convert-excel-to-json DT

1.7.0 • Public • Published 3 years ago

Readme

Code Beta

4 Dependencies

55 Dependents

13 Versions

convert-excel-to-json

Convert Excel to JSON, mapping sheet columns to object keys.

Key features:

- Define a specific Range (e.g. `'A1:E6'`)
- Specify a column to key mapping (e.g. `{porperty1: 'CELLVALUE A1', property2: 'CELLVALUE B1'}`)
- Get just specific sheets (e.g. `{sheets: ['sheet1', 'sheet2']}`)

Install

NPM / Node

```
npm install convert-excel-to-json
```

or to use it via command-line

```
npm install -g convert-excel-to-json
```

Usage / Examples

For all the examples, lets suppose that our excel file has two sheets, named as 'sheet1' and 'sheet2'.

CLI

OBS: All the following examples can be used via command-line, in this case, the `--config` parameter expects a valid JSON string.

```
$ convert-excel-to-json --config='{"sourceFile": "tests/test-data.xlsx"}'
```

In order to use it passing in only the **sourceFile** without extra configuration:

```
$ convert-excel-to-json --sourceFile="tests/test-data.xlsx"
```

To check the help section:

```
$ convert-excel-to-json --help
```

Simple conversion

Just gets all the rows, for each sheet, where each row will be represented by an object with a structure like `{ COLUMN: 'CELLVALUE' }` , e.g. from a sheet with only one column (the column A) and two rows `[{A: 'VALUE OF A1'}, {A: 'VALUE OF A2'}]`

```
'use strict';
const excelToJson = require('convert-excel-to-json');

const result = excelToJson({
    sourceFile: 'SOME-EXCEL-FILE.xlsx'
});

// result will be an Object containing keys with the same name as the sheets found on the excel file. Each o
{
    sheet1: [{
        A: 'data of cell A1',
        B: 'data of cell B1',
        C: 'data of cell C1'
    }],
    sheet2: [{
        A: 'data of cell A1',
        B: 'data of cell B1',
        C: 'data of cell C1'
    }]
}
```

Converting an xlsx that you have as a Buffer

```
'use strict';
const excelToJson = require('convert-excel-to-json');
const fs = require('fs');

const result = excelToJson({
    source: fs.readFileSync('SOME-EXCEL-FILE.xlsx') // fs.readFileSync return a Buffer
});

// result will be an Object containing keys with the same name as the sheets found on the excel file. Each o
{
    sheet1: [{
        A: 'data of cell A1',
        B: 'data of cell B1',
        C: 'data of cell C1'
    }],
    sheet2: [{
        A: 'data of cell A1',
        B: 'data of cell B1',
        C: 'data of cell C1'
    }]
}
```

Identifying header rows

You will notice that if your sheet has some top rows setup as a header (it is very common), the first position of our result will have this data, which in this case it should not be very useful. So we can tell the module how many of the rows are headers, so we can skip them and get only the data.

```
'use strict';
const excelToJson = require('convert-excel-to-json');

const result = excelToJson({
    sourceFile: 'SOME-EXCEL-FILE.xlsx',
    header:{
        // Is the number of rows that will be skipped and will not be present at our result object. Counting
        rows: 1 // 2, 3, 4, etc.
    }
});

// result will be an Object like the previous example, but without the rows that was defined as headers
```

Only to specific sheets

Just gets all the rows for each sheet defined on the config object

```
'use strict';
const excelToJson = require('convert-excel-to-json');

const result = excelToJson({
    sourceFile: 'SOME-EXCEL-FILE.xlsx',
    header:{
        rows: 1
    },
    sheets: ['sheet2']
});

// result will be an Object like:
{
    sheet2: [{
        A: 'data of cell A1',
        B: 'data of cell B1',
        C: 'data of cell C1'
    }]
}
```

Mapping columns to keys

One config to all sheets

Gets all the rows, for each sheet, but defining which columns should be returned and how they should be named on the result object.

```
'use strict';
const excelToJson = require('convert-excel-to-json');

const result = excelToJson({
    sourceFile: 'SOME-EXCEL-FILE.xlsx',
    columnToKey: {
        A: 'id',
        B: 'firstName'
    }
```

```
});

// result will be an Object like:
{
    sheet1: [{
        id: 'data of cell A1',
        firstName: 'data of cell B1'
    }],
    sheet2: [{
        id: 'data of cell A1',
        firstName: 'data of cell B1'
    }]
}
```

Config per sheet

Gets all the rows, for each sheet, but defining which columns should be returned and how they should be named on the result object, **per sheet**.

```
'use strict';
const excelToJson = require('convert-excel-to-json');

const result = excelToJson({
    sourceFile: 'SOME-EXCEL-FILE.xlsx',
    sheets:[{
        name: 'sheet1',
        columnToKey: {
                A: 'id',
                B: 'ProductName'
        }
    },{
        name: 'sheet2',
        columnToKey: {
                A: 'id',
                B: 'ProductDescription'
        }
    }]
});

// result will be an Object like:
{
    sheet1: [{
        id: 'data of cell A1',
        ProductName: 'data of cell B1'
    }],
    sheet2: [{
        id: 'data of cell A1',
        ProductDescription: 'data of cell B1'
    }]
}
```

