

Esdeveniments en NodeJS

TSR 2020 Juansa Sendra, Grup A

Esdeveniments

- Quasi-tot l'API de nodeJS utilitza una arquitectura asincrònica conduïda per esdeveniments
 - Un emisor pot generar diferents tipus d'esdeveniments, identificats amb un nom (string)
 - Example: un stream emiteix 'close', 'drain', 'error', 'finish', 'pipe', 'unpipe'
 - Example: un socket emiteix 'close', 'connection', 'error', 'listening'
 - Podem associar una o més accions a cada esdeveniment
 - cada acció és un callback i representa activitats pendents
 - estableix un ordre: manté un vector d'accions per esdeveniment, i s'activen per ordre
- En els nostres programes podem tractar amb eixos esdeveniments, però també declarar emissors de nous esdeveniments
 - Per a poder definir emissores d'esdeveniments necessitem el mòdul `events`

Esdeveniments.- Estructura aplicació

- Si tots els esdeveniments són externs (API)
 - Example.- esdeveniments de teclat, a través de sockets, etc.
 - associar accions als esdeveniments
- Si desitgem crear esdeveniments interns (des-de el mateix programa)
 - importar la biblioteca `events`
 - crear un emisor (o varios)
 - associar accions als esdeveniments
 - emetre esdeveniments
 - És habitual utilitzar `setTimeout` o `setInterval`

Esdeveniments.- Emisor

- Tots els emisors son instàncies de la classe events.EventEmitter
- Per emetre un esdeveniment utilitza `emisor.emit('esdeveniment', arg, arg,...)`
 - tants arguments com necessitem (possiblement 0)
- En cas de error intern el emisor genera el esdeveniment 'error'

Esdeveniments.- Accions

- `emisor.on('nomEsdeveniment', callback)` associa el callback al esdeveniment
 - El arguments de `emisor.emit(.., args)` són els paràmetres del callback
 - Cada vegada que arriba el esdeveniment activa l'acció
- `emisor.once('nomEsdeveniment', callback)` associa el callback al esdeveniment
 - activa només una vegada i dona la acció de baixa
- podem associar varios callbacks al mateix esdeveniments, i davant la arribada del esdeveniment s'activen en el ordre en que s'han registrat
- si no tenim cap acció associada al esdeveniment `emisor.on('error', ...)` i arriba eixe esdeveniment, el program avorta
- Una acció activada s'executa immediatament (si no estem ejecutan cap altra cosa) o es queda en la cua d'esdeveniments

Esdeveniments.- Example

```
const ev = require('events')
const emisor = new ev.EventEmitter
emisor.on('event', (i)=>{
    console.log('Arriba el esdeveniment '+i)
})
emisor.emit('event',1)
emisor.emit('event',2)
```