**<a>Tự động hóa Google sheet – Phần 1</a>**

**<p><b>Lọc dữ liệu</b></p>**

<p>Lọc dữ liệu là bước quan trọng đầu tiên trong phân tích dữ liệu. Trong google sheet tính năng này gọi là Filter, để minh họa cụ thể mình sẽ sử dụng một file chứa gần 400 mã cổ phiếu hiển thị tổng tài sản của một công ty theo từng quý của năm</p>

<div class="a">

<img src="pics/file.PNG" alt="Trulli" width="500" height="370">

</div>

<p>File cổ phiếu này gồm 6 cột thông tin, cột một ‘stockcode’ là mã cổ phiếu, cột hai ‘year’ là năm, cột 3 ‘asset1’ là tài sản (Vnd) quý 1, cột 4 ‘asset2’ là tài sản (Vnd) quý 2, cột 5 ‘asset3’ (Vnd) là tài sản quý 3 và ‘asset4’ (Vnd) là tài sản quý 4, sẽ có tổng cộng 3019 dòng dữ liệu được đặt trong 6 cột</p>

<p>Để so sánh tài sản của các công ty theo năm, mình sẽ tiến hành lọc dữ liệu theo từng năm từ 2012 đến 2021 và copy ra từng sheet theo các năm</p>

<p>Một vấn đề đặt ra là sheet chính của file sẽ được update liên tục theo từng quý, có nghĩa nó sẽ được cập nhật khi có dữ liệu quý mới, do đó việc lọc dữ liệu ra các năm theo từng sheet cần được lập trình tự động, khi mọi thứ được tự động việc update dữ liệu theo sheet chính sẽ được thực hiện nhanh chóng và đơn giản, lần sau khi bạn đưa thêm dữ liệu vào sheet chính, các sheet phía sau sẽ được cập nhật theo. Để tự động hóa Google sheet bạn sẽ sử dụng Google app script, app script chạy trên ngôn ngữ javascript, nó tương tự như Visual Basic dùng để lập trình tự động Excel</p>

<p>Trước khi sử dụng App script, trước tiên bạn cần hiểu App script sẽ đọc thông tin từ file Google sheet bằng cách nào? Phần nội dung dưới sẽ giải thích các thành phần cơ bản của 1 bảng tính</p>

<p>Đầu tiên thành phần nhỏ nhất của bảng tính sẽ là value nằm trong các ô, nếu trong 1 ô thì value này được gọi trong App script sẽ là getValue(), nếu nhiều hơn 1 ô thì sẽ được gọi là getValues()</p>

<p>Tiếp theo, value sẽ nằm trong các range, range sẽ được gọi trong app script là getRange(). Có 2 cách để thực hiện getRange, giả sử bạn cần chọn phần range A2:C2 là phần tô xanh như hình dưới, app script sẽ dùng câu lệnh là getRange(‘A2:C2’), hoặc có thể viết lại argument đưa vào với dạng số là getRange(2,1,3,3) với hai thông số đầu 2,1 chỉ vị trí giá trị đầu tiên của Range nằm ở dòng 2 cột 1 là A2, hai thông số tiếp theo 3,3 chỉ dữ liệu có tổng cộng 3 dòng và 3 cột bắt đầu từ cột A2</p>

<div class="a">

<img src="pics/getrange.PNG" alt="Trulli" width="500" height="370">

</div>

<p>Mỗi sheet sẽ chứa các giá trị nằm trong các range, để gọi sheet trong app script dùng cú pháp getSheetByName(‘sheetName’) </p>

<p>Tập hợp nhiều sheet lại với nhau sẽ thành một Spreadsheet, mỗi file Gsheet sẽ có 1 spreadsheet duy nhất. Giả sử mình muốn lấy giá trị của ô “A2” là ‘AAA’, theo trình tự mình sẽ phải kích hoạt spreadsheet, sau đó gọi sheet, xác định range và lấy giá trị</p>

<p>const ss = SpreadsheetApp.getActiveSpreadsheet();</p>

<p>const sheet = ss.getSheetByName('asset\_draft'); </p>

<p>const range = sheet.getRange(2, 1, 1, 1); </p>

<p>var a2Value = range.getValue();</p>

<p>Logger.log(a2Value); </p>

<p>Chạy phần code trên trong app script sẽ cho ra được giá trị của a2Value là ‘AAA’</p>

**<p><b>Tính năng Filter trong App script</b></p>**

<p>Để sử dụng tính năng Filter trong app script, thực hiện trình tự theo các bước</p>

<p>Bước 1: đầu tiên cần gán tính năng Filter cho 1 range bất kỳ, giả sử mình muốn lọc cho vùng Range(A2:C2), thực hiện theo cú pháp sau: </p>

<p>var filter = Range.createFilter()</p>

<p>Hoặc nếu Filter đã tồn tại trước đó, sử dụng cú pháp sau: </p>

<p>var filter = range.getFilter()</p>

<p>Bước 2: sau khi đã xác định được vùng cần Filter, tiếp theo cần thiết lập tiêu chuẩn cho vùng đó sẽ lọc nội dung gì bằng cú pháp newFilterCriteria():</p>

<p>filterCriteria\_scope = SpreadsheetApp</p>

<p>.newFilterCreteria()</p>

<p>.whenTextContain(‘AAA’).build();</p>

<p>Bước 3: Gán tiêu chuẩn ở bước 2 lên vị trí cần lọc, giả sử cần lọc ra giá trị “AAA” trong cột đầu tiên A, khi đó cú pháp sẽ là: </p>

<p>Filter.setFilterColumnCriteria(1, filterCriteria\_scope); </p>

<p>Để bỏ lọc, thực hiện cú pháp sau: filter.remove();</p>

**<p><b>Lọc dữ liệu cổ phiếu bằng tính năng Filter trong App script</b></p>**

<p>Tiến hành nhập đoạn code sau vào app script và và cho chạy bằng cách click vào Run, sau khi chạy code Spreadsheet sẽ tạo ra thêm 2 sheet asset2012 và asset2013 chứa toàn bộ thông tin về các mã cổ phiếu theo từng quý của năm 2012 và 2013. Áp dụng tương tự đoạn code trên sẽ lọc thêm được các năm tiếp theo từ 2014 đến 2021, khi đó file Gsheet sẽ có 11 sheet gồm sheet ban đầu và 10 sheet tiếp theo từ năm 2012 đến 2021</p>

<div class="a">

<img src="pics/code.PNG" alt="Trulli" width="500" height="370">

</div>

<p>Giải thích một số cú pháp trong phần code trên: </p>

<p>sheet.getLastRow(): trả về số thứ tự dòng cuối cùng của sheet</p>

<p>sheet.getLastColumn(): trả về số thứ tự cột cuối cùng của sheet</p>

<p>filter = range.getFilter(): gọi tính năng filter trong trường hợp đã set filter trước đó</p>

<p>newRange = sheet.getDataRange(): trả về toàn bộ dữ liệu ở trong range của sheet đang hiển thị trên màn hình, nếu trường hợp này sử dụng getRange().getValues() sẽ trả về toàn bộ dữ liệu bao gồm cả dữ liệu ẩn</p>

<p>.insertSheet(): trả về 1 sheet mới, sheet này sẽ được thêm vào trong Spreadsheet, để đặt tên cho sheet mới này chỉ cần thêm .setName('sheetName') </p>