



CSV JSON

#Node.js Express Notes



CSV JSON



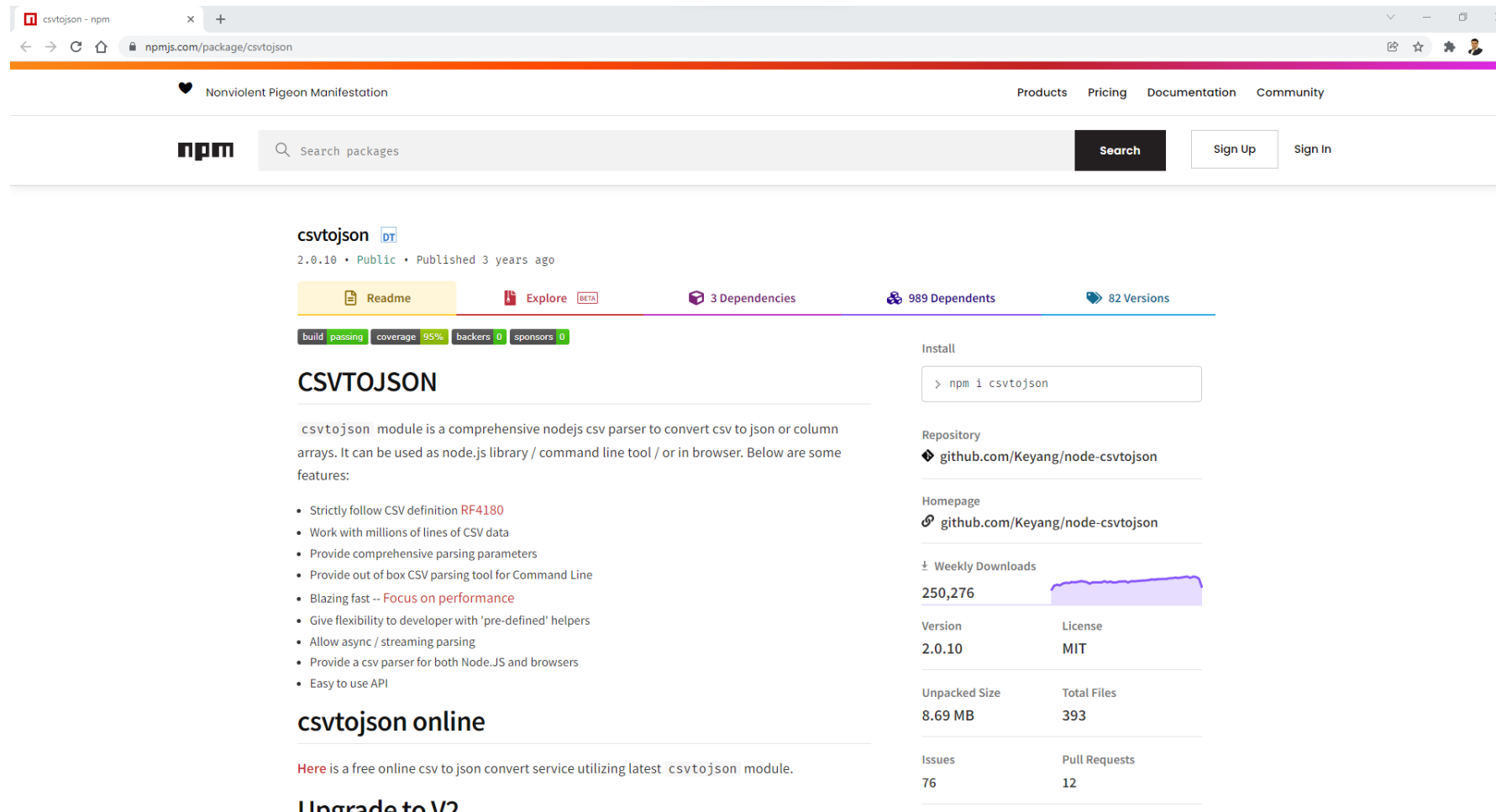
convert CSV to JSON

- CSV (Comma-Separated Values) is a popular data exchange format that stores data in a tabular format where each row consists of one or more fields and each column represents a specific field.
- Due to simplicity and better integration with existing applications, CSV is widely used to export and import large data sets.
- In order to do the CSV to JSON conversion, we'll be using the [csvtojson](#) package from NPM.
- Tool : <https://csv.keyangxiang.com/>



Csvtojson

- <https://www.npmjs.com/package/csvtojson>



The screenshot shows the npm package page for 'csvtojson'. The page header includes the npm logo, a search bar, and links for 'Sign Up' and 'Sign In'. The package name 'csvtojson' is displayed with its version '2.0.10', 'Public' status, and 'Published 3 years ago'. Below this, there are tabs for 'Readme', 'Explore', 'Dependencies', 'Dependents', and 'Versions'. The 'Readme' tab is active, showing the package description: 'csvtojson module is a comprehensive nodejs csv parser to convert csv to json or column arrays. It can be used as node.js library / command line tool / or in browser. Below are some features:'. A list of features follows, including 'Strictly follow CSV definition', 'Work with millions of lines of CSV data', 'Provide comprehensive parsing parameters', 'Provide out of box CSV parsing tool for Command Line', 'Blazing fast -- Focus on performance', 'Give flexibility to developer with 'pre-defined' helpers', 'Allow async / streaming parsing', 'Provide a csv parser for both Node.JS and browsers', and 'Easy to use API'. To the right of the features, there is an 'Install' section with a code block showing 'npm i csvtojson'. Below that, there is a 'Repository' section with a link to 'github.com/Keyang/node-csvtojson'. Further down, there is a 'Homepage' section with a link to 'github.com/Keyang/node-csvtojson'. At the bottom, there is a 'Weekly Downloads' section with a bar chart showing '250,276' downloads. Below this, there is a table with two columns: 'Version' and 'License'. The table shows '2.0.10' and 'MIT'. Below the table, there is another table with two columns: 'Unpacked Size' and 'Total Files'. The table shows '8.69 MB' and '393'. At the bottom, there is a table with two columns: 'Issues' and 'Pull Requests'. The table shows '76' and '12'. At the very bottom, there is a section titled 'csvtojson online' with a link to 'Here is a free online csv to json convert service utilizing latest csvtojson module.' and a link to 'Upgrade to V2'.

