---

Title: BEST GAME TO STREAM

Date: 2016-11-3

Author: Tyler

---

The goal of this project it to:

1. Practice consuming a rest api (or json file) with node
2. Practice building, sorting, and cycling through arrays
3. Figure out which game has the best ratio of viewers to streamers on twitch

To acomplish that we are going to build a node app. So lets get started!

First we need to include our require documents and set up objects in the global scope. I am using FS for things on the file system or request for when I need to make an api request

var fs = require("fs");

var request = require('request');

var contents = {}

Next we want to define the function that "does the work." The first step of the function is to take contents that are passed into it, parses them to json. Also included as look up the word paramaters as the things passed into JS functions paramaters for this fuction are channelMin and viewerMin. I use these to filter out the games where one popular streamer is playing an obscure game and slightly altering the data. I originally didn't have these catches in place and "Gothic II" was the best game to play because rocketbeans was streaming it. I then had them as hard coded but I thought it could be a valid use case where you want to sweep up all the people who realize they like Gothic 2 after rocket beans calls it a day. We loop through every game in the list and if it meets the criteria of the channelMin and viewerMin and add them to a games array.

Once in the games array we apply a sort to them. This spesific sort ranks them by their viewer to channel ratio. Sort is an ES6 function that works by taking a function with two paramaters and applying them against eachother over and over again.

After we have a list of games that meet our criteria and have them sorted we take the top 10 and add them to a string we return so it can be printed to the screen. My original thought was to make this console log directly from with in the function, but if I want to reuse this later having it return a string is a step in the right direction. A sting isn't really reusable, but if I need to have it dump out an array, or an object its easier to build around this string then it consol logging directly from within the function.

var bestGamesToStream = function(contents, channelMin, viewerMin){

var jsonContent = JSON.parse(contents);

var games = []

for(var i = 0; i < jsonContent.top.length; i++){

if(jsonContent.top[i].channels > channelMin && jsonContent.top[i].viewers > viewerMin){

games.push({

"name": jsonContent.top[i].game.name,

"viewersPerChannel": Math.floor(jsonContent.top[i].viewers / jsonContent.top[i].channels)

})

}

}

games.sort(function (a, b) {

if (a.viewersPerChannel < b.viewersPerChannel) {

return 1;

}

if (a.viewersPerChannel > b.viewersPerChannel) {

return -1;

}

// a must be equal to b

return 0;

});

output = ""

for(var i = 0; i < 10; i++){

game = games[i];

output += game.name + "-" + game.viewersPerChannel + "\n"

}

return output

}

The last thing we need to do is pull our data from either a file or directly from the API and pass it to the function we do that in one of two ways.

From a file:

var fileData = fs.readFileSync("twitchdata.json");

console.log(bestGamesToStream(fileData, 4, 4000))

From the API:

request('https://api.twitch.tv/kraken/games/top?limit=100&offset=0', function (error, response, body) {

if (!error && response.statusCode == 200) {

console.log(bestGamesToStream(body, 4, 4000))

}

})

So when we pull it all togeather it looks something like this:

var fs = require("fs");

var request = require('request');

var contents = {}

var bestGamesToStream = function(contents, channelMin, viewerMin){

var jsonContent = JSON.parse(contents);

var games = []

for(var i = 0; i < jsonContent.top.length; i++){

if(jsonContent.top[i].channels > channelMin && jsonContent.top[i].viewers > viewerMin){

games.push({

"name": jsonContent.top[i].game.name,

"viewersPerChannel": Math.floor(jsonContent.top[i].viewers / jsonContent.top[i].channels)

})

}

}

games.sort(function (a, b) {

if (a.viewersPerChannel < b.viewersPerChannel) {

return 1;

}

if (a.viewersPerChannel > b.viewersPerChannel) {

return -1;

}

// a must be equal to b

return 0;

});

output = ""

for(var i = 0; i < 10; i++){

game = games[i];

output += game.name + "-" + game.viewersPerChannel + "\n"

}

return output

}

var fileData = fs.readFileSync("twitchdata.json");

console.log(bestGamesToStream(fileData, 4, 4000))

Not exactly the most complicated bit of code, but it solves a problem, no matter how contrived that problem is.