〈実践報告〉

Google Classroom API を用いた一括得点入力の例

木藤　友規*

An example of how to input the grades of course work using Google Classroom API

Tomonori KITO*

Abstract

本稿では，Google Classroom で，課題の得点入力を API（Application Programming Interface）を使って一括入力する手順や注意点について報告する．API を使用して得点を入力することで，煩雑な作業を短時間で行うことができ，手入力によるミスの発生を防ぐことができるなどの利点があったが，システム上の制約により，処理の手数が多かった．例えば，Google Classroom の画面上の操作で作成した課題には API を使って得点入力できないことや，課題ごとに学生の id が異なることは，プログラムでの処理手順を増加させた．API を用いてアプリケーションの機能を拡張することは，Google Workspace を使用した授業の改善や工夫の幅を広げ，教育活動の効率化にもつながるが，Google Classroom での得点の一括入力については，スクリプト言語などのプログラムをあまり使用しない教員にはハードルが高いかもしれない．

Key words: グーグルクラスルーム，アプリケーションプログラミングインタフェース，スクリプト言語

Ⅰ．はじめに

順天堂大学スポーツ健康科学部では，Google 社の Google Workspace の様々な Web アプリケーションを授業で活用するようになった．それらのアプリケーションのうち Google Classroom（以下「Classroom」という）は，本学部の全ての教員が関わる必修授業科目のスポーツ健康科学総論でも使用されており，使用する機会が増えているアプリケーションの一つである．

授業の様々な情報を一元管理できる Classroom は，画面上のボタンやメニューを操作することで，課題提出や成績管理，学生との情報共有などの機能を使用できるが，画面上の単純な操作だけではできないことや労力を要することもある．例えば，筆記試験の得点を Classroom の採点リストに反映するときには，テキストデータなどの読み込み機能がなく，また，名前の五十音順が一般的な学生名簿の順序と Classroom での受講生の表示（漢字を音読みした場合の五十音順に表示される）が一致しないため，受講生が多い授業では煩雑な作業が発生する．自動で得点を反映させることができる Google Forms を使用することで解決できる場合もあるが，Web 試験よりも筆記試験が適切な場合には筆記試験を実施すべきである．

筆者は，Google 社の Application Programming Interface（以下「API」という）と Google Apps Script（以下「GAS」という）を使用することで，

＊ 順天堂大学スポーツ健康科学部
＊ Faculty of Health and Sports Science, Juntendo University
責任著者：木藤友規
E-mail: t.kito@juntendo.ac.jp

Google Spreadsheet（以下「Spreadsheet」という）に入力した成績一覧をClassroomの採点リストへ反映させる一括処理を行った．本稿では，その処理の手順や注意点について報告する．

Ⅱ．実 践 内 容

1.　概要

図１には，作業手順を示す．APIを使ってClassroomに得点を入力するためには，同じくAPIを使って課題を作成し，課題のidなどの情報を取得することが必要となる．また，今回は，GASで制御しやすいSpreadsheetを使用して，取得したidなどの情報と得点の一覧を作成し，Classroomの採点リストに一括入力する処理を行った．

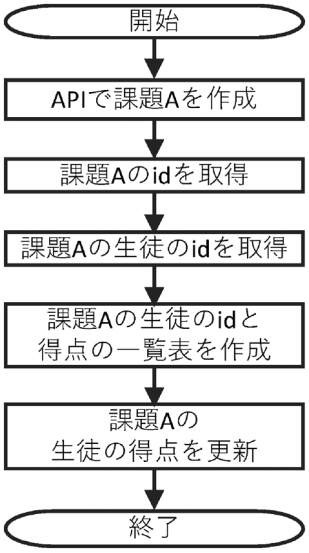

図１　GASとAPIを使って課題を作成してから得点を更新するまでの主な手順
この手順の前に，Classroomにリクエストを行えるようにGASでClassroom APIのサービスを使用できる設定とマニフェストファイルへの認証スコープの追記を完了しなければならない．

2.　作業ごとのコードと注意点

1）　課題の作成

公式リファレンスには，次の記述がある[1)]．

“This request must be made by the Developer Console project of the OAuth client ID used to create the corresponding course work item.

