

The background is a dark navy blue. On the left, there are two overlapping trapezoidal shapes: a blue one in front of a light green one. In the bottom-left corner, there is a circular inset showing a detailed, grayscale image of a printed circuit board (PCB) with various electronic components. In the top-right corner, there is a faint, stylized graphic of circuit traces or a network diagram.

File Upload

Kristiana Atanasova
Tsvetelina Kiriova
Yoanna Vasileva
Ivelin Iliev

Потоци в Java Script

Потоците са една от основните концепции, които захранват приложенията на Node.js. Те са метод за обработка на данни и се използват за последователно четене или запис на вход в изход.

Защо потоци?

- Ефективност на паметта - не е необходимо да зареждате големи количества данни в паметта, преди да можете да ги обработите
- Ефективност на времето - отнема значително по-малко време, за да започнете да обработвате данни веднага щом ги имате, вместо да се налага да чакате с обработката, докато целият пренос бъде осъществен



Има 4 вида потоци в Node.js:

- Записваеми - записваме данни (Пр. `fs.createWriteStream()`)
- Четими - четем данни (Пр. `fs.createReadStream()`)
- Дуплекс - както за четене така и за запис (Пр. `net.Socket`)
- Трансформиращи се - трансформират данните, които се четат или записват в тях



Поточни събития

Readable Streams

Events

- data
- end
- error
- close
- readable

Writable Streams

Events

- drain
- finish
- error
- close
- pipe/unpipe



Четене от поток

```
1  var fs = require("fs");
2  var data = '';
3
4  // Create a readable stream
5  var readerStream = fs.createReadStream('input.txt');
6
7  // Set the encoding to be utf8.
8  readerStream.setEncoding('UTF8');
9
10 // Handle stream events --> data, end, and error
11 readerStream.on('data', function(chunk) {
12     data += chunk;
13 });
14
15 readerStream.on('end', function() {
16     console.log(data);
17 });
18
19 readerStream.on('error', function(err) {
20     console.log(err.stack);
21 });
22
23 console.log("Program Ended");
```