OBS: The config *header.rows* can also be defined per sheet, like in the previous example of *columnToKey*. e.g.

```
{
    sourceFile: 'SOME-EXCEL-FILE.xlsx',
```

```
sheets:[{
    name: 'sheet1',
    header:{
        rows: 1
    },
    columnToKey: {
            A: 'id',
            B: 'ProductName'
    }
},{
    name: 'sheet2',
    header:{
        rows: 3
    },
    columnToKey: {
            A: 'id',
            B: 'ProductDescription'
    }
}]
}
```

Mapping columns to keys :: Special Variables

Cell Variables

A value from a specific cell can be defined as a key name (e.g. `{ A: '{{A1}}' }`). e.g. if we have 3 rows allocated for a header, but the text value is specified at the first row:

```
'use strict';
const excelToJson = require('convert-excel-to-json');

const result = excelToJson({
    sourceFile: 'SOME-EXCEL-FILE.xlsx',
    header:{
        rows: 3
    }
    columnToKey: {
        'A': '{{A1}}',
        'B': '{{B1}}'
    }
});

// result will be an Object like:
{
    sheet1: [{
        THE-VALUE-OF-THE-CELL-A1: 'data of cell A1',
        THE-VALUE-OF-THE-CELL-B1: 'data of cell B1'
    }],
    sheet2: [{
        THE-VALUE-OF-THE-CELL-A1: 'data of cell A1',
        THE-VALUE-OF-THE-CELL-B1: 'data of cell B1'
    }]
}
```

OBS: {{columnHeader}} will follow the config *header.rows* or, in case it is not specified, it will always treat the first row as a header.

Automatic key/property naming following the column header {{columnHeader}}

To return all the data but having the object keys named as a row header found at the excel, instead of the column letters, is just use two special configs. Check the following *columnToKey*:

```
'use strict';
const excelToJson = require('convert-excel-to-json');

const result = excelToJson({
    sourceFile: 'SOME-EXCEL-FILE.xlsx',
    columnToKey: {
        '*': '{{columnHeader}}'
    }
});

// result will be an Object like:
{
    sheet1: [{
        THE-VALUE-OF-THE-HEADER-CELL-A1: 'data of cell A1',
        THE-VALUE-OF-THE-HEADER-CELL-B1: 'data of cell B1',
        THE-VALUE-OF-THE-HEADER-CELL-C1: 'data of cell C1'
    }],
    sheet2: [{
        THE-VALUE-OF-THE-HEADER-CELL-A1: 'data of cell A1',
        THE-VALUE-OF-THE-HEADER-CELL-B1: 'data of cell B1',
        THE-VALUE-OF-THE-HEADER-CELL-C1: 'data of cell C1'
    }]
}
```

OBS: {{columnHeader}} will follow the config *header.rows* or, in case it is not specified, it will always treat the first row as a header.

Range

A specific range can be defined. Also like the previous configs, for all the sheets or per sheet.

One Range for all sheets

```
'use strict';
const excelToJson = require('convert-excel-to-json');

const result = excelToJson({
    sourceFile: 'SOME-EXCEL-FILE.xlsx',
    range: 'A2:B3',
    sheets: ['sheet1', 'sheet2']
});

// result will be an Object like:
{
    sheet1: [{
        A: 'data of cell A2',
        B: 'data of cell B2'
    },{
        A: 'data of cell A3',
        B: 'data of cell B3'
    }],
    sheet2: [{
        A: 'data of cell A2',
```

```
        B: 'data of cell B2'
    },{
        A: 'data of cell A3',
        B: 'data of cell B3'
    }]
}
```

A Range per sheet

```
'use strict';
const excelToJson = require('convert-excel-to-json');

const result = excelToJson({
    sourceFile: 'SOME-EXCEL-FILE.xlsx',
    sheets: [{
        name: 'sheet1',
        range: 'A2:B2'
    },{
        name: 'sheet2',
        range: 'A3:B4'
    }]
});

// result will be an Object like this:
{
    sheet1: [{
        A: 'data of cell A2',
        B: 'data of cell B2'
    ],
    sheet2: [{
        A: 'data of cell A3',
        B: 'data of cell B3'
    },{
        A: 'data of cell A4',
        B: 'data of cell B4'
    }]
}
```

Keywords

excel to json converts excel

Install

```
> npm i convert-excel-to-json
```

Repository

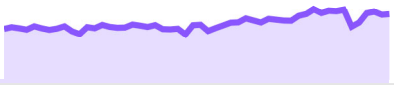
github.com/DiegoZoracKy/convert-excel-to-json

Homepage

github.com/DiegoZoracKy/convert-excel-to-json/

Weekly Downloads

37,112



Version
1.7.0

License
MIT

Unpacked Size
70.3 kB

Total Files
9

Issues
24

Pull Requests
6

Last publish
3 years ago

Collaborators

>_Try on RunKit

Report malware

Support

Help

Advisories

Status

Contact npm

Company

About

Blog

Press

Terms & Policies

Policies

Terms of Use

Code of Conduct

Privacy