Loggers

Consola problema de performance

- Usualmente usamos el console.log cada vez que queremos probar algún flujo.
- Porque evitarlos?

    - Esta es una acción sincrónica, por lo que puede generar un bloqueo en la ejecución de nuestra función.
    - Persistencia de la información que imprimimos en consola. No podemos rescatar mensajes importantes de errores en todo el listado de mensajes que mostramos.

Logger como solución

- Es un mecanismo que nos permite mostrar y registrar información sobre eventos que puedan pasar dentro de nuestra aplicación.
- Podemos segmentar nuestros logs o mensajes por prioridad.
- Podemos mostrar o registrar los logs no solo a nivel de consola, podemos usar archivos.
- Podemos definir niveles de acuerdo al entorno que está nuestra aplicación.

Winston Logger

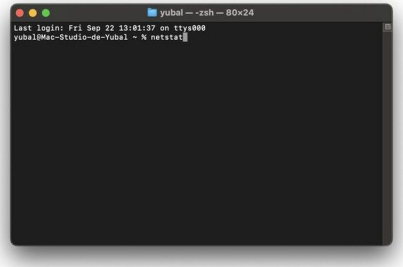
- Es una herramienta o dependencia que nos permite registrar logs de nuestra aplicación en diferentes fuentes (consola, archivos, correos).

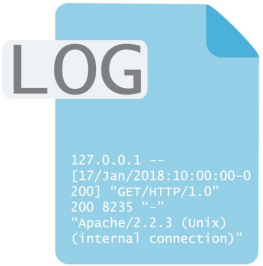
- Conceptos clave dentro de Winston:

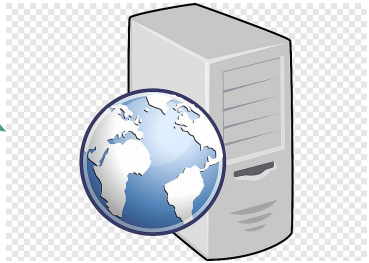
    - Transporte: lugar o fuente donde vamos a almacenar o mostrar nuestros logs.
    - Nivel: prioridad que tiene cada mensaje o log.

Transportes

Logs

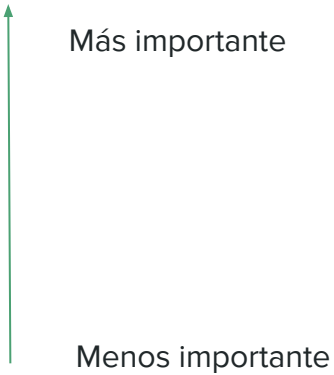






Niveles

- Definen la prioridad de nuestros logs, al final le permitirá a nuestro logger saber cuáles mensajes tomar y cuáles ignorar.
- Niveles por defecto:
  - error: 0
  - warn: 1
  - info: 2
  - http: 3
  - verbose: 4
  - debug: 5
  - silly: 6



Cómo funcionan

Nivel: info

Logguear: “Hola mundo”

- error
- warn
- info
- http
- verbose
- debug
- silly