



Automation Workflow Set-Up Guide: ITEC 2380 Lab File Validation for HTML

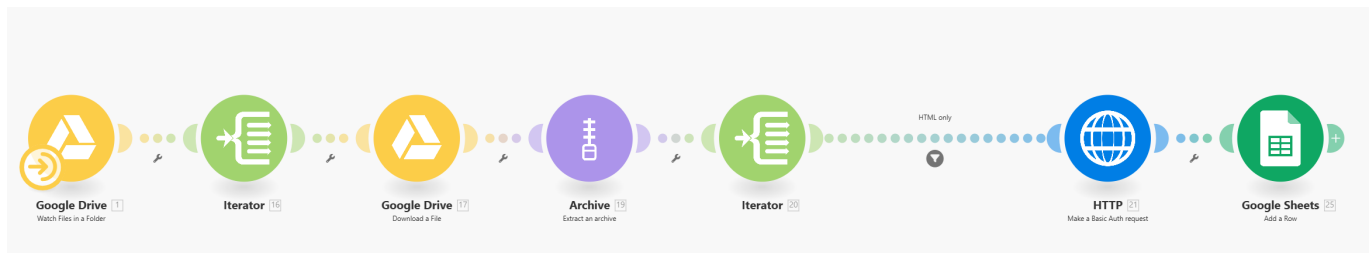
Table of Contents

1. Purpose & Overview
2. Prerequisites
3. Creating the Automation Workflow Diagram
 - Step 1: Watch Files in a Google Drive Folder
 - Step 2: Set Up the First Iterator
 - Step 3: Download the File from Google Drive
 - Step 4: Extract the Archive
 - Step 5: Set Up the Second Iterator
 - Step 6: Set Up the HTTP Request
 - Step 7: Filter for HTML Files
 - Step 8: Add the Results to Google Sheets
4. Running the Automation Workflow
5. Recommendations for Grading Efficiency

Purpose & Overview

The "ITEC 2380 Lab File Validation for HTML" project is designed to automate the process of grading HTML files submitted by students. By grabbing the zip files from students and running each through the W3C HTML Validator, this workflow significantly reduces grading time from 15-45 minutes per student to just 5-10 minutes (50%-78% more efficient). This guide will walk you through creating an automation workflow in Make that monitors a Google Drive folder for new files, processes those files, and validates HTML files using the W3C HTML Validator API.

Below is what the diagram will look like once complete:



Prerequisites

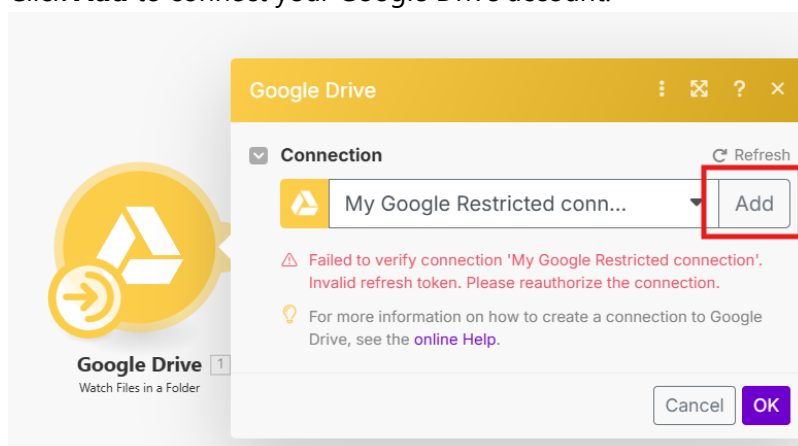
- A Google account with access to Google Drive.
- A client ID and client secret for Google Drive API.
- Basic knowledge of how to navigate and use Make.

