

 無料電子ブック

学習

# google-apps-script

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#google-  
apps-script

.....	1
<b>1: google-apps-script</b> .....	<b>2</b>
.....	2
Examples.....	2
.....	2
.....	2
/.....	3
.....	4
Google Apps Script.....	4
<b>2: AppsWeb Apps</b> .....	<b>6</b>
.....	6
Examples.....	6
Web.....	6
<b>3: DriveApp</b> .....	<b>12</b>
Examples.....	12
Google.....	12
MimeGoogle.....	12
Google.....	12
Google.....	12
-.....	13
-.....	13
.....	14
.....	14
.....	15
<b>4: DriveApp - getFileById</b> .....	<b>17</b>
.....	17
Examples.....	17
IDGoogle.....	17
<b>5: DriveApp</b> .....	<b>18</b>
.....	18
Examples.....	18

Google.....	18
MimeGoogle.....	18
Google.....	18
Google.....	19
-.....	19
-.....	20
<b>6: DriveApp - .....</b>	<b>21</b>
.....	21
Examples.....	21
.....	21
<b>7: FirebaseAppScript.....</b>	<b>23</b>
.....	23
Examples.....	23
GASFirebaseGoogle SpreadsheetFirebase.....	23
AppScriptFirebase.....	23
Firebase.....	25
firebaseURL.....	26
firebaseURL AppScript.....	27
.....	27
1.....	27
2.....	27
3.....	27
4.....	28
<b>8: GmailApp.....</b>	<b>29</b>
.....	29
Examples.....	29
CSV.....	29
<b>9: Google.....</b>	<b>30</b>
.....	30
Examples.....	30
.....	30
.....	31

<b>10: GoogleGoogle Web App</b>	<b>32</b>
.....	32
.....	32
Examples	32
forms.html	32
code.gs	33
.....	34
<b>11: GoogleMailApp</b>	<b>36</b>
.....	36
Examples	36
MailApp	36
.....	36
.....	37
HTML	39
<b>12: SpreadsheetApp</b>	<b>42</b>
.....	42
Examples	42
getActive -	42
<b>13: Google -</b>	<b>43</b>
.....	43
Examples	43
Google	43
<b>14:</b>	<b>44</b>
.....	44
Examples	44
.....	44
1	45
1	45
.....	45
<b>15:</b>	<b>47</b>
.....	47

.....	47
.....	47
Examples.....	47
.....	47
.....	47
.....	<b>49</b>

---

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [google-apps-script](#)

It is an unofficial and free google-apps-script ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official google-apps-script.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

# 1: google-apps-scriptをいめる

Google Apps Scriptのは<http://www.google.com/script/start>にされています。

Google Apps ScriptはJavaScriptクラウドスクリプトで、Googleサービスやサードパーティサービスのタスクをしてウェブアプリケーションをするなをします。

[https://developers.google.com/apps-script/guides/services/#basic\\_javascript\\_features](https://developers.google.com/apps-script/guides/services/#basic_javascript_features)から

Apps ScriptはJavaScript 1.6にえ、1.7と1.8のいくつかのをしています。みみのなGoogleサービスにえて、Array、Date、RegExpなどのオブジェクト、およびMathオブジェクトとObjectグローバルオブジェクトをできます。ただし、AppsスクリプトコードはGoogleのサーバーHTMLサービスページをくクライアントではありません。そのため、DOMやWindow APIなどのブラウザベースのはできません。

## Examples

インストールまたはセットアップ

Google Apps Scriptにはセットアップやインストールはありません。のはGoogleアカウントです。Gmailアカウントは、Google Apps for Work / Education / Governmentアカウントとにします。[accounts.google.com](https://accounts.google.com)にアクセスすると、しいGoogleアカウントをできます

[script.google.com](https://script.google.com)にアクセスして、のスク립トをします。また、Google Apps Scriptには、ドキュメント、ドキュメント、スプレッドシート、フォームなどのくのGoogle Appsのtools -> Script editor...でアクセスすることもできます。Connect more apps..をConnect more apps..と、Google Apps ScriptをGoogleドライブにすることもできます。

のドキュメントは[developers.google.com/apps-script/](https://developers.google.com/apps-script/)にあります。

app-scriptsをするには、code.gsファイルがです。code.gsファイルには、doGetスタンドアロンスクリプトまたはonOpenaddonスクリプトというのがまれているがあります。マニュアルのクイックスタートにはがあります。

app-scriptでapiがになっているは、developers-consoleでもapiをにするがあります。ただし、デベロッパーコンソールにはオンにすることはできませんが、アプリスクリプトインターフェイスにはされないAPIがまれています。たとえば、デベロッパーコンソールでMarketplace SDKをにするがあります。その、Google PlayストアまたはGスイートのドメインにアプリをすることができます。

//けGoogleアプリの、ドメインコンソールには、アプリスクリプトのをまたはするがあります。

スクリプトの

Google Appスクリプトには3あります。

- スタンドアロン
- Google Appsにバインド
- Web Apps

スタンドアロンスクリプト

スタンドアロンスクリプトは、ドキュメント、スプレッドシート、フォームなどの Google アプリにはバインドされていません。スタンドアロンスクリプトは、 [script.google.com](https://script.google.com) にアクセスするか Google アプリスクリプトを Google ドライブにすることができます。スタンドアロンスクリプトは、Google アプリをにプログラムするためにすることも、ウェブアプリケーションとしてすることも、インストールなトリガーからにできるようにすることもできます。スタンドアロンスクリプトの [ドキュメント](#) をしてください。

**Google Apps**にバインド

コンテナバウンドスクリプトともばれる Google Apps にバインドされたスクリプト。スタンドアロンスクリプトとはなり、Google アプリ、つまり Google ドキュメントや Google スプレッドなどにバインドされています。コンテナにバインドされたスクリプトは、Google Apps から `tools> Script editor` してできます。ダイアログ、プロンプト、メニュー、サイドバーなどののは、コンテナにバインドされたスクリプトによってのみされます。さらに、コンテナにバインドされたスクリプトをして [Google アドオン](#) をします。コンテナにバインドされたスクリプトの [ドキュメント](#) をしてください。

**Web Apps**

Google App Script は、ブラウザからアクセスできる Web アプリケーションとしてできます。Web アプリケーションはブラウザのユーザーインターフェイスをし、Google アプリケーションドキュメント、シートなどをすることができます。Google Apps にバインドされたスタンドアロンスクリプトとスクリプトのを Web アプリケーションにできます。スクリプトが Web アプリケーションとしてするには、スクリプトがの2つのをたしているがあります。

- `doGet ()` または `doPost ()` がまれています。
- このは、HTML サービスの `HtmlOutput` オブジェクトまたは Content サービスの `TextOutput` オブジェクトをします。

`InShort`、`doGet ()` および `doPost ()` は、それぞれ `http` の `get` および `post` リクエストハンドラのようにします。

Web アプリケーションのについては、[ドキュメント](#) をしてください。

スクリプトの [デバッグ](#)

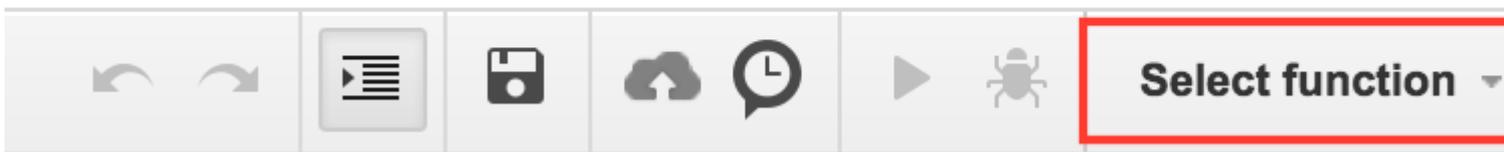
にすように、ツールバーからコードをしてみてください。

File Edit View Run Publish Resources Help



あなたのコードで、のがあるは、それをするに、するをするがあります。えは

File Edit View Run Publish Resources Help



または、キーボードから**Ctrl + r**キーをしてコードをすることもできます。しないとコードをしてからします。しかし、それがするには、ののようになをしておくがあります。

また、スクリプトがアクティビティによってびされたでも、コードのにかをするは、view-> logs をクリックしてログをることができます。

こんにちは

々はメッセージボックスとしてハローとうつもりです。

```
function helloWorld()
{
  Browser.msgBox("Hello World");
}
```

スクリプトをするには、▶をクリックするか、メニュー**Run- > helloWorld**をします

## Google Apps Scriptの

Google Apps Scriptは、にGoogle AppsのとなされるJavaScriptベースのプラットフォームです。Appsスクリプトは、サーバーのプロビジョニングやをとせずGoogleのインフラストラクチャでのみされます。オンラインIDEは、Apps Scriptでなすべてのサービスをするプラットフォームのインターフェイスとしてします。ユーザはOAuth2でプラットフォームにみまれ、スクリプトはコードやをとしません。

Appsスクリプトはサーバーでされますが、HTML、CSS、JavaScript、またはそのブラウザでされたユーザーインターフェイスをつことができます。イベントドリブンであるNodejとはなり、App Scriptsはスレッドモデルでされます。スクリプトへのすべてのびしは、のすべてのインスタンスとはしてされるそのスクリプトののインスタンスをします。スクリプトのインスタンスが

をすると、そのインスタンスはされます。

Appsスクリプトのがブロックされているため、コールバックとのプログラミングパターンがです。ロックは、ファイルIOなどのコードのクリティカルセクションになるインスタンスによってにされないようにするためにされます。

にAppsスクリプトをくのはです。は、テンプレートスプレッドシートからしいスプレッドシートをするなスクリプトです。

```
// Create a new spreadsheet from a template
function createSpreadsheet() {
  var templateFileId = '1Azcz9GwCeHjG19TXf4aUh6g20Eqmgd1UMSdNVjzIZPk';
  var sheetName = 'Account Log for:' + new Date();
  SpreadsheetApp.openById(templateFileId).copy(sheetName);
}
```

オンラインでgoogle-apps-scriptをいめるをむ <https://riptutorial.com/ja/google-apps-script/topic/1154/google-apps-scriptをいめる>