csvtojson 2.0.10 • Public • Published 3 years ago

Readme Explore 3 Dependencies 989 Dependents 82 Versions

build passing coverage 95% backers 0 sponsors 0

CSVTOJSON

csvtojson module is a comprehensive nodejs csv parser to convert csv to json or column arrays. It can be used as node.js library / command line tool / or in browser. Below are some features:

- Strictly follow CSV definition [RF4180](#)
- Work with millions of lines of CSV data
- Provide comprehensive parsing parameters
- Provide out of box CSV parsing tool for Command Line
- Blazing fast -- [Focus on performance](#)
- Give flexibility to developer with 'pre-defined' helpers
- Allow async / streaming parsing
- Provide a csv parser for both Node.JS and browsers
- Easy to use API

csvtojson online

[Here](#) is a free online csv to json convert service utilizing latest csvtojson module.

[Upgrade to V2](#)

Install

```
> npm i csvtojson
```

Repository

[github.com/Keyang/node-csvtojson](#)

Homepage

[github.com/Keyang/node-csvtojson](#)

Weekly Downloads

250,276

Version	License
2.0.10	MIT

Unpacked Size	Total Files
8.69 MB	393

Issues	Pull Requests
76	12



Install NPM

- Install csvtojson Package
- <https://www.npmjs.com/package/csvtojson>
- `npm i csvtojson --save`

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

D:\demo>npm i --save csvtojson
```

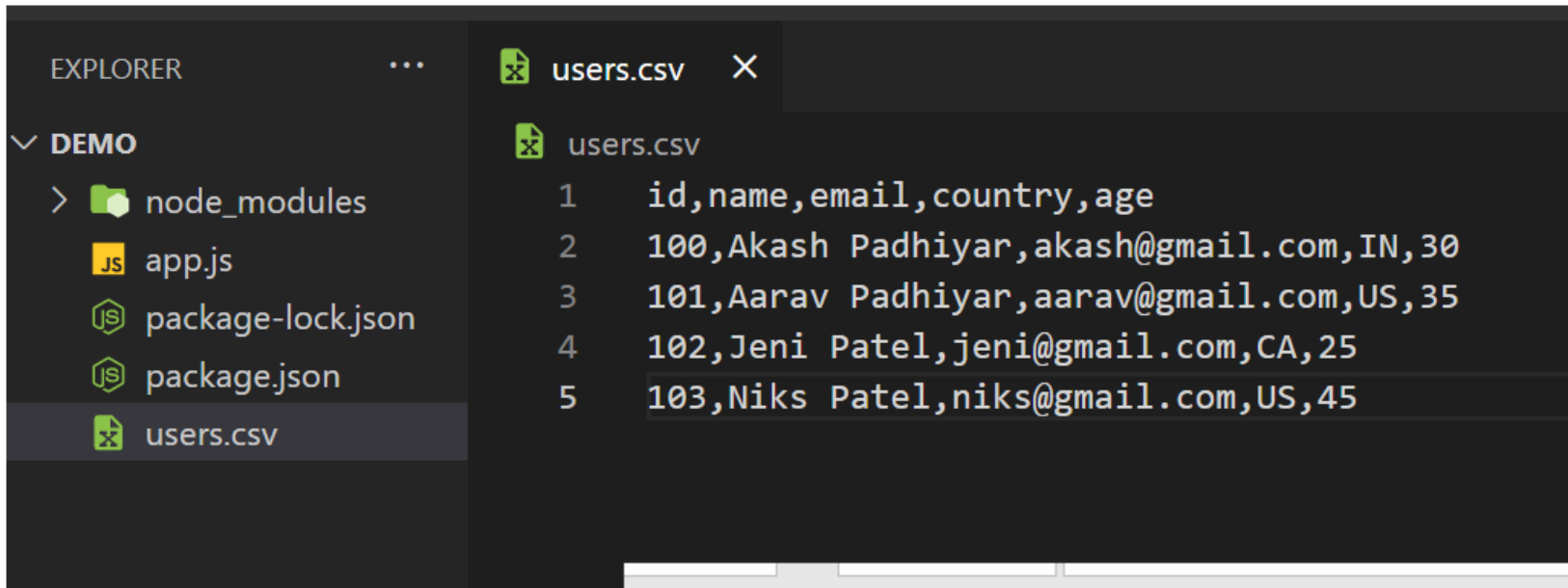


Create CSV File

- create a sample CSV file called **users.csv** with the following content in the root directory:
- id,name,email,country,age
- 100,Akash Padhiyar,akash@gmail.com,IN,30
- 101,Aarav Padhiyar,aarav@gmail.com,US,35
- 102,Jeni Patel,jeni@gmail.com,CA,25
- 103,Niks Patel,niks@gmail.com,US,45



Users.csv



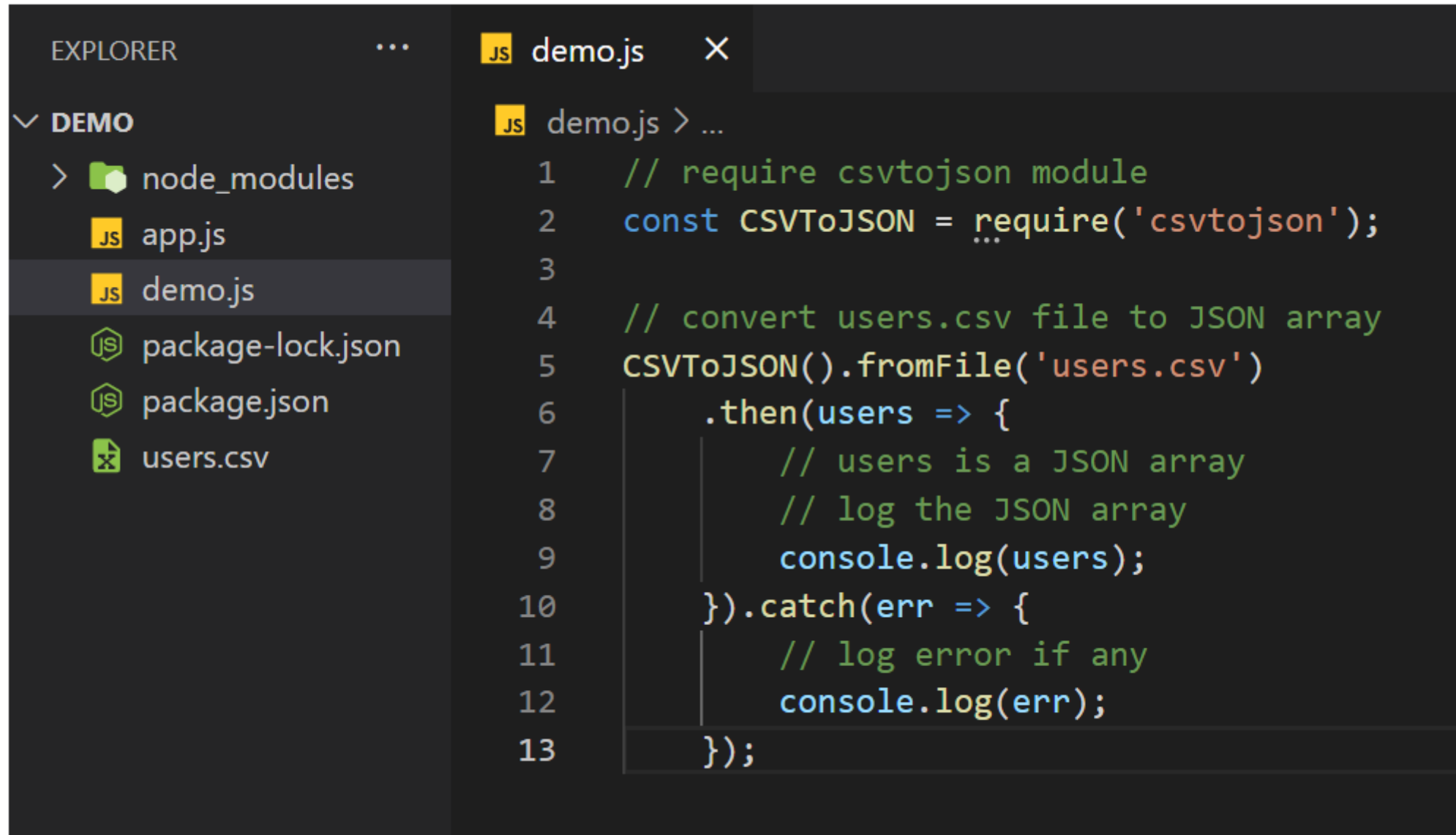
The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays a project structure under 'DEMO' with files: node_modules, app.js, package-lock.json, package.json, and users.csv. The main editor window shows the content of users.csv:

