Sử dụng office script để copy data của các cột trong excel, trả về một mảng. Sau đó sử dụng power automate tách mảng để lấy được từng cột và dán lại vào file excel khác

Tôi sẽ hướng dẫn bạn cách sử dụng Office Script để copy dữ liệu từ các cột trong Excel, trả về một mảng, sau đó dùng Power Automate để tách mảng và dán dữ liệu vào một file Excel khác. Dưới đây là giải pháp chi tiết:

**Bước 1: Tạo Office Script để lấy dữ liệu từ các cột**

Đầu tiên, bạn cần tạo một Office Script để lấy dữ liệu từ các cột trong file Excel nguồn và trả về dưới dạng một mảng.

Dưới đây là đoạn mã Office Script mẫu:

typescript

CollapseWrapCopy

function main(workbook: ExcelScript.Workbook): string[][] {

*// Lấy worksheet hiện tại (hoặc chỉ định worksheet cụ thể)*

let sheet = workbook.getActiveWorksheet(); *// Có thể thay bằng workbook.getWorksheet("Sheet1")*

*// Lấy phạm vi dữ liệu đã sử dụng trong worksheet*

let usedRange = sheet.getUsedRange();

*// Lấy tất cả giá trị từ phạm vi đã sử dụng*

let data = usedRange.getValues();

*// Trả về mảng dữ liệu*

return data;

}

**Giải thích:**

* getUsedRange(): Lấy toàn bộ phạm vi có dữ liệu trong worksheet.
* getValues(): Trả về một mảng 2 chiều chứa tất cả giá trị trong phạm vi.
* Mảng trả về có dạng string[][], trong đó mỗi phần tử là một hàng, và mỗi hàng chứa các giá trị của các cột.

Lưu script này với tên, ví dụ: GetColumnData.

**Bước 2: Tạo Power Automate Flow để xử lý mảng và dán vào file Excel khác**

Sau khi có script trả về mảng, bạn sẽ sử dụng Power Automate để tách mảng thành từng cột và dán vào file Excel đích.

Dưới đây là các bước trong Power Automate:

**1. Trigger (Kích hoạt Flow)**

* Chọn một trigger phù hợp, ví dụ: "Manually trigger a flow" (Kích hoạt thủ công) để thử nghiệm.

**2. Hành động: Run script (Chạy script để lấy dữ liệu)**

* Thêm hành động **"Excel Online (Business) - Run script"**.
* Cấu hình:
  + **Location**: Chọn vị trí file Excel nguồn (OneDrive hoặc SharePoint).
  + **Document Library**: Chọn thư viện chứa file.
  + **File**: Chọn file Excel nguồn.
  + **Script**: Chọn script GetColumnData đã tạo ở bước 1.
* Kết quả từ script sẽ được lưu trong biến result dưới dạng mảng 2 chiều.

**3. Hành động: Parse JSON (Phân tích mảng trả về)**

* Thêm hành động **"Parse JSON"** để xử lý dữ liệu mảng từ script.
* **Content**: Điền body('Run\_script')['result'] (lấy kết quả từ hành động Run script).
* **Schema**: Sử dụng schema sau để phân tích mảng 2 chiều:

json

CollapseWrapCopy

{

"type": "array",

"items": {

"type": "array",

"items": {

"type": "string"

}

}

}

* Hành động này sẽ cho phép bạn truy cập từng hàng và cột trong mảng.

**4. Hành động: Tách từng cột từ mảng**

* Thêm hành động **"Compose"** hoặc **"Select"** để tách dữ liệu thành các cột riêng biệt.
* Ví dụ, nếu bạn muốn lấy từng cột riêng:
  + Sử dụng biểu thức để lấy cột 0 (cột đầu tiên):

plaintext

CollapseWrapCopy

outputs('Parse\_JSON')?[0]

* + Lấy cột 1:

plaintext

CollapseWrapCopy

outputs('Parse\_JSON')?[1]

* + Lặp lại cho các cột khác tùy thuộc vào số cột trong file nguồn.
* Nếu muốn tách toàn bộ cột, bạn có thể dùng **"Apply to each"** để lặp qua các hàng và sử dụng biểu thức như:

plaintext

CollapseWrapCopy

item()[0] (cột 1), item()[1] (cột 2), v.v.

**5. Hành động: Ghi dữ liệu vào file Excel đích**

* Thêm hành động **"Excel Online (Business) - Run script"** lần nữa để ghi dữ liệu vào file đích.
* Tạo một Office Script khác để nhận mảng và ghi vào file Excel đích. Ví dụ:

typescript

CollapseWrapCopy

function main(workbook: ExcelScript.Workbook, columnData: string[][]) {

*// Lấy worksheet đích*

let sheet = workbook.getActiveWorksheet(); *// Hoặc workbook.getWorksheet("Sheet1")*

*// Xác định phạm vi để dán dữ liệu (ví dụ: bắt đầu từ A1)*

let range = sheet.getRange("A1").getResizedRange(columnData.length - 1, columnData[0].length - 1);

*// Gán giá trị từ mảng vào phạm vi*

range.setValues(columnData);

}

* Lưu script này với tên, ví dụ: PasteColumnData.
* Trong Power Automate:
  + **Location**: Chọn vị trí file Excel đích.
  + **Document Library**: Chọn thư viện chứa file.
  + **File**: Chọn file Excel đích.
  + **Script**: Chọn PasteColumnData.
  + **columnData**: Truyền dữ liệu từ bước Parse JSON hoặc Compose (ví dụ: outputs('Parse\_JSON')).

**Lưu ý**

1. **Tách cột riêng lẻ**: Nếu bạn chỉ muốn dán từng cột riêng biệt vào các vị trí khác nhau trong file đích, bạn có thể sửa script PasteColumnData để nhận từng cột và dán vào các phạm vi cụ thể (ví dụ: cột A, cột B, v.v.).
2. **Kiểm tra dữ liệu**: Đảm bảo mảng trả về từ script đầu tiên khớp với cấu trúc file đích để tránh lỗi.
3. **Hiệu suất**: Nếu file lớn, hãy thử nghiệm với dữ liệu nhỏ trước để tối ưu flow.

**Ví dụ kết quả**

Giả sử file nguồn có dữ liệu như sau:

| **Cột A** | **Cột B** | **Cột C** |
| --- | --- | --- |
| 1 | A | X |
| 2 | B | Y |
| 3 | C | Z |

* Script GetColumnData trả về mảng: [["1", "A", "X"], ["2", "B", "Y"], ["3", "C", "Z"]].
* Power Automate tách thành:
  + Cột 1: ["1", "2", "3"]
  + Cột 2: ["A", "B", "C"]
  + Cột 3: ["X", "Y", "Z"]
* Script PasteColumnData dán toàn bộ mảng hoặc từng cột vào file đích theo yêu cầu.

Nếu bạn cần thêm chi tiết hoặc gặp lỗi khi triển khai, hãy cho tôi biết!

Chổ lưu ý số 1 bạn hãy guide chi tiết giúp tôi

DeepSearch

Thinking

Giải thích yêu cầu

Xử lý từng cột

Thinking

Giải thích yêu cầu

* Được rồi, yêu cầu của bạn là "Chổ lưu ý số 1 bạn hãy guide chi tiết giúp tôi". Tôi đang cố gắng hiểu rõ hơn.
* Hm, có vẻ bạn đang nhắc đến lưu ý số 1 từ cuộc trò chuyện trước và muốn hướng dẫn chi tiết.
* Lưu ý số 1 nói về việc tách từng cột riêng lẻ và dán vào các vị trí cụ thể trong file Excel đích bằng Power Automate và Office Scripts.
* Tôi đang nghĩ cách tách mảng 2D thành từng cột để xử lý riêng lẻ trong Power Automate.
* Sau khi chạy script GetColumnData, tôi có mảng 2D, mỗi phần tử là một hàng với các ô.
* Để tách cột, tôi cần thu thập từng phần tử cùng chỉ số từ mỗi hàng, ví dụ cột 0 là phần tử đầu tiên của mỗi hàng.
* Tôi đang tìm cách xác định số cột bằng cách lấy độ dài của hàng đầu tiên từ mảng.
* Trong Power Automate, tôi có thể dùng "Apply to each" trên body('Parse JSON') để lấy item()[i] cho mỗi hàng, thu thập giá trị cột i.
* Tôi đang cân nhắc dùng "Select" action để ánh xạ qua các hàng và trích xuất cột cụ thể, nhưng mảng là mảng các mảng nên hơi phức tạp.
* Tôi đang nghĩ tạo danh sách chỉ số cột từ 0 đến số cột trừ 1 để xử lý từng cột riêng biệt.
* Tôi đang xem xét dùng hàm "Range" trong Power Automate để tạo danh sách chỉ số cột từ 0 đến num\_columns - 1.
* Với mỗi col\_index, tôi có thể dùng "Apply to each" trên body('Parse JSON'), thêm "Compose" với biểu thức @{item()[col\_index]} để thu thập giá trị cột.
* Tôi đang nghĩ cần script Office mới để dán mảng cột đơn vào vị trí cụ thể trong file Excel đích, vì PasteColumnData hiện tại dùng cho mảng 2D.
* Tôi đang cân nhắc chỉnh sửa script PasteColumnData để nhận mảng cột đơn và dán vào phạm vi chỉ định, ví dụ từ ô A1.
* Có thể bạn muốn dán cột 1 vào A1:A3, cột 2 vào D1:D3, và cột 3 vào G1:G3, tùy vị trí.
* Tôi đang nghĩ đến phương án 1: chỉnh sửa script để nhận mảng cột và ô bắt đầu, như vậy linh hoạt hơn.
* Tôi đang xem xét cách người dùng chỉ định vị trí dán cho mỗi cột, nếu cố định thì dễ, nhưng nếu thay đổi thì cần thêm thông tin.
* Tôi đang nghĩ giả sử dán cột theo thứ tự, cột 1 vào A1, cột 2 vào B1, và cứ thế.
* Với chỉ số cột 0, tôi dùng @char(add(col\_index + 65)) để lấy 'A', phù hợp với ASCII 65.
* Với mỗi chỉ số cột, tôi tính ô bắt đầu bằng @concat(char(add(col\_index + 65)), '1'), ví dụ cột 0 là 'A1', cột 1 là 'B1'.
* Tôi đang nghĩ cách chạy script GetColumnData trên file nguồn để lấy mảng 2D, sau đó phân tích JSON.