Creating the Automation Workflow Diagram

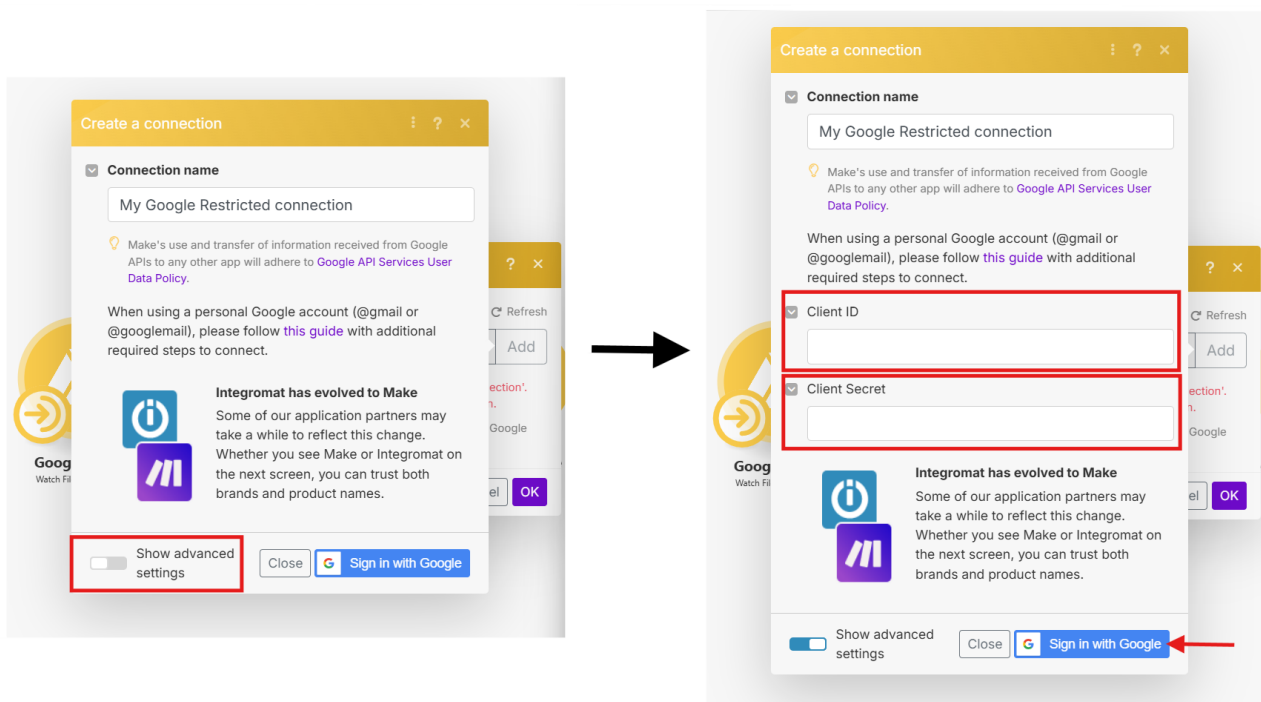
Step 1: Watch Files in a Google Drive Folder

Set up the module to monitor a specific folder in Google Drive.

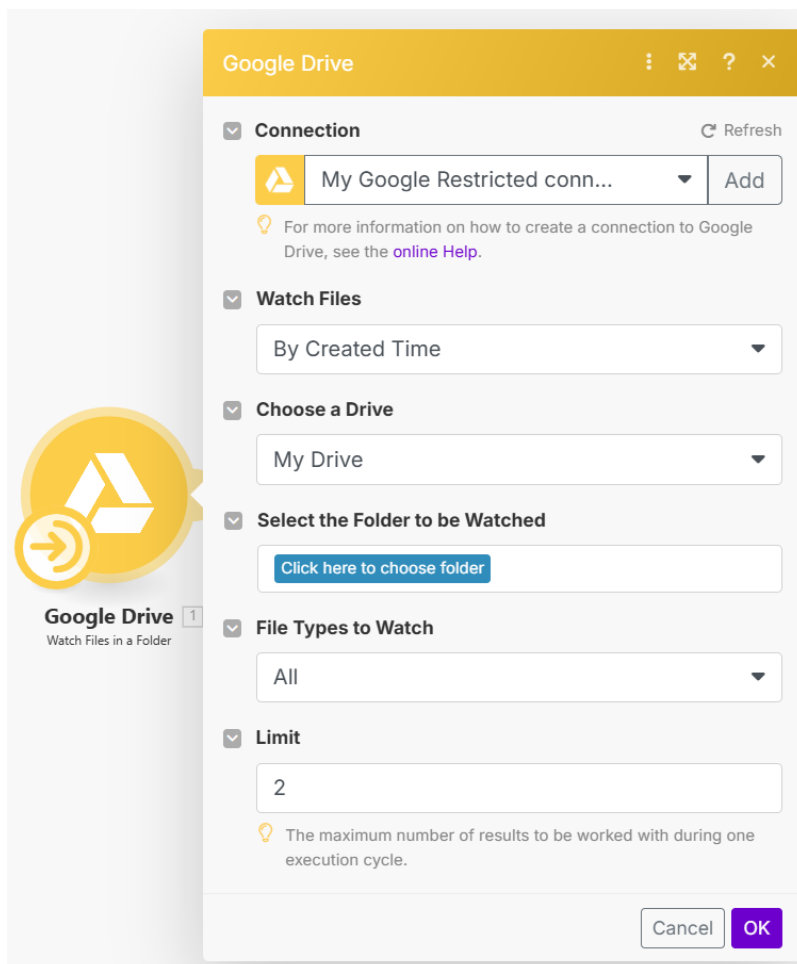
1. Add the **Google Drive** module and select **Watch Files in a Folder**.
2. Click **Add** to connect your Google Drive account.