```
1 id,name,email,country,age
2 100,Akash Padhiyar,akash@gmail.com,IN,30
3 101,Aarav Padhiyar,aarav@gmail.com,US,35
4 102,Jeni Patel,jeni@gmail.com,CA,25
5 103,Niks Patel,niks@gmail.com,US,45
```

	A	B	C	D	E
	id	name	email	country	age
	100	Akash Padhiyar	akash@gmail.com	IN	30
	101	Aarav Padhiyar	aarav@gmail.com	US	35
	102	Jeni Patel	jeni@gmail.com	CA	25
	103	Niks Patel	niks@gmail.com	US	45



Code



The screenshot shows the Visual Studio Code interface. On the left, the 'EXPLORER' sidebar displays a project named 'DEMO'. Inside 'DEMO', there is a 'node_modules' folder and several files: 'app.js', 'demo.js' (which is selected), 'package-lock.json', 'package.json', and 'users.csv'. The main editor area shows the content of 'demo.js'. The code is written in JavaScript and uses the 'csvtojson' module to convert a CSV file into a JSON array. The code includes comments for each step, from requiring the module to logging the final JSON array and handling any potential errors.

```
JS demo.js X
JS demo.js > ...
1  // require csvtojson module
2  const CSVToJson = require('csvtojson');
3
4  // convert users.csv file to JSON array
5  CSVToJson().fromFile('users.csv')
6    .then(users => {
7      // users is a JSON array
8      // log the JSON array
9      console.log(users);
10  }).catch(err => {
11    // log error if any
12    console.log(err);
13  });
```


Details

- Example will load a CSV file asynchronously and convert it into a JSON array.
- The `fromFile()` method takes in the path to the CSV file and returns a new promise.
- Once the promise is resolved successfully, the `then()` method is called with the JSON array as an argument.
- If `csvtojson` fails to convert the CSV file to JSON, it will throw an error that will be caught by the `catch()` method.



output

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
D:\demo>node demo.js
[
  {
    id: '100',
    name: 'Akash Padhiyar',
    email: 'akash@gmail.com',
    country: 'IN',
    age: '30'
  },
  {
    id: '102',
    name: 'Jeni Patel',
    email: 'jeni@gmail.com',
    country: 'CA',
    age: '25'
  },
  {
    id: '103',
    name: 'Niks Patel',
    email: 'niks@gmail.com',
    country: 'US',
    age: '45'
  }
]
```