## 2: Apps スクリプト Web Apps

これはフォームWebアプリケーションです。クライアントのビットは、フォームのようになったボタンや、したのエラーメッセージなどのなUXデザインをします。

Apps Scriptのビットはになものです。htmlをし、フィールドをするのになコードだけがまれています。

はこのサンプルアプリケーションへのリンクです [Example Apps Script Form](#)

Googleアカウントにログインしているがあります。

Apps Scriptのファイルはのとおりです。

- Code.gs
- index.html
- Stylesheet.html
- JavaScript.html

## Examples

### Web アプリケーションフォーム

#### Apps スクリプト

```
//Triggered when the page is navigated to, serves up HTML
function doGet(){
  var template = HtmlService.createTemplateFromFile('index');
  return template.evaluate()
    .setTitle('Example App')
    .setSandboxMode(HtmlService.SandboxMode.IFRAME);
}

//Called from the client with form data, basic validation for blank values
function formSubmit(formData){
  for(var field in formData){
    if(formData[field] == ''){
      return {success: false, message: field + ' Cannot be blank'}
    }
  }
  return {success: true, message: 'Sucessfully submitted!'};
}
```

#### HTML

```
<!DOCTYPE html>
<html>

  <head>
    <base target="_top">
```

```

    <link href="https://ssl.gstatic.com/docs/script/css/add-ons1.css" rel="stylesheet">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js"
type="text/javascript"></script>
</head>

<body>
  <div id="mainForm">
    <h1>Example Form</h1>
    <form>
      <div>
        <div class="inline form-group">
          <label for="name">Name</label>
          <input id="nameInput" style="width: 150px;" type="text">
        </div>
      </div>
      <div>
        <div class="inline form-group">
          <label for="city">City</label>
          <input id="cityInput" style="width: 150px;" type="text">
        </div>
        <div class="inline form-group">
          <label for="state">State</label>
          <input id="stateInput" style="width: 40px;" type="text">
        </div>
        <div class="inline form-group">
          <label for="zip-code">Zip code</label>
          <input id="zip-codeInput" style="width: 65px;" type="number">
        </div>
      </div>
      <div class="block form-group">
        <label for="typeSelect">Type</label>
        <select id="typeSelect">
          <option value="">
            </option>
          <option value="Type 1 ">
            Type 1
          </option>
          <option value="Type 2 ">
            Type 2
          </option>
          <option value="Type 3 ">
            Type 3
          </option>
          <option value="Type 4 ">
            Type 4
          </option>
        </select>
      </div>
      <button class="action" id="submitButton" type="button">Submit</button>
      <button class="clear" id="clearFormButton" type="button">Clear Form</button>
    </form>
    <div class="hidden error message">
      <div class="title">Error:</div>
      <div class="message"></div>
    </div>
    <div class="hidden success message">
      <div class="title">Message:</div>
      <div class="message">Sucessfully submitted</div>
    </div>
  </div>
  <?!= HtmlService.createHtmlOutputFromFile('JavaScript').getContent(); ?>

```

```
        <?!= HtmlService.createHtmlOutputFromFile('Stylesheet').getContent(); ?>
    </body>

</html>
```

## CSS

```
<style>
.hidden {
    display: none;
}

.form-group {
    margin: 2px 0px;
}

#submitButton {
    margin: 4px 0px;
}

body {
    margin-left: 50px;
}

.message {
    padding: 2px;
    width: 50%;
}

.message > * {
    display: inline-block;
}

.message .title {
    font-weight: 700;
    font-size: 1.1em;
}

.success.message {
    border: 1px solid #5c9a18;
    background: #e4ffe4;
    color: #2a8e2a;
}

.error.message {
    background: #f9cece;
    border: 1px solid #7d2929;
}

.error.message .title {
    color: #863030;
}

button.clear {
    background: -moz-linear-gradient(top, #dd6e39, #d17636);
    background: -ms-linear-gradient(top, #dd6e39, #d17636);
    background: -o-linear-gradient(top, #dd6e39, #d17636);
    background: -webkit-linear-gradient(top, #dd6e39, #d17636);
    background: linear-gradient(top, #dd6e39, #d17636);
    border: 1px solid transparent;
```

```

    color: #fff;
    text-shadow: 0 1px rgba(0, 0, 0, .1);
}

button.clear:hover {
    background: -moz-linear-gradient(top, #ca602e, #bd6527);
    background: -ms-linear-gradient(top, #ca602e, #bd6527);
    background: -o-linear-gradient(top, #ca602e, #bd6527);
    background: -webkit-linear-gradient(top, #ca602e, #bd6527);
    background: linear-gradient(top, #ca602e, #bd6527);
    border: 1px solid transparent;
    color: #fff;
    text-shadow: 0 1px rgba(0, 0, 0, .1);
}
</style>

```

## JavaScript

```

<script>
var inputs = [
    'nameInput',
    'cityInput',
    'stateInput',
    'zip-codeInput',
    'typeSelect'
];

$(function(){
    var pageApp = new formApp();
    $('#submitButton').on('click', pageApp.submitForm);
    $('#clearFormButton').on('click', pageApp.clearForm);
});

var formApp = function(){
    var self = this;

    //Clears form input fields, removes message, enables submit
    self.clearForm = function(){
        for(var i = 0; i < inputs.length; i++){
            $('#'+inputs[i]).val('');
        }
        toggleSubmitButton(false);
        setErrorMessage(false);
        setSuccessMessage(false);
    }

    //Submits the form to apps script
    self.submitForm = function(){
        toggleSubmitButton(true);
        setSuccessMessage(false);
        setErrorMessage(false);

        google.script.run
            .withSuccessHandler(self.sucessfullySubmitted)
            .withFailureHandler(self.failedToSubmit)
            .formSubmit(self.getFormData());
    };

    //Retrieves the form data absed on the input fields
    self.getFormData = function(){

```

```

var output = {};
for(var i = 0; i < inputs.length; i++){
    output[inputs[i]] = $('#'+inputs[i]).val();
}
console.log(output)
return output;
}

//When the apps script sucessfully returns
self.successfullySubmitted = function(value){
    if(value.success){
        setSuccessMessage(true, value.message);
    } else {
        setErrorMessage(true, value.message);
        toggleSubmitButton(false);
    }
}

//When the apps script threw an error
self.failedToSubmit = function(value){
    toggleSubmitButton(false);
    setErrorMessage(true, value.message);
}

//Disables/enables the submit button
function toggleSubmitButton(disabled){
    $('#submitButton').prop('disabled', disabled);
}

//Sets the general message box's message and enables or disabled the error box
function setSuccessMessage(show, message){
    if(show){
        $('.success.message').removeClass('hidden');
        $('.success.message .message').text(message);
    } else {
        $('.success.message').addClass('hidden');
        $('.success.message .message').text('');
    }
}

//Sets the error message box's message and enables or disabled the error box
function setErrorMessage(show, message){
    if(show){
        $('.error.message').removeClass('hidden');
        $('.error.message .message').text(message);
    } else {
        $('.error.message').addClass('hidden');
        $('.error.message .message').text('');
    }
}

function getFormData(){
    var output = {};
    for(var i = 0; i < inputs.length; i++){
        output[inputs[i]] = $('#'+inputs[i]).val();
    }
    return output;
}
</script>

```

オンラインでAppsスクリプトWeb Appsをむ <https://riptutorial.com/ja/google-apps-script/topic/4874/appsスクリプトweb-apps>

## 3: DriveApp

### Examples

Googleドライブのルートにしいフォルダをする

```
function createNewFolderInGoogleDrive(folderName) {
  return DriveApp.createFolder(folderName);
}
```

`createNewFolderInGoogleDrive`をして、Googleドライブルートに `Test folder` というのフォルダをします。

```
var newFolder = createNewFolderInGoogleDrive('Test folder');
```

`newFolder`はクラスフォルダタイプがあります

```
// output id of new folder to log
Logger.log(newFolder.getId());
```

のMimeタイプのGoogleドライブにしいファイルをする

```
function createGoogleDriveFileOfMimeType() {
  var content, fileName, newFile; //Declare variable names

  fileName = "Test File " + new Date().toString().slice(0,15); //Create a new file name with date on end
  content = "This is the file Content";

  newFile = DriveApp.createFile(fileName, content, MimeType.JAVASCRIPT); //Create a new file in the root folder
};
```

Googleドライブのルートフォルダにしいテキストファイルをする

```
function createGoogleDriveTextFile() {
  var content, fileName, newFile; //Declare variable names

  fileName = "Test Doc " + new Date().toString().slice(0,15); //Create a new file name with date on end
  content = "This is the file Content";

  newFile = DriveApp.createFile(fileName, content); //Create a new text file in the root folder
};
```

ブロブからGoogleドライブにしいファイルをする

```

function createGoogleDriveFileWithBlob() {
  var blob, character, data, fileName, i, L, max, min, newFile, randomNmbr; //Declare variable names

  fileName = "Test Blob " + new Date().toString().slice(0,15); //Create a new file name with
date on end

  L = 500; //Define how many times to loop
  data = "";
  max = 126;
  min = 55;

  for (i=0; i<L; i+=1) { //Loop to create data
    randomNmbr = Math.floor(Math.random() * (max-min+1) + min); //Create a random number
    //Logger.log('randomNmbr: ' + randomNmbr);
    character = String.fromCharCode(randomNmbr);

    //Logger.log('character: ' + character); //Print the character to the Logs
    data = data + character;
  };

  blob = Utilities.newBlob(data, MIME_TYPE.PLAIN_TEXT, fileName); //Create a blob with random
characters

  newFile = DriveApp.createFile(blob); //Create a new file from a blob

  newFile.setName(fileName); //Set the file name of the new file
};

```

すべてのフォルダをする。フォルダをトークンにれてから、トークンからする

```

function processGoogleDriveFolders() {
  var arrayAllFolderNames, continuationToken, folders, foldersFromToken, thisFolder; //Declare
variable names

  arrayAllFolderNames = []; //Create an empty array and assign it to this variable name

  folders = DriveApp.getFolders(); //Get all folders from Google Drive in this account
  continuationToken = folders.getContinuationToken(); //Get the continuation token

  Utilities.sleep(18000); //Pause the code for 3 seconds

  foldersFromToken = DriveApp.continueFolderIterator(continuationToken); //Get the original
folders stored in the token
  folders = null; //Delete the folders that were stored in the original variable, to prove that
the continuation token is working

  while (foldersFromToken.hasNext()) { //If there is a next folder, then continue looping
    thisFolder = foldersFromToken.next(); //Get the next folder
    arrayAllFolderNames.push(thisFolder.getName()); //Get the name of the next folder
  };

  Logger.log(arrayAllFolderNames); //print the folder names to the Logs
};