This method returns the following error codes:

PERMISSION_DENIED if the requesting developer project did not create the corresponding course work, if the user is not permitted to make the requested modification to the student submission, or for access errors.”

Classroomの画面の操作で作成した課題では，APIを使用して得点を入力したり，書き換えたりすることができないので，予めAPIを用いてその課題を作成している必要がある（図２）．

また，APIを使用して処理を行う際には，GASの「サービス」設定で「Classroom API」を追加してAPIを使用できる状態にすることに加え，対応する認証スコープをGASのマニフェストファイルに追記しておかなければならない（図３）．必要な認証スコープについては，公式リファレンス（https://developers.google.com/classroom/reference/rest）で確認できる．

```
function createCourseWork() {
  Classroom.Courses.CourseWork.create({
    title: '●●●●の課題',          //課題名
    maxPoints: 10,                 //配点を入力する
    workType: "ASSIGNMENT",
    associatedWithDeveloper: true,
    state: "PUBLISHED"
  },
    123456789123                   //課題を作成するクラスのidに置き換える
  );
}
```

図２　課題を作成するコードの例
この例では，最低限必要な課題名と配点を設定している．

```
{
  "timeZone": "Asia/Tokyo",
  "dependencies": {
    "enabledAdvancedServices": [
      {
        "userSymbol": "Classroom",
        "version": "v1",
        "serviceId": "classroom"
      }
    ]
  },
  "exceptionLogging": "STACKDRIVER",
  "oauthScopes": [
    "https://www.googleapis.com/auth/classroom.coursework.students",
    "https://www.googleapis.com/auth/classroom.coursework.students.readonly",
    "https://www.googleapis.com/auth/classroom.coursework.me.readonly",
    "https://www.googleapis.com/auth/classroom.coursework.me",
    "https://www.googleapis.com/auth/classroom.rosters",
    "https://www.googleapis.com/auth/classroom.rosters.readonly",
    "https://www.googleapis.com/auth/classroom.rosters",
    "https://www.googleapis.com/auth/classroom.profile.emails",
  ],
  "runtimeVersion": "V8",
  "webapp": {
    "executeAs": "USER_DEPLOYING",
    "access": "MYSELF“
  }
}
```

図３　マニフェストファイルのコードの例
GAS の「プロジェクトの設定」で「appsscript.json マニフェストファイルをエディタに表示する」を選択することで記述できる．「oauthScopes」の行以降にある https から始まる URL が追記した認証スコープである．

2）　課題の id などの情報の取得

学生は，Google 社のシステムの中で自身のアカウントに紐付いた id（例えば「123456789012345678901」という21桁の数値で，図４に示すコードで取得した）を割り当てられているが，これとは別に Classroom では，課題ごとに学生固有の id が割り当てられる（同一の学生でも課題１では「Cg4I36bt4qoDELnsv_7ZDw」，課題２では「Cg4I36bt4qoDEMSjmvvZDw」というように課題ごとに id が変わる）．

得点を入力する処理は，クラスの id（図５），課題の id（図６），そしてその課題での学生の id（図７）という３種類の id の情報が揃うことで行える．順天堂大学では，学生のアカウントに学生番号が含まれているので，id に対応するアカウントの情報も Classroom から併せて取得し，学生番号を識別子とすることで各学生の id と得点を照合した．この id と得点の照合では，予め Spreadsheet に学生番号とその番号に対応する得点の一覧表を作成しておき，GAS を使用して３種類の id 情報の一覧表へ各学生の得点を追記する処理を実行したが，Spreadsheet の VLOOKUP 関数を使って対応する得点を取得するなど，GAS を使用しなくても照合することはできるので，この処理のコードの報告については省略する．