Code

```
// require csvtojson module
const CSVToJSON = require('csvtojson');

// convert users.csv file to JSON array
CSVToJSON().fromFile('users.csv')
  .then(users => {
    // users is a JSON array
    // log the JSON array
    console.log(users);
  }).catch(err => {
    // log error if any
    console.log(err);
  });
```



JSON.stringify()

- The JSON.stringify() method converts a value to its JSON string representation.
- It's commonly used to convert a JavaScript object but you can also pass primitive values like numbers or booleans and they will also be converted to strings.
- Beside a value to convert, the method also accepts two optional parameters, replacer and space:
- **JSON.stringify(value, replacer, space (0to10));**



Simple Example

JS demo.js X

JS demo.js > ...

```
1  var value = { name: "Akash", age: 30, location: "India" };  
2  var result = JSON.stringify(value);  
3  console.log(result);
```

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

D:\demo>node demo.js

```
{"name": "Akash", "age": 30, "location": "India"}
```



with 2 spaces indentation

```
JS demo.js > ...
1  var value = { name: "Akash", age: 30, location: "India" };
2  var result = JSON.stringify(value);
3  console.log(result);
4  console.log("-----")
5
6  var result1 = JSON.stringify(value, null, 2)
7  console.log(result1);
8
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
D:\demo>node demo.js
{"name":"Akash","age":30,"location":"India"}
-----
{
  "name": "Akash",
  "age": 30,
  "location": "India"
}
```



Example

```
var value = { name: "Akash", age: 30, location: "India" };  
var result = JSON.stringify(value);  
console.log(result);  
console.log("-----")  
  
var result1 = JSON.stringify(value, null, 2)  
console.log(result1);
```



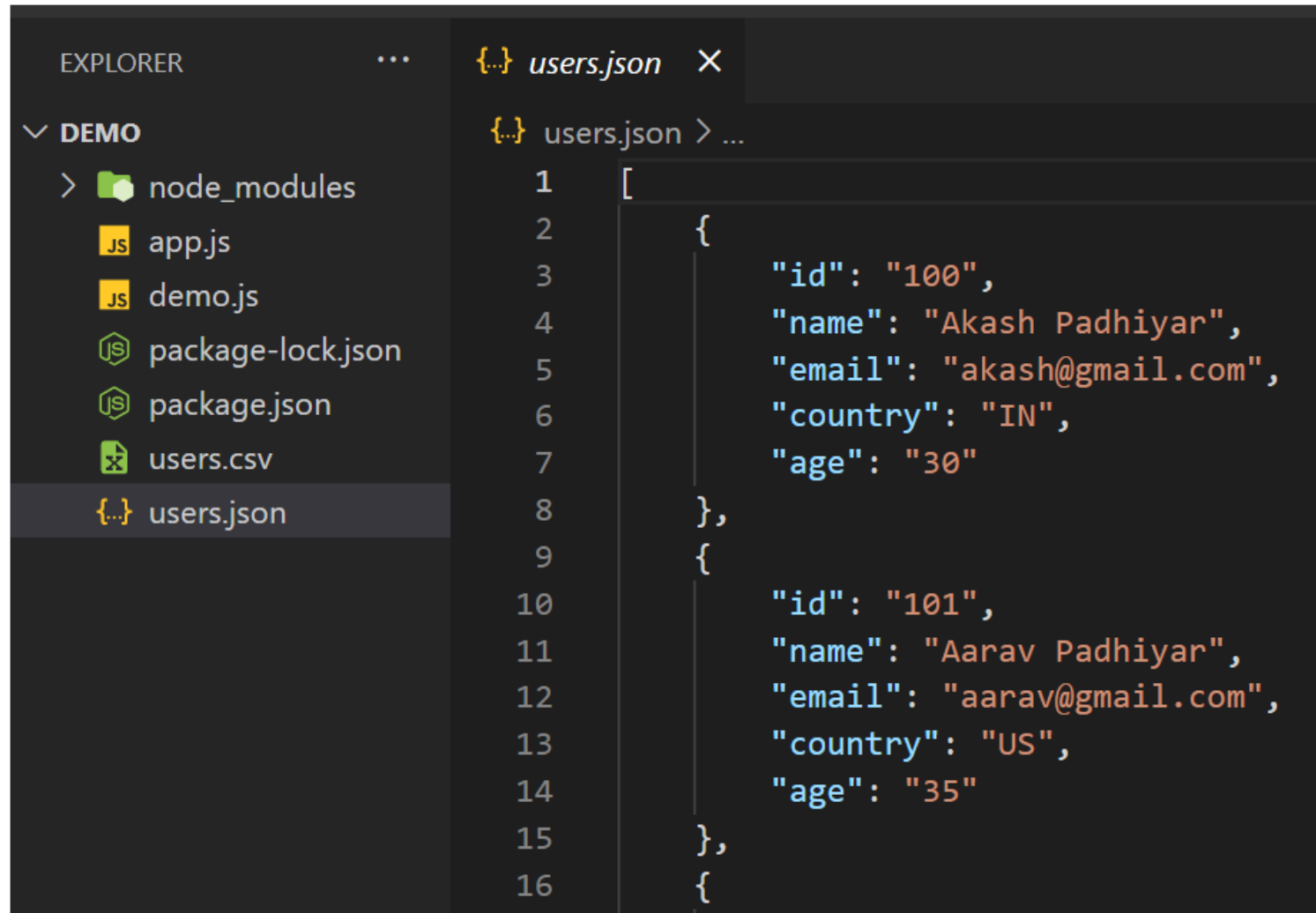
Export CSV to JSON



```
EXPLORER
  DEMO
    node_modules
    app.js
    demo.js
    package-lock.json
    package.json
    users.csv

demo.js
1 // require csvtojson module
2 const CSVToJson = require('csvtojson');
3 const fs = require('fs');
4 // convert users.csv file to JSON array
5 CSVToJson().fromFile('users.csv')
6   .then(users => {
7     // users is a JSON array
8     // log the JSON array
9     console.log(users);
10    // Write JSON array to a file
11    fs.writeFile('users.json', JSON.stringify(users, null, 4), (err) => {
12      if (err) {
13        throw err;
14      }
15      console.log("JSON array is saved.");
16    });
17  }).catch(err => {
18    // log error if any
19    console.log(err);
20  });
```


