

1. GTM: Set up your 404 error event.

1. **Set up a variable** for document.title ([see screenshot](#))
 - a. Variable Type = JavaScript Variable
 - b. Global Variable Name = document.title
 - c. Name the variable "m/a - document.title"
2. **Set up a trigger** to trigger for 404 errors ([see screenshot](#))
 - a. Trigger Type = Page View
 - b. This trigger fires on = insert the variable you created in step 1 "m/a - document.title" and add contains Page not found (or any other text that you find in the document title that indicates an 404 error, such as simply 404)
 - c. Name the trigger "m/a - 404_pages"
3. **Set up a tag** to send the 404 event to GA4 ([see screenshot](#))
 - a. Tag Type = Google Analytics: GA4 Event
 - b. Event name = 404_pages
 - c. Other settings = depends on your setup, not super relevant
 - d. Trigger = select the trigger "m/a - 404_pages" that you created in step 2.

m/a is our internal abbreviation for Midsummer Agency - feel free to replace it with your personal identifier :)

Check in ga4 that your 404_pages event comes in over the following days after your setup is done.

If you don't have GTM you can set it up directly through GA4 too - write me in DM to get those specific instructions if needed :) Click [here](#) to write me.

2. Sheets/Script: Set up the notification based on the 404 error event.

1. Create a new google sheet
2. **Launch Supermetrics Sidebar:** in the top menu in google sheets, select "Extensions">"Supermetrics">"Launch"
3. **Create a new query** ([see screenshot](#))
 - a. data source = google analytics 4
 - b. account = select the account of your client where you have the 404 error event
 - c. dates = Last year & this year to date
 - d. metrics = Event count
 - e. dimensions = Date, Session source / medium, Page referrer, Page location
 - f. filter = Event name equals {{insert the even name of your 404 error event}}
- I usually call them 404_pages but sometimes they already exist and have different names
 - g. options, select:
 - i. Replace blanc metric values with zeros
 - ii. Format results for Looker Studio
 - iii. Show all time values
4. **Add schedule** ([see screenshot](#))
 - a. choose "Automatic refresh"
 - i. action = refresh daily
 - ii. start at = 5:00
5. **Add Apps Script:**
 - a. In your google sheet go to extensions > Apps Script to open Apps Script
 - b. Copy the code you find below into the code section > click run > review the required permissions ([see here](#))
 - c. Go to triggers ([see here](#)) and set up a trigger ([see here](#))
 - i. Choose which function to run = checkEventCounts
 - ii. Which runs at deployment = Head
 - iii. Select event source = Time-driven
 - iv. Select type of time based trigger = Week timer
 - v. Select day of week = Every Monday > please, if you decide to use
 - vi. Select time of day = 8am to 9am
 - vii. Failure notification settings = Notify me daily

Code to insert in App Script, modify:

- line 3: insert sheet name here - copy the exact name of your sheet name in here which you can find at the bottom of your sheet, e.g. "404 checker (all website)" ([see screenshot](#))
- line 12: insert project name
- line 13: insert email address
- Line 32: in case you change your trigger frequency, don't forget to update the "in the last 7 days" text which is not dynamic.

```
function checkEventCounts() {  
  
    // Open the active spreadsheet and get the "404 checker (all  
website)" sheet  
  
    var sheet =  
SpreadsheetApp.getActiveSpreadsheet().getSheetByName("insert sheet  
name here");  
  
    // Get the data range for columns A to E  
  
    var dataRange = sheet.getRange("A:E");  
  
    var data = dataRange.getValues();  
  
    // Define the project name and recipient email  
  
    var projectName = "insert project name";  
  
    var recipientEmail = "insert email address";  
  
    var sheetUrl = SpreadsheetApp.getActiveSpreadsheet().getUrl();  
  
    // Get today's date  
  
    var today = new Date();  
  
    var sevenDaysAgo = new Date();  
  
    sevenDaysAgo.setDate(today.getDate() - 7);  
}
```

```
// Loop through the data rows, starting from the second row to skip
the header

for (var i = 1; i < data.length; i++) {

    // Parse the date in the format "yyyy-MM-dd"

    var date = new Date(data[i][0]);

    var sessionSourceMedium = data[i][1];

    var pageReferrer = data[i][2];

    var pageLocation = data[i][3];

    var eventCount = data[i][4];

    // Check if the date is within the last 7 days and there is a 404
    error (event count > 0)

    if (date >= sevenDaysAgo && date < today && eventCount > 0) {

        // Create the email subject and body

        var subject = "404 Notification: " + projectName;

        var body = "You received some visits to pages that resulted in a
404 error in the last 7 days. Please check it out!\n\n" +

            "You can review the details in the spreadsheet: " +
sheetUrl;

        // Send the email

        MailApp.sendEmail(recipientEmail, subject, body);

        break; // Exit the loop after sending the email once

    }
}
```

```
}
```

```
}
```

3. Read the result of the script in a sheet

Usually these are the columns in the sheet:

- **date** = displays the date when a 404 happened.
- **session source / medium** = displays the source / medium of the session of the specific user that had a 404 error. It doesn't mean that if this says google / cpc, the campaigns send to the 404 page! It just means that a user that arrived through google / cpc, during their session, encountered a 404 page.
- **page location** = the page the user is on at exact moment of getting a 404 error. This is the 404 error page.
- **page referrer** = the page you were browsing before arriving at the . You arrived on page location via a button, link, navigation menu item, whatever.
- **event count** = the number of times a 404 error event happened.