紀要 第 70 号 2018 年 9 月刊行
3. Google Apps Script
<https://developers.google.com/gsuite/aspects/appsscript?hl=ja>
4. Clasp
<https://github.com/google/clasp>
5. Vue.js
<https://jp.vuejs.org/index.html>