

Child Process

```
let { exec } = require("child_process")

exec('ls -R', (error, stdout, stderr) => {
  if (error) {
    console.log('error: ' + error)
    return
  }
  if (stderr) {
    console.log('stderr: ' + stderr)
    return
  }
  console.log(stdout)
})
```

Como variantes le voy a colocar git init
git log, va dar error una vez y luego va a funcionar.

Ideal para ejecutar un .sh de instalacion de node
o integracion continua o algo asi.

```
let {execFile} = require("child_process")

execFile('/opt/misscripts/connect-frankfurt.cmd', (error,
stdout, stderr) => {
  if (error) {
    console.log('error: ' + error)
    return
  }
})
```

```
    if (stderr) {
      console.log('stderr: ' + stderr)
      return
    }
    console.log(stdout)
  })
```

```
let {spawn} = require("child_process")
let asyncForLoop = require('@arteyssoft/asyncforloop')
```

```
let child = spawn('node', ['hijo.js'])
```

```
child.stdout.on('data', (data) => {
  console.log(`ok ${data}`)
})
```

```
child.stderr.on('data', (data) => {
  console.log(`error ${data}`)
})
```

```
child.on('error', (error) => {})
child.on('exit', (code, signal) => {})
```

```
asyncForLoop(1000000, (i, next, abort) => {
  console.log('-----')
  console.log(i)
  next()
})
```

Luego el proceso hijo

```
for (let z = 0;; z++) {  
  if (z % 111111111 === 0) {  
    console.log(z)  
  }  
}
```

FORK