Xử lý từng cột

* Sau khi thu thập giá trị cột, tôi dùng "Collect data from all responses" để tạo mảng cho cột đó, gọi là \_column\_data.
* Trong Power Automate, tôi cần truyền hai tham số cho PasteColumnData: dữ liệu và ô bắt đầu.
* Tôi đang nghĩ thêm tham số "data" với giá trị \_column\_data và "startCell" với giá trị startCell cho action "Run script".
* Tôi đang tìm hiểu cách truyền nhiều tham số trong action "Run script" của Power Automate, vì script Office dùng tham số vị trí.
* Trong Office Script, hàm main lấy 'workbook' làm tham số đầu tiên, các tham số khác truyền theo thứ tự.
* Power Automate cho phép truyền tham số bằng cách chỉ định tên và giá trị, nhưng có thể dựa vào vị trí nếu không đặt tên.
* Tôi đang nghĩ script Office có thể định nghĩa với tham số tên, như data: string[]; startCell: string, để Power Automate ánh xạ dễ hơn.
* Tôi đang nghĩ cách thiết lập flow Power Automate: chạy GetColumnData, phân tích JSON, lấy số cột, tạo danh sách chỉ số cột.
* Với mỗi col\_index, tôi dùng "Apply to each" trên parsed\_data, thêm "Compose" với @item()[col\_index] để lấy giá trị ô.
* Tôi đang nghĩ cách tối ưu hóa, vì hành động lồng nhau có thể phức tạp. Có thể dùng "Select" action để trích xuất phần tử cụ thể mà không cần vòng lặp lồng.
* Tôi vừa tìm ra cách đơn giản hơn: dùng @outputs('Parse JSON')?[\*][col\_index] để lấy mảng cột mà không cần vòng lặp lồng.
* Với cách này, tôi không cần vòng lặp lồng, chỉ cần "Compose" với biểu thức trên để lấy mảng cột.
* Nếu họ muốn dán cột theo thứ tự từ A1, cách này đủ dùng. Nếu cần vị trí khác, họ có thể điều chỉnh biểu thức ô bắt đầu.
* Tôi nghĩ hướng dẫn này đã bao quát lưu ý số 1. Nếu bạn cần làm rõ thêm, cứ hỏi nhé!
* Tôi đang nghĩ thêm về cách tạo script PasteColumnData để xử lý mảng cột đơn và ô bắt đầu.
* Tôi đang cân nhắc cách đảm bảo file Excel đích mở đúng và worksheet hoạt động khi chạy script qua Power Automate.
* Tôi đang nghĩ cách tối ưu hóa flow Power Automate để giảm số hành động lồng nhau, làm quy trình mượt mà hơn.

**Key Points**

* Dường như có thể tách từng cột từ mảng dữ liệu Excel và dán vào các vị trí khác nhau trong file đích bằng Office Scripts và Power Automate.
* Quá trình này có thể phức tạp, đòi hỏi tạo script để xử lý mảng và điều chỉnh flow trong Power Automate.
* Nghiên cứu cho thấy cách tiếp cận này hiệu quả nếu bạn biết số cột và vị trí dán, nhưng cần thử nghiệm với dữ liệu nhỏ trước.

**Hướng Dẫn Tách và Dán Từng Cột**

**Chuẩn Bị Office Scripts**

* Tạo script GetColumnData để lấy dữ liệu từ các cột trong file Excel nguồn và trả về mảng 2 chiều.  
  Ví dụ:

javascript

CollapseWrapCopy

function main(workbook: ExcelScript.WORKBOOK): string[][] {

let sheet = workbook.getActiveWorksheet();

let usedRange = sheet.getUsedRange();

let data = usedRange.getValues();

return data;

}

* Tạo script PasteColumnData để dán một cột vào vị trí cụ thể trong file đích, nhận mảng 1 chiều và địa chỉ ô bắt đầu.  
  Ví dụ:

javascript

CollapseWrapCopy

function main(workbook: ExcelScript.WORKBOOK; data: string[]; startCell: string) {

let sheet = workbook.getActiveWorksheet();

let range = sheet.getRange(startCell).getResizedRange(data.length - 1);

range.setValues(data.map((value) => [value]));

}

**Cấu Hình Power Automate Flow**

* **Bước 1: Chạy Script Lấy Dữ Liệu**  
  Sử dụng hành động "Run script" để chạy GetColumnData trên file nguồn, lưu kết quả vào biến (ví dụ: source\_data).
* **Bước 2: Phân Tích JSON**  
  Sử dụng "Parse JSON" để xử lý mảng trả về, với schema là mảng 2 chiều của chuỗi, lưu vào biến parsed\_data.
* **Bước 3: Xác Định Số Cột**  
  Tạo biến first\_row lấy hàng đầu tiên từ parsed\_data, sau đó tính num\_columns bằng độ dài của first\_row.
* **Bước 4: Tạo Danh Sách Chỉ Số Cột**  
  Tạo biến column\_indices với giá trị @range(0,@sub(num\_columns,-1)) để có danh sách chỉ số cột (0, 1, 2, ...).
* **Bước 5: Xử Lý Mỗi Cột**  
  Sử dụng "Apply to each" trên column\_indices:
  + Tạo hành động "Compose" để lấy dữ liệu cột với biểu thức @outputs('Parse JSON')?[\*][col\_index], lưu vào column\_data.
  + Tạo hành động "Compose" để tính địa chỉ ô bắt đầu với @concat(char(add(col\_index +65)),'1'), lưu vào startCell (ví dụ: cột 0 là "A1", cột 1 là "B1", v.v.).
  + Chạy "Run script" với PasteColumnData trên file đích, truyền tham số:
    - data: Kết quả từ "Compose" đầu tiên.
    - startCell: Kết quả từ "Compose" thứ hai.

**Lưu Ý**

* Đảm bảo file đích có worksheet phù hợp và không bị ghi đè dữ liệu quan trọng.
* Nếu muốn dán vào vị trí khác, điều chỉnh biểu thức startCell cho phù hợp.
* Thử nghiệm với dữ liệu nhỏ để đảm bảo flow hoạt động đúng.

**Báo Cáo Chi Tiết**

Dưới đây là hướng dẫn chi tiết và toàn diện về cách sử dụng Office Scripts và Power Automate để tách từng cột từ mảng dữ liệu Excel và dán vào các vị trí khác nhau trong file đích, dựa trên yêu cầu của bạn. Báo cáo này bao gồm tất cả các bước, giải thích và ví dụ để đảm bảo bạn có thể triển khai hiệu quả.

**Giới Thiệu**

Quá trình tách từng cột từ mảng dữ liệu Excel và dán vào các vị trí cụ thể trong file đích đòi hỏi sự phối hợp giữa Office Scripts để xử lý dữ liệu và Power Automate để quản lý luồng công việc. Phương pháp này đặc biệt hữu ích khi bạn cần tổ chức lại dữ liệu theo cột riêng lẻ, chẳng hạn như dán cột 1 vào cột A, cột 2 vào cột B, v.v., trong file Excel đích. Tuy nhiên, mức độ phức tạp có thể tăng nếu số cột thay đổi hoặc vị trí dán không cố định.

**Chuẩn Bị Office Scripts**

Trước tiên, bạn cần tạo hai script trong Office Scripts để xử lý dữ liệu:

1. **Script GetColumnData**  
   Script này lấy toàn bộ dữ liệu từ phạm vi đã sử dụng trong worksheet và trả về dưới dạng mảng 2 chiều. Ví dụ:

javascript

CollapseWrapCopy

function main(workbook: ExcelScript.WORKBOOK): string[][] {

let sheet = workbook.getActiveWorksheet();

let usedRange = sheet.getUsedRange();

let data = usedRange.getValues();

return data;

}

* + Mảng trả về có dạng như [["1", "A", "X"], ["2", "B", "Y"], ["3", "C", "Z"]] nếu file nguồn có 3 cột và 3 hàng.

1. **Script PasteColumnData**  
   Script này nhận một mảng 1 chiều (dữ liệu của một cột) và địa chỉ ô bắt đầu, sau đó dán dữ liệu vào worksheet đích. Ví dụ:

javascript

CollapseWrapCopy

function main(workbook: ExcelScript.WORKBOOK; data: string[]; startCell: string) {

let sheet = workbook.getActiveWorksheet();

let range = sheet.getRange(startCell).getResizedRange(data.length - 1);

range.setValues(data.map((value) => [value]));

}

* + Ví dụ, nếu data là ["1", "2", "3"] và startCell là "A1", dữ liệu sẽ được dán vào A1:A3.

**Cấu Hình Power Automate Flow**

Để tự động hóa quá trình, bạn cần tạo một flow trong Power Automate với các bước sau:

**Bước 1: Kích Hoạt và Chạy Script Lấy Dữ Liệu**

* Sử dụng trigger "Manually trigger a flow" để thử nghiệm (hoặc trigger phù hợp khác).
* Thêm hành động "Excel Online (Business) - Run script" để chạy GetColumnData trên file Excel nguồn.
* Cấu hình:
  + Location: Vị trí file nguồn (OneDrive hoặc SharePoint).
  + Document Library: Thư viện chứa file.
  + File: Chọn file Excel nguồn.
  + Script: Chọn GetColumnData.
* Kết quả sẽ được lưu trong biến body('Run\_script')['result'].

**Bước 2: Phân Tích JSON**

* Thêm hành động "Parse JSON" để xử lý mảng trả về.
* **Content**: body('Run\_script')['result'].
* **Schema**:

json

CollapseWrapCopy

{

"type": "array",

"items": {

"type": "array",

"items": {

"type": "string"

}

}

}

* Kết quả được lưu trong biến outputs('Parse\_JSON'), là mảng 2 chiều đại diện cho dữ liệu nguồn.

**Bước 3: Xác Định Số Cột**

* Thêm hành động "Initialize variable" để lấy hàng đầu tiên:
  + Name: first\_row.
  + Value: outputs('Parse\_JSON')[0].
* Thêm hành động "Initialize variable" để tính số cột:
  + Name: num\_columns.
  + Value: length(outputs('Parse\_JSON')[0]).
* Điều này giúp xác định có bao nhiêu cột cần xử lý (ví dụ: nếu hàng đầu tiên là ["1", "A", "X"], thì num\_columns là 3).

**Bước 4: Tạo Danh Sách Chỉ Số Cột**

* Thêm hành động "Initialize variable" để tạo danh sách chỉ số cột:
  + Name: column\_indices.
  + Value: @range(0,@sub(num\_columns,-1)).
* Kết quả là một mảng các số nguyên từ 0 đến num\_columns - 1, ví dụ: [0, 1, 2] nếu có 3 cột.

**Bước 5: Xử Lý Mỗi Cột và Dán Vào File Đích**

* Thêm hành động "Apply to each" trên column\_indices.
* Trong mỗi lần lặp (cho mỗi col\_index):
  + **Hành động 1: Lấy Dữ Liệu Cột**
    - Thêm "Compose" với biểu thức @outputs('Parse\_JSON')?[\*][col\_index].
    - Kết quả (body('Compose\_column\_data')) là mảng 1 chiều chứa tất cả giá trị của cột đó, ví dụ: nếu col\_index là 0, kết quả là ["1", "2", "3"].
  + **Hành động 2: Tính Địa Chỉ Ô Bắt Đầu**
    - Thêm "Compose" với biểu thức @concat(char(add(col\_index +65)),'1').
    - Kết quả (body('Compose\_startCell')) là địa chỉ ô bắt đầu, ví dụ: col\_index=0 cho "A1", col\_index=1 cho "B1", v.v. (dựa trên mã ASCII, 'A' là 65).
  + **Hành động 3: Dán Dữ Liệu Vào File Đích**
    - Thêm "Excel Online (Business) - Run script" để chạy PasteColumnData trên file Excel đích.
    - Cấu hình:
      * Location: Vị trí file đích.
      * Document Library: Thư viện chứa file.
      * File: Chọn file Excel đích.
      * Script: Chọn PasteColumnData.
      * Parameters:
        + Name: data, Value: body('Compose\_column\_data').
        + Name: startCell, Value: body('Compose\_startCell').

**Ví Dụ Minh Họa**

Giả sử file nguồn có dữ liệu như sau:

| **Cột A** | **Cột B** | **Cột C** |
| --- | --- | --- |
| 1 | A | X |
| 2 | B | Y |
| 3 | C | Z |

* Sau khi chạy GetColumnData, mảng trả về là [["1", "A", "X"], ["2", "B", "Y"], ["3", "C", "Z"]].
* Power Automate sẽ:
  + Tách cột 0: ["1", "2", "3"], dán vào A1:A3.
  + Tách cột 1: ["A", "B", "C"], dán vào B1:B3.
  + Tách cột 2: ["X", "Y", "Z"], dán vào C1:C3.