3. Show advanced settings, then enter the client ID and client secret from your Google account. Then sign in.



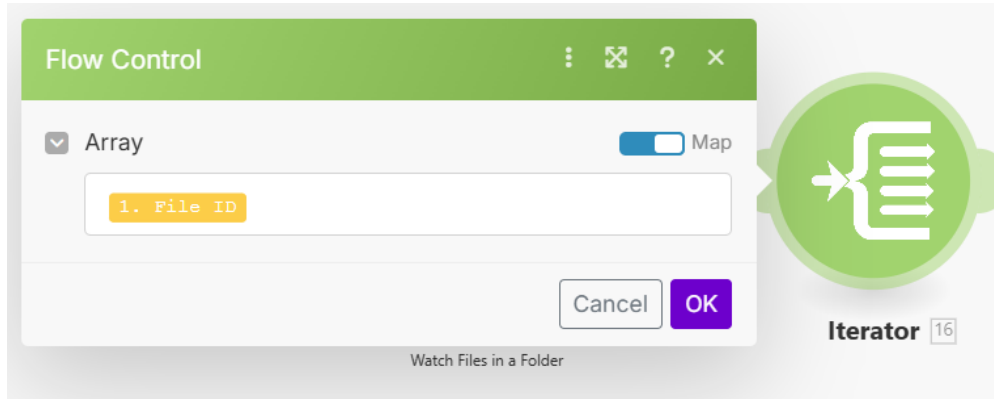
4. Once complete, set up the Google Drive as below. Remember to select the correct folder from the Drive.



Step 2: Set Up the First Iterator

This iterator will allow you to process multiple files.

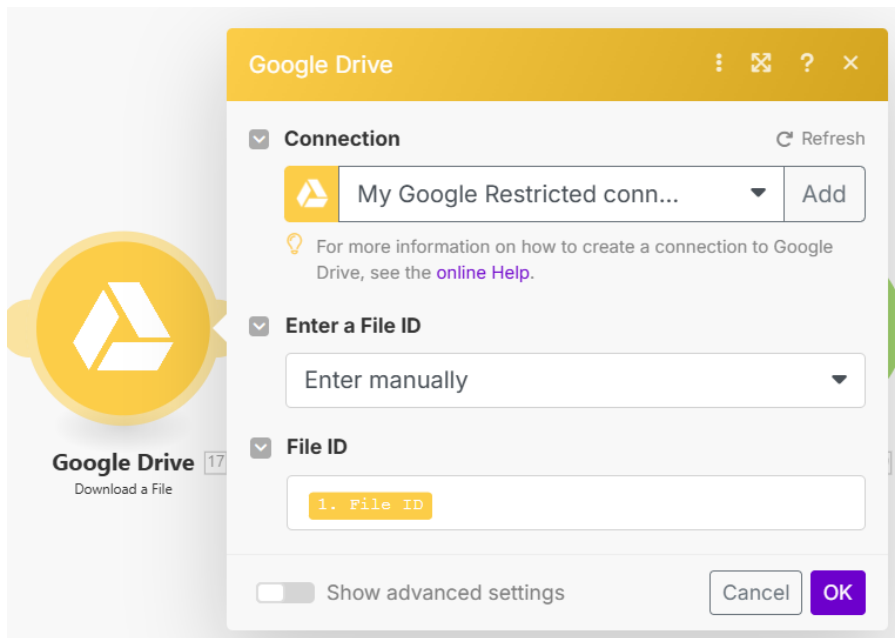
1. Add an **Iterator** module.
2. In **Array**, grab the File ID (**1. File ID**), with the map option toggled **On**. Make sure the number and color are matching with the screenshot below.