```

すべてのファイルをする。トークンにれてからそれらをする

```

function processGoogleDriveFiles() {

```

```

var arrayAllFileNames, continuationToken, files, filesFromToken, fileIterator, thisFile; //Declare
variable names

arrayAllFileNames = []; //Create an empty array and assign it to this variable name

files = DriveApp.getFiles(); //Get all files from Google Drive in this account
continuationToken = files.getContinuationToken(); //Get the continuation token

Utilities.sleep(18000); //Pause the code for 3 seconds

filesFromToken = DriveApp.continueFileIterator(continuationToken); //Get the original files
stored in the token
files = null; //Delete the files that were stored in the original variable, to prove that the
continuation token is working

while (filesFromToken.hasNext()) { //If there is a next file, then continue looping
  thisFile = filesFromToken.next(); //Get the next file
  arrayAllFileNames.push(thisFile.getName()); //Get the name of the next file
};

Logger.log(arrayAllFileNames);
};

```

## ルートドライブにフォルダをする

```

function DriveAppAddFolder(child) { //Adds file to the root drive in Google Drive
  var body, returnedFolder; //Declare variable names

  if (!child) {
    body = "There is no folder";
    MailApp.sendEmail(Session.getEffectiveUser().getEmail(), "", "Error Adding Folder!", body)
    return;
  };

  returnedFolder = DriveApp.addFolder(child); //Add a folder to the root drive

  Logger.log('returnedFolder: ' + returnedFolder); //Print the folder results to the Logs
};

function createNewFolderInGoogleDrive() {
  var folder, newFolderName, timeStamp, dateTimeAsString;

  timeStamp = new Date(); //Create a new date
  dateTimeAsString = timeStamp.toString().slice(0, 15);

  newFolderName = 'Test Folder Name ' + dateTimeAsString; //Create new folder name with
date/time appended to name

  folder = DriveApp.createFolder(newFolderName); //Create a new folder
  DriveAppAddFolder(folder); //Call a function and pass a folder to the function
};

```

## しいテキストファイルをし、それをルートフォルダにする

```

function DriveAppAddFile(child) { //Adds file to the root drive in Google Drive
  var body, returnedFolder; //Declare variable names

```

```

if (!child) {
  body = "There is no file";
  MailApp.sendEmail(Session.getEffectiveUser().getEmail(), "", "Error Adding File!", body)
  return;
};

returnedFolder = DriveApp.addFile(child);

Logger.log('returnedFolder: ' + returnedFolder);
};

function createNewFileInGoogleDrive() {
  var content, file, newFileName, timeStamp, dateTimeAsString;

  timeStamp = new Date();//Create a new date
  dateTimeAsString = timeStamp.toString().slice(0,15);

  content = "This is test file content, created at: " + dateTimeAsString;//Create content for
new file
  newFileName = 'Test File ' + dateTimeAsString;//Create new file name with date/time appended
to name

  file = DriveApp.createFile(newFileName, content);//Create a new file
  DriveAppAddFile(file);//Call a function and pass a file to the function
};

```

## ドライブフォルダのすべてのファイルをする

```

function onOpen() {

  // Add a custom menu to run the script
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var searchMenuEntries = [ {name: "Run", functionName: "search"}];
  ss.addMenu("Get Files", searchMenuEntries);
}

function getFiles() {

  // Get the active spreadsheet and the active sheet
  var ss = SpreadsheetApp.getActiveSpreadsheet();
  var ssid = ss.getId();

  // Look in the same folder the sheet exists in. For example, if this template is in
  // My Drive, it will return all of the files in My Drive.
  var ssparents = DriveApp.getFileById(ssid).getParents();
  var sheet = ss.getActiveSheet();

  // Set up the spreadsheet to display the results
  var headers = [ ["Last Updated", "File Owner", "File Name", "File URL"]];
  sheet.getRange("A1:D").clear();
  sheet.getRange("A1:D1").setValues(headers);

  // Loop through all the files and add the values to the spreadsheet.
  var folder = ssparents.next();
  var files = folder.getFiles();
  var i=1;
  while(files.hasNext()) {

```

```
var file = files.next();
if(ss.getId() == file.getId()){
    continue;
}
sheet.getRange(i+1, 1, 1,
4).setValues([[file.getLastUpdated(), file.getOwner().getName(), file.getName(),
file.getUrl()]]);
    i++;
}
}
```

オンラインでDriveAppをむ <https://riptutorial.com/ja/google-apps-script/topic/5363/driveapp>

---

## 4: DriveApp - getFileById

また、ファイルのURLでファイルをすることもできます。ファイルのIDはURLにあるため、URLではなくIDをすると、パラメータがなくなります。IDではなくURLをすると、よりくのスペースになります。

### Examples

ファイルIDをしてGoogleドライブからファイルをする

```
function getGoogleDriveFileById(id) {
  var file;

  file = DriveApp.getFileById(id); //Returns a file - The "id" must be a string

  //One way to manually get a file ID
  // - Open the file from Google Drive
  // - The file ID is in the URL in the browsers address bar
  //https://docs.google.com/spreadsheets/d/File_ID_is_here/edit#gid=0
};
```

オンラインでDriveApp - getFileByIdをむ <https://riptutorial.com/ja/google-apps-script/topic/6087/driveapp-----getfilebyid-id->

## 5: DriveApp サービス

Google Mimeタイプは、Mimeタイプの3のパラメータにはできません。 Google Mimeタイプをすると、のようなエラーがします。

「DriveApp.createFile」をしてGoogle MIMEタイプをすることはできません。アドバンスドドライブサービスをごください

MimeType.GOOGLE\_APPS\_SCRIPT

MimeType.GOOGLE\_DOCS

MimeType.GOOGLE\_DRAWINGS

MimeType.GOOGLE\_FORMS

MimeType.GOOGLE\_SHEETS

MimeType.GOOGLE\_SLIDES

### Examples

Googleルートドライブにしいフォルダをする

```
function createNewFolderInGoogleDrive() {
  var folderName,newFolder;//Declare variable names

  folderName = "Test Folder " + new Date().toString().slice(0,15);//Create a new folder name
with date on end
  newFolder = DriveApp.createFolder(folderName);//Create a new folder in the root drive
};
```

のMimeタイプのGoogleドライブにしいファイルをする

```
function createGoogleDriveFileOfMimeType() {
  var content,fileName,newFile;//Declare variable names

  fileName = "Test File " + new Date().toString().slice(0,15);//Create a new file name with
date on end
  content = "This is the file Content";

  newFile = DriveApp.createFile(fileName,content,MimeType.JAVASCRIPT);//Create a new file in
the root folder
};
```

Googleルートドライブフォルダにしいテキストファイルをする

```
function createGoogleDriveTextFile() {
```

```

var content,fileName,newFile;//Declare variable names

fileName = "Test Doc " + new Date().toString().slice(0,15);//Create a new file name with
date on end
content = "This is the file Content";

newFile = DriveApp.createFile(fileName,content);//Create a new text file in the root folder
};

```

## ブラウザからGoogleドライブにしいファイルをする

```

function createGoogleDriveFileWithBlob() {
  var blob,character,data,fileName,i,L,max,min,newFile,randomNmbr;//Declare variable names

  fileName = "Test Blob " + new Date().toString().slice(0,15);//Create a new file name with
date on end

  L = 500;//Define how many times to loop
  data = "";
  max = 126;
  min = 55;

  for (i=0;i<L;i+=1) { //Loop to create data
    randomNmbr = Math.floor(Math.random()*(max-min+1)+min);//Create a random number
    //Logger.log('randomNmbr: ' + randomNmbr);
    character = String.fromCharCode(randomNmbr);

    //Logger.log('character: ' + character);//Print the character to the Logs
    data = data + character;
  };

  blob = Utilities.newBlob(data, MimeType.PLAIN_TEXT, fileName);//Create a blob with random
characters

  newFile = DriveApp.createFile(blob);//Create a new file from a blob

  newFile.setName(fileName);//Set the file name of the new file
};

```

## すべてのフォルダをする。フォルダをトークンにれてから、トークンからする

```

function processGoogleDriveFolders() {
  var arrayAllFolderNames,continuationToken,folders,foldersFromToken,thisFolder;//Declare
variable names

  arrayAllFolderNames = [];//Create an empty array and assign it to this variable name

  folders = DriveApp.getFolders();//Get all folders from Google Drive in this account
  continuationToken = folders.getContinuationToken();//Get the continuation token

  Utilities.sleep(18000);//Pause the code for 3 seconds

  foldersFromToken = DriveApp.continueFolderIterator(continuationToken);//Get the original
folders stored in the token
  folders = null;//Delete the folders that were stored in the original variable, to prove that
the continuation token is working

```

```
while (foldersFromToken.hasNext()) { //If there is a next folder, then continue looping
    thisFolder = foldersFromToken.next(); //Get the next folder
    arrayAllFolderNames.push(thisFolder.getName()); //Get the name of the next folder
};

Logger.log(arrayAllFolderNames); //print the folder names to the Logs
};
```

すべてのファイルをする。トークンに入れてからそれらをする

```
function processGoogleDriveFiles() {
    var arrayAllFileNames, continuationToken, files, filesFromToken, fileIterator, thisFile; //Declare variable names

    arrayAllFileNames = []; //Create an empty array and assign it to this variable name

    files = DriveApp.getFiles(); //Get all files from Google Drive in this account
    continuationToken = files.getContinuationToken(); //Get the continuation token

    Utilities.sleep(18000); //Pause the code for 3 seconds

    filesFromToken = DriveApp.continueFileIterator(continuationToken); //Get the original files stored in the token
    files = null; //Delete the files that were stored in the original variable, to prove that the continuation token is working

    while (filesFromToken.hasNext()) { //If there is a next file, then continue looping
        thisFile = filesFromToken.next(); //Get the next file
        arrayAllFileNames.push(thisFile.getName()); //Get the name of the next file
    };

    Logger.log(arrayAllFileNames);
};
```

オンラインでDriveAppサービスをむ <https://riptutorial.com/ja/google-apps-script/topic/6395/driveapp> サービス

