

Apps Script Projects

Create a Doc with Apps Script	2
DocumentApp Create and update docs	3
Update element attributes	3
Use the id of a doc to open and update	4
Copy Google Doc with Code	4
Create a new document with DriveApp	4
Replace text in a document	5
Use the template to duplicate it and create a new updated file with object data.	5
Spreadsheet Data	6
New Docs with Sheet Data	7
Docs to PDF and Email	8
Create Doc within the destination folder and create PDF version.	8
Create only PDF version no docs copies in destination folder	9
Create Doc as PDF and email to Sheet data email address	9

This section covers how to get started with Google Apps Script and how to create common functionality with just a few lines of code. Create files, select existing files and update them. Use them within your code. How to send out emails and add attachments. Use of blobs to create PDF files, copy and use Docs content in brand new dynamically created Docs. Explore how to get Google Spreadsheet sheet data, and then use the data from the sheet to populate a Google Doc Template, creating new Docs with the populated content dynamically. Take those new files and convert them into PDFs, create PDF files dynamically from Doc files and email them to addresses coming from a Spreadsheet. The upcoming lessons will help you learn more about how to connect the workspace services together to create a fully functional application that can use Sheet data, create files and email files as PDFs.

Explore Google Apps Script - connect the power of Google Services together!

- Create Documents - Select and Update Existing Google Docs
- Apply Styling to Documents
- Copy a Google Doc to new Locations
- Get Data from your Spreadsheet and populate a Google Doc with Sheet data
- Create a Doc as a template
- Convert the Doc to a PDF version
- Using Sheet data and a Doc Template create PDFs and send them to email addresses
- Send emails and dynamically generated attachments

Create a Doc with Apps Script

How to get started with Google Apps Script Create a Doc

Create a script at <https://script.google.com/home>

Press the “New Project” button

This will create a new script file on the root folder in your drive.

The screenshot shows the Google Apps Script editor interface. At the top, there are navigation icons and buttons for 'Run', 'Debug', and 'maker1'. A red arrow points to the 'Run' button, and a purple arrow points to the 'Execution log' button. Below the toolbar, the code editor contains the following JavaScript code:

```
1 function maker1() {  
2   const docName = 'Laurence Svekis';  
3   const doc = DocumentApp.create(docName);  
4   Logger.log(doc.getId());  
5 }
```

Annotations include: '2. Execute the function code' pointing to the code, '1. Select the function to Run' pointing to the 'maker1' dropdown, and 'Logger.log Output into the execution log' pointing to the log output. Below the code editor is the 'Execution log' section, which contains three entries:

Time	Level	Message
1:25:09 PM	Notice	Execution started
1:25:10 PM	Info	1h8UqxMLpBcWdpxs3B87CRXxRPdj7RNJMraIJ6FI5m7o
1:25:11 PM	Notice	Execution completed

Code will create a Google Doc in the root of the drive.

The screenshot shows the Google Drive interface. At the top, there is a 'My Drive' dropdown and several action icons (link, share, view, trash, etc.). Below the toolbar is a table listing the contents of the drive:

Name	Last modified
Private	Nov 21, 2020
test 1	1:20 PM
Laurence Svekis	1:22 PM

```
function maker1() {  
  const docName = 'Laurence Svekis';  
  const doc = DocumentApp.create(docName);
```

```
    Logger.log(doc.getId());  
}
```

DocumentApp Create and update docs

```
function maker2() {  
    const docName = 'Test 2';  
    const doc = DocumentApp.create(docName);  
    const id = doc.getId();  
    const body = doc.getBody();  
    let temp = `ID ${id}`;  
    body.appendParagraph(temp);  
    Logger.log(doc.getId());  
}
```

Update element attributes

```
//{HORIZONTAL_ALIGNMENT=Left, FONT_SIZE=null,  
BACKGROUND_COLOR=null, FOREGROUND_COLOR=null,  
INDENT_FIRST_LINE=0.0, SPACING_BEFORE=0.0, STRIKETHROUGH=null,  
INDENT_START=0.0, LINE_SPACING=1.15, ITALIC=null, BOLD=null,  
INDENT_END=0.0, FONT_FAMILY=null, HEADING=Normal,  
LEFT_TO_RIGHT=true, SPACING_AFTER=0.0, LINK_URL=null,  
UNDERLINE=null}
```

Use the id of a doc to open and update

```
function update1(){
  const id = '1*****U';
  const doc = DocumentApp.openById(id);
  const body = doc.getBody();
  let temp = `ID6 ${id}`;
  const ele = body.appendParagraph(temp);
  Logger.log(ele.getAttributes());
  const style = {
    BACKGROUND_COLOR : '#ff0000',
    FOREGROUND_COLOR : '#ffffff',
    BOLD : true
  };
  ele.setAttributes(style);
}
```

Copy Google Doc with Code

Create a new document with DriveApp

```
function update3(){
  const id = '1Tdp-4Yp30nLbp9H-zbrztt7vdBU8TLt1esuD8T6aJOU';
  const tarDoc = DriveApp.getFileById(id);
```

```
const newDoc = tarDoc.makeCopy('New Doc');
//Logger.log(newDoc.getId());
//newDoc.setTrashed(true);
}
```

Replace text in a document

```
function update4(){
  const id = '1***1';
  const doc = DocumentApp.openById(id);
  const body = doc.getBody();
  //body.clear();
  const user = {first:'Laurence',last:'Svekis',id:100};
  body.replaceText('{first}',user.first);
  body.replaceText('{last}',user.last);
  body.replaceText('{id}',user.id);
}
```