Output



The screenshot shows a VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'DEMO' with several files: 'node_modules', 'app.js', 'demo.js', 'package-lock.json', 'package.json', 'users.csv', and 'users.json'. The 'users.json' file is selected. The code editor shows the content of 'users.json', which is a JSON array containing two user objects. The first object has an id of '100', name 'Akash Padhiyar', email 'akash@gmail.com', country 'IN', and age '30'. The second object has an id of '101', name 'Aarav Padhiyar', email 'aarav@gmail.com', country 'US', and age '35'.

```
EXPLORER  ...  {..} users.json X
  v DEMO
    > node_modules
      app.js
      demo.js
      package-lock.json
      package.json
      users.csv
      {..} users.json

  {..} users.json > ...
  1  [
  2    {
  3      "id": "100",
  4      "name": "Akash Padhiyar",
  5      "email": "akash@gmail.com",
  6      "country": "IN",
  7      "age": "30"
  8    },
  9    {
 10      "id": "101",
 11      "name": "Aarav Padhiyar",
 12      "email": "aarav@gmail.com",
 13      "country": "US",
 14      "age": "35"
 15    },
 16    {
```

Code

```
// require csvtojson module
const CSVToJSON = require('csvtojson');

const fs = require('fs');

// convert users.csv file to JSON array
CSVToJSON().fromFile('users.csv')

  .then(users => {

    // users is a JSON array

    // log the JSON array
    console.log(users);

    // Write JSON array to a file
    fs.writeFile('users.json', JSON.stringify(users, null, 4), (err) => {

      if (err) {

        throw err;

      }

      console.log("JSON array is saved.");

    });

  }).catch(err => {

    // log error if any
    console.log(err);

  });
```



JSON to CSV



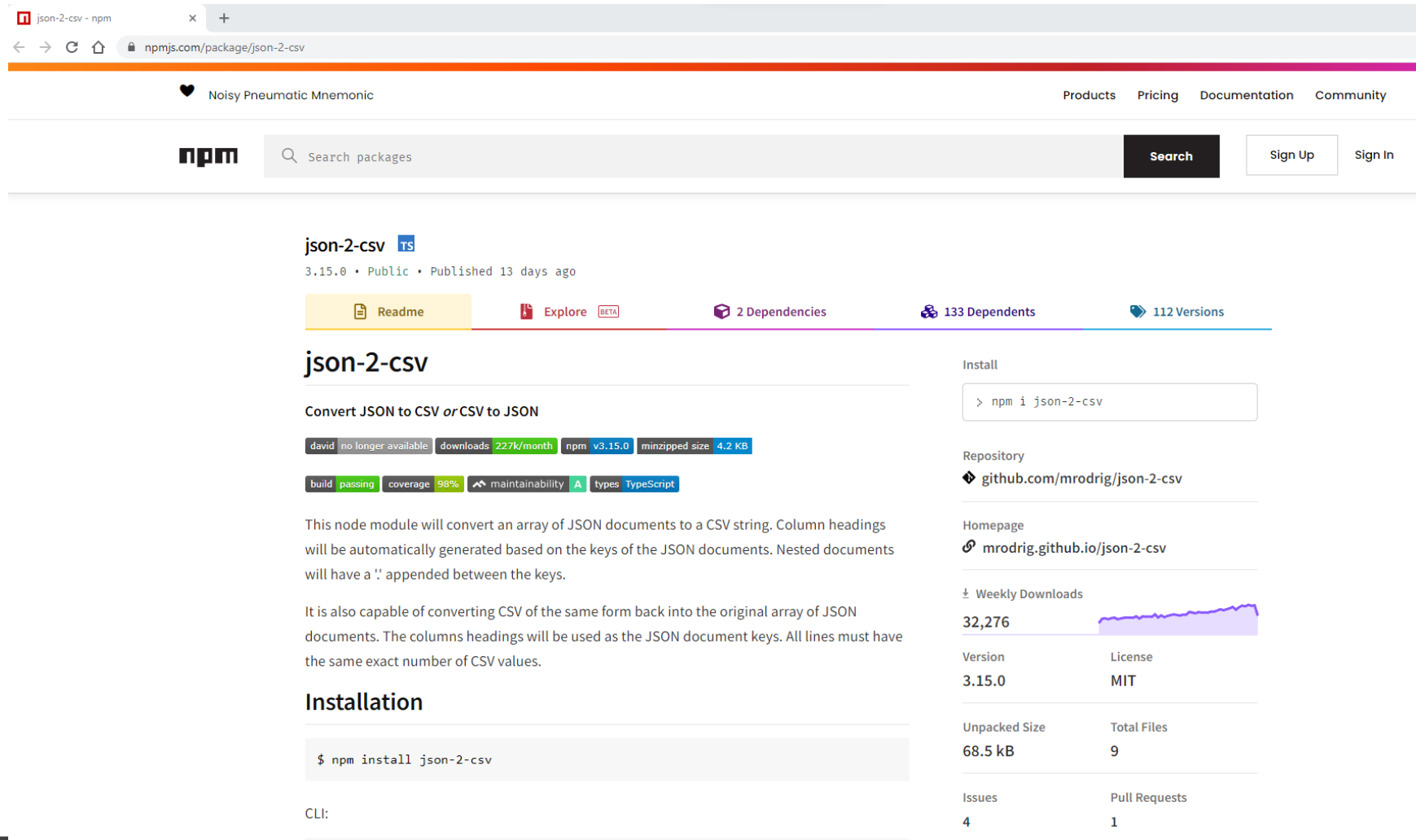
JSON to CSV

- To do the JSON to CSV conversion, we'll be using the json-2-csv module from Node Package Manager (NPM).
- This package converts an array of JSON documents into a CSV string. It automatically generates column headings based on the keys of the JSON documents.
- Nested documents will have a '.' appended between the keys.