## 6: DriveApp サービス - タイプとによるファイル

### パラメーター

パラメータ	のためにします
ファイルにつかる	

### Examples

ファイルにするをむファイルタイプでファイルをする

ファイルに「」というですべてのGoogleフォームをします。

```
function mainSearchFunction(searchStr) {
  var fileInfo, arrayFileIDs, arrayFileNames, arrayOfIndexNumbers,
      allFileIDsWithStringInName, i, searchStr, thisID; //Declare variables

  if (!searchStr) {
    searchStr = "Untitled"; //Assign a string value to the variable
  };

  fileInfo = getFilesOfType(); //Run a function that returns files information
  arrayFileNames = fileInfo[1]; //Get the array of file names
  arrayOfIndexNumbers = searchFileNamesForString(arrayFileNames, searchStr);

  //Logger.log('searchStr: ' + searchStr)
  //Logger.log(arrayOfIndexNumbers)

  allFileIDsWithStringInName = [];
  arrayFileIDs = fileInfo[0];

  for (i=0; i<arrayOfIndexNumbers.length; i+=1) {
    thisID = arrayFileIDs[arrayOfIndexNumbers[i]];
    allFileIDsWithStringInName.push(thisID);
  };

  Logger.log(allFileIDsWithStringInName)
};

function getFilesOfType() {
  var allFormFiles, arrFileName, arrFileID, arrFileUrls, thisFile;

  allFormFiles = DriveApp.getFilesByType(MimeType.GOOGLE_FORMS);
  arrFileName = [];
  arrFileID = [];
  arrFileUrls = [];

  while (allFormFiles.hasNext()) {
    thisFile = allFormFiles.next();
    arrFileName.push(thisFile.getName());
    arrFileID.push(thisFile.getId());
  }
}
```

```
    arrFileUrls.push(thisFile.getUrl());
};

//Logger.log(arrFileName)
return [arrFileID,arrFileName];
};

function searchFileNamesForString(arrayFileNames,searchStr) {
    var arrayIndexNumbers,i,L,thisName;

    arrayIndexNumbers = [];

    L = arrayFileNames.length;

    for (i=0;i<L;i+=1){
        thisName = arrayFileNames[i];
        Logger.log(thisName);
        Logger.log('thisName.indexOf(searchStr): ' + thisName.indexOf(searchStr));

        if (thisName.indexOf(searchStr) !== -1) {
            arrayIndexNumbers.push(i);
        };
    };

    return arrayIndexNumbers;
};
```

オンラインでDriveAppサービス - タイプとによるファイルをもむ <https://riptutorial.com/ja/google-apps-script/topic/4049/driveappサービス---タイプとによるファイル>

## 7: FirebaseとAppScriptはじめに

き

FirebaseとGoogle AppScriptをして、Firebaseデータベースのデータを読み取ります。

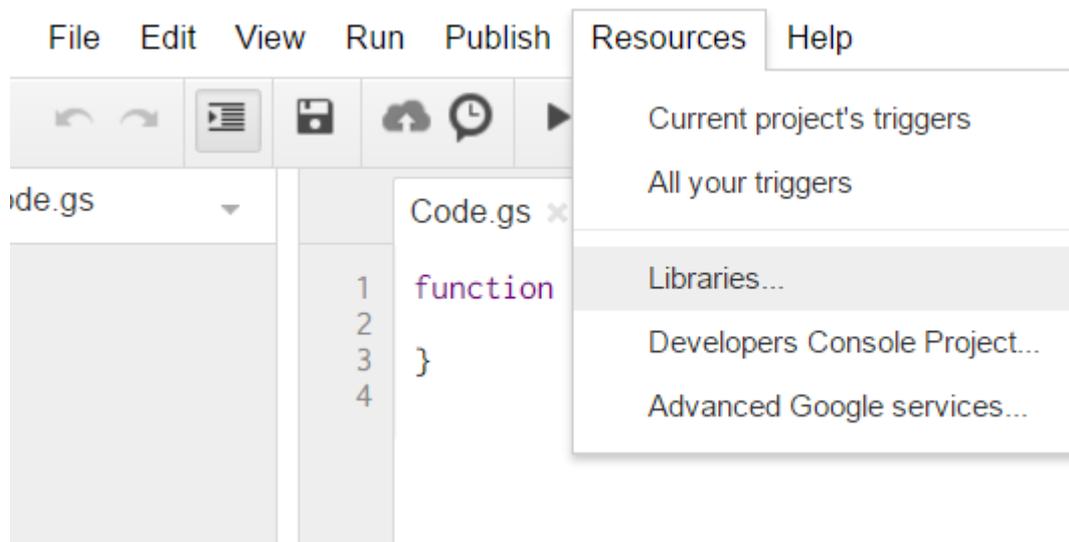
Firebaseは、GoogleのNoSQLデータベースシステムで、リアルタイムデータベースをしてモバイル、デスクトップ、およびタブレットデバイスにアプリケーションをおよびホストするのに使われます。NoSQLデータベースは、JSONオブジェクトをしてデータを保存します。

### Examples

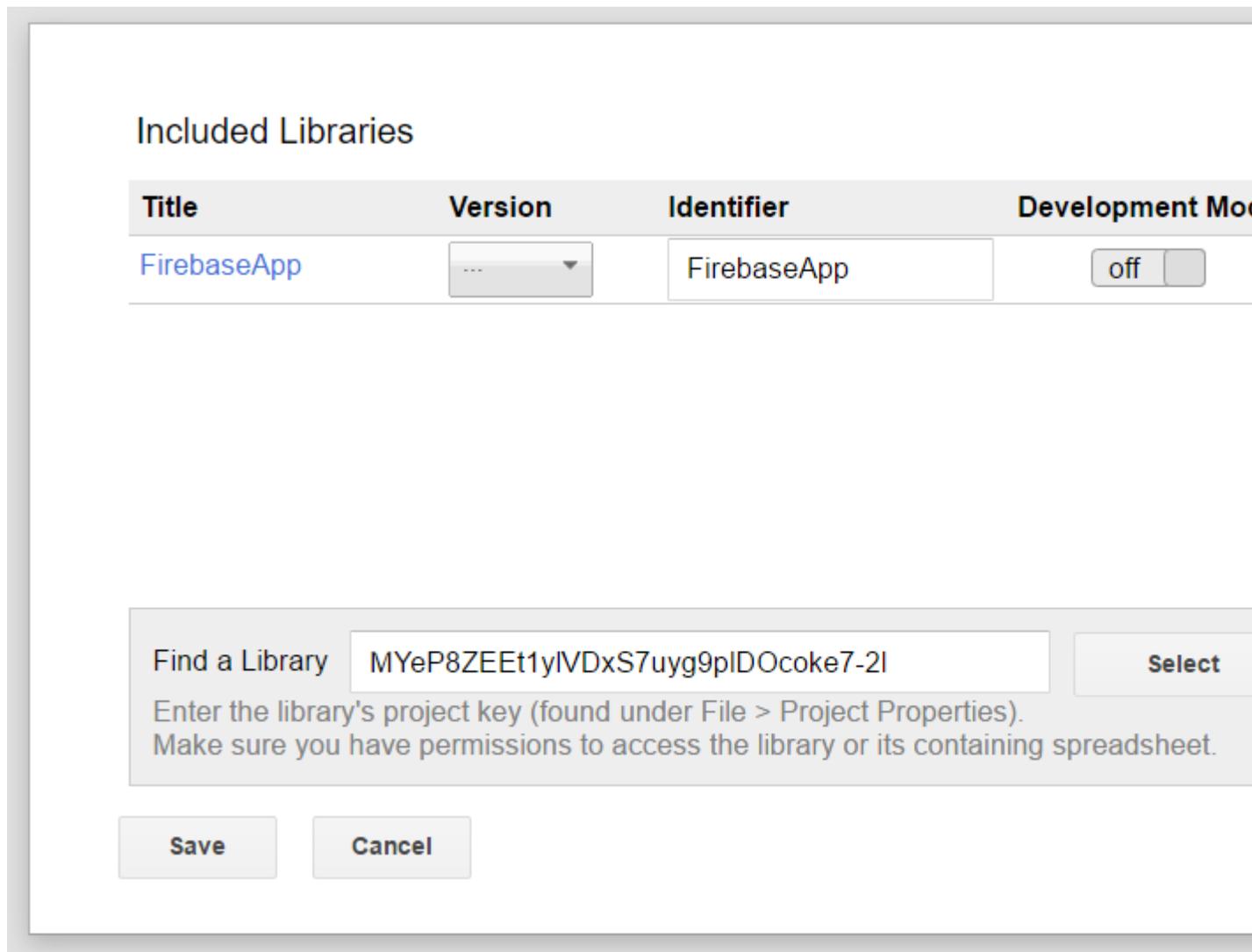
**GAS**の**Firebase**プロジェクトにし、Google Spreadsheetから**Firebase**にデータを送る