Use the template to duplicate it and create a new updated file with object data.

```
function filltemp1(){
  const id = '1***1';
  const tempDoc = DriveApp.getFileById(id);
  const newDoc = tempDoc.makeCopy('New Doc');
  const doc = DocumentApp.openById(newDoc.getId());
  const body = doc.getBody();
```

```

Logger.log(body);
const user = {first:'Laurence',last:'Svekis',id:100};
body.replaceText('{first}',user.first);
body.replaceText('{last}',user.last);
body.replaceText('{id}',user.id);
}

```

Spreadsheet Data

```

function getSheetDataValues(){
  const id = '1****1';
  const ss = SpreadsheetApp.openById(id);
  const sheet = ss.getSheetByName('data');
  const data = sheet.getDataRange().getValues();
  const rows = data.slice(1);
  rows.forEach((row,index)=>{
    Logger.log(row);
    const temp = sheet.getRange(index+2,5,1,1);
    temp.setValue(false);
    //Logger.log(temp.getValue());
  })
  //Logger.log(rows);
}

```

Table

first	last	id	email	created
Laurence	Svekis	1	gappscourses+1@gmail.com	FALSE

Laurence Svekis <https://basescripts.com/>

Jane	Doe	2 gappscourses+5@gmail.com	FALSE
Jack	Smith	3 gappscourses+6@gmail.com	FALSE

New Docs with Sheet Data

Hello {first} {last}

Welcome to the website! Your id is {id}
Looking forward to seeing you.

Thank you



```
function docMaker1(){
  const fid = '1****1';
  const sid = '1****1';
  const tid = '1****1';
  const tempDoc = DriveApp.getFileById(tid);
  const mainFolder = DriveApp.getFolderById(fid);
  const sheet =
SpreadsheetApp.openById(sid).getSheetByName('data');
  const data = sheet.getDataRange().getValues().slice(1);
  data.forEach((row, index)=>{
    const newDoc = tempDoc.makeCopy(row[0]);
    mainFolder.addFile(newDoc);
  const doc = DocumentApp.openById(newDoc.getId());
    const body = doc.getBody();
```

```

body.replaceText('{first}', row[0]);
body.replaceText('{last}', row[1]);
body.replaceText('{id}', row[2]);
sheet.getRange(index+2, 5, 1, 1).setValue(doc.getUrl());
})
}

```

Docs to PDF and Email

Create Doc within the destination folder and create PDF version.

```

function manyPDFs(){
  const sourceID = '1****1';
  const sourceF = DriveApp.getFolderById(sourceID);
  const destID = '2****2';
  const destF = DriveApp.getFolderById(destID);
  const files = sourceF.GetFiles();
  while(files.hasNext()){
    const file = files.next();
    const blob = file.makeCopy(destF).getAs(MimeType.PDF);
    const pdf =
destF.createFile(blob).setName(file.getName()+'.pdf');
  }
}

```


Create only PDF version no docs copies in destination folder

```
function manyPDFs1(){
  const sourceID = '1***1';
  const sourceF = DriveApp.getFolderById(sourceID);
  const destID = '2***2';
  const destF = DriveApp.getFolderById(destID);
  const files = sourceF.GetFiles();
  while(files.hasNext()){
    const file = files.next();
    const blob = file.getBlob();
    const pdf =
destF.createFile(blob).setName(file.getName()+'.pdf');
  }
}
```

Create Doc as PDF and email to Sheet data email address

Final Coding Project Exercise :

1. Create a Spreadsheet with headings that you want to use replacing the values in a Google Doc.
2. Create a Google Doc to use as a template for the PDF. Within the Doc for the dynamic values that will be populated from the spreadsheet add the keys using the curly brackets around the heading words from the Sheet heading row contents. {first} Add additional template text, images and setup to prepare to send out.
3. In Drive create a folder that you can use to save temporary files into.
4. Create a new Apps Script standalone application. Give it a name

5. Get the id of the Spreadsheet. Select the sheet data and get the heading row contents as an array.
6. Get the folder for the temporary files, Get the Doc file within DriveApp as a file object.
7. Loop through all the rows of content from the Spreadsheet data. Create a file making a copy of the template Doc.
8. Get the document object using DocumentApp of the file that was just created.
9. Open the doc and get the body.
10. Loop through the Sheet heading names, do a replaceText() of the heading name in the curly brackets to the row using the index value.
11. Set the name of the newly created doc to be the first and last name of the column data from the sheet.
12. Get the document as a blob.
13. Create the email subject and body contents.
14. Using MailApp send an email with the blob as a PDF attachment.

Hi, Laurence Welcome check out the attachment



```
function sender1(){
  const id = '1***1';
  const sheet =
SpreadsheetApp.openById(id).getSheetByName('data');
  const data = sheet.getDataRange().getValues();
  const rows = data.slice(1);
  const doid = '1***1';
  const temp = DriveApp.getFileById(doid);
```

```

const folder = DriveApp.getFolderById('1****1');
rows.forEach((row)=>{
  const file = temp.makeCopy(folder);
  const doc = DocumentApp.openById(file.getId());
  const body = doc.getBody();
  data[0].forEach((heading, i) =>{
    body.replaceText(`${heading}`, row[i]);
  })
  doc.setName(row[0]+row[1]);
  const blob = doc.getAs(MimeType.PDF);
  const email = row[3];
  const subject = row[0]+row[1] + ' New file';
  const hbody = `Hi, ${row[0]} Welcome check out the
attachment`;
  doc.saveAndClose();
  MailApp.sendEmail({
    to:email,
    subject:subject,
    htmlBody:hbody,
    attachments:[blob.getAs(MimeType.PDF)]
  });
  file.setTrashed(true);
})
//Logger.log(data);
}

```

