

Error Logging

An error log is a file that contains detailed records of error conditions a computer software encounters when it's running. [1] There are many error logging libraries for Node.JS. We will be looking at some of them over here.

Winston:

- Winston is a universal logging library with support for multiple transports (devices to store log files).
- Each can have multiple transports configured at different levels.
- It is developed in such a way as to make the different different parts of the logging process decoupled from each other.
- A logger can be created by using the **createLogger** function.

Some of its features are explored below:

- We create a logger by using the **createLogger** function and at least define some **transports**.
- Levels can be defined in ascending order of importance with 0 being the most important and n being the least.

The following properties are accepted by a logger:

- level (Level of the log message)
- levels (Levels (and colors) representing log priorities)
- format (Formatting for **info** messages)
- Transports (Where the logs will be stored)
- exitOnError
- silent

Transports can be added or removed from a logger once it has been created.

A logger can also be reconfigured using **logger.configure**.

To set up a simple logger in Winston with 3 levels: error, warn and info, we need to do the following:

- Create a logger instance using `createLogger`.
- Assign the **levels** property to it with 3 levels: error, warn & info.
- Assign format settings to the format property.
- Select the variables (datetime, message, level) to be output.
- Assign the transports in the transport property.

Pino:

Pino is a very fast logger for JS frameworks.

Here, we will be going through some of the key features of **Pino**.

- The default log levels in pino are: **trace**, **debug**, **info**, **warn**, **error**, and **fatal** (in ascending order of severity).
- The minimum log level in pino is **info** and it is set at the time of creating the logger. This means that only events of this severity and below will be logged.
- Custom log levels can be created and assigned using the **customLevels** property in **pino**.
- Pino binds two extra properties to each log entry by default: the program's process ID (pid), and the current machine's hostname. These can be changed by changing the **bindings** property.
- It is possible to add **contextual messages** to our logs. This is done through the **message** parameter on a level method.
- Contextual messages about where a log message is being produced (function, endpoint, etc) can be added using child loggers.
- It is possible to transport **pino** logs to multiple destinations.