

Power Up Reports with Google

Tips and Hints

Finding Email Attachments

- Set up a filter in Gmail that automatically adds a label to the emails that will have your reports attached to them.
- Every message in Gmail is part of a thread, even if the thread contains just that message.
 - Once you find the thread or threads that match your label, you then have to loop through each of the messages in that thread to access an attachment.
- There are a couple of steps to accessing the data in the attachment.
 - First grab the attachment as a blob (which is the actual term in Javascript for some "thing" of data).
 - Then use the built-in parseCsv function to turn it into an array of data.

Ranges vs. Arrays

- A range of cells in Google Sheets is defined by where it starts (row and column), as well as the number of rows long and the number of columns wide.
 - Start counting at 1: A range containing just cell A1 would be defined as `getRange(1, 1, 1, 1)`.
 - This is unlike Javascript arrays, which start at 0: The first element in the first "row" of an array would be defined as `array[0][0]`.
- Use Javascript's length function to get around knowing the exact size your range needs to be.
 - `getRange(1, 1, array.length, array[0].length)` means start at row 1, column 1, and then make the range as many rows long as the array has "rows", and as many columns wide as the first "row" of the array (`[0]`) has "columns."

Dates

- Get to know Javascript dates. (https://www.w3schools.com/js/js_dates.asp is a good place to start...)
- To add and subtract dates in Google Apps Script, get the time equivalent first, then create another new date that adds or subtracts the appropriate number of milliseconds:

```
today = new Date();
theDayBeforeYesterday = new Date(today.getTime() - (2 * 24 * 60 * 60 * 1000));
```

- If you want to include date variables as part of your SQL query (like, I want results from yesterday to the day before, whenever the script happens to run), you'll have to do some conversion. Here's a function you can use:

```
function convertToJTS(date) {
  var dateString = Utilities.formatDate(date, '-0400', 'dd-MMM-yyyy');
  return dateString.toUpperCase();
}
```

Security

You may need to allow access to your data source from Google Apps Script's IPs through your firewall. (see Google's JDBC documentation> Other Databases > Creating other database connections:

https://developers.google.com/apps-script/guides/jdbc#other_databases)