Package

- <https://www.npmjs.com/package/json-2-csv>



The screenshot shows the npm package page for `json-2-csv`. The page header includes the npm logo, a search bar, and links for Products, Pricing, Documentation, and Community. The package name `json-2-csv` is displayed with a TypeScript icon. Below the name, it shows the version `3.15.0`, the package is `Public`, and it was published 13 days ago. A navigation bar contains links for Readme, Explore (marked BETA), 2 Dependencies, 133 Dependents, and 112 Versions. The main content area describes the package as a tool to convert JSON to CSV or CSV to JSON. It lists statistics: `no longer available`, `227k/month` downloads, `v3.15.0` npm version, `4.2 KB` minzipped size, `build passing`, `coverage 98%`, `maintainability A`, and `types TypeScript`. A description states that the module converts an array of JSON documents to a CSV string with automatically generated column headings. It also mentions the reverse conversion from CSV to JSON. The installation section shows the CLI command `$ npm install json-2-csv`. On the right, there is an 'Install' section with the command `> npm i json-2-csv`, a 'Repository' section linking to `github.com/mrodrig/json-2-csv`, a 'Homepage' section linking to `mrodrig.github.io/json-2-csv`, a 'Weekly Downloads' chart showing `32,276` downloads, and a table with package details.

Version	License
3.15.0	MIT

Unpacked Size	Total Files
68.5 kB	9

Issues	Pull Requests
4	1



Install NPM

- Install json-2-csv Package
- <https://www.npmjs.com/package/json-2-csv>
- `npm i json-2-csv --save`



A screenshot of a Visual Studio Code terminal window. The terminal has tabs for 'PROBLEMS', 'OUTPUT', 'TERMINAL' (which is active and underlined), and 'DEBUG CONSOLE'. The command prompt shows the directory 'D:\demo' and the command 'npm i json-2-csv --save' being entered, with a cursor at the end of the line.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

D:\demo>npm i json-2-csv --save
```

```
// require json-2-csv module
const converter = require('json-2-csv');
// declare a JSON array
const products = [
  {
    "id": 1,
    "title": "iPhone",
    "price": 50000
  },
  {
    "id": 2,
    "title": "macbook pro",
    "price": 125000
  },
  {
    "id": 3,
    "title": "apple watch",
    "price": 26000
  }
];
// convert JSON array to CSV string
converter.json2csv(products, (err, csv) => {
  if (err) {
    throw err;
  }
  // print CSV string
  console.log(csv);
});
```

Code

```
JS demo.js > converter.json2csv() callback
1 // require json-2-csv module
2 const converter = require('json-2-csv');
3 // declare a JSON array
4 const products = [
5   {
6     "id": 1,
7     "title": "iPhone",
8     "price": 50000
9   },
10  {
11    "id": 2,
12    "title": "macbook pro",
13    "price": 125000
14  },
15  {
16    "id": 3,
17    "title": "apple watch",
18    "price": 26000
19  }
20 ];
21 // convert JSON array to CSV string
22 converter.json2csv(products, (err, csv) => {
23   if (err) {
24     throw err;
25   }
26   // print CSV string
27   console.log(csv);
28 });
```



output

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

D:\demo>node demo.js
id,title,price
1,iPhone,50000
2,macbook pro,125000
3,apple watch,26000

D:\demo>
```



Convert json to csv

■ converter.json2csv(array, callback, options)

- `array` - An array of JSON documents to be converted to CSV.
- `callback` - A function of the form `function (err, csv);`
 - This function will receive any errors and/or the string of CSV generated.
- `options` - (Optional) A JSON document specifying any of the following key value pairs:
 - `checkSchemaDifferences` - Boolean - Should all documents have the same schema?
 - Default: `false`
 - Note: An error will be thrown if some documents have differing schemas when this is set to `true`.
 - `delimiter` - Document - Specifies the different types of delimiters
 - `field` - String - Field Delimiter.
 - Default: `,`
 - `wrap` - String - Wrap values in the delimiter of choice (e.g. wrap values in quotes).
 - Default: `"`
 - `eol` - String - End of Line Delimiter.
 - Default: `\n`
 - `emptyFieldValue` - Any - Value that, if specified, will be substituted in for field values that are `undefined`, `null`, or an empty string.
 - Default: `none`
 - `excelBOM` - Boolean - Should a unicode character be prepended to allow Excel to open a UTF-8 encoded file with non-ASCII characters present.
 - `excludeKeys` - Array - Specify the keys that should be excluded from the output.
 - Default: `[]`
 - Note: When used with `unwindArrays`, arrays present at excluded key paths will not be unwound.
 - `expandArrayObjects` - Boolean - Should objects in array values be deep-converted to CSV?
 - Default: `false`
 - Example:



Write CSV to File

```
demo.js > ...
1 // declare a JSON array
2 const products = [
3   {
4     "id": 1,
5     "title": "iPhone",
6     "price": 50000
7   },
8   {
9     "id": 2,
10    "title": "macbook pro",
11    "price": 125000
12  },
13  {
14    "id": 3,
15    "title": "apple watch",
16    "price": 26000
17  }
18 ];
19 // require json-2-csv module
20 const converter = require('json-2-csv');
21 const fs = require('fs');
22 converter.json2csv(products, (err, csv) => {
23   if (err) {
24     throw err;
25   }
26   // print CSV string
27   console.log(csv);
28   // write CSV to a file
29   fs.writeFileSync('products.csv', csv);
30 });
```

```
// require json-2-csv module
const converter = require('json-2-csv');
const fs = require('fs');
converter.json2csv(products, (err, csv) => {
  if (err) {
    throw err;
  }
  // print CSV string
  console.log(csv);
  // write CSV to a file
  fs.writeFileSync('products.csv', csv);
});
```