Kết quả trong file đích sẽ là:

| **A** | **B** | **C** |
| --- | --- | --- |
| 1 | A | X |
| 2 | B | Y |
| 3 | C | Z |

**Lưu Ý và Tối Ưu Hóa**

* **Vị Trí Dán Linh Hoạt**: Nếu bạn muốn dán vào vị trí khác (ví dụ: cột 1 vào D1, cột 2 vào G1), điều chỉnh biểu thức startCell trong Power Automate, ví dụ: @concat(char(add(col\_index + 68)),'1') để bắt đầu từ "D1" cho col\_index=0.
* **Kiểm Tra Dữ Liệu**: Đảm bảo file đích không bị ghi đè dữ liệu quan trọng và worksheet đích đã được chọn đúng.
* **Hiệu Suất**: Với file lớn, hãy thử nghiệm với dữ liệu nhỏ trước để đảm bảo flow không bị lỗi hoặc chậm.
* **Số Cột Biến Đổi**: Phương pháp này tự động xử lý số cột thay đổi, miễn là dữ liệu đầu vào có cấu trúc đồng nhất.

**Bảng Tóm Tắt Các Bước**

| **Bước** | **Hành Động** | **Ghi Chú** |
| --- | --- | --- |
| 1. Lấy Dữ Liệu | Chạy GetColumnData trên file nguồn | Trả về mảng 2 chiều |
| 2. Phân Tích JSON | Parse JSON với schema phù hợp | Lưu vào parsed\_data |
| 3. Xác Định Số Cột | Tính num\_columns từ hàng đầu tiên | Đảm bảo dữ liệu có cấu trúc |
| 4. Tạo Chỉ Số Cột | Sử dụng @range(0,@sub(num\_columns,-1)) | Tạo danh sách chỉ số cột |
| 5. Xử Lý Mỗi Cột | Lặp qua từng cột, tách dữ liệu, tính vị trí, dán | Dán vào file đích với vị trí tương ứng |

**Kết Luận**

Phương pháp trên cho phép bạn tách từng cột từ mảng dữ liệu Excel và dán vào các vị trí khác nhau trong file đích một cách tự động và linh hoạt. Hãy đảm bảo kiểm tra kỹ lưỡng và điều chỉnh flow nếu cần thiết để phù hợp với yêu cầu cụ thể của bạn.

**Key Citations**

* [Hướng dẫn sử dụng Office Scripts trong Excel](https://support.microsoft.com/en-us/office/using-office-scripts-with-excel-on-the-web-9aee1032-021e-40e3-aa6a-1eb96d74c892)
* [Tài liệu Power Automate với Excel Online](https://learn.microsoft.com/en-us/connectors/excelonlinebusiness/)

Dựa trên những dữ liệu từ cuộc trò chuyện trước. Tối muốn bạn guide cho tôi cách thực hiện với yêu cầu như sau. Tôi có col1,col2,col3,...coln nằm ở vị trí khác nhau trong file gốc, tôi cần copy những col này, sau đó dán vào một file khác có cùng tên với tên col trong file gốc, vị trí col bị thay đổi.

DeepSearch

Thinking

Khám phá yêu cầu

Xử lý quy trình

Thinking

Khám phá yêu cầu

* Yêu cầu là về việc sử dụng Office Script và Power Automate để sao chép dữ liệu từ các cột cụ thể trong file Excel. Sau đó, dán dữ liệu đó vào một file Excel khác, mỗi cột sẽ vào một file riêng, đặt tên theo tên cột.
* Các cột trong file gốc có tên như col1, col2, đến coln, và vị trí của chúng có thể thay đổi. Nhiệm vụ là sao chép từng cột và dán vào các file Excel riêng, mỗi file mang tên cột, ví dụ "col1.xlsx".
* Tôi đang xem xét việc mỗi cột trong file gốc, như "col1", sẽ tạo một file mới "col1.xlsx" và dán dữ liệu vào đó. Phần "vị trí col bị thay đổi" có thể có nghĩa là cột trong file mới không ở cùng vị trí như file gốc.
* Hmm, có vẻ như người dùng muốn mỗi cột, bao gồm cả tiêu đề, được sao chép và dán vào file riêng. Tôi đang nghĩ liệu trong file mới, dữ liệu có nên bắt đầu từ cột A không.
* Mỗi file mới sẽ chỉ có một cột, là dữ liệu từ cột tương ứng trong file gốc. Vì vậy, vị trí của cột trong file mới luôn là cột A, có lẽ không có vấn đề gì.
* Nhưng trước đây, người dùng nói về việc dán tất cả cột vào một file, mỗi cột vào vị trí khác nhau trong file đó. Bây giờ, họ muốn tạo file riêng cho mỗi cột, nên cần điều chỉnh cách tiếp cận.
* Tôi đang tìm hiểu cách Power Automate có thể tạo nhiều file mới. Có vẻ như nó có thể dùng hành động "Create file" trong OneDrive hoặc SharePoint để làm điều này.
* Được rồi, tôi nghĩ cách hiệu quả hơn là dùng Office Script để lấy toàn bộ dữ liệu ban đầu. Sau đó, trong Power Automate, phân tách từng cột dựa trên chỉ số của nó.
* Tôi đang xem xét cách phân tích mảng JSON để lấy hàng đầu tiên làm tên cột. Sau đó, cho mỗi chỉ số cột, trích xuất dữ liệu và tạo file Excel mới mang tên cột đó.
* Trong Power Automate, hành động "Create Excel Online document" dưới "Excel Online (Business)" có thể tạo file Excel mới. Tôi đang nghĩ cách sử dụng nó để tạo từng file cho mỗi cột.
* Tôi đang xây dựng một kịch bản Office Script để lấy dữ liệu cột dưới dạng mảng 1D và dán vào file mới, bắt đầu từ A1. Có vẻ như cần sửa lại để dùng workbook.getActiveWorksheet() thay vì worksheet.
* Khi tạo file Excel Online mới, nó sẽ là một file trống. Sau đó, chạy script trên file này để ghi dữ liệu vào, nhưng có thể có vấn đề về thời gian, vì script cần file được tải đầy đủ trước khi chạy.
* Tôi đang đảm bảo mỗi file mới được đặt tên duy nhất dựa trên tên cột. Giả sử tên cột không có ký tự đặc biệt, mọi thứ sẽ ổn.
* Trong Power Automate, tôi sẽ chạy script "GetAllData" để lấy toàn bộ dữ liệu từ file gốc. Sau đó, lưu kết quả vào biến source\_data để xử lý tiếp.
* Tôi đang nghĩ đến việc dùng "Apply to each" trong Power Automate để lặp qua các chỉ số cột. Sẽ cần lấy tên cột từ hàng đầu tiên và dữ liệu cột từ mảng 2D.

Xử lý quy trình

* Tôi đang lập kế hoạch cho quy trình Power Automate, bắt đầu bằng việc kích hoạt thủ công. Sau đó, chạy script để lấy toàn bộ dữ liệu từ file gốc và lưu vào biến source\_data.
* Khi tạo file mới qua "Create document", nó trả về ID hoặc thông tin để tham chiếu sau. Tôi đang kiểm tra xem có thể dùng thông tin này để chạy script trên file mới không.
* Hành động "Create document" dưới "Excel Online (Business)" trả về đối tượng với thuộc tính như "id", "name", v.v. Tôi đang nghĩ cách lưu output vào biến new\_file để dùng sau.
* Mỗi lần lặp sẽ tạo file riêng và chạy script riêng, không ảnh hưởng đến các lần lặp khác. Tôi nghĩ điều này ổn, miễn là mỗi lần lặp có ngữ cảnh riêng.
* Về việc làm sạch tên file, nếu tên cột có ký tự đặc biệt hoặc khoảng trắng, có thể cần thay thế khoảng trắng bằng dấu gạch dưới. Nhưng hiện tại, tôi giả định tên cột phù hợp để làm tên file.
* Nếu parsed\_data là [["col1", "col2", "col3"], ["1", "A", "X"], ["2", "B", "Y"], ["3", "C", "Z"]], thì @outputs('Parse JSON')?[\*][col\_index] sẽ cho mảng chứa tất cả giá trị cột, bao gồm hàng tiêu đề, phù hợp để dán vào file mới từ A1.
* Tôi tin rằng khi tạo file Excel Online mới, nó sẽ là một file trống. Sau đó, chạy script trên file này sẽ hoạt động như mong đợi, miễn là người dùng có quyền tạo file và chạy script.
* Tôi nhận ra rằng trong bước trước, tôi đã dùng đuôi ".xls" cho file mới, nhưng thường thì file Excel là ".xlsx". Nên sửa lại để dùng ".xlsx" cho định dạng mới hơn.
* Đuôi file ".xls" là cho định dạng Excel 97-2003, trong khi ".xlsx" là cho định dạng từ Excel 2007 trở đi. Vì đang dùng Excel Online, nên ".xlsx" là lựa chọn phù hợp hơn.
* Đuôi ".xls" là cho định dạng Excel 97-2003, còn ".xlsx" là cho Excel 2007 trở đi. Vì đang dùng Excel Online, nên ".xlsx" là lựa chọn phù hợp hơn.
* Tôi đang xác nhận rằng ".xls" là cho định dạng cũ, còn ".xlsx" là cho định dạng mới từ Excel 2007. Vì vậy, trong Power Automate, cần dùng ".xlsx" để đảm bảo tương thích.
* Khi tạo file Excel Online qua Power Automate, có thể chỉ cần tên mà không cần đuôi, hệ thống tự thêm ".xlsx". Tôi đang kiểm tra xem điều này có đúng không để đảm bảo quy trình mượt mà.
* Trong hành động "Create document" của "Excel Online (Business)", chỉ cần cung cấp tên mà không cần đuôi. Hệ thống sẽ tự thêm ".xlsx", tránh tạo file như "col1.xlsx.xlsx" nếu thêm đuôi thủ công.
* Trong Office 365 và OneDrive/SharePoint Online, file Excel mới được tạo với đuôi ".xls" mặc định qua UI hoặc API như Microsoft Graph hoặc Power Automate. Vì vậy, trong "Excel Online (Business) - Create document", chỉ cần tên, hệ thống tự thêm ".xls".
* Nếu đặt File Name là "col1" trong "Create document", thì body('Create document')['name'] sẽ là "col1.xls". Tôi đang xác nhận cách Power Automate xử lý phần mở rộng để tránh nhầm lẫn.
* Tôi đang xem xét liệu Power Automate có cho phép chạy script ngay trên file mới tạo không. Có thể cần thêm hành động "Delay" để đảm bảo file sẵn sàng, ví dụ 5 giây, nhưng nên thử không có delay trước.
* Nếu file gốc không có dữ liệu hoặc chỉ có hàng tiêu đề, mỗi file mới sẽ chỉ có một ô với tên cột. Điều này vẫn chấp nhận được, vì nó là dữ liệu hợp lệ.
* Nếu tên cột có ký tự đặc biệt hoặc không phù hợp làm tên file, cần làm sạch chúng trước khi dùng, như đã đề cập trước đây.
* Với số lượng cột hợp lý, ví dụ lên đến 26, quy trình nên hoạt động tốt. Tôi đang đảm bảo không vượt quá giới hạn thời gian chạy hoặc số lần lặp trong Power Automate.
* Office Scripts cần được bật cho tổ chức và cần quyền phù hợp để chạy qua Power Automate. Tôi đang đảm bảo kết nối được cấu hình đúng trong phần cài đặt Power Automate.
* Đuôi ".xls" là cho định dạng Excel 97-2003, còn ".xlsx" là cho Excel 2007 trở đi. Vì đang dùng Excel Online, nên ".xlsx" là lựa chọn phù hợp hơn.
* Tôi đang nghĩ đến việc kiểm tra xem có cần thêm hành động "Delay" sau khi tạo file mới để đảm bảo file sẵn sàng trước khi chạy script. Có thể thử không có delay trước, nhưng nếu có lỗi, sẽ thêm delay khoảng 5 giây.
* Tôi đang lập kế hoạch cho bước tiếp theo, khởi tạo biến first\_row với @outputs('Parse JSON')[0] để lấy tên cột. Sau đó, khởi tạo num\_columns với @length(outputs('Parse JSON')[0]) để xác định số cột.

