Idea to WebService in hours Using Google Apps Script

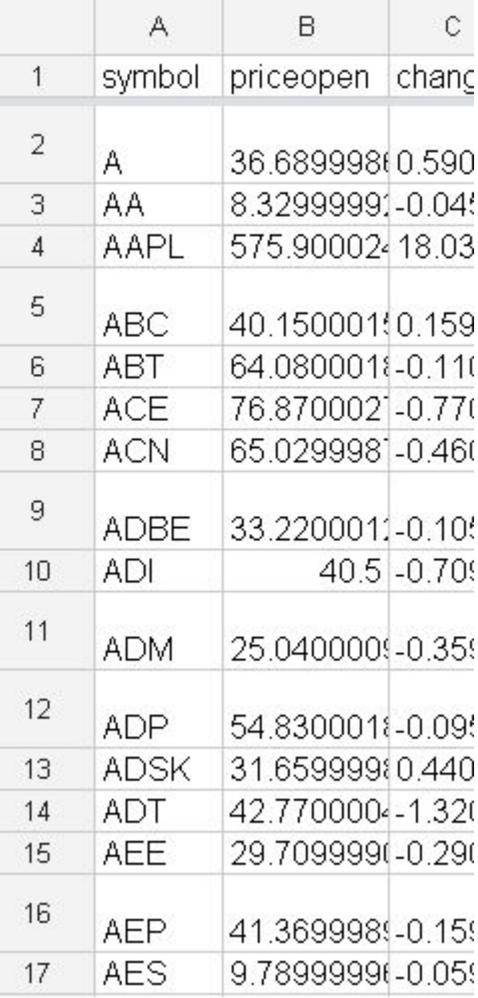
Matt Kaufman
CTO
MK Partners, Inc.

The Idea

Build a web service that will provide information about a publicly traded stock based on its symbol.

Step 1 - Creating the Google Spreadsheet

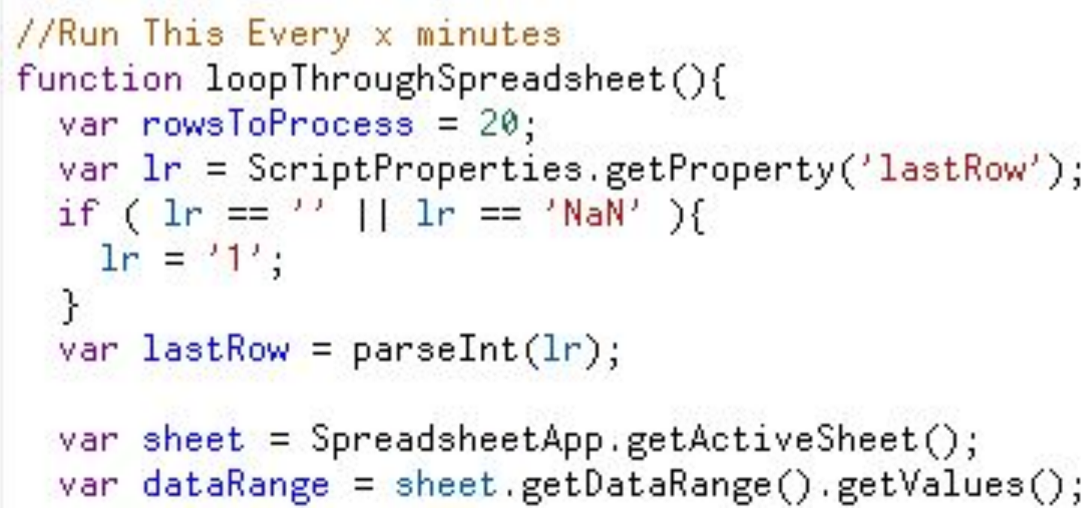
Create a Google Spreadsheet, filling the first column with stock symbols from the S&P 500



	A	B	C
1	symbol	priceopen	chang
2	A	36.689998	0.590
3	AA	8.3299999	-0.045
4	AAPL	575.90002	18.03
5	ABC	40.150001	0.159
6	ABT	64.080001	-0.110
7	ACE	76.870002	-0.770
8	ACN	65.029998	-0.460
9	ADBE	33.220001	-0.105
10	ADI	40.5	-0.709
11	ADM	25.040000	-0.359
12	ADP	54.830001	-0.095
13	ADSK	31.659999	0.440
14	ADT	42.770000	-1.320
15	AEE	29.709999	-0.290
16	AEP	41.369998	-0.159
17	AES	9.7899999	-0.059

Step 2 - Getting the Stock Info from Google

Write a function to loop through the spreadsheet and use Google Apps Script's financeApp class to append the latest stock info for each symbol. Setup a trigger to run the function every x minutes.

