# CSV

Additionaly, you might want to look at the [iconv-lite](https://github.com/ashtuchkin/iconv-lite) module for alternative encoding support.

This project provides CSV generation, parsing, transformation and serialization for Node.js.

It has been tested and used by a large community over the years and should be considered reliable. It provides every option you would expect from an advanced CSV parser and stringifier.

The csv package is itself split into 4 packages:

* [csv-generate](https://github.com/wdavidw/node-csv-generate), a flexible generator of CSV string and Javascript objects.
* [csv-parse](https://github.com/wdavidw/node-csv-parse), a parser converting CSV text into arrays or objects.
* [stream-transform](https://github.com/wdavidw/node-stream-transform), a transformation framework.
* [csv-stringify](https://github.com/wdavidw/node-csv-stringify), a stringifier converting records into a CSV text.

**Documentation**

The full documentation for the current version is available [here](http://csv.adaltas.com/) while the previous documentation is still available [here](http://csv.adaltas.com/legacy/).

**Usage**

Installation command is npm install csv.

Each package is fully compatible with the stream 2 and 3 specifications. Also, a simple callback-based API is always provided for convenience.

**Callback example**

Execute this script with the command node samples/callback.js.

var csv = require('csv');

csv.generate({seed: 1, columns: 2, length: 20}, function(err, data){

csv.parse(data, function(err, data){

csv.transform(data, function(data){

return data.map(function(value){return value.toUpperCase()});

}, function(err, data){

csv.stringify(data, function(err, data){

process.stdout.write(data);

});

});

});

});

**Stream example**

Execute this script with the command node samples/stream.js.

var csv = require('csv');

var generator = csv.generate({seed: 1, columns: 2, length: 20});

var parser = csv.parse();

var transformer = csv.transform(function(data){

return data.map(function(value){return value.toUpperCase()});

});

var stringifier = csv.stringify();

generator.on('readable', function(){

while(data = generator.read()){

parser.write(data);

}

});

parser.on('readable', function(){

while(data = parser.read()){

transformer.write(data);

}

});

transformer.on('readable', function(){

while(data = transformer.read()){

stringifier.write(data);

}

});

stringifier.on('readable', function(){

while(data = stringifier.read()){

process.stdout.write(data);

}

});

**Pipe example**

Execute this script with the command node samples/pipe.js.

var csv = require('csv');

csv.generate({seed: 1, columns: 2, length: 20})

.pipe(csv.parse())

.pipe(csv.transform(function(record){

return record.map(function(value){

return value.toUpperCase()

});

}))

.pipe(csv.stringify())

.pipe(process.stdout);