How to use winston for the typescript app

井民全, Jing, mqjing@gmail.com

Table of Contents

1. Install

npm install winston

2. Quick

Code

```
# install
npm install winston

# code
import winston = require('winston');
import {format} from 'winston';

export default class LoggerService {
   public static logger:winston.Logger;

   public static init(strLevel:string='info'){
      let config = {
            level: 'info',
            format: format.combine(
               format.timestamp({
                  format: 'YYYY-MM-DD HH:mm:ss'
                  }),
               format.printf(info => `${info.timestamp} ${info.level}: ${info.message}`)
            ),
      }
      LoggerService.logger = winston.createLogger(config);

      const transportsConsole = new winston.transports.Console(
         {level: strLevel,
          format: winston.format.combine(
            winston.format.colorize(),
            winston.format.simple()
           )}
      );

      const transportsErrorFile = new winston.transports.File(
         {filename: 'error.log', level: 'error', options:{flag: 'w'}},
      );

      const transportsCombinedFile = new winston.transports.File(
         {filename: 'combined.log', level: strLevel, options:{flag: 'w'}},
      );

      if (process.env.NODE_ENV !== 'production') {
         LoggerService.logger.add(transportsConsole);
         LoggerService.logger.add(transportsErrorFile);
         LoggerService.logger.add(transportsCombinedFile);
      }
   }
}
```

2.1. Code (verbose)

```
# install
npm install winston

# code
import winston = require('winston');
import {format} from 'winston';

const logger = winston.createLogger({
  level: 'info',
  //format: winston.format.json(),
  format: format.combine(
    format.timestamp({
      format: 'YYYY-MM-DD HH:mm:ss'
    }),
    // format.label({ label: '[my-label]' }),
    //
    // The simple format outputs
    // `${level}: ${message} ${[Object with everything else]}`
    // format.simple()
    format.printf(info => `${info.timestamp} ${info.level}: ${info.message}`)
  ),
  // defaultMeta: { service: 'user-service' },
  transports: [
    new winston.transports.File({ filename: 'error.log', level: 'error' }),
    new winston.transports.File({ filename: 'combined.log' }),
  ],
});
if (process.env.NODE_ENV !== 'production') {
  logger.add(new winston.transports.Console({
     format: winston.format.combine(
                  winston.format.colorize(),
                  winston.format.simple()
                )

  }));
}

logger.info('this is an info');
logger.error('this is an error');
logger.warn('this is a warning')

Output
info: this is an info {"timestamp":"2020-10-06 12:14:53"}
error: this is an error {"timestamp":"2020-10-06 12:14:53"}
warn: this is a warning {"timestamp":"2020-10-06 12:14:53"}
```

Log Files
combined.log
2020-10-06 12:14:53 info: this is an info
2020-10-06 12:14:53 error: this is an error
2020-10-06 12:14:53 warn: this is a warning

Error.log
2020-10-06 12:14:53 error: this is an error

2.2. Output

info: this is an info {"timestamp":"2020-10-06 12:14:53"}
error: this is an error {"timestamp":"2020-10-06 12:14:53"}
warn: this is a warning {"timestamp":"2020-10-06 12:14:53"}

2.3. Log Files

combined.log
2020-10-06 12:14:53 info: this is an info
2020-10-06 12:14:53 error: this is an error
2020-10-06 12:14:53 warn: this is a warning

Error.log
2020-10-06 12:14:53 error: this is an error

3. Module Version

3.1. Code

LoggerService.ts

```
import {createLogger, format, transports} from 'winston'

export default class LoggerService {
 public static logger = createLogger({
   level: 'info',
   //format: winston.format.json(),
   format: format.combine(
       format.timestamp({
       format: 'YYYY-MM-DD HH:mm:ss'
       }),
       // format.label({ label: '[my-label]' }),
```

```
        //
        // The simple format outputs
        // `${level}: ${message} ${[Object with everything else]}`
        // format.simple()
        format.printf(info => `${info.timestamp} ${info.level}:
${info.message}`)
    ),
    // defaultMeta: { service: 'user-service' },
    transports: [
        // clean the log file when start
        new transports.File({ filename: 'error.log', level: 'error',
options:{flag: 'w'} }),
        new transports.File({ filename: 'combined.log', options:{flag:
'w'} })
        ],
    });

    public static init(){
        if (process.env.NODE_ENV !== 'production') {
            this.logger.add(new transports.Console({
            format: format.simple(),
            }));
        }
    }
}
```

Usage.ts

```
import {LoggerService} from './LoggerService'
LoggerService.init();
LoggerService.logger.info('this is an info2');
LoggerService.logger.error('this is an error2');
LoggerService.logger.warn('this is a warning2');
```

3.2. Output

info: this is an info {"timestamp":"2020-10-06 12:14:53"}
error: this is an error {"timestamp":"2020-10-06 12:14:53"}

warn: this is a warning {"timestamp":"2020-10-06 12:14:53"}

3.3. Log Files

combined.log
2020-10-06 12:14:53 info: this is an info
2020-10-06 12:14:53 error: this is an error
2020-10-06 12:14:53 warn: this is a warning

Error.log
2020-10-06 12:14:53 error: this is an error

4. References

1. Official site, https://github.com/winstonjs/winston
2. Quick start and more example, https://github.com/winstonjs/winston/blob/master/examples/quick-start.js