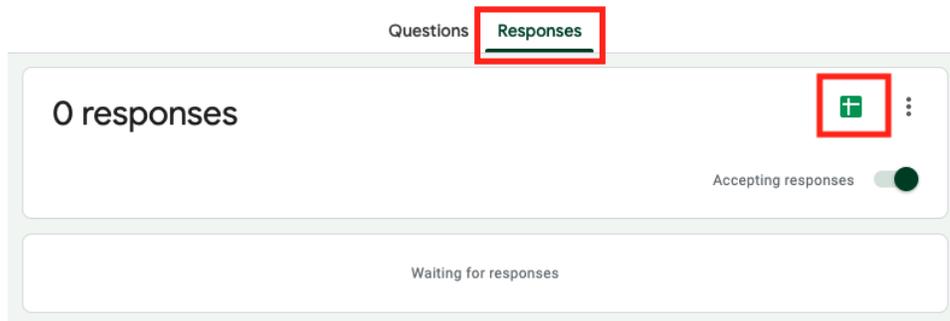


# How do I receive Google Form submission responses via email?

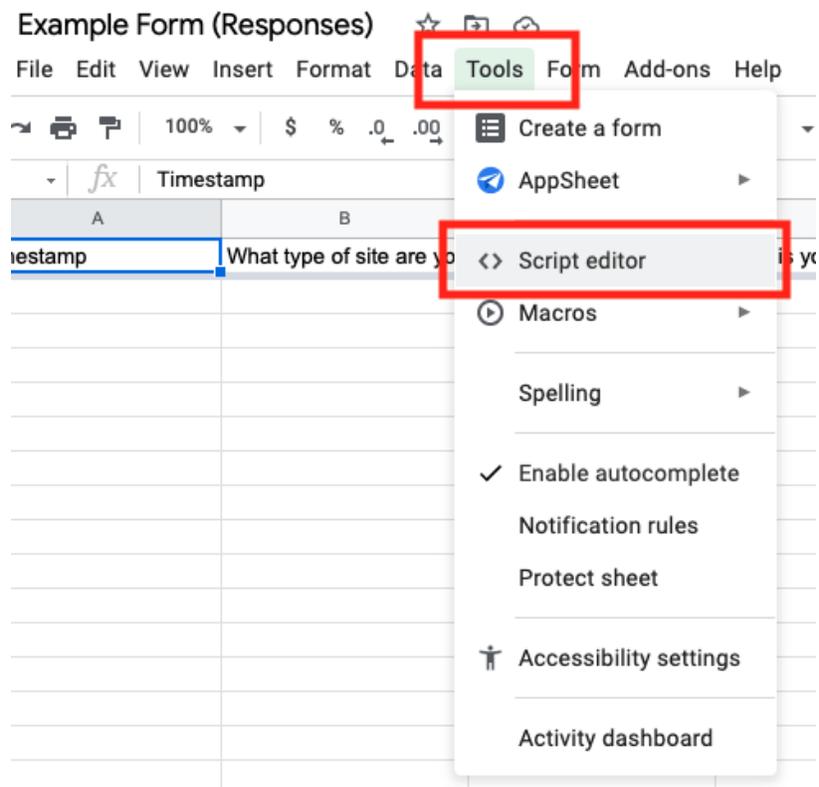
## Tell Me

In order to receive the Google Form responses via email, you will need to create a Google Apps Script to email responses each time someone submits a form.

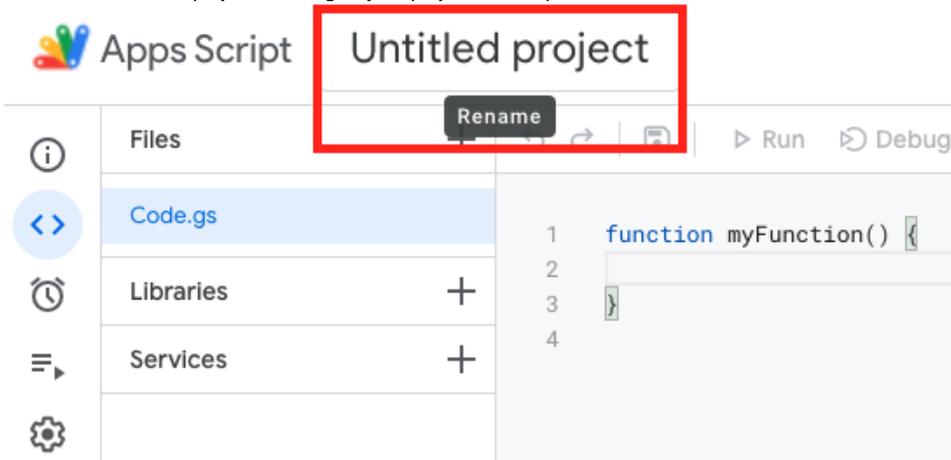
1. Start by creating a responses Google Sheet if you have not already done so:
  - a. Go to your Google Form
  - b. Click on the **Responses** tab
  - c. Click the **Create Spreadsheet** (or "View responses in Sheets" if already created)



2. From the Form responses Google Sheet, go to "Tools" > "Script Editor" to bring up the Google Apps Script console and create a new project.