JSON.parse()

- we can convert a JSON string to plain JavaScript object using JSON.parse().
- JSON.parse(text[, reviver]);

```
const object = JSON.parse('{"key":"value"}');  
// { key: "value" }
```



Read JSON from File

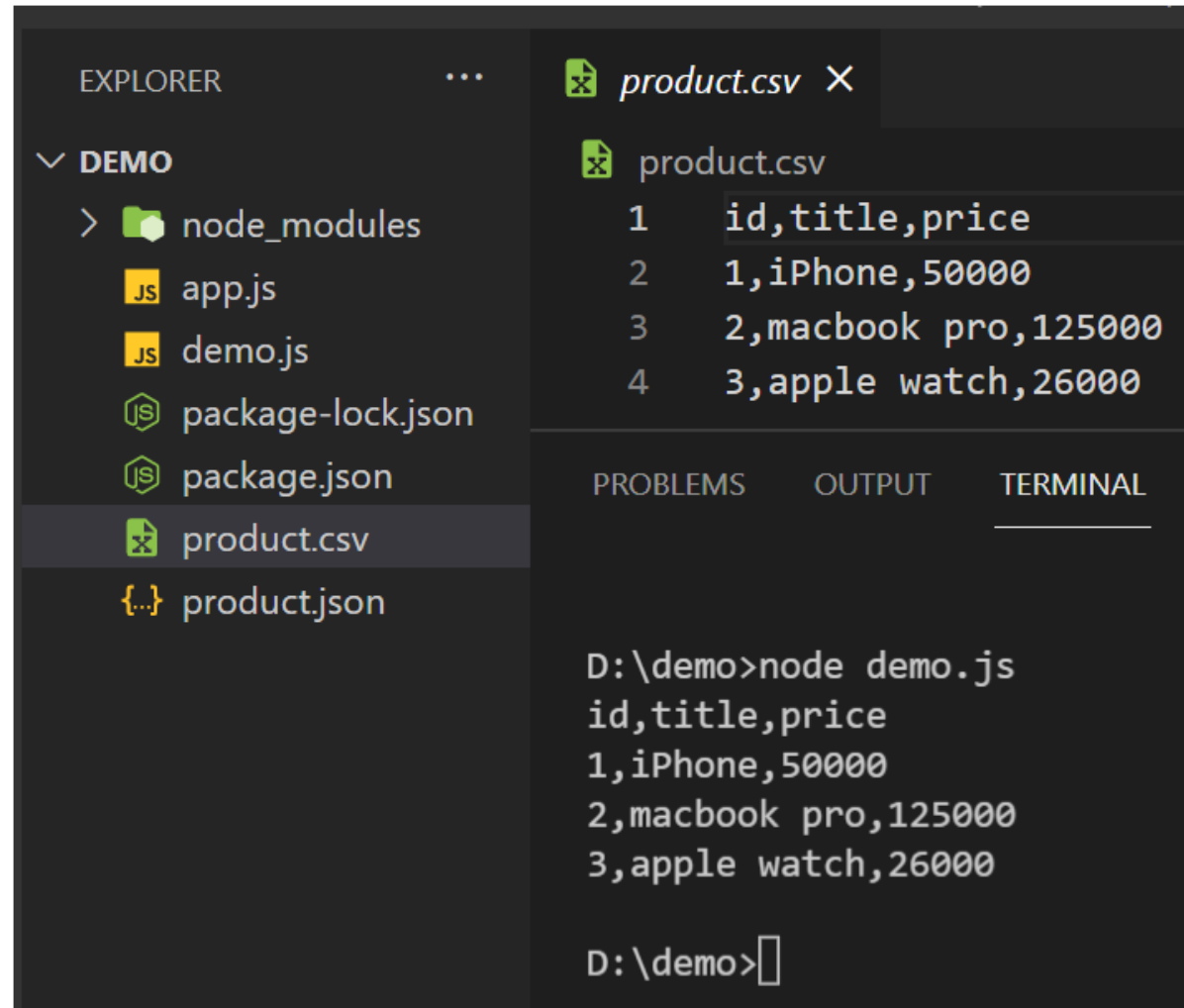
```
{...} product.json ×
{...} product.json > ...
1  [{
2    "id": 1,
3    "title": "iPhone",
4    "price": 50000
5  },
6  {
7    "id": 2,
8    "title": "macbook pro",
9    "price": 125000
10 },
11 {
12   "id": 3,
13   "title": "apple watch",
14   "price": 26000
15 }
16 ]

EXPLORER
DEMO
> node_modules
  JS app.js
  JS demo.js
  package-lock.json
  package.json
  {...} product.json

JS demo.js ×
JS demo.js > ...
1  // require json-2-csv module
2  const converter = require('json-2-csv');
3  const fs = require('fs');
4
5  // read JSON from a file
6  const todos = JSON.parse(fs.readFileSync('product.json'));
7
8  // convert JSON array to CSV string
9  converter.json2csv(todos, (err, csv) => {
10    if (err) {
11      throw err;
12    }
13    // print CSV string
14    console.log(csv);
15    // write CSV to a file
16    fs.writeFileSync('product.csv', csv);
17  });
18  });
```



Output



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a project named 'DEMO' with files: 'node_modules', 'app.js', 'demo.js', 'package-lock.json', 'package.json', 'product.csv' (selected), and 'product.json'. The main editor area shows 'product.csv' with the following content:

	id	title	price
1	1	iPhone	50000
2	2	macbook pro	125000
3	3	apple watch	26000

Below the editor, the Output panel is active, showing the command 'D:\demo>node demo.js' and its output:

```
id,title,price
1,iPhone,50000
2,macbook pro,125000
3,apple watch,26000
```

The terminal prompt 'D:\demo>' is visible at the bottom.

