//open new file – updateCSV.js

//node -v

//npm -v

//terminal – node updateCSV.js

//node updateCSV.js .... in order to keep updating the CSV file

//npm init - in order to define Papaparse and tell nodejs from where to get it .... this will create a package where you can store the dependencies – just go with the default

// npm install papaparse –save .... in order to install Papaparse dependency – it’s a third-party library called Papaparse – it will update the package as well

//npm install ..... install all the dependencies required for continues delivery of the project .... the node\_module will hold/store these dependencies

//npm install newman -save .... To install newman runner dependencies

const fs = require('fs'); //import java library for file system. It helps to read and write file

const Papa = require('papaparse'); //how to update the file itself – import java script library to parse a csv file and as well to generate a csv file from java script object ..

const newman = require('newman'); //import newman runner dependency

newman.run({ //newman runner

collection: require('./collection.json'), //collection ... postman collection

reporters: 'cli', // reporter is cli

iterationData: './data.csv' //data.cvs .... its the cvs file where the data is stored

}, (error) => { //error handling method

if (error) {

throw error;

}

console.log('Collection run complete.'); //to see the test result in cli/terminal as well for each iteration and assertion

}).on('beforeDone', (error, data) => { //beforeDone ... before it writes back to csv file, will show the test results in terminal

if (error) {

throw error;

}

const findFailures = (a, c) => {

return a && (c.error === null || c.error === undefined);

}

const result = data.summary.run.executions.reduce((a, c) => {

if (a[c.cursor.iteration] !== 'FAILED') {

a[c.cursor.iteration] = c.assertions.reduce(findFailures, true) ? 'PASSED' : 'FAILED'; //go thru each assertion in the test section of the collection and if any of the asssertion failed, throw error/failed even if the statusCode is 200

}

return a;

}, []);

updateCsvFile(result); //update the data.updated.csv file with test results

});

function updateCsvFile(result) { //in order to put everything in java script function and update the file with result after the run

fs.readFile('./data.csv', 'utf8', (error, data) => {

//fs.read – to read from csv file

//utf8 to handle special characters

//callback function (//error: to handle the error ....AND.... data: handling the data itself from csv file)

if (error) { //error handling

throw error;

}

const jsonData = Papa.parse(data, { header: true }); //telling the papa parse that the first line is not the data but just the header

jsonData.data.map((item, index) => item.result = result[index]); //mapping the data by index wise .... go thru each line/row and run all the assertions and results back in same order

const updatedCsv = Papa.unparse(jsonData.data); //generate a new data file with results called "updatedCsv"

console.log(updatedCsv); //to see the Passed/Failed results in terminal as well

fs.writeFile('./data-updated.csv', updatedCsv, (error) => { //write the Passed/Failed results back in to the updatedCsv file

if (error) { //error handling

throw error;

}

console.log('Updated CSV file.'); //Updated CSV file. Error will be shown in case of any issue

});

});

}