Logging in Node JS Application

There are several tools found searching on internet and Listed below.

1. Loggly ( Third party software, require purchase)
2. Morgan -- npm install - -save morgan
3. Winston – npm install - -save Winston
4. Bunyan --- npm install - -save bunyan
5. Log -- npm install - -save log
6. log4js – npm install - -save log4js (<https://www.npmjs.com/package/log4js>)
7. More logging options in node JS (https://www.npmjs.com/browse/keyword/log4j)
8. Logging also can be done by fs (file stream)

**First, we need to figure out what are we logging and which would fulfill our requirements.**

* **Loggly:**

would be the mostly out of the consideration as it is a third party and require purchase, unless no other logging mechanism works and we desperately need logging in application

* **Morgan**: (<https://github.com/expressjs/morgan>; https://www.npmjs.com/package/morgan)

Morgan can be used with predefined format string. The format function will be called with three arguments tokens, req, and res, where tokens is object with all defined tokens, req is the HTTP request and res is the HTTP response. The function is expected to return a string that will be the log line, or undefined / null to skip logging. There are various pre-defined formats provided. We need to dig in more for using in our application. Morgan is a middleware, which will help us to identify the clients who are accessing our application

* **Winston: (https://www.npmjs.com/package/express-winston)**

Winston designed to be a simple and universal logging library with support for multiple transports, transports are storage device where logs being stored. Winston logger have multiple transporters configured at different levels and can be configured via constructor options. Winston supports multiple loggers, so for multiple logs on different category can be logged.

* **Bunyan:(https://www.npmjs.com/package/bunyan)**

Bunyan logs differently than Winston, logs from Bunyan is one line of JSON file output. Bunyan works well to log complex context and object than Winston. Bunyan has a concept of child logger, which allows specializing a logger for a sub-component of application, which helps to scope logs from different component.

* **Log:(** **https://www.npmjs.com/package/log)**

Light-weight logging for NodeJS, including a streaming log reader. It has different log levels mirroring Sys.logs.

**Log levels:**

* + - 0 EMERGENCY system is unusable
* 1 ALERT action must be taken immediately
* 2 CRITICAL the system is in critical condition
* 3 ERROR error condition
* 4 WARNING warning condition
* 5 NOTICE a normal but significant condition
* 6 INFO a purely informational message
* 7 DEBUG messages to debug an application
* **Log4j:**

The converted version of Log4j, which is working in node, is available (<https://www.npmjs.com/package/log4js>) and full documentation is available at <https://nomiddlename.github.io/log4js-node/>

* + - Out of the box it supports the following features:
    - colored console logging to stdout or stderr
    - File appender, with configurable log rolling based on file size or date
    - SMTP appender
    - GELF appender
    - Loggly appender
    - Logstash UDP appender
    - LogFaces (UDP and HTTP) appender
    - Multiprocess appender (useful when you've got multiple servers)
    - a logger for connect/express servers
    - Configurable log message layout/patterns
    - Different log levels for different log categories (make some parts of your app log as DEBUG, others only ERRORS, etc.)

**Many more logging tools in Node js available at following link:**

[**https://www.npmjs.com/browse/keyword/log4j**](https://www.npmjs.com/browse/keyword/log4j)