### AppScriptに**Firebase**リソースをインストールする

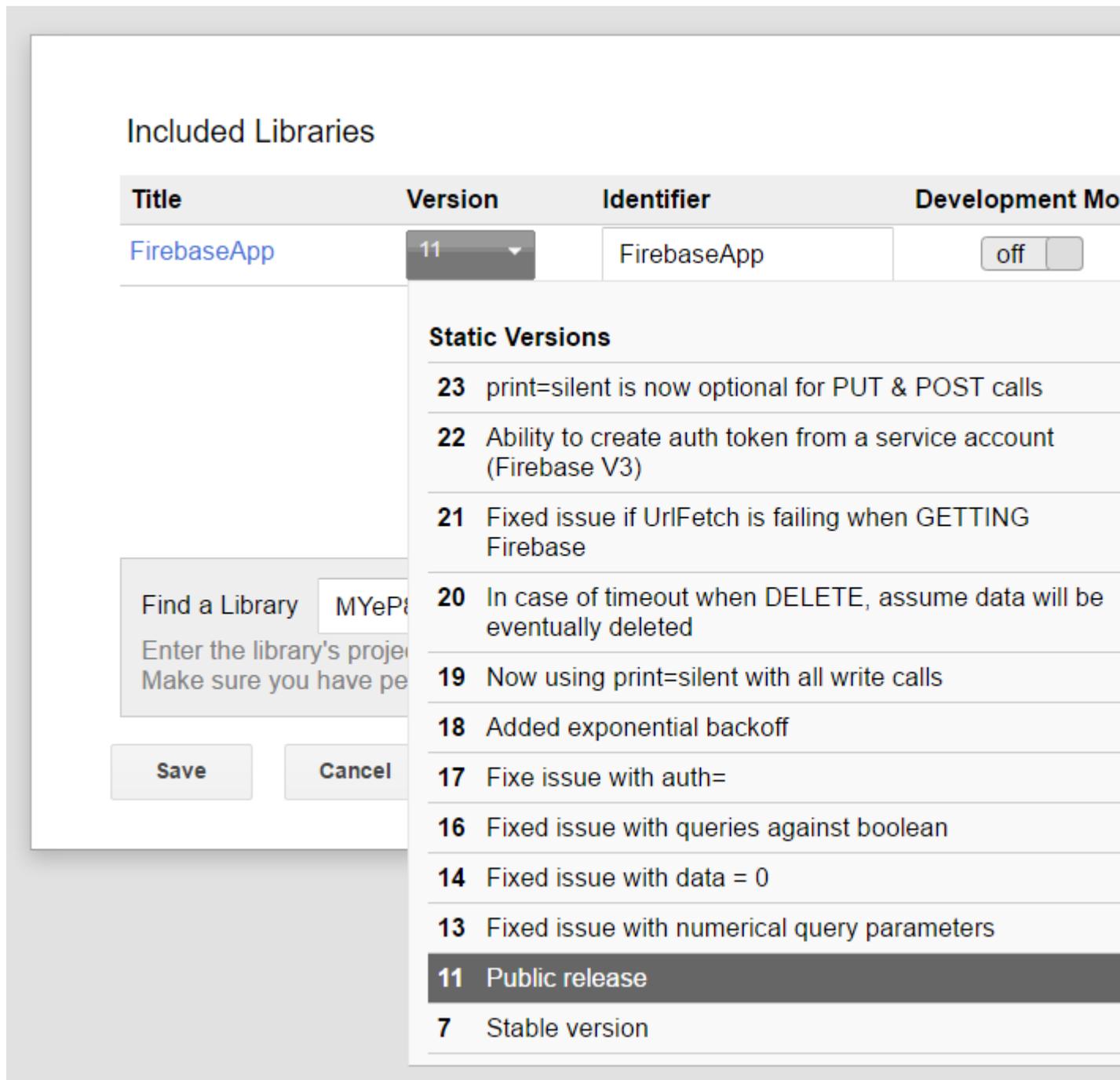
- これを行うには、[リソース]をクリックし、に[ライブラリ]をクリックします。
- Firebaseには、AppScriptにインストールする必要があるプロジェクトライブラリキーがあります。



- ライブラリをクリックすると、このポップアップが表示されます。テキストボックスにプロジェクトキーを入力します。 **MYeP8ZEEt1yIVDxS7uyg9pIDOCoke7-2I**これはFirebaseのプロジェクトライブラリキーです。



- バージョンでは、したをします。



- [ ]をクリックします。これで、FirebaseはあなたのAppScriptにインストールされ、できます。

に、Firebaseからデータをみきするためのをえてみましょう。

- に、Googleスプレッドシートでしたサンプルをします。

A	B	C	D	E	
First Name	Last Name	Email Address	Phone Number	Semester	Departn
Vishal	vishwakarma	vishal.vishwakar	9594852468		7 INFT
Yash	Udasi		75395185246		7 INFT

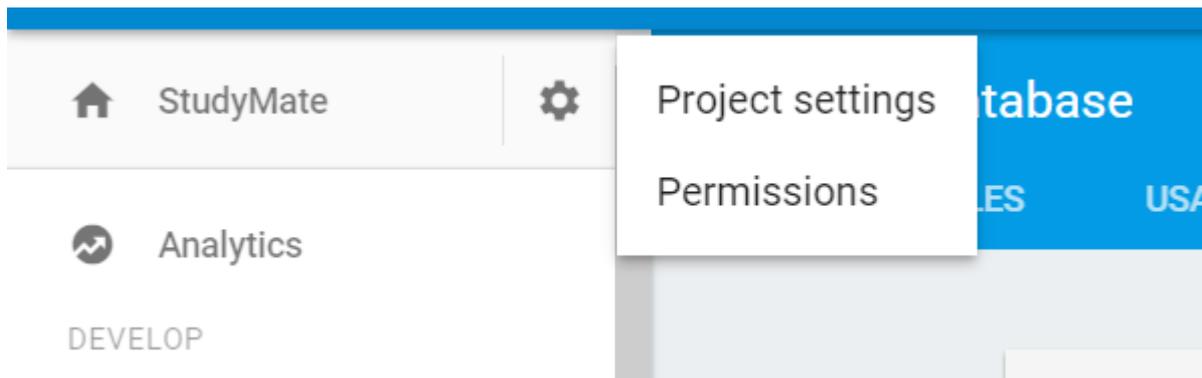
- このテーブルを使ってFirebaseにデータベースをします。AppScriptにのコードをします。

```
function writeToFirebase() {
  var ss = SpreadsheetApp.openById("1LACsj0s3syAa9gvORdRWBhJ_YcXHybjQfHPgw3TLQ6g");
  var sheet = ss.getSheets()[0];
  var data = sheet.getDataRange().getValues();
  var dataToImport = {};
  for(var i = 1; i < data.length; i++) {
    var firstName = data[i][0];
    var lastName = data[i][1];
    dataToImport[firstName + '-' + lastName] = {
      firstName:firstName,
      lastName:lastName,
      emailAddress:data[i][2],
      semester:data[i][4],
      department:data[i][5],
    };
  }
  var firebaseUrl = "https://example-app.firebaseio.com/";
  var secret = "secret-key";
  var base = FirebaseApp.getDatabaseByUrl(firebaseUrl, secret);
  base.setData("", dataToImport);
}
```

スプレッドシートIDとfirebaseURLとをきえます。

## firebaseURL とのは

- あなたのFirebaseダッシュボードにき、のギアをクリックしてください。[プロジェクト]をクリックします。



- 「サービスアカウント」セクションにすると、databaseURLが見つかります。これは

firebaseURLとしてします。

- Database Secretsタブをクリックすると、をつけることができます。

これで**firebaseURL**とがされました。あなたはすべてくつもりです。 **AppScript**エンジンでコードをクリックします。

- あなたがしたときににをするようにめられます。
- [のと]をクリックします。
- これでがされ、**Firestore**データベースでされたテーブルがされます。

データベースをるには**Firestore** ダッシュボードにき、データベースをクリックするとデータベースがされます。

みきをするかいくつかあります。

## 1.がしているかどうかをテストするためのなデータをきむ。

```
function myFunction(){
  var firebaseUrl = "https://example-app.firebaseio.com/";
  var secret = "secret-key";
  var base = FirebaseApp.getDatabaseByUrl(firebaseUrl, secret);
  base.setData("test", "Hello Firestore");
}
```

## 2.すべてのデータをみむには

```
function getAllData() {
  var firebaseUrl = "https://example-app.firebaseio.com/";
  var secret = "secret-key";
  var base = FirebaseApp.getDatabaseByUrl(firebaseUrl, secret);
  var data = base.getData();
  for(var i in data) {
    Logger.log(data[i].firstName + ' ' + data[i].lastName);
  }
}
```

みまれたデータはログにされます。ログをするには、[View]→[Logs]をクリックするか、に **Control + Enter** をします。

## 3.のレコードをみるには

```
function getContact() {
  var firebaseUrl = "https://example-app.firebaseio.com/";
  var secret = "secret-key";
  var base = FirebaseApp.getDatabaseByUrl(firebaseUrl, secret);
  var contact = base.getData("Yash-Udasi");
}
```

```
Logger.log(contact);  
}
```

みまれたデータはログにされます。ログをするには、[View]→[Logs]をクリックするか、に **Control + Enter** をします。

#### 4. のレコードをする。

```
function updateData() {  
  var firebaseUrl = "https://example-app.firebaseio.com/";  
  var secret = "secret-key";  
  var base = FirebaseApp.getDatabaseByUrl(firebaseUrl, secret);  
  base.updateData("Yash-Udasi/emailAddress", "yash.udasi@fyuff.com");  
}
```

オンラインでFirebaseとAppScriptはじめにをむ <https://riptutorial.com/ja/google-apps-script/topic/9417/firebaseとappscript-はじめに>

## 8: GmailApp

なメソッドについては、GmailAppの[APIリファレンス](#)もしてください。

### Examples

メールにされた**CSV**ファイルをする

された**CSV**ファイルのでメールで々のレポートをし、これらのファイルにアクセスしたいというシステムがあるとします。

```
function getCsvFromGmail() {
  // Get the newest Gmail thread based on sender and subject
  var gmailThread = GmailApp.search("from:noreply@example.com subject:\"My daily report\"", 0,
  1)[0];

  // Get the attachments of the latest mail in the thread.
  var attachments = gmailThread.getMessages()[gmailThread.getMessageCount() -
  1].getAttachments();

  // Get and parse the CSV from the first attachment
  var csv = Utilities.parseCsv(attachments[0].getDataAsString());
  return csv;
}
```

オンラインでGmailAppをむ <https://riptutorial.com/ja/google-apps-script/topic/5899/gmailapp>

## 9: Google スプレッドシートのカスタムをする

き

Googleドキュメントのカスタムは、のドキュメントにびけられていますしたがって、そのドキュメントでのみできます。

したがって、そののスク립トでするがあります[ツール]->[スク립トエディタ]。すると、ののスプレッドシートとじようにできます。

### Examples

カスタム

このは、されたでなをします cm /s<sup>2</sup>のは1、 ft /s<sup>2</sup>のは2、 m /s<sup>2</sup>のは3

```
/**
 * Returns the standard gravity constant in the specified acceleration units
 * Values taken from https://en.wikipedia.org/wiki/Standard_gravity on July 24, 2016.
 *
 * @param {number} input 1 for cm/s2, 2 for ft/s2, 3 for m/s2
 *
 * @customfunction
 */
function sg(units_key) {
  var value;
  switch(units_key) {
    case 1:
      value = 980.665;
      break;
    case 2:
      value = 32.1740;
      break;
    case 3:
      value = 9.80665;
      break;
    default:
      throw new Error('Must to specify 1, 2 or 3');
  }
  return value;
}
```