3）　Classroom への得点入力

Classroom への得点入力は，patch メソッドの「updateMask」というパラメータによる更新処理によって行った（図８）．得点を入力するフィールドには，学生へ返却される「assignedGrade」と下書きの「draftGrade」があり，「updateMask」自体は１つ以上のフィールドを指定すれば使用できるが，公式リファレンスでは「draftGrade」なしに「assignedGrade」だけを更新することを避けるように推奨されているため，入力する得点を「assignedGrade」と「draftGrade」の両方に入力して更

```
function getStudentList() {

  //  例はstudentListという名前のシートを指定（実際のシート名に置き換える）
  const ss = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("studentList");

  let courseId = '123456789123';  //実際のクラスのidに置き換える
  let pageToken = "";

  do {
    let lastRow = ss.getRange('A:A').getValues().filter(String).length;
    let students = Classroom.Courses.Students.list(courseId, { "pageToken": pageToken });

    //  生徒一覧を出力
    for (i = 0; i < students['students'].length; i++) {
      let student = students['students'][i];
      let email = student.profile.emailAddress;
      let name = student.profile.name.fullName;
      let studentId = student.profile.id;

      //  スプレッドシートに記録
      let rowNum = i + lastRow + 1;
      ss.getRange(rowNum, 1).setValue(email);
      ss.getRange(rowNum, 2).setValue(name);
      ss.getRange(rowNum, 3).setValue(studentId);

    }

    pageToken = students.nextPageToken;

  } while (pageToken != null);

}
```

図４　クラスに参加している各学生のアカウントに紐付けられた id を取得するコードの例
　著者の担当授業では，学生数が30人以上で，一度に取得できる初期値（リストの１ページあたりのアイテム数）を超えていたので，30人分を取得した後，次のページのリストの情報を取得する処理を繰り返した．Spreadsheet のシート名「studentList」の A 列にアカウント（メールアドレス），B 列に学生の名前，C 列に id を入力する場合の処理の例である．

新処理を行った．

Ⅲ．おわりに

本稿は，日常的な教育活動で生じた課題の解決のために「こんな方法もある」という活用事例を蓄積するための報告の一つである．今日では，インターネット上に膨大な情報があるが，Classroom の API については，必要な情報を得ることに苦労し，公式リファレンスの情報をもとに動作を確認してはじめて理解できることもあったため，本稿で報告することにした．

実際に API を使用して得点を入力することで，煩雑な作業を短時間で行うことができ，手入力によるミスの発生を防ぐことができるなどの利点があったが，システム上の制約により，処理の手数が多いと感じた．例えば，Classroom の画面上の操作で作成した課題には API を使って得点入力できないことや，課題ごとに学生の id が異なることは，プログラムでの処理を増加させた．処理の手数が少なく，また GAS のコードが短ければ利用する教員も増えるだろうが，現状の仕様では GAS などの言語をはじめて使う教員にはハードルが高いかもしれない．

Classroom のように操作しやすいアプリケーションを活用することは ICT を使った授業の普及に寄与するが，簡便なシステムでできることには制限が多い．今回報告した得点入力の一括処理のように，API を用いてアプリケーションの機能を拡張することは，Google Workspace を使用した授業の改善や工夫の幅を広げ，教育活動の効率化にもつながる

```
function courseList() {

  let array = [ ];

  for (const course of Classroom.Courses.list().courses) {

    // 停止していない有効なクラスの情報だけを取得する
    if (course.courseState == 'ACTIVE') {
      // 配列に格納
      array.push([
      course.id,
      course.name,
      ])
    }

  }

  console.log(array);

}
```

図５　Classroom にある各クラスの id を取得するコードの例
　この例では，主要なコードだけを示すために配列に格納した情報を console.log で表示する処理を示している（図６も同様）．

```
function getCourseWork() {

  let courseId = '123456789123';  //実際のクラスのidに置き換える
  let works = Classroom.Courses.CourseWork.list(courseId);

  let array = new Array();

  //   課題の名前とidを取得
  for (i = 0; i < works['courseWork'].length; i++) {

    let work = works['courseWork'][i];
    let title = work.title;
    let id = work.id;

    array[i] = [title, id];

  }

  console.log(array);  //画面に表示する

}
```

