

## Develop highly interactive web charts with SAS®

Rajesh Inbasekaran, Naren Mudivarthy, Neetha Sindhu  
Kavi Associates LLC, Barrington IL

### ABSTRACT

Very often there is a need to present the analysis output from SAS through web applications. On these occasions, it would make a lot of difference to have highly interactive charts over static image charts and graphs. Not only this is visually appealing, with features like zooming, filtering etc., enables consumers to have a better understanding of the output.

There are a lot of charting libraries available in the market which enables us to develop cool charts without much effort. Some of the packages are Highcharts, Highstock, KendoUI etc. They are developed in JavaScript and use the latest HTML5 components and also support a variety of chart types such as line, spline, area, area spline, column, bar, pie, scatter, angular gauges, area range, area spline range, column range, bubble, box plot, error bars, funnel, waterfall and polar chart types etc.

This paper demonstrates how we can combine the data processing and analytic powers of SAS with the visualization abilities of these charting libraries. Since most of them consume JSON formatted data, the emphasis is on JSON producing capabilities of SAS, both with Proc JSON and other custom programming methods. The example would show how easy it is to develop a stored process which produces JSON data which would be consumed by the charting library with minimum change to the sample program.

### INTRODUCTION

In some instances, comparing to traditional static charts generated by SAS, highly dynamic and interactive charts generated by HTML5/JavaScript would be more desirable way of presenting data. There are multiple benefits of using HTML5/JavaScript charts:

- Visually enjoyable. More colorful and dynamic animation effects.
- Real time. It's possible to show real time time-series data.
- Interactive to explore data. It's possible to zoom in to see more detail or zoom out to check the overall trends.
- Time series data can be represented on time slide and zoom graphs

The example of zoom in of time series of data is shown below.

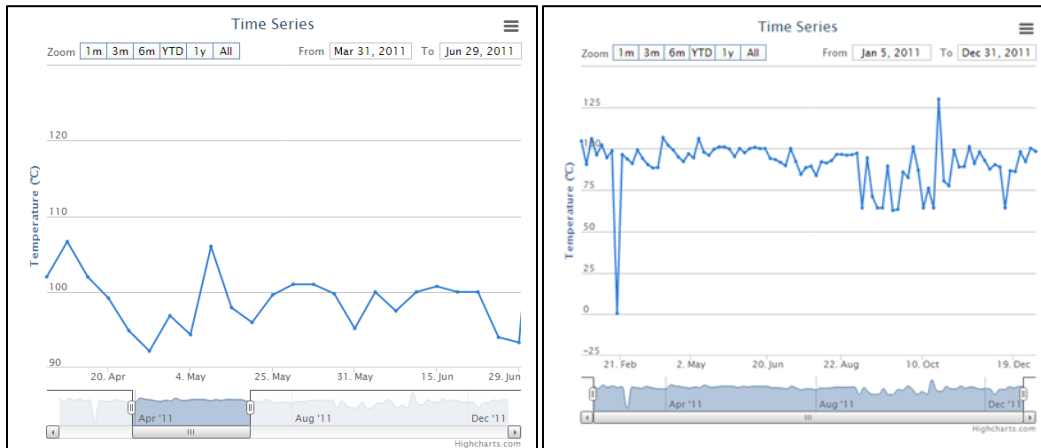


Figure 1. Sample Time Series Highchart

## CONCEPTS OF HTML5 AND JAVASCRIPT

### HTML5

HTML5 is the latest standard for HTML. HTML5 was designed to replace HTML 4, XHTML, and the HTML DOM Level 2. It was specially designed to deliver rich content without the need for additional plugins. The current version delivers everything from animation to graphics, music to movies, and can also be used to build complicated web applications. HTML5 is also cross-platform. It is designed to work whether you are using a PC, or a Tablet, a Smartphone, or a Smart TV. [1]

### JAVASCRIPT

JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It has also become common in server-side programming, game development and the creation of desktop applications. [2]

### JSON FORMAT

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language. [3]

### CASE

We have created a very simple demonstration project that consists of a SAS stored process that returns data in JSON format to the client application in HTML5 which then uses JQuery getJson() method to pass the data in the required format to Highcharts. The highcharts require for data to be given in a certain structured JSON.

We have created a stored process that returns sales price data of product lines over a certain period of time. SAS stores process queries the database, converts the date into the format required by Highcharts and prepares the data to represent the X and Y coordinates of the graph. It then outputs this prepared data in JSON format using our custom JSON builder macro and passes it back to the client HTML5 application.

For the Stock charts, a type of Highcharts which is a time series chart, X axis value will be date which could be represented in two formats as shown below.

1. As a Date.UTC(2014, 2, 17), a method that accepts the date as parameters and returns the number of milliseconds in a Date object since January 1, 1970, 00:00:00, universal time
2. Actual number of milliseconds since January 1, 1970, 00:00:00

SAS® stored process code takes care of converting date into the second format.

### SAS® STORED PROCESS

We have developed custom code to generate JSON object out of the prepared data. This is now possible to accomplish using PROC JSON provided in SAS® 9.4. We developed two macros that is used in this stored process- one to create the JSON object and the other to stream the JSON object as output. Shown below is a code snippet of these macros.