このをするには、スク립トエディタツール ->スク립トエディタ...をしてスプレッドシートにバインドするがあります。がされると、セルのをびすことによって、のGoogleシートとにできます。

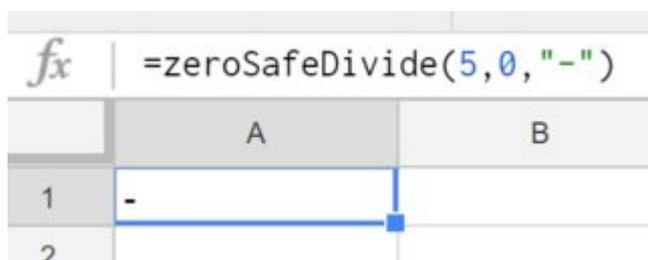
にすると、がオートコンプリートでどのようにされるかにしてください。これは、がJSDocおよびJavadocにていることをするためにされる、ののコメントによるものです。オートコンプリートでをするには、コメントに@customfunctionタグをするがあります。

な

スプレッドシートでしい#DIV/0エラーを避けるために、カスタムをすることができます。

```
/**
 * Divides n by d unless d is zero, in which case, it returns
 * the given symbol.
 *
 * @param {n} number The numerator
 * @param {d} number The divisor
 * @param {symbol} string The symbol to display if `d == 0`
 * @return {number or string} The result of division or the given symbol
 *
 * @customfunction
 */
function zeroSafeDivide(n, d, symbol) {
  if (d == 0)
    return symbol;
  else
    return n / d;
}
```

このをするには、スクリプトエディタ ツール ->スクリプトエディタ... をしてスプレッドシートにバインドするがあります。がされると、セルのをびすことによって、のGoogleシートとにできます。



にすると、がオートコンプリートでどのようにされるかにしてください。これは、がJSDocおよびJavadocにていることをするためにされる、ののコメントによるものです。オートコンプリートでをするには、コメントに@customfunctionタグをするがあります。

オンラインでGoogleスプレッドシートのカスタムをするをむ <https://riptutorial.com/ja/google-apps-script/topic/5572/google-spreadsheet-custom>

# 10: Googleドライブからダウンロードする Google Web Appスクリプト

き

このシンプルなGoogle App Webスクリプトスタンドアロンをすると、ユーザーのローカルPCにファイルをダウンロードするためにGoogleドライブをリポーリングすることができます。1つのアプリケーションスクリプトをして、ののをするをします。1.ユーザーインターフェイスこのではなもの2.ファイルダウンロードページ。どのようにするかについてのなは、「どのようにするか」のをしてください。

Web Scriptは、させるためにするがあります。

ポップアップは<https://script.google.com>でにするがあります

## Examples

### forms.html

```
<!DOCTYPE html>
<html>
  <head>
    <base target="_top">
    <script>

    setInterval(
    function ()
    {
      document.getElementById('messages').innerHTML = 'Event Timer Fetching';
      google.script.run
        .withSuccessHandler(onSuccess)
        .withFailureHandler(onFailure)
        .fetchFromGoogleDrive();
    }, 60000);

    function callFetchGoogleDrive() {
      document.getElementById('messages').innerHTML = 'Fetching';
      google.script.run
        .withSuccessHandler(onSuccess)
        .withFailureHandler(onFailure)
        .fetchFromGoogleDrive();
    }

    function onSuccess(sHref)
    {
      if(new String(sHref).valueOf() == new String("").valueOf())
      {
        document.getElementById('messages').innerHTML = 'Nothing to download';
      }
    }
  }
</script>
</head>
</html>
```

```

else
{
  document.getElementById('messages').innerHTML = 'Success';
  document.getElementById('HiddenClick').href = sHref;
  document.getElementById('HiddenClick').click(); // Must enable Pop Ups for
https://script.google.com
}
}

function onFailure(error)
{
  document.getElementById('messages').innerHTML = error.message;
}

</script>
</head>
<body>
<div id="messages">Waiting to Download!</div>
<div>
  <a id="HiddenClick" href="" target="_blank" onclick="google.script.host.close"
style="visibility: hidden;">Hidden Click</a>
</div>
<div>
  <button type="button" onclick='callFetchGoogleDrive();' id="Fetch">Fetch Now!</button>
</div>
</body>
</html>

```

## code.gs

```

function doGet(e) {
  var serveFile = e.parameter.servefile;
  var id = e.parameter.id;

  if(serveFile)
  {
    return downloadFile(id); // and Hyde
  }

  return HtmlService.createHtmlOutputFromFile('form.html'); // Jekyll
}

function fetchFromGoogleDrive() { // Jekyll
  var fileList = DriveApp.searchFiles("your search criteria goes here + and trashed =
false"); // the 'and trashed = false' prevents the same file being download more than once

  if (fileList.hasNext()) {
    var afile = fileList.next();
    var html = ScriptApp.getService().getUrl()+"?servefile=true&id="+afile.getId();
    return html;
  }
  else
  {
    return '';
  }
}

function downloadFile(id){ // and Hyde
  try

```

```

{
  var afile = DriveApp.getFileById(id);

  var aname = afile.getName();
  var acontent = afile.getAs('text/plain').getDataAsString();

  var output = ContentService.createTextOutput();
  output.setMimeType(ContentService.MimeType.CSV);
  output.setContent(acontent);
  output.downloadAsFile(aname);
  afile.setTrashed(true);
  return output;
}
catch (e) {
  return ContentService.createTextOutput('Nothing To Download')
}
}

```

い

GoogleドライブスタンドアロンWeb Appをして、ドライブからユーザーのローカルPCダウンロードフォルダにファイルをダウンロードポッピングできます。

DriveAppは、ファイルのとダウンロードのためのメカニズムをします。しかし、ダウンロードメカニズムには、Google Appsがしたクライアント/サーバーアーキテクチャのためにいくつかのがあります。Googleのはありません

サーバDriveAppは、クライアントがどこにいるのかというをたず、サーバにファイルをダウンロードすることがであるため、ローカルPCにダウンロードするをしません。

サーバーコードには、クライアントコードにファイルデータまたはファイルへのリンクをするメカニズムがです。どちらのメカニズムもされていますが、からのデータはクライアントのコードによってされることにされています。クライアントは、したデータをローカルPCにするみがありません。したがって、Webページにデータをするためにできます。

2のメカニズムでは、スクリプトのURLそれまたはドライブファイルのURLをすことができます。ドライブファイルのURLは、クライアントブラウザでファイルをダウンロードするためにすることはできません。このURLをアンカーにいてクリックするだけで、いてもにはもしませんおそらくファイルをオンラインでることをいて。

それはスクリプトのURLをします。しかし、スクリプトのURLは、スクリプトではなく、ファイルをします。

ダウンロードをするには、Driveサービスからのファイルを、Googleオンラインガイドにされているように、ContentService createTextOutputをしてサーバーサイドスクリプトのdoGet / doPostからすがあります。しかしこれは、doGet / doPostによってされたによってされたWebページにのUIがしないことをします。

これはたちになをします。ページをダウンロードするユーザUIのないのWebページがじて、のダウンロードがなときにでくがあります。

らかにのホスティングWebページは、このをするためにUIとWeb Appダウンロードスクリプトへのリンクをすることができます。

このスクリプトでは、このをするためにDr JekyllとHydeのアプローチをしています。

スクリプトがGETdoGetへのパラメータなしでオープンされているは、デフォルトでフォームがされます。これは、されたアプリがユーザーによってにかれたときのになります。このでされるフォームはにです。

パラメータがservefile = trueのでスクリプトをくと、スクリプトはドライブファイルのダウンロードとしてします。

クライアントサイドのjavascriptには、にサーバーサイドスクリプトをびしてダウンロードするのファイルのをチェックするポーリングメカニズムイベントタイマーsetIntervalがまれています。

にするドライブファイルがつかったにサーバーサイドスクリプトがされると、スクリプトのURLにパラメータがされたURLがされます。

```
servefile = trueid = the_id_of_the_google_drive_file
```

\*このなのは、サーバーサイドスクリプトにハードコードされています。にじてクライアントからサーバーににすことができます。

このは、されたwithSuccessHandlerメカニズムをしてとしてクライアントにされます。

クライアントのJavaスクリプトは、このされたでれたアンカーのHREFをし、アンカーをにクリックします。

これにより、app / scriptののびしがされます。アプリケーションのしいびしがすると、doGetはservefileパラメータをし、ユーザーインターフェイスをすわりにファイルをブラウザにします。されるファイルは、のによってにされたされたIDパラメータによってされるファイルになります。

されたIDをつファイルがまだする、それはダウンロードされ、アプリケーションのしいびしがじられ、のびしでこのプロセスがりされます。

ユーザー/テスターがタイマーをつことにな、なインターフェースにボタンがされますが、それはではなく、そうでなければすることができます。

シンプルなフォームはもちろん、にじてよりかなユーザーインターフェイスをするようにすることもできます。ファイルをするなど。

オンラインでGoogleドライブからダウンロードするGoogle Web Appスクリプトをむ  
<https://riptutorial.com/ja/google-apps-script/topic/8212/google> ドライブからダウンロードする  
google-web-appスクリプト

# 11: Googleのシート MailApp

き

このサービスをする、ユーザーはメールのをにしてメールをできます。 GmailAppとはなり、MailAppののはメールをすることです。 MailAppはユーザーのGmailトレイにアクセスできません。

GmailAppをしてされたスクリプトのは、MailAppスクリプトよりもユーザーからのリクエストをトリガーするがくなります。

## Examples

なMailAppの

MailAppはGoogle App Scriptのapiで、メールのにできます

```
function sendEmails() {  
  
    var subject = "A subject for your new app!";  
    var message = "And this is the very first message";  
    var recipientEmail = "abc@example.com";  
  
    MailApp.sendEmail(recipientEmail, subject, message);  
}
```

MailAppクラスは、あなたのGoogleアカウントについてされています

- コンシューマユーザー—Gmailアカウント1あたり100の
- Google Appsの1あたり100の
- GSuite/ Gov / Edu / Business1500/

MailAppのメールクォータをできます

```
function checkQuota() {  
    Logger.log(MailApp.getRemainingDailyQuota());  
}
```

