

WRITING DATA IN EXCEL FILE

for documentation reference

<https://www.npmjs.com/package/excel4node>

JS FirstExcelFile.js X

Javascript > Lecture 28 > JS FirstExcelFile.js > ...

```
1 // node FirstExcelFile.js --source=teams.json --dest=teams.csv
2 // npm install excel4node
3 // npm install minimist
4 // npm init
5
6 let minimist = require("minimist");
7 let fs = require("fs");
8 let excel = require("excel4node");
9
10 let args = minimist(process.argv);
11 console.log(args.source);
12 console.log(args.dest);
```

library used
→ excel4node

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 28 git:(main) ✘ node FirstExcelFile.js --source=teams.json --dest=teams.csv
teams.json
teams.csv
→ Lecture 28 git:(main) ✘
```

JS FirstExcelFile.js X

```
1 // node FirstExcelFile.js --source=teams.json --dest=team
2 // npm install excel4node
3 // npm install minimist
4 // npm init
5
6 let minimist = require("minimist");
7 let fs = require("fs");
8 let excel = require("excel4node");
9
10 let args = minimist(process.argv);
11
12 let teamsJSON = fs.readFileSync(args.source, "utf-8");
13 let teams = JSON.parse(teamsJSON);
14
15 console.log(teams[0].matches[0].vs);
```

Conversion of JSON
to JSO

{} teams.json X

```
1 [
2   {
3     "name": "India",
4     "rank": 1,
5     "matches": [
6       {
7         "vs": "England",
8         "result": "Win"
9       },
10      {
11        "vs": "Australia",
12        "result": "Win"
13      }
14    ],
15  },
16  {
17    "name": "Australia",
18    "rank": 3,
19    "matches": [
20      {
21        "vs": "India",
22        "result": "Loss"
23      },
24      {
25        "vs": "England",
26        "result": "Win"
27      }
28    ],
29  }
]
```

{} teams.json X

```
30  {
31    "name": "England",
32    "rank": 2,
33    "matches": [
34      {
35        "vs": "Australia",
36        "result": "Loss"
37      },
38      {
39        "vs": "India",
40        "result": "Loss"
41      }
42    ],
43  }
44 ]
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 28 git:(main) ✘ node FirstExcelFile.js --source=teams.json --dest=teams.csv
England
→ Lecture 28 git:(main) ✘
```

zsh + ↻

EXPLORER

PEPCODING-FJP1... Javascript Lecture 28 > JS FirstExcelFile.js > ...

```

1 // node FirstExcelFile.js --source=teams.json --dest=teams.csv
2 // npm install excel4node
3 // npm install minimist
4 // npm init
5
6 let minimist = require("minimist");
7 let fs = require("fs");
8 let excel = require("excel4node");
9
10 let args = minimist(process.argv);
11
12 let teamsJSON = fs.readFileSync(args.source, "utf-8");
13 let teams = JSON.parse(teamsJSON); // Conversion of JSON to JSO
14
15 let wb = new excel.Workbook(); // Create a new instance of a Workbook class
16 for(let i = 0; i < teams.length; i++){
17     // Add Worksheets to the workbook
18     wb.addWorksheet(teams[i].name);
19 }
20
21 wb.write(args.dest);
22

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```

> Lecture 28 git:(main) ✘ node FirstExcelFile.js --source=teams.json --dest=teams.csv
> Lecture 28 git:(main) ✘

```

zsh +

Javascript > Lecture 28 > JS FirstExcelFile.js > ...

