Node.js Stream

npm init --yes

https://www.npmjs.com/package/stream-filter
npm install stream-filter --save

在使用pipe的时候先对数据做下过滤。

Stream

Stability: 2 - Stable

A stream is an abstract interface for working with streaming data in Node.js. The `stream` module provides a base API that makes it easy to build objects that implement the stream interface.

There are many stream objects provided by Node.js. For instance, a request to an HTTP server and `process.stdout` are both stream instances.

Streams can be readable, writable, or both. All streams are instances of `EventEmitter`.

The `stream` module can be accessed using:

```
const stream = require('stream');
```

File System

Stability: 2 - Stable

File I/O is provided by simple wrappers around standard POSIX functions. To use this module do `require('fs')`. All the methods have asynchronous and synchronous forms.

The asynchronous form always takes a completion callback as its last argument. The arguments passed to the completion callback depend on the method, but the first argument is always reserved for an exception. If the operation was completed successfully, then the first argument will be `null` or `undefined`.

When using the synchronous form any exceptions are immediately thrown. You can use try/catch to handle exceptions or allow them to bubble up.

request.end([data][, encoding][, callback])#

Added in: v0.1.90
Finishes sending the request. If any parts of the body are unsent, it will flush them to the stream. If the request is chunked, this will send the terminating '0\r\n\r\n'.
If data is specified, it is equivalent to calling response.write(data, encoding) followed by request.end(callback).
If callback is specified, it will be called when the request stream is finished.

The request implements the Writable Stream interface. This is an `EventEmitter` with the following events:

req
The request implements the Writable Stream interface.