**Key Points**

* Nghiên cứu cho thấy bạn có thể sử dụng Office Script và Power Automate để sao chép dữ liệu từ các cột cụ thể trong file Excel gốc và dán vào các file mới, mỗi file được đặt tên theo tên cột (ví dụ: "col1.xlsx" cho cột "col1").
* Có vẻ như quy trình này sẽ tự động hóa việc tạo file mới và dán dữ liệu, nhưng cần thử nghiệm để đảm bảo không có lỗi, đặc biệt với số lượng cột lớn.
* Vị trí cột trong file mới sẽ thay đổi, vì mỗi file chỉ chứa một cột và thường bắt đầu từ cột A.

**Chuẩn Bị**

Trước tiên, bạn cần tạo hai script Office Script:

* **Script "GetAllData"**: Lấy toàn bộ dữ liệu từ file Excel gốc dưới dạng mảng 2 chiều.
* **Script "PasteColumnData"**: Dán dữ liệu cột vào file mới, bắt đầu từ ô A1.

**Thiết Lập Power Automate Flow**

Dưới đây là các bước để thiết lập flow trong Power Automate:

1. **Kích hoạt**: Sử dụng trigger như "Kích hoạt thủ công".
2. **Chạy script "GetAllData"** trên file Excel gốc để lấy dữ liệu.
3. **Phân tích JSON**: Xử lý mảng dữ liệu trả về.
4. **Xác định tên cột**: Lấy hàng đầu tiên để xác định tên cột (col1, col2, v.v.).
5. **Lặp qua từng cột**:
   * Lấy dữ liệu cột, tạo file mới với tên là tên cột (ví dụ: "col1.xlsx").
   * Chạy script "PasteColumnData" để dán dữ liệu vào file mới, bắt đầu từ cột A.

**Lưu Ý**

* Đảm bảo tên cột không có ký tự đặc biệt gây lỗi khi đặt tên file.
* Có thể cần thêm độ trễ nhỏ sau khi tạo file mới để tránh lỗi do đồng bộ hóa.
* Kiểm tra quyền truy cập để tạo file và chạy script trong Power Automate.

**Báo Cáo Chi Tiết**

Dựa trên yêu cầu của bạn, dưới đây là hướng dẫn chi tiết về cách sử dụng Office Scripts và Power Automate để sao chép dữ liệu từ các cột cụ thể (col1, col2, ..., coln) trong file Excel gốc, sau đó dán vào các file mới, mỗi file được đặt tên theo tên cột và vị trí cột có thể thay đổi. Báo cáo này bao gồm tất cả các bước, giải thích và ví dụ để đảm bảo bạn có thể triển khai hiệu quả.

**Giới Thiệu**

Yêu cầu của bạn là sao chép dữ liệu từ các cột có tên col1, col2, ..., coln trong file Excel gốc, với vị trí cột có thể khác nhau, và dán dữ liệu này vào các file Excel mới, mỗi file được đặt tên theo tên cột tương ứng (ví dụ: "col1.xlsx" cho cột "col1"). Trong file mới, vị trí cột sẽ thay đổi, có thể hiểu là mỗi file mới chỉ chứa một cột và thường bắt đầu từ cột A. Phương pháp này sử dụng Office Scripts để xử lý dữ liệu và Power Automate để tự động hóa quy trình tạo file và dán dữ liệu.

**Chuẩn Bị Office Scripts**

Trước tiên, bạn cần tạo hai script trong Office Scripts để xử lý dữ liệu:

1. **Script "GetAllData"** Script này lấy toàn bộ dữ liệu từ phạm vi đã sử dụng trong worksheet và trả về dưới dạng mảng 2 chiều. Ví dụ:

javascript

CollapseWrapCopy

function main(workbook: ExcelScript.WORKBOOK): string[][] {

let sheet = workbook.getActiveWorksheet();

let usedRange = sheet.getUsedRange();

let data = usedRange.getValues();

return data;

}

* + Mảng trả về có dạng như [["col1", "col2", "col3"], ["1", "A", "X"], ["2", "B", "Y"], ["3", "C", "Z"]] nếu file gốc có 3 cột và 3 hàng dữ liệu.

1. **Script "PasteColumnData"** Script này nhận một mảng 1 chiều (dữ liệu của một cột, bao gồm header) và dán vào worksheet mới, bắt đầu từ ô A1. Ví dụ:

javascript

CollapseWrapCopy

function main(workbook: ExcelScript.WORKBOOK; data: string[]) {

let sheet = workbook.getActiveWorksheet();

let range = sheet.getRange("A1").getResizedRange(data.length - 1);

range.setValues(data.map((value) => [value]));

}

* + Ví dụ, nếu data là ["col1", "1", "2", "3"], dữ liệu sẽ được dán vào cột A, bắt đầu từ A1:

| **A** |
| --- |
| col1 |
| 1 |
| 2 |
| 3 |

**Cấu Hình Power Automate Flow**

Để tự động hóa quy trình, bạn cần tạo một flow trong Power Automate với các bước sau:

**Bước 1: Kích Hoạt và Chạy Script Lấy Dữ Liệu**

* Sử dụng trigger "Manually trigger a flow" để thử nghiệm (hoặc trigger phù hợp khác).
* Thêm hành động "Excel Online (Business) - Run script" để chạy GetAllData trên file Excel gốc.
* Cấu hình:
  + Location: Vị trí file Excel gốc (OneDrive hoặc SharePoint).
  + Document Library: Thư viện chứa file.
  + File: Chọn file Excel gốc.
  + Script: Chọn GetAllData.
* Kết quả sẽ được lưu trong biến body('Run\_script')['result'].

**Bước 2: Phân Tích JSON**

* Thêm hành động "Parse JSON" để xử lý mảng trả về.
* **Content**: body('Run\_script')['result'].
* **Schema**:

json

CollapseWrapCopy

{

"type": "array",

"items": {

"type": "array",

"items": {

"type": "string"

}

}

}

* Kết quả được lưu trong biến outputs('Parse\_JSON'), là mảng 2 chiều đại diện cho dữ liệu gốc.

**Bước 3: Xác Định Tên Cột và Số Cột**

* Thêm hành động "Initialize variable" để lấy hàng đầu tiên (chứa tên cột):
  + Name: first\_row.
  + Value: outputs('Parse JSON')[0].
* Thêm hành động "Initialize variable" để tính số cột:
  + Name: num\_columns.
  + Value: length(outputs('Parse JSON')[0]).
* Điều này giúp xác định có bao nhiêu cột cần xử lý (ví dụ: nếu hàng đầu tiên là ["col1", "col2", "col3"], thì num\_columns là 3).

**Bước 4: Tạo Danh Sách Chỉ Số Cột**

* Thêm hành động "Initialize variable" để tạo danh sách chỉ số cột:
  + Name: column\_indices.
  + Value: @range(0,@sub(num\_columns,-1)).
* Kết quả là một mảng các số nguyên từ 0 đến num\_columns - 1, ví dụ: [0, 1, 2] nếu có 3 cột.

**Bước 5: Xử Lý Mỗi Cột và Tạo File Mới**

* Thêm hành động "Apply to each" trên column\_indices.
* Trong mỗi lần lặp (cho mỗi col\_index):
  + **Hành động 1: Lấy Tên Cột và Dữ Liệu Cột**
    - Thêm "Compose" với biểu thức @outputs('first\_row')[col\_index] để lấy tên cột, lưu vào column\_name.
    - Thêm "Compose" với biểu thức @outputs('Parse JSON')?[\*][col\_index] để lấy tất cả dữ liệu của cột đó (bao gồm header), lưu vào column\_data.
    - Ví dụ: Nếu col\_index là 0 và parsed\_data là [["col1", "col2", "col3"], ["1", "A", "X"], ["2", "B", "Y"], ["3", "C", "Z"]], thì column\_data sẽ là ["col1", "1", "2", "3"].
  + **Hành động 2: Tạo File Mới**
    - Thêm "Excel Online (Business) - Create document" để tạo file mới.
    - **Document Library**: Thư viện nơi lưu file mới.
    - **File Name**: @body('Compose\_column\_name') (không cần thêm đuôi ".xlsx", Power Automate sẽ tự động thêm ".xlsx").
    - Kết quả được lưu, ví dụ, nếu column\_name là "col1", file mới sẽ là "col1.xlsx".
  + **Hành động 3: Dán Dữ Liệu Vào File Mới**
    - Thêm "Excel Online (Business) - Run script" để chạy PasteColumnData trên file mới.
    - Cấu hình:
      * Location: Vị trí thư viện chứa file mới.
      * Document Library: Thư viện đã chọn.
      * File: @body('Create document')['name'] (ví dụ: "col1.xlsx").
      * Script: Chọn PasteColumnData.
      * Parameters:
        + Name: data, Value: @body('Compose\_column\_data').

**Ví Dụ Minh Họa**

Giả sử file gốc "data.xlsx" có dữ liệu như sau:

| **A** | **B** | **C** |
| --- | --- | --- |
| col1 | col2 | col3 |
| 1 | A | X |
| 2 | B | Y |
| 3 | C | Z |

* Sau khi chạy flow, Power Automate sẽ:
  + Tạo file "col1.xlsx" với nội dung:

| **A** |
| --- |
| col1 |
| 1 |
| 2 |
| 3 |

* + Tạo file "col2.xlsx" với nội dung:

| **A** |
| --- |
| col2 |
| A |
| B |
| C |

* + Tạo file "col3.xlsx" với nội dung:

| **A** |
| --- |
| col3 |
| X |
| Y |
| Z |

**Lưu Ý và Tối Ưu Hóa**

* **Xử Lý Tên File**: Đảm bảo tên cột không chứa ký tự đặc biệt (như /, \, ?, v.v.) gây lỗi khi đặt tên file. Nếu cần, bạn có thể thêm bước "Compose" để làm sạch tên, ví dụ: thay khoảng trắng bằng dấu gạch dưới với biểu thức @replace(body('Compose\_column\_name'), ' ', '\_').
* **Đồng Bộ Hóa**: Có thể xảy ra lỗi nếu script chạy trước khi file mới được tạo hoàn tất. Để an toàn, thêm hành động "Delay" (ví dụ: 5 giây) sau "Create document" và trước "Run script" để đảm bảo file sẵn sàng.
* **Hiệu Suất**: Với số lượng cột lớn, Power Automate có thể gặp giới hạn về thời gian chạy hoặc số lần lặp. Kiểm tra giới hạn của tổ chức bạn tại [Microsoft Power Automate Limits](https://learn.microsoft.com/en-us/power-platform/admin/limits-allocations).
* **Quyền Truy Cập**: Đảm bảo tài khoản Power Automate có quyền tạo file và chạy script trong thư viện đích.