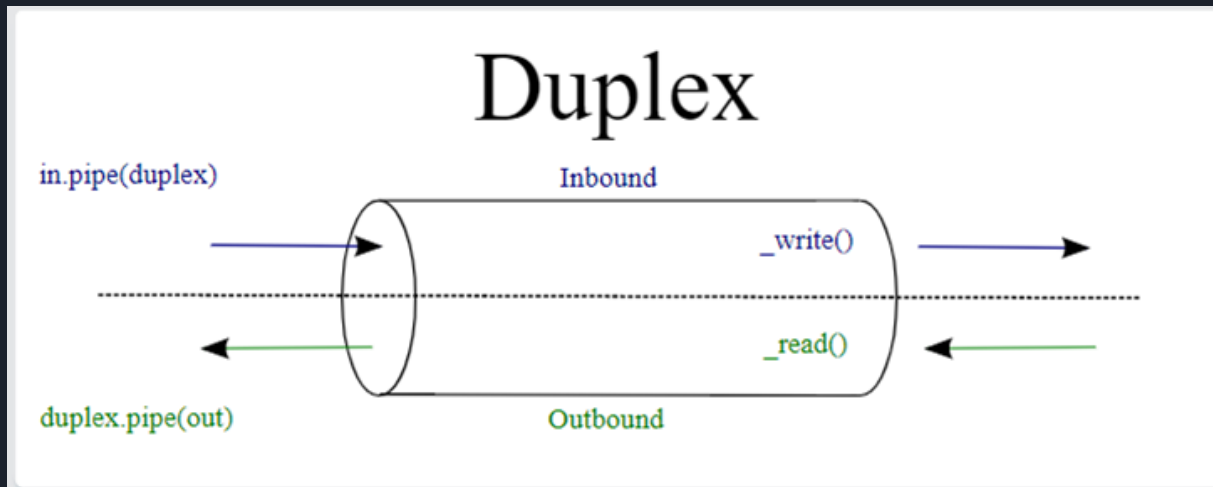


Записване в поток

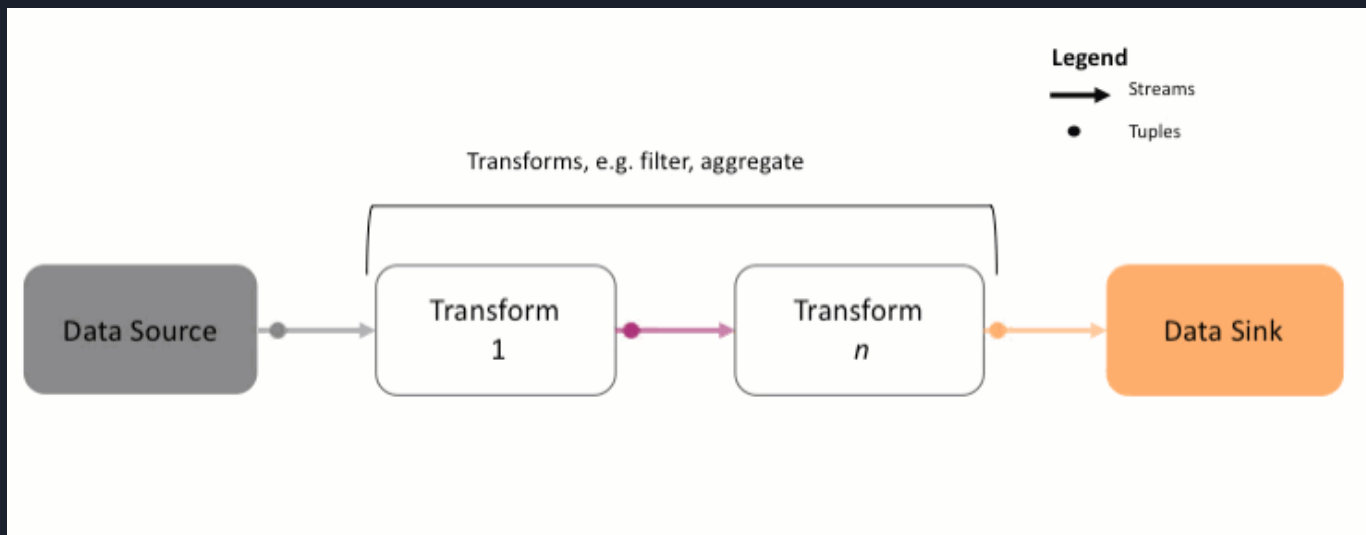
```
1  var fs = require("fs");
2  var data = 'Simply Easy Learning';
3
4  // Create a writable stream
5  var writerStream = fs.createWriteStream('output.txt');
6
7  // Write the data to stream with encoding to be utf8
8  writerStream.write(data, 'UTF8');
9
10 // Mark the end of file
11 writerStream.end();
12
13 // Handle stream events --> finish, and error
14 writerStream.on('finish', function() {
15     console.log("Write completed.");
16 });
17
18 writerStream.on('error', function(err) {
19     console.log(err.stack);
20 });
21
22 console.log("Program Ended");
```

Имплементиране на дуплексни/трансформиращи потоци

1. Дуплексни потоци - основна категория потоци в **Node.js**. Този тип поток е хибриден, което означава, че очаква метод за четене и запис при имплементацията му.



2. Трансформиращ поток - Потокът за трансформиране се състои от двойка потоци: поток за запис, известен като неговата страна за запис, и поток за четене, известен като неговата страна за четене. Може да бъде използван и за вмъкване на данни.





Обектен режим на потоците

По подразбиране потоците очакват стойности на Buffer/String. Има флаг на `objectMode`, който можем да настроим така, че потокът да приема всеки JavaScript обект.


Пример: следната комбинация от потоци за трансформиране прави функция за картографиране на низ от стойности, разделени със запетая, в обект на JavaScript. Така „a,b,c,d“ става {a: b, c: d}.

Качване на файлове

Качването на файлове е често срещано за уеб проектите. Днес ще обсъдим как можем да качим файл от нашето устройство с помощта на `node.js`. Налични са много различни модули за качване на файлове, но днес ще се съсредоточим основно върху два - `formidable` и `multer`.

Avatar

Upload Avatar



Choose File

No file chosen

No local avatar is set. Use the upload field to add a local avatar.

MULTER

Multer е междинен софтуер на node.js за обработка на мултичастни/формулярни данни, който се използва основно за качване на файлове.

```
npm install multer
```

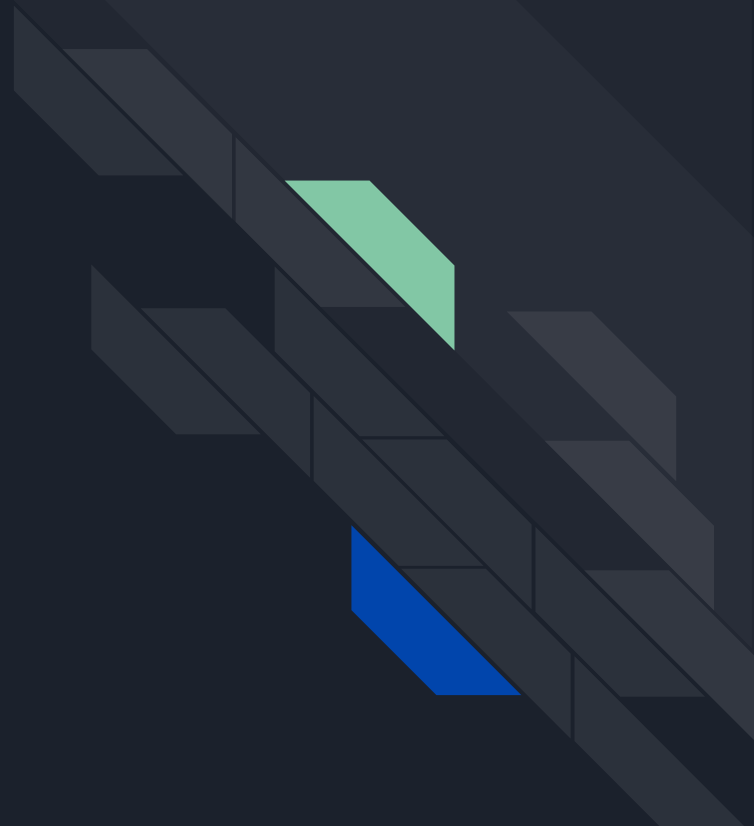
MULTER

Multer има три типа съхранение:

Дисково съхранение (DiskStorage)

Памет за съхранение (MemoryStorage)

Персонализирано съхранение (CustomStorage)



FORMIDABLE

Formidable е Node.js модул за анализиране на данни от формуляри, особено качвания на файлове.

```
npm install formidable
```