シートからデータにアクセスする

```
function getSheetData() {  
  
    var sheet = SpreadsheetApp.getActiveSheet();  
  
    var startRow = 2; // First row of data to process  
    var numRows = 100; // Number of rows to process  
    var startCol = 1; //First column of data to process
```

```

var numCols = 15;    // Number of columns to process

var dataRange = sheet.getRange(startRow, startCol, numRows, numCols);

// Fetch values for each row in the Range.
var data = dataRange.getValues();

return data;
}

```

また、のようにのをして、シートにするコンテンツからデータをにすることもできます。

```

function getDataSheet() {

    sheet = SpreadsheetApp.getActiveSheet();

    //Get data range based on content
    var dataRange = sheet.getDataRange();

    // Fetch values for each row in the Range.
    var data = dataRange.getValues();

    return data;
}

```

シートデータをしてメールをする

えられた - いしをしたのシートをとっています。

- いしがされるときににメールをするがあります

シートはのようになります

A	B	C
Name	Email Address	Reimberseme amount
Ramesh	ramesh@sample.com	200
Vidhita	vidhita@sample.com	50
Jhanvi	jhanvi@sample.com	30

メールをするはのとおりです。

```

function getDataSheet() {

    sheet = SpreadsheetApp.getActiveSheet();

```

```

startRow = 2; // First row of data to process
numRows = 100; // Number of rows to process
startCol = 1; //First column of data to process
numCols = 15; // Number of columns to process

var dataRange = sheet.getRange(startRow, startCol, numRows, numCols);

// Fetch values for each row in the Range.
var data = dataRange.getValues();

return data;
}

function getMessage(name, amount) {
    return "Hello " + name + ", Your reimbursement for amount " + amount + " is processed
successfully";
}

function sendEmail() {

    var emailSent = "Yes";
    var reimbursed = "Yes";
    var emailCol = 5;

    var data = getDataSheet();

    for (var i = 0; i < data.length; i++) {

        var row = data[i];

        var isReimbursed = row[3];
        var isEmailSent = row[4];
        var name = row[0];
        var amount = row[2];

        if(isReimbursed == reimbursed && isEmailSent != emailSent) {

            var subject = "Reimbursement details";
            var message = getMessage(name, amount);

            var recipientEmail = row[1];

            MailApp.sendEmail(recipientEmail, subject, message);

            //Sheet range starts from index 1 and data range starts from index 0
            sheet.getRange(1 + i, emailCol).setValue(emailSent);
        }
    }
}
}

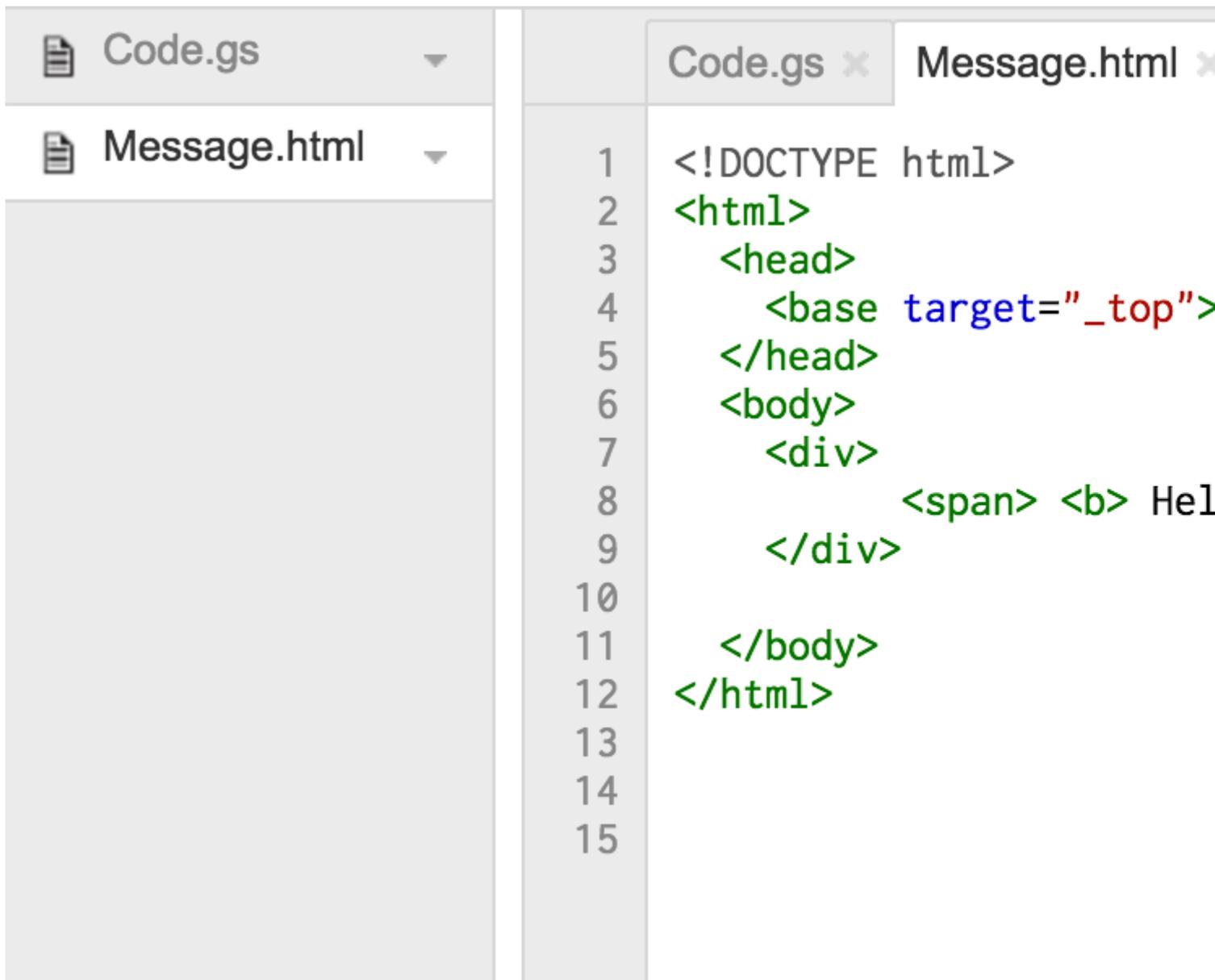
```

A	B	C
Name	Email Address	Reimberseme amount
Ramesh	ramesh@sample.com	20
Vidhita	vidhita@sample.com	5
Jhanvi	jhanvi@sample.com	3

メールで**HTML**コンテンツをする

のでは、HTMLコンテンツをメールのメッセージとしてするは、[ファイル] -> [] -> [HTMLファイル]をして**HTML**ファイルをします

これでgsファイルのHTMLファイルがのようにされます。



```
Code.gs x Message.html x
Code.gs
Message.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <base target="_top">
5   </head>
6   <body>
7     <div>
8       <span> <b> Hel
9     </div>
10
11   </body>
12 </html>
13
14
15
```

に、のから `getMessage` メソッドをのようになります。

```
function getMessage(name, amount) {
  var htmlOutput = HtmlService.createHtmlOutputFromFile('Message'); // Message is the name of
the HTML file

  var message = htmlOutput.getContent();
  message = message.replace("%name", name);
  message = message.replace("%amount", amount);

  return message;
}
```

*MailApp* APIへのびしにもするがあります

```
MailApp.sendEmail(recipientEmail, subject, message, {htmlBody : message});
```

コードはのようになります

```

function getDataSheet() {

    sheet = SpreadsheetApp.getActiveSheet();

    startRow = 2; // First row of data to process
    numRows = 100; // Number of rows to process
    startCol = 1; //First column of data to process
    numCols = 15; // Number of columns to process

    var dataRange = sheet.getRange(startRow, startCol, numRows, numCols);

    // Fetch values for each row in the Range.
    var data = dataRange.getValues();

    return data;
}

function getMessage(name, amount) {
    var htmlOutput = HtmlService.createHtmlOutputFromFile('Message');

    var message = htmlOutput.getContent()
    message = message.replace("%name", name);
    message = message.replace("%amount", amount);

    return message;
}

function sendEmail() {

    var emailSent = "Yes";
    var reimbursed = "Yes";
    var emailCol = 5;

    var data = getDataSheet();

    for (var i = 0; i < data.length; i++) {

        var row = data[i];

        var isReimbursed = row[3];
        var isEmailSent = row[4];
        var name = row[0];
        var amount = row[2];

        if(isReimbursed == reimbursed && isEmailSent != emailSent) {

            var subject = "Reimbursement details";
            var message = getMessage(name, amount);

            var recipientEmail = row[1];

            MailApp.sendEmail(recipientEmail, subject, message, {htmlBody : message});

            sheet.getRange(startRow + i, emailCol).setValue(emailSent);
        }
    }
}

```

オンラインでGoogleのシートMailAppをむ <https://riptutorial.com/ja/google-apps-script/topic/5298/googleのシートmailapp>

---

## 12: SpreadsheetApp アクティブシート

メソッド `getActive`

りの [スプレッドシート](#)

### Examples

**getActive** - アクティブなスプレッドシートをする

アクティブなスプレッドシートをします。スプレッドシートがないは `null` をします。

```
var currentSheet = SpreadsheetApp.getActive();
var url = currentSheet.getUrl();
Logger.log( url );
```

オンラインで [SpreadsheetApp](https://riptutorial.com/ja/google-apps-script/topic/5861/spreadsheetapp) アクティブシートをむ <https://riptutorial.com/ja/google-apps-script/topic/5861/spreadsheetapp> アクティブシート

# 13: クライアントからGoogleアプリへのびし - スクリプト

き

Googleのappscriptは、スタンドアロンのプラットフォームとして、Googleのドキュメント、シート、フォームのアドオンでします。ただし、らかのアクションをするためにクライアントブラウザからGoogleアプリをびすがじることがあります。

そのため、GoogleではクライアントのリクエストをGoogleのアプリスクリプトにしました。このをするために、Googleは[クライアントサイドライブラリ](#)