**Bảng Tóm Tắt Các Bước**

| **Bước** | **Hành Động** | **Ghi Chú** |
| --- | --- | --- |
| 1. Lấy Dữ Liệu | Chạy GetAllData trên file gốc | Trả về mảng 2 chiều |
| 2. Phân Tích JSON | Parse JSON với schema phù hợp | Lưu vào parsed\_data |
| 3. Xác Định Tên Cột | Lấy hàng đầu tiên để xác định tên cột | Đảm bảo dữ liệu có cấu trúc |
| 4. Tạo Chỉ Số Cột | Sử dụng @range(0,@sub(num\_columns,-1)) | Tạo danh sách chỉ số cột |
| 5. Xử Lý Mỗi Cột | Lặp qua từng cột, tách dữ liệu, tạo file, dán | Dán vào file mới với vị trí bắt đầu A1 |

**Kết Luận**

Phương pháp trên cho phép bạn tự động hóa việc sao chép dữ liệu từ các cột cụ thể trong file Excel gốc và dán vào các file mới, mỗi file được đặt tên theo tên cột và chứa dữ liệu cột tương ứng, bắt đầu từ cột A. Hãy đảm bảo kiểm tra kỹ lưỡng và điều chỉnh flow nếu cần thiết để phù hợp với yêu cầu cụ thể, đặc biệt với số lượng cột lớn hoặc tên cột có ký tự đặc biệt.

**Key Citations**

* [Microsoft Power Automate Limits](https://learn.microsoft.com/en-us/power-platform/admin/limits-allocations)
* [Hướng dẫn sử dụng Office Scripts trong Excel](https://learn.microsoft.com/en-us/office/dev/scripts/overview/excel-scripts)
* [Tài liệu Power Automate với Excel Online](https://learn.microsoft.com/en-us/connectors/excelonlinebusiness/)

Bạn hiểu sai câu hỏi. Tôi yêu cầu bạn đọc lại yêu cầu của tôi ở phía dưới. Dựa trên những dữ liệu từ cuộc trò chuyện trước. Tối muốn bạn guide cho tôi cách thực hiện với yêu cầu như sau. Tôi có các tên cột col1,col2,col3,...col thứ n nằm ở vị trí khác nhau trong file gốc, tôi cần copy những col này, sau đó dán vào một file khác có chứa cùng tên cột với tên cột trong file gốc, vị trí cột bị thay đổi trong file mới.

DeepSearch

Thinking

Khám phá yêu cầu

Đánh giá yêu cầu

Xử lý dữ liệu

Xác định định dạng

Xử lý script

Tối ưu hóa quy trình

Thinking

Khám phá yêu cầu

* Yêu cầu là về việc sử dụng Office Script và Power Automate để sao chép dữ liệu từ các cột cụ thể trong file Excel.
* Dữ liệu sẽ được dán vào một file khác, nơi các cột có cùng tên nhưng vị trí có thể thay đổi.
* Tôi đang nghĩ liệu người dùng có muốn sắp xếp lại thứ tự cột trong file mới không, vì họ nói "vị trí cột bị thay đổi".
* Hmm, trước đây, người dùng nói về việc sao chép dữ liệu cột vào mảng, sau đó tách mảng để dán từng cột vào file khác.
* Bây giờ, người dùng nói về việc sao chép các cột col1, col2, v.v. từ file gốc, nơi vị trí cột khác nhau.
* Tôi đang tự hỏi liệu có sự nhầm lẫn nào giữa việc dán vào một file mới với cùng tên cột nhưng vị trí khác, so với việc tách từng cột vào file riêng.
* Tôi đang cân nhắc liệu người dùng có muốn sao chép các cột cụ thể và dán chúng vào file mới với thứ tự khác.
* Có lẽ người dùng muốn sắp xếp lại thứ tự cột, ví dụ, col3 trước, rồi col1, col2, v.v., trong file mới.
* Tôi đang nghĩ rằng người dùng có thể muốn chọn các cột cụ thể và đặt chúng theo thứ tự mới trong file khác.
* Tôi đang tự hỏi liệu họ có muốn tạo file mới với các cột cùng tên, nhưng sắp xếp lại vị trí khi dán dữ liệu từ file gốc không.
* Tôi đang nghĩ rằng người dùng có thể muốn sao chép các cột như col2 và col3, rồi sắp xếp lại vị trí của chúng, ví dụ col2 ở vị trí 1, col3 ở vị trí 2, trong file mới.
* Tôi đang tự hỏi liệu họ có muốn dán dữ liệu col2 vào cột A và col3 vào cột B trong file mới không, vì họ có thể quyết định thứ tự khi tạo file mới.
* Tôi đang nghĩ rằng người dùng có thể muốn biết cách dùng Office Script để sao chép các cột cụ thể, rồi dùng Power Automate để sắp xếp chúng theo thứ tự mong muốn trong file mới.
* Có lẽ người dùng muốn sao chép các cột cụ thể từ file gốc, ví dụ col2 và col3, rồi dán vào file mới với thứ tự khác.
* Tôi đang nghĩ rằng họ có thể muốn sắp xếp lại vị trí cột, như đặt col2 ở vị trí 1 và col3 ở vị trí 2 trong file mới.

Đánh giá yêu cầu

* Người dùng có thể muốn sao chép tất cả các cột nhưng sắp xếp lại thứ tự trong file mới.
* Tôi đang tự hỏi liệu "file khác có chứa cùng tên cột" có nghĩa là file mới đã có cột sẵn hay họ sẽ tạo cột mới.
* Tôi đang nghĩ rằng người dùng có thể muốn xác định thứ tự mới cho các cột cụ thể khi dán vào file mới.
* Tôi đang thử phác thảo giải pháp, bắt đầu bằng việc xác định các cột cần sao chép từ file gốc.
* Script này sẽ trả về một mảng ba chiều, nơi mỗi mảng con tương ứng với dữ liệu của một cột.
* Trong ExcelScript, bạn có thể chỉ định phạm vi để đặt giá trị từ các mảng, giúp sắp xếp cột theo thứ tự mong muốn.
* Power Automate có thể xử lý việc quyết định mảng nào dán vào vị trí nào dựa trên logic hoặc tham số do người dùng định nghĩa.
* Tôi đang nghĩ rằng người dùng có thể định nghĩa danh sách tên cột theo thứ tự mong muốn, ví dụ ["col3", "col2"], để sắp xếp trong file mới.
* Tôi đang nghĩ rằng khi dán vào file mới, người dùng có thể muốn sắp xếp các cột liên tiếp từ A1, không cần lo về vị trí cụ thể.
* Trong Power Automate, sau khi lấy danh sách mảng tương ứng với mỗi cột đã sao chép, người dùng có thể quyết định mảng nào dán vào vị trí cột nào dựa trên thứ tự mong muốn.
* Có thể dùng hành động "Excel Online (Business) - Update range" để đặt giá trị cho từng phạm vi phù hợp.
* Ngoài ra, có thể dùng một Office Script khác, nhận danh sách mảng và chỉ định vị trí dán dựa trên chỉ số hoặc tham số khác.
* Power Automate cho phép truyền tham số vào script, như danh sách mảng, nhưng có thể có hạn chế với cấu trúc dữ liệu phức tạp.
* Tôi đang nghĩ rằng người dùng cần cách sắp xếp mảng theo thứ tự mong muốn trước khi dán vào file mới, có thể cần chuyển đổi dữ liệu để phù hợp.
* Có lẽ cách tốt hơn là script đầu tiên trả về dữ liệu dưới dạng mảng hai chiều, mỗi hàng tương ứng với một hàng trên tất cả các cột được chọn.
* Tôi đang nghĩ rằng người dùng có thể muốn ghép các mảng cột riêng lẻ, ví dụ col2 và col3, thành một mảng hai chiều để dán vào file mới.
* Tôi đang nghĩ rằng người dùng có thể sắp xếp lại các mảng nếu cần, trước khi ghép ngang để tạo mảng hai chiều.
* Script Office lấy danh sách tên cột và trả về danh sách mảng, mỗi mảng là dữ liệu của một cột.
* Tôi đang nghĩ rằng tạo một dictionary ánh xạ tên cột với mảng tương ứng sẽ giúp sắp xếp dễ dàng hơn trong Power Automate.

Xử lý dữ liệu

* Power Automate có thể dùng "Select" hoặc "Compose" để sắp xếp lại các mảng theo thứ tự mong muốn.
* Trong Power Automate, dùng "Apply to each" trên chỉ số để xây dựng mảng kết hợp, ví dụ [["A", "X"], ["B", "Y"], ["C", "Z"]]. Sau đó, dùng "Excel Online (Business) - Update range" để đặt giá trị từ A1 với mảng này.
* Tôi đang nghĩ rằng người dùng có thể muốn dùng "Run script" để lấy danh sách mảng cho các cột đã chọn, rồi sắp xếp lại dựa trên thứ tự mong muốn.
* Trong Power Automate, hành động "Update range" cần chuỗi JSON định dạng mảng hai chiều, nên cần chuyển đổi mảng kết hợp thành định dạng này.
* Trong JSON, mảng hai chiều như [["X","A"], ["Y","B"], ["Z","C"]] được biểu diễn đúng, không phải là mảng ba chiều như [[["X","A"]], [["Y","B"]], [["Z","C"]]].
* Mảng hai chiều trong JSON là [[ "X", "A" ], [ "Y", "B" ], [ "Z", "C" ]], mỗi mảng con đại diện cho một hàng.

Xác định định dạng

* Trong Power Automate, "Update range" yêu cầu chuỗi JSON đại diện cho mảng hai chiều, như [[ "X", "A" ], [ "Y", "B" ], [ "Z", "C" ]].
* Tôi đang nghĩ rằng người dùng có thể dùng biểu thức @string(combined\_data) để chuyển đổi mảng kết hợp thành chuỗi JSON phù hợp.
* Tôi đang thử tóm tắt với ví dụ: file gốc có col2 và col3, muốn dán vào file mới với col3 trước, col2 sau.
* Trong Power Automate, sau khi sắp xếp lại, ghép mảng theo thứ tự mong muốn, ví dụ col3 trước col2, tạo mảng như [["X", "A"], ["Y", "B"], ["Z", "C"]].
* Trong Power Automate, giá trị "Values" cho "Update range" cần chuỗi JSON, ví dụ "[[\"X\",\"A\"],[\"Y\",\"B\"],[\"Z\",\"C\"]]" với chuỗi đã thoát.
* Power Automate có hạn chế với mảng lồng nhau, nhưng dùng biểu thức cẩn thận sẽ quản lý được.
* Script đầu tiên có thể trả về một object, với tên cột làm key và dữ liệu mảng làm value, giúp dễ dàng xử lý trong Power Automate.
* Tôi đang nghĩ rằng người dùng có thể muốn sắp xếp lại các cột dựa trên thứ tự mong muốn, ví dụ ["col3", "col2"], bằng cách tạo danh sách mảng đã sắp xếp.
* Script thứ hai trên file mới có thể nhận danh sách mảng đã sắp xếp và xử lý việc đặt giá trị.
* Tôi đang nghĩ rằng cách sạch hơn là dùng script thứ hai để xử lý logic phức tạp, Power Automate chỉ truyền tham số.