Step 3: Download the File from Google Drive

Set up the module to download the detected files.

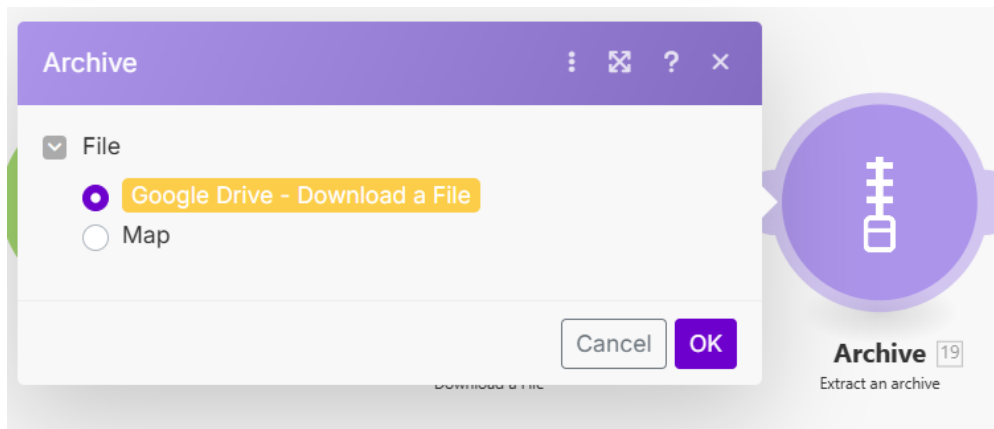
1. Add another **Google Drive** module and select **Download a File**.
2. After connecting to the Google account (See **Step 1**), **Enter the File ID** manually.
3. The **File ID** is **1. File ID**. Make sure the number and color are matching with the screenshot below.



Step 4: Extract the Archive

The files are in a zip, so this step will extract them.

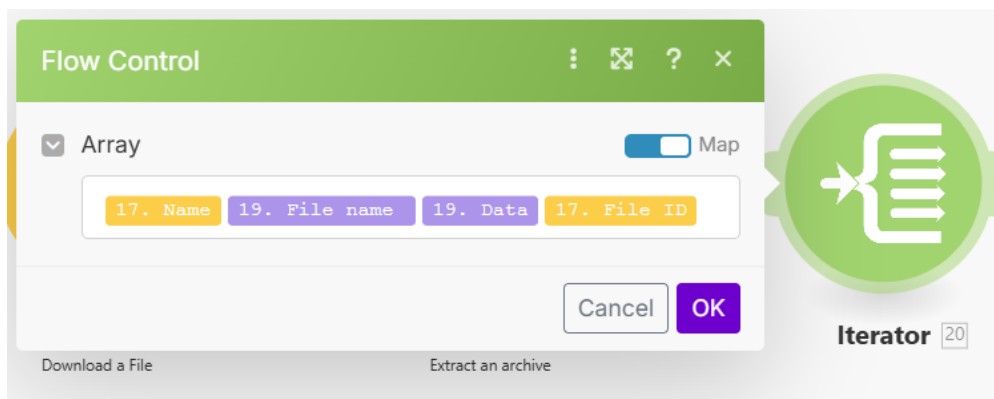
1. Add the **Archive** module.
2. Select **Google Drive - Download a File**.



Step 5: Set Up the Second Iterator

This iterator will process each item within the extracted archive.

1. Add another **Iterator** module.
2. In **Array**, grab the **17. Name**, **19. File name**, **19. data**, and **17. File ID**, with the map option toggled **On**. Make sure the numbers and colors are matching with the screenshot below.



Step 6: Set Up the HTTP Request

Use the W3C HTML Validator to validate the HTML files.

1. Add an **HTTP** module and select **Make Basic Auth Request**.
2. Add **Credentials** as W3C HTML Validator for the name, with no username or password.
3. Enter the URL: <https://validator.w3.org/nu/?out=gnu>.
4. Set the method to **POST**.
5. Set the body type to **Raw**, and content type to **HTML (text/html)**.
6. Set the request content to **19. Data**. Make sure the number and color are matching with the screenshot below.

