

Kindling

a simple logging utility for your Node project.

Installation

```
npm install 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

- INFO
- DEBUG
- WARN
- ERROR
- FAILURE

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}
 - This logs out to a file. No fanciness here. I don't currently recommend this for anything other than debugging. 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. If you don't know what that means that this option is not for you.
- **Custom**
 - {name, type, logLevel}
 - Pretty much do whatever you like here.

TODOs

- ☐ It's terribly documented. I need to clean all of that up.
- ☐ Fix the custom endpoint
- ☐ Add multicast handling to UDP logger
- ☐ Add rotation options to FILE logger - this may be better suited as a new type
- ☐ Possibly add a websocket logger?
- ☐ Add a custom output type
- ☐ Add bind parameter to UDP - this allows logging through a particular interface