図６　クラス内の各課題の id を取得するコードの例

であろう．ただし，スキルの修得に必要な時間は人それぞれである．各教員は，時間コストと効果とのバランスを考慮しながら，使用するシステムに対するリテラシー能力を身に付けるべきである．令和３年８月に学校教育法施行規則が改正され，小学校などの学校には「情報通信技術支援員」という職員を置くことができるようになり[2)]，専門的なスタッフと教員が協力したチームでの教育活動が目指されている[3)]．大学においては教育活動での個々の教員の裁量が大きいとされる[4)]が，ICT を活用し，教育活動の充実につなげるためには，他の学校種と同様に，専門的なスタッフの協力を得たチームとしての授業づくりに努めることが益々必要になるだろう．

利益相反

　本稿に関して，開示すべき利益相反関連事項はない．

```
function listStudentSubmissionId() {

  //  例はstudentListという名前のシートを指定（実際のシート名に置き換える）
  const ss = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("studentList");

  //  C列に入力した学生idの一覧を配列に格納する
  var arrayStudentId = ss.getRange('C:C').getValues().flat();

  var students = Classroom.Courses.CourseWork.StudentSubmissions.list(
    '123456789123',  //実際のクラスのidに置き換える
    '987654321987'   //実際の課題のidに置き換える
  );

  //  当該課題での学生のidを記録する
  for (i = 0; i < students['studentSubmissions'].length; i++) {
    let student = students['studentSubmissions'][i];
    let submissionId = student.id;
    let studentId = student.userId;

    let rowNum = arrayStudentId.indexOf(studentId) + 1;
    ss.getRange(rowNum, 4).setValue(submissionId);

  }
}
```

図７　当該課題での各学生の id を取得するコードの例
図４で示したものと同じシート名「studentList」の D 列に当該課題の学生の id を入力する例である．

```
function updateScores() {

  //  例はstudentListという名前のシートを指定（実際のシート名に置き換える）
  const ss = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("studentList");
  const lastRow = ss.getRange('A:A').getValues().filter(String).length;

  for (i = 1; i < lastRow; i++) {
    var courseId = '123456789123';  //実際のクラスのidに置き換える
    var courseWorkId =  '987654321987' ;  //実際の課題のidに置き換える
    var submissionId = ss.getRange(i, 4).getValues().toString();
    var score = ss.getRange(i, 5).getValues().toString(); //E列に得点を入力した場合の例

    var resource = {
      "assignedGrade": score,
      "draftGrade": score
    };

    Classroom.Courses.CourseWork.StudentSubmissions.patch(
      resource,
      courseId,
      courseWorkId,
      submissionId,
      { "updateMask": "assignedGrade,draftGrade" }
    );

  }
}
```

図８　課題の得点を入力するコードの例
図４と図７で作成したシート名「studentList」の一覧表の E 列に得点が入力されている場合の例である．

文　　献

1） Google LLC（2022）Google Classroom API Reference. https://developers.google.com/classroom/reference/rest/v1/courses.courseWork.studentSubmissions/patch（閲覧日：2022年10月６日）
2） 文部科学省（2021） ３文科初第861号学校教育法施行規則の一部を改正する省令の施行について（通知）.
3） 文部科学省（2020年） 教育の情報化に関する手引（追補版）第８章学校及びその設置者等における教育の情報化に関する推進体制．https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/mext_00117.html（閲覧日：2022年10月６日）
4） 中井俊樹（2019年） 大学教員の教育活動における倫理とは．アルカディア学報，No. 647. https://www.shidaikyo.or.jp/riihe/research/647.html（閲覧日：2022年10月６日）

（令和４年10月７日　受付
令和４年10月31日　採録決定
令和４年11月21日　早期公開）