HTTP

☒ **Credentials**

W3C HTML Validator ▼ Add

☒ **URL**

☒ **Method** Map

▼

☒ **Headers** Map

+ Add a header

☒ **Query String** Map

+ Add parameter

☒ **Body type**

▼

☒ **Content type**

▼

Sets the **Content-Type** request header.

☒ **Request content**

☒ **Parse response**

☐ Yes ☒ No

Automatically parses responses and converts JSON and XML responses so you don't need to use JSON or XML parser. Before you can use parsed JSON or XML content, run the module once manually so that the module can recognize the response content and allows you to map it in subsequent modules.

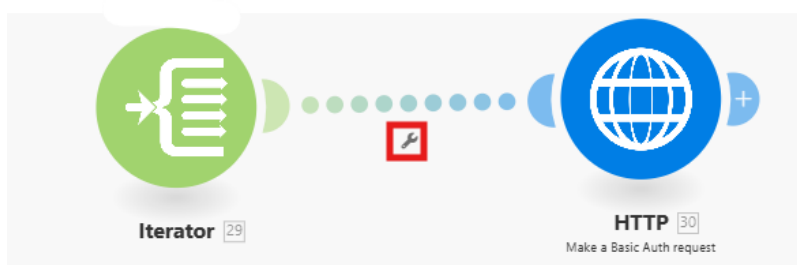
☐ Show advanced settings Cancel OK

LS

Step 7: Filter for HTML Files

Add a filter to only process HTML files.

1. Click the wrench located under the connection between the second Iterator module and the HTTP module to **Set up a Filter** and label it **HTML Only**.



2. Set the **Condition** to **19. File Name**, with the text operator **Ends With** and input **.html**. Make sure the number and color are matching with the screenshot below.
3. Add an **OR rule** with the **Condition 19. File Name**, the text operator **Equals To**, and input **.htm**. Make sure the number and color are matching with the screenshot below.

Set up a filter

Label

HTML only

Condition

19. File name

Text operators: Ends with

.html

Add AND rule Add OR rule

or

19. File name

Text operators: Equal to

.htm

Add AND rule Add OR rule

Cancel OK

Step 8: Add the Results to Google Sheets

Add a module to log the results to Google Sheets.

1. Add a **Google Sheets** module and select **Add a Row**.
2. After connecting to the Google account (See **Step 1**), set the **Search Methods** to **Search by Path**.
3. Connect to the proper **Drive, Spreadsheet ID, and Sheet Name**. The sheet should be set up with three columns: Student Name, File name, Validator Outcomes.
4. Set the **Table** to contain headers.
5. Set the **Values** as follows: Student name (**17. Name**), File Name (**19. File Name**), Validator Outcomes (**21. Data**). Make sure the numbers and colors are matching with the screenshot below.

Google Sheets

My Google connection

Add

For more information on how to create a connection to Google Sheets, see the [online Help](#).

Search Method

Search by path

Drive

My Drive

Spreadsheet ID

Sheet Name

Table contains headers

Refresh

Yes

Values

Student Name (A)

17. Name

File Name (B)

19. File name

Validator Outcomes (C)

21. Data

(D)

(E)

(F)

Show advanced settings

Cancel

OK

Google Sheets

Add a Row

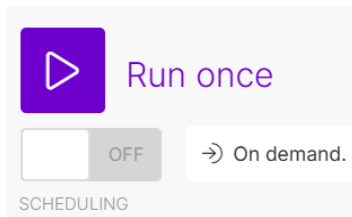
AI

FAVORITES

8 / 9

Running the Automation Workflow

1. Set the **Scheduling** to Off so that it runs on demand. Press the **Run Once** button for the automation to begin.



This concludes the step-by-step setup. Make sure to test the workflow thoroughly to ensure it processes files correctly. It will look like this once complete:



Recommendations for Grading Efficiency

After the names and errors have been populated into the Google Sheet, you can significantly cut down on grading time by color-coding the errors based on their severity:

- **Acceptable Errors:** Use a green highlight to indicate minor issues that do not significantly affect the quality of the HTML file.
- **Minor Errors:** Use a yellow highlight to indicate errors that are noticeable but do not drastically impact functionality.
- **Major Errors:** Use a red highlight for critical errors that affect the usability or presentation of the HTML content.
- **Chained Errors:** If an error results from a previous issue, mark it in a lighter shade of the corresponding severity color.

This color-coding system allows for quick visual assessment, making it easier to provide constructive feedback and grades.