

Script (Google Apps Script) to beautify Google Form submissions by merging submission data with a Spreadsheet template to create sheet & PDF.

Automatically send the beautified submission form/sheet PDF to yourself & the submitter on every form submission.

Why use this script ?

- **Convert to offline submitted form to maintain office records.**
- Send a copy to the submitter to get signed or other legal reasons.
- View submissions better & clear , similar to an offline version of the form
- Convert offline forms into Google forms & back again to submitted form. You just need to create the template.

Practical Use Cases

- Tenant Form : Generate offline copy & get signed. Keep a legal record.
- Paying Guest Application
- Membership Forms
- Convert any offline form to Google Form & back to offline!

This script works for any template & form!

- Works fine for any template
- Works with any number of fields in the form
- No code changes required

Why you should get this script ?

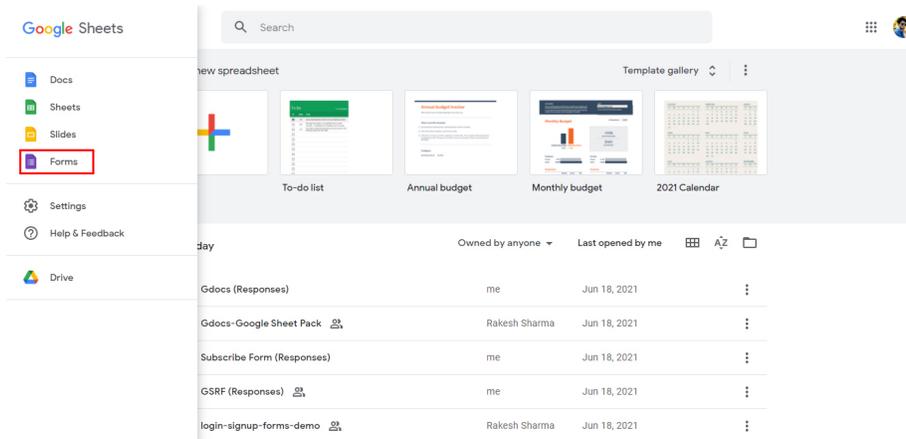
- **Zero Apps Script knowledge** required
- Generate Multiple/Bulk Sheet (& PDF)
- No Code Changes Required
- Working with new template & change/addition of merge fields is super easy. Only changes required in a sheet containing metadata like template id, destination folder id, merge fields & additional config.
- Awesome logs to debug.
- **Lifetime Usage**
- You are getting the full code , so you can modify accordingly.
- **Getting Started Video + PDF guide**
- **Source Code:** The actual Apps Script code.
- **Documentation:** PDF file containing all the details about the script, usage and additional information.

- Sample Sheets & Templates: The sheets that I use in the demo & getting started video, so you can play around or follow along.

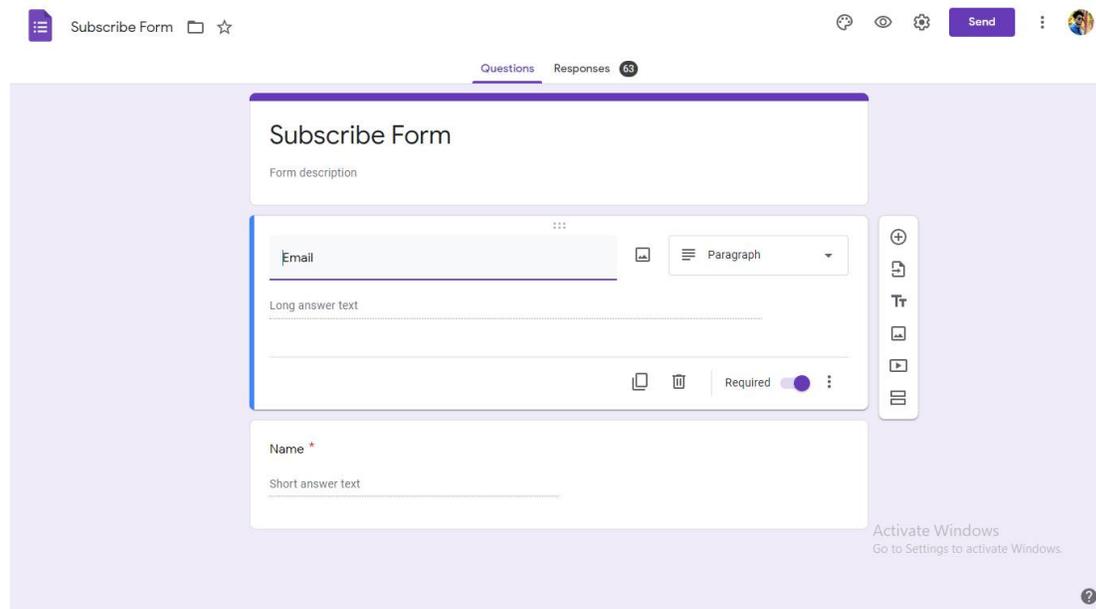
Product Demo=====

Guide=====

1. Create a Google Form. Ignore if you already have an existing form.



2. Create a Spreadsheet template where you want the values from the form submission merged.



3. Submit the form with dummy values so you can know the columns of the responses in the response sheet. Ignore if you already have responses.