3. Click on the "Untitled project" text to give your project a descriptive name.



4. Replace the content in the editor with with the following code, update the email recipients and email subject values in between the quotes, then click the "Save project" icon:

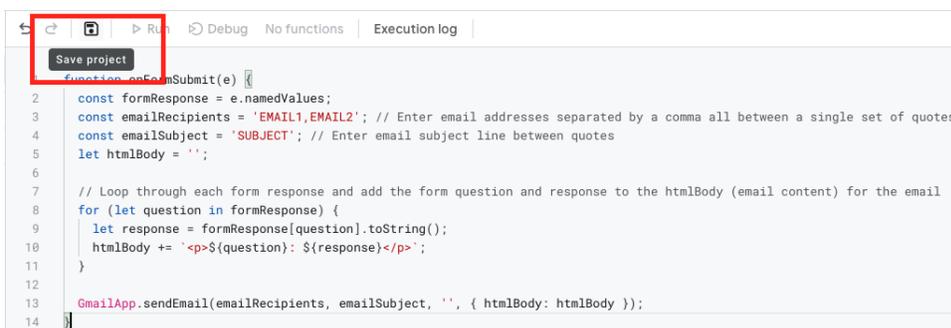
```
function onFormSubmit(e) {
  const formResponse = e.namedValues;
  const emailRecipients = 'EMAIL1,EMAIL2'; // Enter email addresses separated by a comma all between a
  single set of quotes
  const emailSubject = 'SUBJECT'; // Enter email subject line between quotes
  let htmlBody = '';

  // Loop through each form response and add the form question and response to the htmlBody (email
  content) for the email
  for (let question in formResponse) {
    let response = formResponse[question].toString();
    htmlBody += `<p>${question}: ${response}</p>`;
  }

  GmailApp.sendEmail(emailRecipients, emailSubject, '', { htmlBody: htmlBody });
}
```

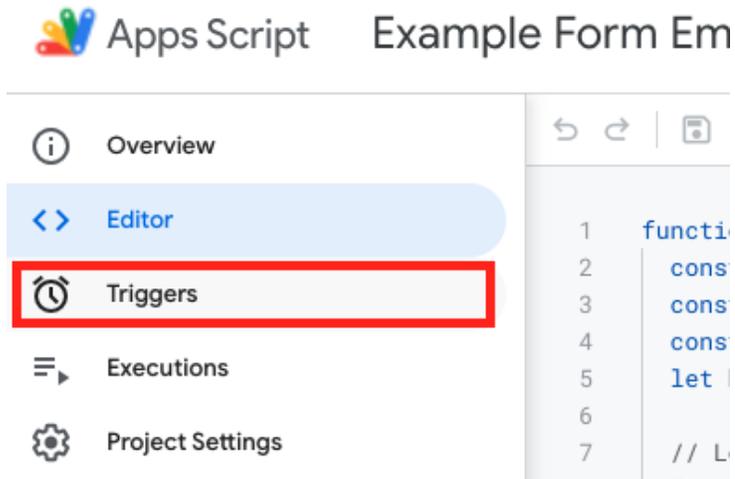
**i** Be sure to update the 'EMAIL1,EMAIL2' values with the actual email addresses you want to receive the form submission responses. You can enter only one email address or multiple email addresses separated by a comma. Your email addresses must be in between a single set of quotes.

Be sure to update the 'SUBJECT' value to the text you want to include in your email subject line.



5. Next you need to create a trigger to run the script each time the form is submitted:

a. Hover over the clock face icon on the left hand side of the screen and click “Triggers” to enter into the triggers listing page.



b. Create a new trigger by clicking the “Add Trigger” button on the bottom right side of the screen.



c. In the “Add Trigger” model, update the “Select event type” dropdown to “On form submit” and click “Save”.



Since we only have one function and we created it from the Google Sheet, the “Choose which function to run” and “Select event source” should be set correctly by default. However, please confirm that “Choose which function to run” is set to “onFormSubmit” (the name of the function we pasted above) and “Select event source” is set to “From spreadsheet” since this script will need to be triggered each time a new form is submitted and the response is added to the Google Sheet.

Because the script will need access to your Google account to send email, you will need to grant the script project access to your Google account in order to proceed and save the trigger. Please follow the prompts to allow access.

### Add Trigger for Example Form Email Responses on For...

Choose which function to run

onFormSubmit

Choose which deployment should run

Head

Select event source

From spreadsheet

Select event type

- On open
- On edit
- On change
- On form submit**

Failure notification settings +

Notify me daily

Cancel **Save**

6. After saving the trigger, click on the Editor "< >" icon on the left side of the screen to return to the code editor
7. From the editor, click the "Run" button to initialize onFormSubmit function.

**i** Ignore the "Error" that appears in the "Execution log". We are simply initiating the function so it will run on subsequent form submissions.

8. Now each time someone submits a response to your form, the recipients you listed in the script will receive an email from you containing a list of all the form questions along with their response.