Code

```
[{
  "id": 1,
  "title": "iPhone",
  "price": 50000
},
{
  "id": 2,
  "title": "macbook pro",
  "price": 125000
},
{
  "id": 3,
  "title": "apple watch",
  "price": 26000
}
]
```

```
// require json-2-csv module
const converter = require('json-2-csv');
const fs = require('fs');

// read JSON from a file
const todos = JSON.parse(fs.readFileSync('product.json'));

// convert JSON array to CSV string
converter.json2csv(todos, (err, csv) => {
  if (err) {
    throw err;
  }
  // print CSV string
  console.log(csv);
  // write CSV to a file
  fs.writeFileSync('product.csv', csv);
});
```



Async

```
JS demo.js > ...
1  // require json-2-csv module
2  const converter = require('json-2-csv');
3  const fs = require('fs');
4
5  // read JSON from a file
6  const todos = JSON.parse(fs.readFileSync('product.json'));
7
8  // convert JSON array to CSV string
9  converter.json2csvAsync(todos).then(csv => {
10
11      // print CSV string
12      console.log(csv);
13
14      // write CSV to a file
15      fs.writeFileSync('product.csv', csv);
16
17  }).catch(err => console.log(err));
```



Code

```
// require json-2-csv module
const converter = require('json-2-csv');
const fs = require('fs');

// read JSON from a file
const todos = JSON.parse(fs.readFileSync('product.json'));

// convert JSON array to CSV string
converter.json2csvAsync(todos).then(csv => {

    // print CSV string
    console.log(csv);

    // write CSV to a file
    fs.writeFileSync('product.csv', csv);

}).catch(err => console.log(err));
```

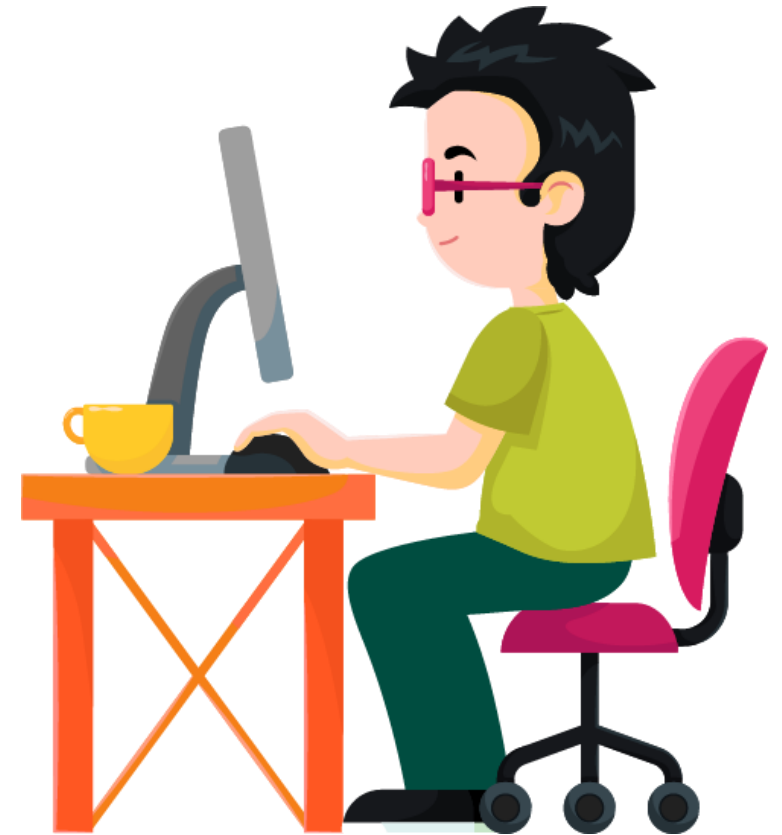


Task

- Take a CSV File from user using File upload and Convert into Json
- Take a JSON File from user using File upload and Convert into CSV



Get Exclusive Video Tutorials



www.apptutorials.com

<https://www.youtube.com/user/Akashtips>





Get More Details

www.akashsir.com



If You Liked It !

Rating Us Now



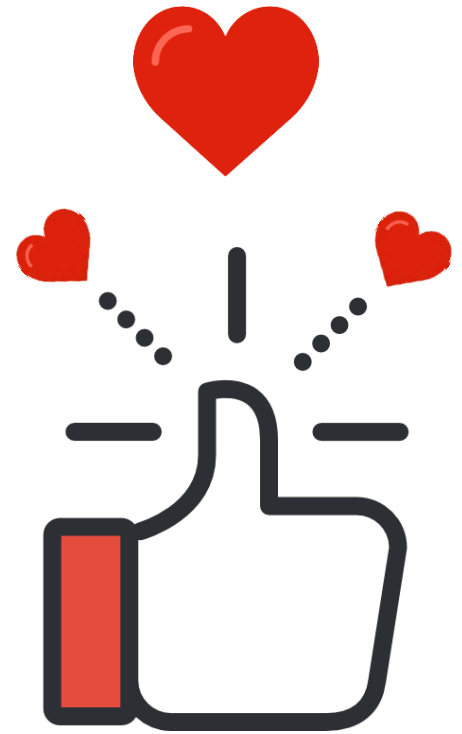
Just Dial

https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/O79PXX79-XX79-170615221520-S5C4_BZDET



Sulekha

<https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad>



Connect With Me



Akash Padhiyar
#AkashSir

www.akashsir.com
www.akashtechlabs.com
www.akashpadhiyar.com
www.apptutorials.com

Social Info



Akash.padhiyar



Akashpadhiyar



Akash_padhiyar



+91 99786-21654



#Akashpadhiyar
#apptutorials