## Examples

これは、Googleアプリスクリプトへのクライアントサイドコールのです

```
<script src="https://apis.google.com/js/api.js"></script>
<script>
function start() {
  // 2. Initialize the JavaScript client library.
  gapi.client.init({
    'apiKey': 'YOUR_API_KEY',
    // clientId and scope are optional if auth is not required.
    'clientId': 'YOUR_WEB_CLIENT_ID.apps.googleusercontent.com',
    'scope': 'profile',
  }).then(function() {
    // 3. Initialize and make the API request.
    return gapi.client.request({
      'path': 'https://people.googleapis.com/v1/people/me',
    })
  }).then(function(response) {
    console.log(response.result);
  }, function(reason) {
    console.log('Error: ' + reason.result.error.message);
  });
};
// 1. Load the JavaScript client library.
gapi.load('client', start);
</script>
```

オンラインでクライアントからGoogleアプリへのびし - スクリプトをむ

<https://riptutorial.com/ja/google-apps-script/topic/8875/クライアントからgoogleアプリへのびし---スクリプト>

## 14: スプレッドシートサービス

スプレッドシートサービスのAPIリファレンスは<https://developers.google.com/apps-script/reference/spreadsheet/>にあります。

### Examples

シート

きシートタブへのの

```
var spread_sheet = SpreadsheetApp.getActiveSpreadsheet(); // Get active spreadsheet
var sheet_with_name_a = spread_sheet.getSheetByName("sheet_tab_name");
```

アクティブシートタブの

```
var spread_sheet = SpreadsheetApp.getActiveSpreadsheet();
var active_sheet = spread_sheet.getActiveSheet();
```

を

```
var spread_sheet = SpreadsheetApp.getActiveSpreadsheet();
var active_sheet = spread_sheet.getActiveSheet();
active_sheet.insertColumnAfter(1); // This inserts a column after the first column position
active_sheet.insertColumnBefore(1); // This inserts a column in the first column position
active_sheet.insertColumns(1); // Shifts all columns by one
active_sheet.insertColumns(1, 3); // Shifts all columns by three
active_sheet.insertColumnsAfter(1); // This inserts a column in the second column position
active_sheet.insertColumnsBefore(1, 5); // This inserts five columns before the first column
```

を

```
var spread_sheet = SpreadsheetApp.getActiveSpreadsheet();
var active_sheet = spread_sheet.getActiveSheet();
active_sheet.insertRowAfter(1); // This inserts a row after the first row position
active_sheet.insertRowBefore(1); // This inserts a row in the first row position
active_sheet.insertRows(1); // Shifts all rows by one
active_sheet.insertRows(1, 3); // Shifts all rows by three
active_sheet.insertRowsAfter(1); // This inserts a row in the second row position
active_sheet.insertRowsBefore(1, 5); // This inserts five rows before the first row
```

セル

```
var spread_sheet = SpreadsheetApp.getActiveSpreadsheet();
var active_sheet = spread_sheet.getActiveSheet();
var cell = range.getCell(1, 1);
var cell_value = cell.getValue();
cell.setValue(100);
```

## をコピーする

```
var spread_sheet = SpreadsheetApp.getActiveSpreadsheet();
var active_sheet = spread_sheet.getActiveSheet();
var rangeToCopy = active_sheet.getRange(1, 1, sheet.getMaxRows(), 5);
rangeToCopy.copyTo(sheet.getRange(1, 6));
```

```
var spread_sheet = SpreadsheetApp.getActiveSpreadsheet();
var active_sheet = spread_sheet.getActiveSheet();
var range = active_sheet.getRange("B5");
var formula = range.getFormula();
range.setFormula("=SUM(B3:B4)");
```

## 1のシートからのシートにをコピーする

Googleののスプレッドシートがあるとし、のシートのセルD5にB2セルのをあるとします

。

```
function copyValueandPaste()
{
    var source = SpreadsheetApp.openById('spread sheet id is here'); //Separate spreadsheet
    book
    var sourcesheet = source.getSheetByName('Sheet1'); //Sheet tab with source data
    var sourceCellValue = sourcesheet.getRange('B2').getValue(); // get B2 cell value

    var thisBook = SpreadsheetApp.getActive(); // Active spreadsheet book
    var thisSheet = thisBook.getSheetByName('Sheet1'); // Target sheet
    thisSheet.getRange('D5').setValue(sourceCellValue); //Set value to target sheet D5 cell
}
```

あなたのURLからスプレッドシートIDをつけることができます。

## 1つののを

```
function lastRowForColumn(sheet, column){
    // Get the last row with data for the whole sheet.
    var numRows = sheet.getLastRow();

    // Get all data for the given column
    var data = sheet.getRange(1, column, numRows).getValues();

    // Iterate backwards and find first non empty cell
    for(var i = data.length - 1 ; i >= 0 ; i--){
        if (data[i][0] != null && data[i][0] != ""){
            return i + 1;
        }
    }
}
```

## をとして

スプレッドシートののをするのはです

```
var someSheet = SpreadsheetApp.getActiveSpreadsheet().getSheets()[0];
someSheet.appendRow(["Frodo", "Baggins", "Hobbit", "The Shire", 33]);
```

これはのでないのにをすることにしてください。

んにをするのはもうしです

```
var someSheet = SpreadsheetApp.getActiveSpreadsheet().getSheets()[0];

var newRowIndex = 2;
var row = ["Gandalf", "?", "Wizard", "?", 2019];
someSheet.insertRowBefore(newRowIndex);
// getRange(row, col, numRows, numCols)
someSheet.getRange(newRowIndex, 1, 1, row.length).setValues([row]); // Note 2D array!
```

このなコードのくは、ヘルパーにすることができます

```
function insertRowBefore(sheet, rowIndex, rowData) {
  sheet.insertRowBefore(rowIndex);
  sheet.getRange(rowIndex, 1, 1, rowData.length).setValues([rowData]);
}
```

これはたちのをのようにならします

```
var someSheet = SpreadsheetApp.getActiveSpreadsheet().getSheets()[0];
insertRowBefore(someSheet, 2, ["Gandalf", "?", "Wizard", "?", 2019]);
```

オンラインでスプレッドシートサービスをむ <https://riptutorial.com/ja/google-apps-script/topic/2688/スプレッドシートサービス>

# 15: スプレッドシートのメニュー

## 1. addMenu、subMenus

### パラメーター

するメニューの	
サブメニュー	JavaScriptマップの

、スプレッドシートがロードされたときにメニューがにされるように、onOpenからaddMenuをびすがあります。

```
// The onOpen function is executed automatically every time a Spreadsheet is loaded
function onOpen() {
  var activeSheet = SpreadsheetApp.getActiveSpreadsheet();
  var menuItems = [];
  // When the user clicks on "addMenuExample" then "Menu 1", the function Myfunction1 is
  executed.
  menuItems.push({name: "Menu 1", functionName: "Myfunction1"});
  menuItems.push(null); // adding line separator
  menuItems.push({name: "Menu 2", functionName: "Myfunction2"});

  activeSheet.addMenu("addMenuExample", menuEntries);
}
```

## Examples

### しいメニューをする

スプレッドシートUIにしいメニューをします。メニューはユーザーをします。

```
var activeSheet = SpreadsheetApp.getActiveSpreadsheet();
var menuItems = [];
// When the user clicks on "addMenuExample" then "Menu 1", the function Myfunction1 is
executed.
menuItems.push({name: "Menu 1", functionName: "Myfunction1"});
menuItems.push(null); // adding line separator
menuItems.push({name: "Menu 2", functionName: "Myfunction2"});

activeSheet.addMenu("addMenuExample", menuEntries);
```

### カスタムメニューの

/\*

## メソッドカスタムメニューをするこれは、まずApp Loads

---

\*/

```
function onOpen() {
  var ui = SpreadsheetApp.getUi();
  // Or DocumentApp or FormApp.
  ui.createMenu('My HR')
    .addItem('Send Form to All', 'sendIDPForm_All')
    .addItem('Trigger IDP System', 'applyCategory')
    .addToUi();
}
```

オンラインでスプレッドシートのメニューをむ <https://riptutorial.com/ja/google-apps-script/topic/4253/スプレッドシートのメニュー>

## クレジット

S. No		Contributors
1	google-apps-scriptを いめる	<a href="#">Albert Portnoy</a> , <a href="#">Community</a> , <a href="#">Douglas Gaskell</a> , <a href="#">iJay</a> , <a href="#">MShoaib91</a> , <a href="#">Rubén</a> , <a href="#">Saloni Vithalani</a> , <a href="#">Shyam Kansagra</a> , <a href="#">Spencer Easton</a> , <a href="#">sudo bangbang</a> , <a href="#">Supertopoz</a>
2	AppsスクリプトWeb Apps	<a href="#">Douglas Gaskell</a>
3	DriveApp	<a href="#">Brian</a> , <a href="#">Kos</a> , <a href="#">nibarius</a> , <a href="#">Sandy Good</a> , <a href="#">Wolfgang</a>
4	DriveApp - getFileById	<a href="#">Sandy Good</a>
5	DriveAppサービス	<a href="#">Sandy Good</a>
6	DriveAppサービス - タイプとによるファ イル	<a href="#">nibarius</a> , <a href="#">Sandy Good</a>
7	FirebaseとAppScript はじめに	<a href="#">Joseba</a> , <a href="#">Vishal Vishwakarma</a>
8	GmailApp	<a href="#">nibarius</a>
9	Googleスプレッドシ ートのカスタムをす る	<a href="#">Francky_V</a> , <a href="#">Joshua Dawson</a> , <a href="#">Pierre-Marie Richard</a> , <a href="#">Rubén</a>
10	Googleドライブから ダウンロードする Google Web Appス クリプト	<a href="#">Walter</a>
11	Googleのシート MailApp	<a href="#">Bhupendra Piprava</a> , <a href="#">Brian</a> , <a href="#">Jordan Rhea</a> , <a href="#">Kos</a> , <a href="#">nibarius</a> , <a href="#">Saloni Vithalani</a>
12	SpreadsheetAppアク ティブシート	<a href="#">iJay</a>
13	クライアントから Googleアプリへのび し - スクリプト	<a href="#">Supertopoz</a>

14	スプレッドシートサービス	<a href="#">cdrini</a> , <a href="#">iJay</a> , <a href="#">nibarius</a> , <a href="#">Sandy Good</a> , <a href="#">sudo bangbang</a>
15	スプレッドシートのメニュー	<a href="#">Bishal</a> , <a href="#">iJay</a> , <a href="#">nibarius</a>