Xử lý script

* Script đầu tiên trên file gốc trả về object với tên cột làm key và dữ liệu mảng làm value.
* Script này giúp dễ dàng ánh xạ tên cột với dữ liệu, hỗ trợ sắp xếp trong Power Automate.
* Tôi đang nghĩ rằng script thứ hai trên file mới sẽ nhận chuỗi JSON của các mảng đã sắp xếp và xử lý việc dán dữ liệu.
* Trong Power Automate, sau khi lấy object từ script đầu tiên, người dùng có thể sắp xếp lại dựa trên thứ tự mong muốn, ví dụ ["col3", "col2"].
* Tôi đang nghĩ rằng cách tốt nhất là dùng script thứ hai để nhận chuỗi JSON và xử lý logic, tránh hạn chế của Power Automate với mảng lồng nhau.
* Script thứ hai nhận chuỗi JSON, phân tích và ghép mảng ngang để dán vào file mới từ A1.
* Tôi đang nghĩ rằng cách này đảm bảo linh hoạt và hiệu quả trong việc quản lý luồng công việc dữ liệu Excel của bạn.
* Script thứ hai trên file mới nhận chuỗi JSON, phân tích và ghép mảng ngang để dán vào file mới từ A1.

Tối ưu hóa quy trình

* Script thứ hai kiểm tra xem orderedArrays có phải là mảng không, nếu không thì báo lỗi.
* Tôi đang nghĩ rằng cách này giúp đảm bảo dữ liệu đầu vào hợp lệ, tránh lỗi khi xử lý.

**Key Points**

* Nghiên cứu cho thấy bạn có thể sử dụng Office Script và Power Automate để sao chép dữ liệu từ các cột cụ thể (col1, col2, ..., coln) trong file Excel gốc, sau đó dán vào một file mới với cùng tên cột, nhưng vị trí cột có thể được sắp xếp lại theo ý muốn.
* Có vẻ như quy trình này sẽ tự động hóa việc sắp xếp lại thứ tự cột, nhưng cần thử nghiệm để đảm bảo không có lỗi, đặc biệt với số lượng cột lớn.
* Vị trí cột trong file mới có thể thay đổi so với file gốc, tùy thuộc vào thứ tự bạn chọn.

**Chuẩn Bị**

* Đảm bảo bạn có quyền truy cập vào file Excel gốc và file mới trong OneDrive hoặc SharePoint.
* Cài đặt Office Scripts trong Excel Online và Power Automate.

**Các Bước Thực Hiện**

**Bước 1: Tạo Office Script để Lấy Dữ Liệu Cột**

* Tạo script "GetColumnData" để lấy dữ liệu từ các cột được chỉ định trong file gốc và trả về dưới dạng một đối tượng, với tên cột làm khóa và dữ liệu cột làm giá trị.  
  Ví dụ:

javascript

CollapseWrapCopy

function main(workbook: ExcelScript.WORKBOOK; columnNames: string[]): { [key: string]: string[][] } {

let sheet = workbook.getActiveWorksheet();

let allData: { [key: string]: string[][] } = {};

for (let name of columnNames) {

let range = sheet.getRangeByColumns(name);

let data = range.getValues();

allData[name] = data;

}

return allData;

}

**Bước 2: Tạo Office Script để Dán Dữ Liệu Kết Hợp**

* Tạo script "PasteCombinedData" để nhận danh sách mảng đã sắp xếp (dưới dạng chuỗi JSON), kết hợp chúng theo chiều ngang và dán vào file mới bắt đầu từ ô A1.  
  Ví dụ:

javascript

CollapseWrapCopy

function main(workbook: ExcelScript.WORKBOOK; jsonOrderedArrays: string) {

let sheet = workbook.getActiveWorksheet();

let orderedArrays = JSON.parse(jsonOrderedArrays);

if (!Array.isArray(orderedArrays)) {

throw new Error("Invalid input: expected an array");

}

let numRows = orderedArrays[0].length;

let numCols = orderedArrays.length;

let combinedData = [];

for (let i = 0; i < numRows; i++) {

let row = [];

for (let j = 0; j < numCols; j++) {

if (orderedArrays[j][i] && orderedArrays[j][i].length > 0) {

row.push(orderedArrays[j][i][0]);

} else {

row.push("");

}

}

combinedData.push(row);

}

let range = sheet.getRange("A1").getResizedRange(numRows - 1, numCols - 1);

range.setValues(combinedData);

}

**Bước 3: Cấu Hình Power Automate Flow**

* **Kích hoạt:** Sử dụng trigger "Kích hoạt thủ công" để thử nghiệm.
* **Hành động 1:** Đặt biến cho columnNames (các cột cần sao chép, ví dụ: ["col2", "col3"]) và desiredOrder (thứ tự mong muốn trong file mới, ví dụ: ["col3", "col2"]).
* **Hành động 2:** Chạy script "GetColumnData" trên file gốc với tham số columnNames.
* **Hành động 3:** Sắp xếp lại mảng dựa trên desiredOrder:
  + Khởi tạo orderedArrays là danh sách rỗng.
  + Sử dụng vòng lặp "For each" trên desiredOrder để thêm dữ liệu tương ứng từ kết quả của hành động 2 vào orderedArrays.
* **Hành động 4:** Chuyển orderedArrays thành chuỗi JSON bằng biểu thức @json(orderedArrays).
* **Hành động 5:** Chạy script "PasteCombinedData" trên file mới với tham số jsonOrderedArrays là chuỗi JSON từ hành động 4.

**Ví Dụ**

* File gốc có:

| **col1** | **col2** | **col3** |
| --- | --- | --- |
| 1 | A | X |
| 2 | B | Y |
| 3 | C | Z |

* Sao chép cột col2 và col3, với thứ tự mong muốn là ["col3", "col2"].
* Kết quả trong file mới sẽ là:

| **A** | **B** |
| --- | --- |
| X | A |
| Y | B |
| Z | C |

**Báo Cáo Chi Tiết**

Dựa trên yêu cầu của bạn, dưới đây là hướng dẫn chi tiết về cách sử dụng Office Scripts và Power Automate để sao chép dữ liệu từ các cột cụ thể (col1, col2, ..., coln) trong file Excel gốc, sau đó dán vào một file mới với cùng tên cột, nhưng vị trí cột có thể được sắp xếp lại theo ý muốn. Báo cáo này bao gồm tất cả các bước, giải thích và ví dụ để đảm bảo bạn có thể triển khai hiệu quả.

**Giới Thiệu**

Yêu cầu của bạn là sao chép dữ liệu từ các cột có tên col1, col2, ..., coln trong file Excel gốc, với vị trí cột có thể khác nhau, và dán dữ liệu này vào một file Excel mới, trong đó file mới cũng có cùng tên cột như file gốc, nhưng thứ tự vị trí của các cột có thể thay đổi so với file gốc. Phương pháp này sử dụng Office Scripts để xử lý dữ liệu và Power Automate để tự động hóa quy trình sắp xếp và dán dữ liệu.

**Chuẩn Bị**

Trước tiên, bạn cần đảm bảo:

* Có quyền truy cập vào file Excel gốc và file mới trong OneDrive hoặc SharePoint.
* Đã cài đặt Office Scripts trong Excel Online và Power Automate.
* File gốc có các cột với tên như col1, col2, v.v., và bạn biết danh sách các cột cần sao chép cũng như thứ tự mong muốn trong file mới.

**Tạo Office Scripts**

Bạn cần tạo hai script trong Office Scripts để xử lý dữ liệu:

1. **Script "GetColumnData"** Script này lấy dữ liệu từ các cột được chỉ định trong file gốc và trả về dưới dạng một đối tượng, với tên cột làm khóa và dữ liệu cột làm giá trị. Ví dụ:

javascript

CollapseWrapCopy

function main(workbook: ExcelScript.WORKBOOK; columnNames: string[]): { [key: string]: string[][] } {

let sheet = workbook.getActiveWorksheet();

let allData: { [key: string]: string[][] } = {};

for (let name of columnNames) {

let range = sheet.getRangeByColumns(name);

let data = range.getValues();

allData[name] = data;

}

return allData;

}

* + Mảng trả về có dạng như { "col2": [["A"], ["B"], ["C"]], "col3": [["X"], ["Y"], ["Z"]] } nếu bạn chọn cột col2 và col3.

1. **Script "PasteCombinedData"** Script này nhận danh sách mảng đã sắp xếp (dưới dạng chuỗi JSON), phân tích chúng, kết hợp theo chiều ngang và dán vào file mới bắt đầu từ ô A1. Ví dụ:

javascript

CollapseWrapCopy

function main(workbook: ExcelScript.WORKBOOK; jsonOrderedArrays: string) {

let sheet = workbook.getActiveWorksheet();

let orderedArrays = JSON.parse(jsonOrderedArrays);

if (!Array.isArray(orderedArrays)) {

throw new Error("Invalid input: expected an array");

}

let numRows = orderedArrays[0].length;

let numCols = orderedArrays.length;

let combinedData = [];

for (let i = 0; i < numRows; i++) {

let row = [];

for (let j = 0; j < numCols; j++) {

if (orderedArrays[j][i] && orderedArrays[j][i].length > 0) {

row.push(orderedArrays[j][i][0]);

} else {

row.push("");

}

}

combinedData.push(row);

}

let range = sheet.getRange("A1").getResizedRange(numRows - 1, numCols - 1);

range.setValues(combinedData);

}

* + Script này sẽ dán dữ liệu kết hợp vào file mới, với mỗi cột được sắp xếp theo thứ tự đã chỉ định.

**Cấu Hình Power Automate Flow**

Để tự động hóa quy trình, bạn cần tạo một flow trong Power Automate với các bước sau:

**Bước 1: Kích Hoạt và Đặt Biến**

* Sử dụng trigger "Manually trigger a flow" để thử nghiệm (hoặc trigger phù hợp khác).
* Thêm hành động "Initialize variable" để đặt columnNames (danh sách cột cần sao chép, ví dụ: ["col2", "col3"]) và desiredOrder (thứ tự mong muốn trong file mới, ví dụ: ["col3", "col2"]).

**Bước 2: Chạy Script Lấy Dữ Liệu**

* Thêm hành động "Excel Online (Business) - Run script" để chạy GetColumnData trên file Excel gốc.
* Cấu hình:
  + Location: Vị trí file Excel gốc (OneDrive hoặc SharePoint).
  + Document Library: Thư viện chứa file.
  + File: Chọn file Excel gốc.
  + Script: Chọn GetColumnData.
  + Parameters: Đặt columnNames là biến columnNames đã khởi tạo.
* Kết quả sẽ được lưu trong biến body('Run\_script')['result'], là một đối tượng với tên cột làm khóa và dữ liệu cột làm giá trị.

**Bước 3: Sắp Xếp Lại Mảng Theo Thứ Tự Mong Muốn**

* Khởi tạo biến orderedArrays là danh sách rỗng.
* Sử dụng hành động "Apply to each" trên desiredOrder:
  + Trong mỗi lần lặp, thêm dữ liệu tương ứng từ body('Run\_script')['result'][current\_item] vào orderedArrays.
  + Ví dụ: Nếu desiredOrder là ["col3", "col2"], thì thứ tự mảng sẽ là dữ liệu của col3 trước, sau đó là col2.

**Bước 4: Chuyển Đổi Thành Chuỗi JSON**

* Sử dụng hành động "Compose" với biểu thức @json(orderedArrays) để chuyển orderedArrays thành chuỗi JSON, lưu vào biến jsonOrderedArrays.

