

Streams

Los Streams son una colección de datos como los arrays o strings sólo que se van procesando pedazo por pedazo, esta capacidad los hace muy poderosos porque podemos manejar una gran cantidad de datos de manera óptima.

```
const fs = require('fs')
const file = fs.createWriteStream('./big')

for (let i = 0; i <= 1e6; i++) {
  file.write(
    'Estaba la pájara pinta sentada en un verde limón. Con el pico cortaba la rama, con la rama cortaba la flor. Ay, ay, ay!...'
  )
}

file.end()
```

fs permite leer archivos mediante el método `createWriteStream`

`file.write()` me permite escribir lo que se agregara al archivo y con `file.end()` lo finalizamos

```
const fs = require('fs')
const server = require('http').createServer()

server.on('request', (req, res) => {
  fs.readFile('./big', (err, data) => {
    if (err) {
      console.log('error', err)
    }
    res.end(data)
  })
})

server.listen(3000)
```

Creamos un servidor

`server.on()` es un evento que nos permite recibir una petición con el request y el response

`fs.readFile()` nos permite cargar el archivo

puerto del servidor

Nuestro archivo ocupa toda la memoria de nuestro servidor

```
const fs = require('fs')
const server = require('http').createServer()

server.on('request', (req, res) => {
  const src = fs.createReadStream('./big')
  src.pipe(res)
})

server.listen(3000)
```

`CreateReadStream('./big')` nos permite leer nuestro archivo como un stream

`src.pipe(res)` nos limita el almacenamiento en buffer de datos a niveles aceptables de modo que no sobrecargue la memoria

Nuestra memoria esta estable