4. Open settings of form & link spreadsheet , select create a new sheet. You can also use an existing empty spreadsheet if already created. This will link the form responses with a spreadsheet.

5. This response sheet is the important one where you have to use the script.

6. Create a new sheet/tab in this response spreadsheet & rename it "METADATA" (not the quotes).

7. There is a sample METADATA sheet provided to you. Copy the data (first two columns & the rows containing data) to the METADATA sheet you have created in step 6.

8. You need to edit & change the existing values according to your responses & the template. Also, you need to put your metadata like template id & other config/preferences. You also need to provide four additional columns at the end for process, error & email sending status updates from the script. This is optional but recommended. Read the documentation for in depth details of the METADATA sheet.

9. Now, it's time to add the script. Click on Extensions=>Apps Script. Now, you should see a code editor. Delete existing code. You should have got a file called "code.txt" from Gumroad. Open this & copy the code. Paste this code in the editor & save. Refresh your response sheet now. The editor will close automatically but after refresh, in the response sheet, you will see a new menu item called FormSubBeautify.

10. You can run now by clicking on any cell of a row in the response sheet. Click on the menu item & click "Generate Selected" . This will run for one row. If you select more rows, it will process the selected rows. Running & sending email automatically on form submission. Check the video on trigger to create a trigger to run the script automatically on every new form submission. You have to select the function called generateSubmitted to run. After the trigger gets created, the script will run for the new submission automatically. If email sending is opted, you should get emails with the PDF file.

Merge Google Form submission data with Sheets template to create dynamic sheets. Generate PDFs & email to submitter & yourself. You don't need a form, you can merge existing sheet data as well.

You can manually generate sheets & PDFs in bulk as well. To get started, check the getting started guide & the video. AUTH You will need to authorize the script to run.

DOCUMENTATION How does it work? You need to create a METADATA sheet in the spreadsheet which contains the form responses. The script reads this sheet & generates the sheet from the template sheet & also PDF if opted for. It also sends email according to the config on this sheet. METADATA Sheet I have included a sample METADATA sheet for reference. Check sample spreadsheet. Every METADATA sheet that you create needs to be exactly like this except the values in it. These values will change according to the number of fields & the sheet template. It also depends on the config that you want the script to read.

Information of all the fields: Cell B1 is the response sheet name. Because you can have multiple sheets in a single spreadsheet, you have to provide a name here. Cell B2 is the sheet template id. Cell B3 is the folder id of the folder where you want the generated/individual sheets to store. Cell B4 is an option to generate PDF. If Yes or y (case insensitive), PDF will be generated. Cell B5 is PDF folder id, i.e. where you want PDFs to be stored. Cell B6 is an option to delete the generated sheet. If Yes or y (case insensitive), will be deleted. Cell B7 is the file name of the generated sheet or PDF. This can be dynamic. To make dynamic, add the column (alphabet) of response sheet surrounding ##. Example ##B##_submission . This will check value of B (only if it is a merge field) & replace it with ##B##. Cell B8 is a column to update processing status. You should see a Yes in this column in the response sheet. Cell B9 is the error update column. Cell B10 is an option to send email (with PDF) to the form submitter. Cell B11 is the column (alphabet) where the script will get the email of the form submitter. Relevant only if you have selected B10 yes. B12 is the email sending status column to the form submitter. B13 is whether you want to receive email. If Yes or y, you will receive an email. B14 contains your email addresses separated by comma (if more than one). B15 is the sending status column (alphabet) of this email. From row 17, you need to provide your merge fields. Column A is the comuns (alphabet) of the response sheet, i.e. the actual data. Column B is the corresponding cell in the template where you want the value to be written in the template. So, the script will check all the rows from 17 to get data from the response sheet by reading the value from the column specified in column A & put it in the cell specified in column B. In short, column A is the source & B is the destination cell

RUNNING By default the script does not run automatically on form submission. You need to create a trigger to run the script on form submission. Refer getting started guide & the trigger video. Manually it can be run by clicking on the menu button called FormSubBeautify .

LOGS Check execution logs to debug any issue. The script logs most of the important things, making it easy to debug yourself.

NOTES 1. PDF is always generated in portrait mode & default size. You may need other PDF exporting code available on the internet to generate in landscape mode or with other config.

2. Check the print preview of your template & the wrapping of words inside the cells. Otherwise some data may be incomplete & hidden. You should prefer merging of cells to avoid this problem.
3. If merging cells in template, provide only the first leftmost cell in column B in METADATA sheet. You don't need to provide the entire cell range.
4. The processing update column, error update column & email sending status columns are not mandatory. Leave them empty if you want. But keeping them is recommended because you can check the status without checking the execution logs.
5. Google Apps Script has an execution limit of 6 minutes. If you are manually generating sheets & sending the PDFs , select a few rows at a time, otherwise it will time out.
6. Don't write anything below the merge fields at the end of the METADATA sheet. The script detects the last row automatically. Anything written below will mess up the script.
7. Test the PDFs & sheets generated with dummy data.

The email subject & body is hard coded in the code. You can change it in the script. Check the last part of the script. Variable names are self explanatory. Currently, there is no provision to send dynamic email body or subject. Feel free to modify according to your needs. If email sending is opted, PDF will be generated for the attachment even if PDF generation is not Yes. In this case, you won't find the PDF stored in the PDF destination folder.