

Kindling

a simple logging utility for your Node project.

Change-log

- 1.6.0
 - Moved to Typescript
 - Types are now included when compiled!
 - All enum names are now the correct case
 - This ***may*** break older implementations.
 - Exposes godly and ungodly log levels
 - Removed dependency on BoxToolsJS
 - Updated Readme

Installation

```
npm install kindling-logger
```

OR

```
yarn add kindling-logger
```

Usage

Create the logging object

```
const { Logger, LogLevels, logEndpoints } = require('kindling-logger')
const myLog = new Logger()
```

Add an endpoint for the log messages

```
const consoleEndpoint = {
  name: 'Console',
  type: LogEndpoints.console,
  logLevel: LogLevels.INFO,
}
```

```
myLog.addEndpoint(consoleEndpoint)
myLog.info('Hello World!')

// 2019-05-19 10:10:48.374 INFO:: Hello World!
```

Log Levels

- UNGODLY
- INFO
- DEBUG
- WARN
- ERROR
- FAILURE
- GODLY

Endpoints

- **Console**
 - {name, type, LogLevel}
 - This logs out to your local console, or the JS console in your browser (if you like that sort of thing).
- **File**
 - {name, type, LogLevel, filePath, fileName, rotating [hourly, daily, weekly, monthly, yearly]}
 - This logs out to a file. No fanciness here. If A file path or file name is not set it will output to the local directory as 'log.txt'
- **UDP**
 - {name, type, logLevel, ipAddress, port}
 - This will output a UDP message to anywhere you like. It's up to you what to do with it from there.
- **Electron Console**
 - {name, type, LogLevel}
 - If you're building an Electron app this will output to the main console in your electron app.
- **Custom**
 - {name, type, LogLevel}
 - Pretty much do whatever you like here.

TODOs

- ☐ Fix the custom endpoint
- ☐ Possibly add a web-socket logger?
- ☐ Add a custom output type
 - ☐ Needs testing
- ☐ Add bind parameter to UDP - this allows logging through a particular interface
- ☐ Thinking about making this a singleton class