

@apiclient.xyz/logdn

a

Documentation for @apiclient.xyz/logdna

- [readme.md for @apiclient.xyz/logdna](#)

readme.md for @apiclient.xyz/logdna @mojoio/logdna

an unofficial package for the logdna api

Availability and Links

- npmjs.org (npm package)
- gitlab.com (source)
- github.com (source mirror)
- [docs](#) (typedoc)

Status for master

Status Category	Status Badge
GitLab Pipelines	pipeline status
GitLab Pipeline Test Coverage	coverage report
npm	npm downloads per month
Snyk	Known Vulnerabilities
TypeScript Support	TypeScript
node Support	node
Code Style	Code Style
PackagePhobia (total standalone install weight)	PackagePhobia
PackagePhobia (package size on registry)	PackagePhobia

Status Category	Status Badge
BundlePhobia (total size when bundled)	BundlePhobia
Platform support	Supports Windows 10 Supports Mac OS X

Usage

Use TypeScript for best in class instellisense.

This package is an unofficial package for the logdna.com service. It comes with the following features:

- aggregate logs that require the same uri query parameters and sends them as bundle. This ensures the correct order of logs within logdna.
- resend logs that failed to send.
- support smartlog messages and the smartlog ecosystem
- support giraffe.cloud ecosystem

```
import { ILogPackage } from '@pushrocks/smartlog-interfaces';

import { LogdnaAccount, LogdnaMessage } from '@mojoio/logdna';

// lets create a logDnaAccount
const logDnaAccount = new LogdnaAccount(process.env.LOGDNA_APIKEY);

// lets create a smartlog message (smartlog normally takes care of creating those objects)
const smartlogPackage: ILogPackage = {
  timestamp: Date.now(),
  type: 'log',
  level: 'info',
  context: {
    company: 'Lossless GmbH',
    companyunit: 'lossless.cloud',
    containerName: 'ci-mojoio-logdna',
    environment: 'test',
    runtime: 'node',
    zone: 'shipzone',
  },
  message: 'this is an awesome log message sent by the tapbundle test',
```

```
};

const logdnaMessage = LogdnaMessage.fromSmartLogPackage(smartlogPackage);

logDnaAccount.sendLogdnaMessage(logdnaMessage);

// alternatively simply send the smartlogPackage
// creation of the LogdnaMessage is done for you
logDnaAccount.sendSmartlogPackage(smartlogPackage);

// most of the above funtions return promises should you want to wait for a log to be fully
sent
```

Contribution

We are always happy for code contributions. If you are not the code contributing type that is ok. Still, maintaining Open Source repositories takes considerable time and thought. If you like the quality of what we do and our modules are useful to you we would appreciate a little monthly contribution: You can [contribute one time](#) or [contribute monthly](#). :)

For further information read the linked docs at the top of this readme.

“ MIT licensed | © [Lossless GmbH](#) | By using this npm module you agree to our [privacy policy](#)

repo-footer