**Bước 5: Dán Dữ Liệu Vào File Mới**

* Thêm hành động "Excel Online (Business) - Run script" để chạy PasteCombinedData trên file mới.
* Cấu hình:
  + Location: Vị trí file mới.
  + Document Library: Thư viện chứa file.
  + File: Chọn file Excel mới (nếu chưa tồn tại, Power Automate sẽ tạo mới).
  + Script: Chọn PasteCombinedData.
  + Parameters: Đặt jsonOrderedArrays là kết quả từ hành động "Compose" ở trên.

**Ví Dụ Minh Họa**

Giả sử file gốc "data.xlsx" có dữ liệu như sau:

| **col1** | **col2** | **col3** |
| --- | --- | --- |
| 1 | A | X |
| 2 | B | Y |
| 3 | C | Z |

* Bạn muốn sao chép cột col2 và col3, với thứ tự mong muốn là ["col3", "col2"].
* Sau khi chạy flow, file mới sẽ có nội dung:

| **A** | **B** |
| --- | --- |
| X | A |
| Y | B |
| Z | C |

**Lưu Ý và Tối Ưu Hóa**

* **Xử Lý Dữ Liệu Thiếu:** Đảm bảo script "PasteCombinedData" xử lý trường hợp mảng rỗng hoặc dữ liệu không đồng nhất bằng cách thêm kiểm tra (như đã có trong script).
* **Hiệu Suất:** Với số lượng cột lớn, Power Automate có thể gặp giới hạn về thời gian chạy hoặc số lần lặp. Kiểm tra giới hạn tại [Microsoft Power Automate Limits](https://learn.microsoft.com/en-us/power-platform/limits-and-configurations).
* **Quyền Truy Cập:** Đảm bảo tài khoản Power Automate có quyền tạo và chỉnh sửa file trong thư viện đích.
* **Kiểm Tra Tên Cột:** Đảm bảo tên cột trong columnNames và desiredOrder khớp với tên cột trong file gốc để tránh lỗi.

**Bảng Tóm Tắt Các Bước**

| **Bước** | **Hành Động** | **Ghi Chú** |
| --- | --- | --- |
| 1. Kích Hoạt và Đặt Biến | Đặt columnNames và desiredOrder | Xác định cột cần sao chép và thứ tự mới |
| 2. Lấy Dữ Liệu | Chạy GetColumnData trên file gốc | Trả về đối tượng với dữ liệu cột |
| 3. Sắp Xếp Lại | Sắp xếp mảng dựa trên desiredOrder | Tạo orderedArrays theo thứ tự mong muốn |
| 4. Chuyển Đổi JSON | Chuyển orderedArrays thành chuỗi JSON | Sẵn sàng để truyền vào script tiếp theo |
| 5. Dán Dữ Liệu | Chạy PasteCombinedData trên file mới | Dán dữ liệu kết hợp vào file mới |

**Kết Luận**

Phương pháp trên cho phép bạn sao chép dữ liệu từ các cột cụ thể trong file Excel gốc, sắp xếp lại thứ tự cột theo ý muốn và dán vào một file mới với cùng tên cột, nhưng vị trí cột được thay đổi. Hãy đảm bảo kiểm tra kỹ lưỡng và điều chỉnh flow nếu cần thiết để phù hợp với yêu cầu cụ thể, đặc biệt với số lượng cột lớn hoặc dữ liệu phức tạp.

**Key Citations**

* [Microsoft Power Automate Limits](https://learn.microsoft.com/en-us/power-platform/limits-and-configurations)

Vẫn là yêu cầu cũ của tôi. Nhưng file mới đã có sẵn tên cột và tôi muốn dán data các cột đã copy từ file gốc qua file mới theo tên cột đã có sẵn trong file mới

DeepSearch

Thinking

Giải thích yêu cầu

Xử lý tham số script

Quản lý lỗi script

Thinking

Giải thích yêu cầu

* Yêu cầu của bạn hơi khó hiểu với tôi. Hãy cùng phân tích từng bước nhé.
* Bạn nói về col1, col2, col3,...coln ở vị trí khác nhau trong file gốc. Có phải bạn muốn nói các cột này trong file Excel gốc không theo thứ tự, như col2 ở cột B, col1 ở cột C, v.v. không?
* Có lẽ bạn muốn copy dữ liệu từ các cột này, được xác định bởi tên như col1, col2, v.v., từ file gốc. Sau đó, dán vào file mới, nơi các cột đã có cùng tên nhưng vị trí có thể khác.
* Tôi đang nghĩ rằng nếu file mới đã có các cột với cùng tên, dữ liệu sẽ tự động khớp đúng cột, bất kể vị trí. Nhưng bạn nhấn mạnh "vị trí cột bị thay đổi", nên có thể bạn muốn dán vào vị trí cụ thể.
* Nếu cả hai file đều có cùng tên cột như col1, col2, v.v., việc copy và paste sẽ dựa trên tên cột. Tôi tự hỏi tại sao vị trí lại quan trọng.
* Có lẽ file mới có cùng tên cột nhưng thứ tự khác, như col2 ở cột A, col1 ở cột B. Dữ liệu từ col1 file gốc sẽ dán vào cột col1 file mới, dù vị trí thay đổi.
* Tôi đang cân nhắc liệu bạn có muốn sắp xếp lại thứ tự cột khi dán không. Có vẻ như vấn đề nằm ở cách Excel xử lý tên cột.
* Có thể tôi đang làm phức tạp vấn đề. Hãy nghĩ khác đi, bạn muốn copy các cột cụ thể từ file gốc và dán vào file mới, nơi các cột đã được thiết lập với cùng tên nhưng thứ tự khác.
* File mới đã có các cột với cùng tên như col1, col2, v.v. Vậy có lẽ bạn chỉ cần copy dữ liệu từ các cột cụ thể trong file gốc và dán vào cột tương ứng trong file mới.
* Nếu đúng như vậy, việc này khá đơn giản: copy dữ liệu từ col1 file gốc và dán vào col1 file mới, và cứ thế tiếp tục. Tôi đang nghĩ liệu bạn có muốn tự động hóa quy trình này không, như dùng Office Scripts và Power Automate.
* Có lẽ bạn muốn một script chạy qua Power Automate để copy các cột cụ thể từ file này sang file khác. Tôi đang nghĩ cách tiếp cận vấn đề này, bắt đầu bằng việc xác định các cột cần copy, như col1, col2, v.v.
* Office Scripts có thể làm việc với file trên OneDrive hoặc SharePoint qua Power Automate. Tôi đang nghĩ dùng Power Automate để chạy script trên các file khác nhau lần lượt.
* Trong Power Automate, khi chạy script trên file bằng "Excel Online (Business) - Run script", nó có thể trả về kết quả để dùng cho hành động tiếp theo. Tôi đang nghĩ tạo hai hành động: một để lấy dữ liệu từ file gốc, một để dán vào file mới.
* Tôi cần đảm bảo cả hai file có cùng tên cột và cấu trúc. Hãy thử phác thảo từng bước, bắt đầu bằng việc tạo script để lấy dữ liệu từ các cột cụ thể trong file gốc.
* Trong Office Scripts, getRangeByColumns(name) trả về một Range object cho cột đó. Sau đó, getValues() sẽ cho mảng 2D, mỗi phần tử là giá trị của hàng trong cột.
* Vì cả hai file có cùng tên cột và cấu trúc, tôi có thể giả định thứ tự columnNames giống nhau giữa hai script. Tôi đang nghĩ tạo script thứ hai để dán dữ liệu vào cột tương ứng trong file mới, dựa trên tên cột.
* Tôi đang nghĩ cách truyền columnNames cho cả hai script trong Power Automate. Hành động "Run script" cho phép truyền tham số, nên tôi có thể truyền mảng tên cột cho script đầu tiên.
* Tôi đang nghĩ định nghĩa columnNames như một biến trong Power Automate trước khi chạy script. Ví dụ, khởi tạo biến columnNames với giá trị ["col1", "col2", "col3"].
* Tôi đang nghĩ về một vấn đề tiềm ẩn. Trong Excel Script, getRangeByColumns(name) cần tên cột là một chuỗi, và setValues mong đợi mảng 2D, mỗi phần tử là giá trị hàng cho cột đó.
* Tôi đang nghĩ liệu cả hai file có header không. Nếu cả hai có header, khi copy dữ liệu, tôi nên chỉ lấy từ hàng thứ hai trở đi, tránh ghi đè header.
* Để loại bỏ header, tôi có thể dùng range.getOffsetRange(1) để lấy dữ liệu từ hàng thứ hai trở đi. Điều này sẽ giúp lấy đúng dữ liệu, không bao gồm hàng tiêu đề.
* Tôi đang nghĩ script đầu tiên sẽ lấy dữ liệu từ hàng thứ hai trở đi, dùng getOffsetRange(1). Script thứ hai cũng cần dán từ hàng thứ hai, đảm bảo không ghi đè header file mới.
* Khi dùng dataRange.setValues(data[i]), dữ liệu sẽ bắt đầu từ hàng thứ hai, phù hợp với số hàng của data[i]. Tôi cần đảm bảo data[i] khớp với cấu trúc của dataRange.
* Nếu số hàng trong hai file không giống nhau sau header, có thể xảy ra vấn đề khi dán dữ liệu không khớp chiều dài. Tôi đang nghĩ xử lý trường hợp file có nhiều hoặc ít hàng hơn.
* Để đơn giản, giả sử cả hai file có đủ hàng hoặc chấp nhận ghi đè hoặc mở rộng khi cần. Tôi đang cân nhắc nếu một cột trong file gốc có nhiều hàng hơn cột tương ứng trong file mới, việc dán có thể gặp vấn đề.
* Dữ liệu dán sẽ ghi đè bất kỳ dữ liệu hiện có trong các hàng đó của các cột trong file mới. Nếu mảng dữ liệu dài hơn số hàng hiện có sau header, các cột sẽ được mở rộng tương ứng.
* Tôi đang nghĩ về cách triển khai trong Power Automate. Đầu tiên, kích hoạt flow thủ công hoặc dùng trigger khác nếu cần. Sau đó, khởi tạo biến columnNames với danh sách tên cột mong muốn.

Xử lý tham số script

* Trong Power Automate, truyền tham số cho script bằng cách dùng @variables('columnNames'). Điều này sẽ tự động xử lý mảng chuỗi cho script đầu tiên.
* Tôi đang nghĩ đảm bảo dataFromOriginal, là string[][][], được truyền đúng cho script thứ hai. Cần kiểm tra cả hai script được cấu hình đúng trong thư viện Office Scripts và có thể truy cập qua Power Automate.
* Nếu tên cột không tồn tại trong file gốc, getRangeByColumns(name) sẽ báo lỗi. Tôi cần kiểm tra trước khi lấy dữ liệu, tránh lỗi không cần thiết.
* Tương tự, khi dán vào file mới, nếu tên cột không tồn tại, cũng sẽ báo lỗi. Phải đảm bảo tất cả tên cột được chỉ định đều có trong cả hai file.
* Office Scripts hỗ trợ try-catch để xử lý lỗi. Tôi đang nghĩ thêm try-catch vào script để kiểm tra cột tồn tại và xử lý lỗi nếu cột không tìm thấy hoặc rỗng.
* Trong script lấy dữ liệu, tôi sẽ thêm try-catch để xử lý lỗi nếu cột không tồn tại. Nếu lỗi xảy ra, tôi sẽ ghi log và bỏ qua cột đó, tiếp tục với cột tiếp theo.

