**json2csv-stream**

Transform json to csv data. The difference to my other module [json2csv](https://github.com/zemirco/json2csv) is json2csv-stream uses streams for transforming the incoming data. The module is built with the new streaming API from Node.js v0.10.0 but maintains backwards compatibility to earlier Node.js versions.

Listen for header and line events or pipe the data directly to a readable stream.

Install with

$ npm install json2csv-stream

**Transform and pipe data to readable stream**

Input - data.json

[

{"car": "Audi","price": 40000,"color": "blue"},

{"car": "BMW","price": 35000,"color": "black"},

{"car": "Mercedes","price": 80000,"color": "red"},

{"car": "Porsche","price": 60000,"color": "green"}

]

Transformation process

var fs = require('fs');

var MyStream = require('json2csv-stream');

// create the transform stream

var parser = new MyStream();

// create the read and write streams

var reader = fs.createReadStream('data.json');

var writer = fs.createWriteStream('out.csv');

reader.pipe(parser).pipe(writer);

Output - out.csv

car,price,color

Audi,40000,blue

BMW,35000,black

Mercedes,80000,red

Porsche,60000,green

**Use header and line events**

If you want to further manipulate your data listen on the custom events.

parser.on('header', function(data) {

console.log(' ++ yeah header found ++');

console.log(data);

});

parser.on('line', function(data) {

console.log(' ++ yeah line found ++');

console.log(data);

});

**Usage**

var MyStream = require('json2csv-stream');

// create the parsing stream with default options

var parser = new MyStream();

// create the stream with custom options. All options are optional.

var parser = new MyStream(options);

The following options are supported

* del: Delimiter for csv values. Default is ,.
* keys: Specify the keys you'd like to output. In the default setting all keys are exported.
* eol: End-of-line marker. Default is the one used by the operating system.
* showHeader: If you don't want the header line in your csv set to false. Default is true.

**Use optional custom delimiter**

The default delimiter is , (comma). If you want to have tab-seperated values \t or semilocon-seperated values ; you can specify an optional delimiter using the del property.

var parser = new MyStream({

del: ';'

});

**Use optional specific keys**

You can specify which key-value pairs you'd like to include in your .csv file. Use the keys property.

var parser = new MyStream({

keys: ['car', 'color']

});

**Use optional end-of-line markers**

The default end-of-line marker is os.eol. That means \n on unix systems and \r\n on windows machines. You can specify your own end-of-line markers with the eol property in the options.

var parser = new MyStream({

eol: '\r\n'

});

**Use without writing the header line**

If you want your csv data without the header line set showHeader to false;

var parser = new MyStream({

showHeader: false

});

**Benchmark**

Go into the /benchmark folder and run

$ node benchmark.js

Results:

Executed benchmark against node module: "json2csv-stream"

Count (34), Cycles (3), Elapsed (6.179 sec), Hz (563.3422353498144 ops/sec)

Executed benchmark against node module: "json2csv"

Count (38), Cycles (5), Elapsed (6.241 sec), Hz (652.5024167610189 ops/sec)

Module: "json2csv" wins.