The macro code we used to convert the data into JSON is as shown below.

```
%macro convert_to_json(input_data,series_name,output_data);

proc contents data=&input_data. out=mtemp1 order=varnum noprint;
run;

data mtemp2;
set &input_data.;
ROW_ID=_n_;
run;

proc sql noprint;
select upper(name) into :variables separated by ' ' from mtemp1;
quit;
```

```

proc TRANSPOSE data=mtemp2 out=mtemp3 prefix=Column name=Source
  lable=Label;
  BY ROW_ID;
  VAR &variables.;
quit;

proc sql;
create table mtemp4(drop=varnum) as
select a.*,b.type,b.varnum from mtemp3 a, mtemp1 b
where a.source=b.name
order by b.varnum;
quit;

data mtemp5;
  set mtemp4;
  format values $2000.;
  keep row_id source values;
  if type eq 1 or substr(column1,1,8)="Date.UTC" then
  values=strip(source)||":"||strip(column1);
  else if type eq 2 then
  values=strip(source)||":"||"'"||strip(column1)||"'";
run;

proc sort data=mtemp5;
  by row_id;
run;

data mtemp6(keep=Series_name Json_value);
  set mtemp5;
  length series_name $50.;
  length json_value $2000.;
  retain json_value;
  by row_id;
  series_name="&series_name.";
  if first.row_id then do;
    json_value='{ '||strip(values);
  end;
  else do;
    json_value=strip(json_value)||","||strip(values);
  end;
  if last.row_id then do;
    json_value=strip(json_value)||'}';
    output;
  end;
run;

proc append data=mtemp6 base=&output_data. force nowarn;
run;

%mend convert_to_json;

```

The macro to stream the JSON object as an output is shown below.

```

%macro stream_json_table(input_data);

data _null_;
set &input_data. end=endvar;
file _webout;

if _n_ eq 1 then do;
put "{graphData:[{name:"'"series_name'"', data:[" json_value;
end;

```

```

else do;
put ', ' json_value;
end;

if endvar=1 then do;
put ']]]';
end;

run;

%mend;

```

## STORED PROCESS RESPONSE

The SAS stored process produces the output in JSON format and makes it available as a url service to the client application. Below is the example of the response.

```

{
  "name": "Sales Price ",
  "data":[
    {"y":77.829909772,"x": 1147651200000},
    {"y":77.830027617,"x": 1147737600000},
    {"y":77.829717291,"x": 1147824000000},
    {"y":77.829956448,"x": 1147910400000},
    {"y":77.830036489,"x": 1147996800000},
    {"y":73.905233111,"x": 1148256000000}
  ]
}

```

## JAVASCRIPT

getJSON() is a JQuery method which loads JSON-encoded data from the server using a the url provided by the stored process. This method reads the JSON object from the URL provided and triggers the callback function which then renders the chart with appropriate values passed to the chart.

```

// Set URL variable
Var sasUrl =
"http://xx.xxx.xxx.xx:8080/SASStoredProcess/do?_program=%2FProjects%2FIUN%20Dev%2FStored_Process%2FSP_Price_Plot_Test&_username=*****&_password=*****";

// getJSON JQuery method call to read URL and render Highchart
$.getJSON (sasUrl, function (data) {
  $('#container').highcharts('StockChart', {
    chart: {
      zoomType: 'x'
    },
    yAxis: {
      title: {
        text: 'Temperature (°C)'
      }
    },
    title: {
      text: 'Hourly temperatures in Vik i Sogn, Norway, 2004-2010'
    },
  },

```

```

        subtitle: {
text: 'Built chart at...'
        },
        plotOptions: {
            series: {
                turboThreshold: 50000
            }
        },
        series: [data]
    });
});

```

The below HTML5 code snippet shows how the Highcharts renders a graph to the User Interface via the DIV element.

```

<body>
<div id="container" style="height: 500px; min-width: 500px"></div>
</body>

```

## CONCLUSION

The integration of SAS and these very flexible, robust, interactive charts becomes very easy with the usage of SAS stored process and JSON objects using HTML5 and javascript.

## REFERENCES

[1] [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)

[2] <http://en.wikipedia.org/wiki/JavaScript>

[3] <http://json.org/>

## ACKNOWLEDGMENTS

We would like to thank Kavi Associates LLC to provide us with all the necessary infrastructure and resources to be able to strengthen our skillset in SAS. We are also grateful to SAS for giving us this opportunity to present and publish our work and learnings.

## CONTACT

Please reach out to us for any comments, feedback or questions.

Name: Rajesh Inbasekaran  
Enterprise: Kavi Associates LLC  
1250 S Grove Ave Suite 300  
Barrington, IL - 60010  
Work Phone: (847)387-6760 [extn: 234]  
E-mail: [rajesh.inbasekaran@kaviglobal.com](mailto:rajesh.inbasekaran@kaviglobal.com)

Name: Naren Mudivarthy  
Enterprise: Kavi Associates LLC  
1250 S Grove Ave Suite 300  
Barrington, IL - 60010  
Work Phone: (847)387-6760  
E-mail: [naren.mudivarthy@kaviglobal.com](mailto:naren.mudivarthy@kaviglobal.com)

Name: Neetha Sindhu  
Enterprise: Kavi Associates LLC  
1250 S Grove Ave Suite 300  
Barrington, IL - 60010  
Work Phone: (847)387-6760 [extn: 201]  
E-mail: [neetha.sindhu@kaviglobal.com](mailto:neetha.sindhu@kaviglobal.com)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.