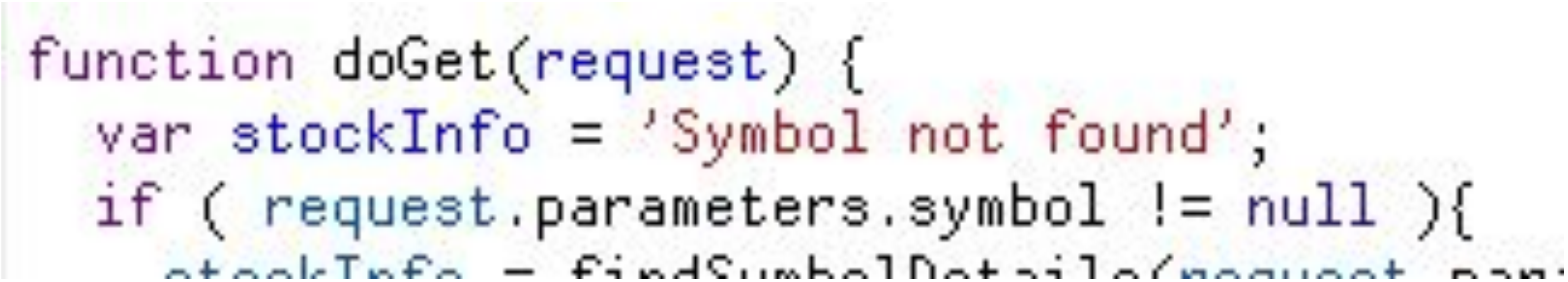


```
//Run This Every x minutes
function loopThroughSpreadsheet(){
  var rowsToProcess = 20;
  var lr = ScriptProperties.getProperty('lastRow');
  if ( lr == '' || lr == 'NaN' ){
    lr = '1';
  }
  var lastRow = parseInt(lr);

  var sheet = SpreadsheetApp.getActiveSheet();
  var dataRange = sheet.getDataRange().getValues();
```

Step 3 - Writing the WebService

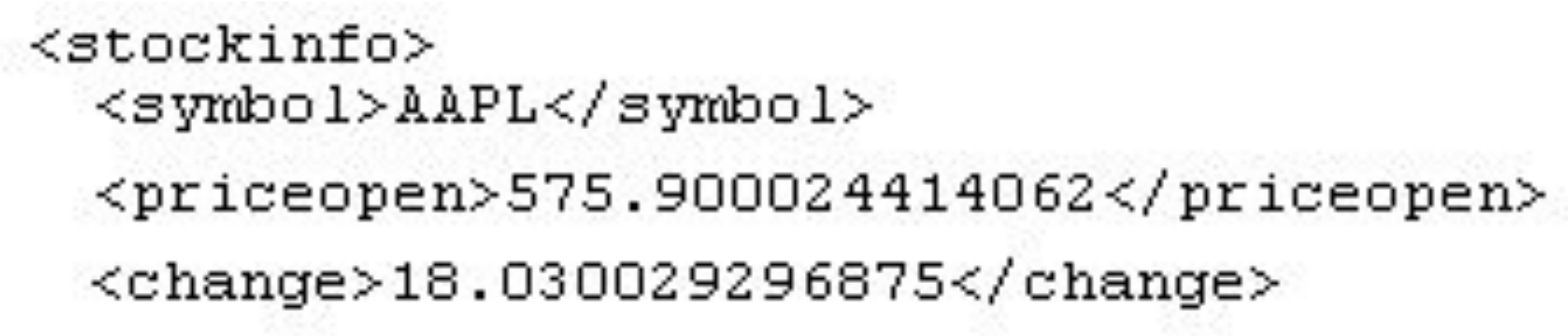
Write a doGet() function that reads the symbol parameter in the URL, gets the corresponding stock info from the spreadsheet and displays it as xml on the page.



```
function doGet(request) {
  var stockInfo = 'Symbol not found';
  if ( request.parameters.symbol != null ){
    stockInfo = findSymbolDetails(request.param
```

Step 4 - Test it out

?symbol=aapl



```
<stockinfo>
  <symbol>AAPL</symbol>
  <priceopen>575.900024414062</priceopen>
  <change>18.030029296875</change>
```