```

1 // node FirstExcelFile.js --source=teams.json --dest=teams.csv
2 // npm install excel4node
3 // npm install minimist
4 // npm init
5
6 let minimist = require("minimist");
7 let fs = require("fs");
8 let excel = require("excel4node");
9
10 let args = minimist(process.argv);
11
12 let teamsJSON = fs.readFileSync(args.source, "utf-8");
13 let teams = JSON.parse(teamsJSON); // Conversion of JSON to JSO
14
15 let wb = new excel.Workbook(); // Create a new instance of a Workbook class
16 for(let i = 0; i < teams.length; i++){
17     // Add Worksheets to the workbook
18     wb.addWorksheet(teams[i].name);
19 }
20
21 wb.write(args.dest);

```

A workbook is whole excel file.

We are adding corresponding team name worksheets in the workbook team.csv

India	Australia	England

teams														
Home		Insert		Draw		Page Layout		Formulas		Data		Review		View
Cut	Paste	Copy	Format	Calibri	12	A	A	=	=	=	Wrap Text	General	Con	
B	I	U									Merge & Centre	0.00	0.00	
												%		
A1	X	V	fx											
1														
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														
18														
19														
20														
21														
22														
23														
24														
25														
26														
27														
28														
29														
30														
31														
32														
33														
34														
35														
36														
37														
38														
39														
40														
41														

```

JS FirstExcelFile.js ✘
J Javascript > Lecture 28 > JS FirstExcelFile.js > sheet
1 // node FirstExcelFile.js --source=teams.json --dest=teams.csv
2 // npm install excel4node
3 // npm install minimist
4 // npm init
5
6 let minimist = require("minimist");
7 let fs = require("fs");
8 let excel = require("excel4node");
9
10 let args = minimist(process.argv);
11
12 let teamsJSON = fs.readFileSync(args.source, "utf-8");
13 let teams = JSON.parse(teamsJSON); // Conversion of JSON to JSO
14
15 let wb = new excel.Workbook(); // Create a new instance of a Workbook class
16 for(let i = 0; i < teams.length; i++){
17   // Add Worksheets to the workbook
18   let sheet = wb.addWorksheet(teams[i].name);
19   sheet.cell(1,1).string("Opponent");
20   sheet.cell(1,2).string("Result");
21 }
22
23 wb.write(args.dest);
24
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
→ Lecture 28 git:(main) ✘ node FirstExcelFile.js --source=teams.json --dest=teams.csv
→ Lecture 28 git:(main) ✘

```

JS FirstExcelFile.js X

Javascript > Lecture 28 > JS FirstExcelFile.js > ...

```

1 // node FirstExcelFile.js --source=teams.json --dest=teams.csv
2 // npm install excel4node
3 // npm install minimist
4 // npm init
5
6 let minimist = require("minimist");
7 let fs = require("fs");
8 let excel = require("excel4node");
9
10 let args = minimist(process.argv);
11
12 let teamsJSON = fs.readFileSync(args.source, "utf-8");
13 let teams = JSON.parse(teamsJSON); // Conversion of JSON to JSO
14
15 let wb = new excel.Workbook(); // Create a new instance of a Workbook class
16 for(let i = 0; i < teams.length; i++){
17     // Add Worksheets to the workbook
18     let sheet = wb.addWorksheet(teams[i].name);
19     sheet.cell(1,1).string("Opponent");
20     sheet.cell(1,2).string("Result");
21
22     for(let j = 0; j < teams[i].matches.length; j++){
23         let vs = teams[i].matches[j].vs; → taking out opponent from object
24         let result = teams[i].matches[j].result; → taking out result from
25
26         sheet.cell(2 + j, 1).string(vs);
27         sheet.cell(2 + j, 2).string(result);
28     }
29 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

→ Lecture 28 git:(main) ✘ node FirstExcelFile.js --source=teams.json --dest=teams.csv
 → Lecture 28 git:(main) ✘

	A	B	C
1	Opponent	Result	
2	England	Win	
3	Australia	Win	
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			

	A	E	C
1	Opponent	Result	
2	India	Loss	
3	England	Win	
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			

	A	B	C
1	Opponent	Result	
2	Australia	Loss	
3	India	Loss	
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			