Quản lý lỗi script

* Tôi đang nghĩ thêm try-catch vào script dán dữ liệu để xử lý lỗi nếu cột không tồn tại. Nếu lỗi xảy ra, tôi sẽ ghi log và bỏ qua cột đó, không dừng toàn bộ quy trình.
* Tôi đang nghĩ cách đảm bảo quy trình hoạt động mượt mà, ngay cả khi có lỗi với một vài cột. Điều này sẽ giúp bạn hoàn thành công việc mà không bị gián đoạn.
* Tôi đang nghĩ tạo hai Office Scripts: một để lấy dữ liệu từ các cột cụ thể, loại bỏ header, và một để dán dữ liệu vào file mới từ hàng thứ hai trở đi.
* Trong Power Automate, tôi sẽ khởi tạo biến columnNames, chạy script đầu tiên trên file gốc, sau đó chạy script thứ hai trên file mới, truyền kết quả và tham số cần thiết.
* Tôi đang nghĩ về các trường hợp lỗi tiềm ẩn, như cột không tồn tại hoặc số hàng không khớp. Cần xử lý để quy trình không bị gián đoạn.
* Tôi đang nghĩ cách đảm bảo quy trình hoạt động mượt mà, ngay cả khi có lỗi với một vài cột. Điều này sẽ giúp bạn hoàn thành công việc mà không bị gián đoạn.

**Key Points**

* Nghiên cứu cho thấy bạn có thể sử dụng Office Scripts và Power Automate để sao chép dữ liệu từ các cột cụ thể (col1, col2, ..., coln) trong file Excel gốc và dán vào file mới, nơi file mới đã có sẵn các cột với cùng tên, nhưng vị trí có thể khác.
* Có vẻ như quy trình này sẽ tự động hóa việc dán dữ liệu vào đúng cột dựa trên tên, nhưng cần thử nghiệm để đảm bảo không có lỗi, đặc biệt với số lượng cột lớn hoặc dữ liệu không đồng nhất.

**Hướng Dẫn Tóm Tắt**

**Chuẩn Bị**

* Đảm bảo bạn có quyền truy cập vào cả file Excel gốc và file mới trong OneDrive hoặc SharePoint.
* Cài đặt Office Scripts trong Excel Online và Power Automate.

**Tạo Office Scripts**

1. **Script "GetColumnData"**: Lấy dữ liệu từ các cột được chỉ định trong file gốc, bắt đầu từ hàng thứ 2 (loại bỏ header), và trả về dưới dạng mảng.
   * Ví dụ: Nếu chọn cột "col1" và "col2", script trả về mảng 2D cho mỗi cột, ví dụ [["1"], ["2"], ["3"]] cho cột "col1".
2. **Script "PasteColumnData"**: Nhận dữ liệu từ script trước và dán vào các cột tương ứng trong file mới, bắt đầu từ hàng thứ 2 (giữ header hiện có).

**Cấu Hình Power Automate**

* Khởi tạo biến columnNames với danh sách cột cần sao chép (ví dụ: ["col1", "col2"]).
* Chạy script "GetColumnData" trên file gốc, lưu kết quả.
* Chạy script "PasteColumnData" trên file mới, truyền dữ liệu và danh sách cột.

**Báo Cáo Chi Tiết**

Dựa trên yêu cầu của bạn, dưới đây là hướng dẫn chi tiết về cách sử dụng Office Scripts và Power Automate để sao chép dữ liệu từ các cột cụ thể (col1, col2, ..., coln) trong file Excel gốc và dán vào một file mới, nơi file mới đã có sẵn các cột với cùng tên, nhưng vị trí có thể khác. Báo cáo này bao gồm tất cả các bước, giải thích và ví dụ để đảm bảo bạn có thể triển khai hiệu quả.

**Giới Thiệu**

Yêu cầu của bạn là sao chép dữ liệu từ các cột có tên col1, col2, ..., coln trong file Excel gốc, với vị trí cột có thể khác nhau, và dán dữ liệu này vào một file Excel mới, trong đó file mới đã có sẵn các cột với cùng tên (col1, col2, v.v.), nhưng thứ tự vị trí của các cột có thể thay đổi so với file gốc. Phương pháp này sử dụng Office Scripts để xử lý dữ liệu và Power Automate để tự động hóa quy trình dán dữ liệu vào đúng cột dựa trên tên.

**Chuẩn Bị**

Trước tiên, bạn cần đảm bảo:

* Có quyền truy cập vào file Excel gốc và file mới trong OneDrive hoặc SharePoint.
* Đã cài đặt Office Scripts trong Excel Online và Power Automate.
* File gốc và file mới đều có header (hàng đầu tiên chứa tên cột như col1, col2, v.v.), và file mới đã có sẵn cấu trúc cột với cùng tên.

**Tạo Office Scripts**

Bạn cần tạo hai script trong Office Scripts để xử lý dữ liệu:

1. **Script "GetColumnData"** Script này lấy dữ liệu từ các cột được chỉ định trong file gốc, bắt đầu từ hàng thứ 2 (loại bỏ header), và trả về dưới dạng một mảng 2D cho mỗi cột. Ví dụ:

javascript

CollapseWrapCopy

function main(workbook: ExcelScript.WORKBOOK; columnNames: string[]): string[][][] {

let sheet = worksheet.getActiveWorksheet();

let allData = [];

for (let name of columnNames) {

try {

let range = sheet.getRangeByColumns(name);

if (!range || range.getCellCount() == 0) {

throw new Error(`Column ${name} not found or empty.`);

}

let dataRange = range.getOffsetRange(1); *// Bắt đầu từ hàng 2, loại bỏ header*

let data = dataRange.getValues();

allData.push(data);

} catch (error) {

console.error(`Error getting data for column ${name}: ${error.message}`);

*// Bỏ qua cột này nếu có lỗi*

}

}

return allData;

}

* + Mảng trả về có dạng như [ [["1"], ["2"], ["3"]], [["A"], ["B"], ["C"]] ] nếu bạn chọn cột "col1" với giá trị 1, 2, 3 và cột "col2" với giá trị A, B, C.

1. **Script "PasteColumnData"** Script này nhận mảng dữ liệu từ script trước và danh sách tên cột, sau đó dán dữ liệu vào các cột tương ứng trong file mới, bắt đầu từ hàng thứ 2 (giữ header hiện có). Ví dụ:

javascript

CollapseWrapCopy

function main(workbook: ExcelScript.WORKBOOK; data: string[][][]; columnNames: string[]) {

let sheet = worksheet.getActiveWorksheet();

for (let i = 0; i < data.length; i++) {

let name = columnNames[i];

try {

let range = sheet.getRangeByColumns(name);

if (!range || range.getCellCount() == 0) {

throw new Error(`Column ${name} not found or empty.`);

}

let dataRange = range.getOffsetRange(1); *// Bắt đầu từ hàng 2*

dataRange.setValues(data[i]);

} catch (error) {

console.error(`Error pasting data for column ${name}: ${error.message}`);

*// Bỏ qua cột này nếu có lỗi*

}

}

}

* + Script này đảm bảo dán dữ liệu vào đúng cột dựa trên tên, bất kể vị trí cột trong file mới khác với file gốc.

**Cấu Hình Power Automate Flow**

Để tự động hóa quy trình, bạn cần tạo một flow trong Power Automate với các bước sau:

**Bước 1: Kích Hoạt và Đặt Biến**

* Sử dụng trigger "Manually trigger a flow" để thử nghiệm (hoặc trigger phù hợp khác).
* Thêm hành động "Initialize variable" để đặt columnNames (danh sách cột cần sao chép, ví dụ: ["col1", "col2", "col3"]).

**Bước 2: Chạy Script Lấy Dữ Liệu**

* Thêm hành động "Excel Online (Business) - Run script" để chạy GetColumnData trên file Excel gốc.
* Cấu hình:
  + Location: Vị trí file Excel gốc (OneDrive hoặc SharePoint).
  + Document Library: Thư viện chứa file.
  + File: Chọn file Excel gốc.
  + Script: Chọn GetColumnData.
  + Parameters: Đặt columnNames là biến columnNames đã khởi tạo.
* Kết quả sẽ được lưu trong biến body('Run\_script')['result'], là một mảng 2D cho mỗi cột.

**Bước 3: Dán Dữ Liệu Vào File Mới**

* Thêm hành động "Excel Online (Business) - Run script" để chạy PasteColumnData trên file Excel mới.
* Cấu hình:
  + Location: Vị trí file Excel mới.
  + Document Library: Thư viện chứa file.
  + File: Chọn file Excel mới (đảm bảo file mới đã có sẵn header với cùng tên cột).
  + Script: Chọn PasteColumnData.
  + Parameters:
    - Name: data, Value: body('Run\_script')['result'] (kết quả từ bước trước).
    - Name: columnNames, Value: @variables('columnNames').

**Ví Dụ Minh Họa**

Giả sử file gốc "data.xlsx" có dữ liệu như sau:

| **col1** | **col2** | **col3** |
| --- | --- | --- |
| 1 | A | X |
| 2 | B | Y |
| 3 | C | Z |

Và file mới "new\_data.xlsx" có cấu trúc như sau (với header đã có sẵn):

| **col3** | **col1** | **col2** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

* Bạn muốn sao chép cột col1 và col2 từ file gốc.
* Sau khi chạy flow, dữ liệu sẽ được dán vào file mới như sau:

| **col3** | **col1** | **col2** |
| --- | --- | --- |
|  | 1 | A |
|  | 2 | B |
|  | 3 | C |

Lưu ý rằng cột col3 trong file mới không bị ảnh hưởng vì không được sao chép từ file gốc.

**Lưu Ý và Tối Ưu Hóa**

* **Xử Lý Dữ Liệu Thiếu:** Đảm bảo script xử lý trường hợp cột không tồn tại hoặc rỗng bằng cách ghi log lỗi và tiếp tục (như đã có trong try-catch).
* **Số Hàng Không Đồng Nhất:** Nếu file mới có số hàng khác nhau sau header, dữ liệu sẽ được ghi đè hoặc mở rộng tùy theo kích thước mảng dữ liệu. Đảm bảo kiểm tra trước nếu cần.
* **Hiệu Suất:** Với số lượng cột lớn, Power Automate có thể gặp giới hạn về thời gian chạy. Kiểm tra giới hạn tại [Microsoft Power Automate Limits](https://learn.microsoft.com/en-us/power-platform/limits-and-configurations).
* **Quyền Truy Cập:** Đảm bảo tài khoản Power Automate có quyền chỉnh sửa file trong thư viện đích.

**Bảng Tóm Tắt Các Bước**

| **Bước** | **Hành Động** | **Ghi Chú** |
| --- | --- | --- |
| 1. Kích Hoạt và Đặt Biến | Đặt columnNames với danh sách cột cần sao chép | Xác định cột cần xử lý |
| 2. Lấy Dữ Liệu | Chạy GetColumnData trên file gốc | Trả về mảng dữ liệu từ các cột |
| 3. Dán Dữ Liệu | Chạy PasteColumnData trên file mới | Dán dữ liệu vào cột tương ứng, giữ header |

**Kết Luận**

Phương pháp trên cho phép bạn sao chép dữ liệu từ các cột cụ thể trong file Excel gốc và dán vào file mới, nơi file mới đã có sẵn header với cùng tên cột, bất kể vị trí cột khác nhau. Hãy đảm bảo kiểm tra kỹ lưỡng và điều chỉnh flow nếu cần thiết để phù hợp với yêu cầu cụ thể, đặc biệt với số lượng cột lớn hoặc dữ liệu phức tạp.

**Key Citations**

* [Microsoft Power Automate Limits](https://learn.microsoft.com/en-us/power-platform/limits-and-configurations)

Top of Form