JS FirstExcelFile.js ×

Javascript > Lecture 28 > JS FirstExcelFile.js > ...

```
1 // node FirstExcelFile.js --source=teams.json --dest=teams.csv
2 // npm install excel4node
3 // npm install minimist
4 // npm init
5
6 let minimist = require("minimist");
7 let fs = require("fs");
8 let excel = require("excel4node");
9
10 let args = minimist(process.argv);
11
12 let teamsJSON = fs.readFileSync(args.source, "utf-8");
13 let teams = JSON.parse(teamsJSON); // Conversion of JSON to JSO
14
15 let wb = new excel.Workbook(); // Create a new instance of a Workbook class
16 for(let i = 0; i < teams.length; i++){
17     // Add Worksheets to the workbook
18     let sheet = wb.addWorksheet(teams[i].name);
19     sheet.cell(1,1).string("Opponent");
20     sheet.cell(1,2).string("Result");
21
22     sheet.cell(1,4).string("Rank");
23     sheet.cell(1,5).number([teams[i].rank]);
24
25     for(let j = 0; j < teams[i].matches.length; j++){
26         let vs = teams[i].matches[j].vs; // taking out opponent from object
27         let result = teams[i].matches[j].result; // taking out result from object
28
29         sheet.cell(2 + j, 1).string(vs);
30         sheet.cell(2 + j, 2).string(result);
31     }
32 }
33
34 wb.write(args.dest);
35
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 28 git:(main) ✘ node FirstExcelFile.js --source=teams.json --dest=teams.csv
→ Lecture 28 git:(main) ✘
```

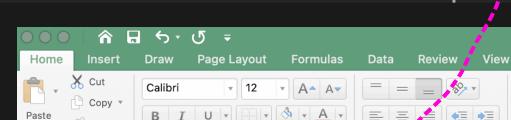
	A	B	C	D	E
1	Opponent	Result	Rank	1	1
2	England	Win			
3	Australia	Win			
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18	India	Australia	England	+	

Styling worksheet

JS FirstExcelFile.js U X

Javascript > Lecture 28 > JS FirstExcelFile.js > ...

```
1 // node FirstExcelFile.js --source=teams.json --dest=teams.csv
2 // npm install excel4node
3 // npm install minimist
4 // npm init
5 let minimist = require("minimist");
6 let fs = require("fs");
7 let excel = require("excel4node");
8 let args = minimist(process.argv);
9 let teamsJSON = fs.readFileSync(args.source, "utf-8");
10 let teams = JSON.parse(teamsJSON); // Conversion of JSON to JSO
11
12 let wb = new excel.Workbook(); // Create a new instance of a Workbook class
13 // Create a reusable style
14 let hstyle = wb.createStyle({ // for styling inside work book <- hstyle (header style)
15     font: {
16         color: "red",
17     },
18     fill: {
19         type: "pattern",
20         patternType: "solid",
21         fgColor: "#00FF00" //foreground color
22     }
23
24 });
25 for(let i = 0; i < teams.length; i++){
26     // Add Worksheets to the workbook
27     let sheet = wb.addWorksheet(teams[i].name);
28     sheet.cell(1,1).string("Opponent").style(hstyle);
29     sheet.cell(1,2).string("Result").style(hstyle);
30
31     sheet.cell(1,4).string("Rank").style(hstyle);
32     sheet.cell(1,5).number(teams[i].rank);
33
34     for(let j = 0; j < teams[i].matches.length; j++){
35         let vs = teams[i].matches[j].vs; // taking out opponent from object
36         let result = teams[i].matches[j].result; // taking out result from object
37         sheet.cell(2 + j, 1).string(vs);
38         sheet.cell(2 + j, 2).string(result);
39     }
40 }
41 wb.write(args.dest);
```



	A	B	C	D	E
1	Opponent	Result		Rank	1
2	England	Win			
3	Australia	Win			
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16	India	Australia	England	+	