

<https://www.linkedin.com/in/call-me-milad/>



Top 100 useful Node.js modules

91 - 100

Share, Like and Comment
Follow me for more knowledge

<https://www.linkedin.com/in/call-me-milad/>

91.NCONF

92.THROTTLE

93.MAILGUN-JS

94.MONGOOSE-PAGINATE-V2

95.NODE-PERSIST

96.DOTENV-FLOW

97.JSONFILE

98. NODEMAILER-SENDGRID-TRANSPORT

99.SHELLJS

100.DOTENV-SAFE

91.NCONF

- **Description:** Nconf is a robust configuration management library for Node.js that simplifies handling configuration settings in your applications. It allows you to load configuration data from various sources such as environment variables, command-line arguments, and configuration files.
- **Use Case:** Nconf is particularly useful for managing application settings in a structured manner. It enables you to centralize configuration management and easily switch between different configuration sources for various environments (e.g., development, staging, production).

```
Nconf

const nconf = require('nconf');
// Set defaults
nconf.defaults({
  'database:host': 'localhost',
  'database:port': 27017
});
// Access a configuration setting
const dbHost = nconf.get('database:host');
const dbPort = nconf.get('database:port');
console.log(`Database Host: ${dbHost}, Port: ${dbPort}`);
```

92. THROTTLE

- **Description:** Throttle is a Node.js library that provides a convenient way to limit the rate at which a function is called. This is valuable for controlling resource-intensive operations or preventing abuse of APIs by limiting the number of requests over time.
- **Use Case:** You can use Throttle to prevent your application from making too many requests to external services, ensuring compliance with rate limits and avoiding performance issues.

```
Throttle

const throttle = require('throttle');

const throttledFunction = throttle(myFunction, 5000); // Allow one call every
5 seconds

// Call the throttled function
throttledFunction();
```

93. MAILGUN-JS

- **Description:** Mailgun-js is a Node.js library that simplifies sending and managing email messages through the Mailgun email service. It offers features such as sending plain text and HTML emails, handling attachments, and tracking email deliveries.
- **Use Case:** You can use Mailgun-js to incorporate email functionality into your Node.js applications, such as sending notifications, newsletters, or transactional emails.

```
Mailgun-js

const mailgun = require('mailgun-js')({
  apiKey: 'your-api-key',
  domain: 'your-domain.com'
});

const data = {
  from: 'you@your-domain.com',
  to: 'recipient@example.com',
  subject: 'Hello',
  template: 'welcome-email', // Use a predefined template
  'v:username': 'John', // Template variables
};

mailgun.messages().send(data, (error, body) => {
  console.log(body);
});
```

94. MONGOOSE-PAGINATE-V2

- **Description:** Mongoose-paginate-v2 is a pagination plugin designed for Mongoose, a popular MongoDB object modeling library for Node.js. It extends Mongoose models to provide easy-to-use pagination functionality.
- **Use Case:** When working with large datasets in MongoDB, Mongoose-paginate-v2 simplifies the process of paginating query results, making it easier to implement paginated APIs or data browsing features.

```
Mongoose-paginate-v2

const mongoose = require('mongoose');
const mongoosePaginate = require('mongoose-paginate-v2');

const schema = new mongoose.Schema({ /* Your schema definition */ });
schema.plugin(mongoosePaginate);

const Model = mongoose.model('Model', schema);

Model.paginate({}, { page: 1, limit: 10, sort: { createdAt: -1 } })
  .then((result) => {
    console.log(result);
  });
```

95. NODE-PERSIST

- **Description:** Node-persist is a simple and efficient key-value storage library for Node.js. It allows you to persistently store data on the local filesystem, making it suitable for applications that need basic data persistence.
- **Use Case:** Node-persist is often used in situations where a full-fledged database is not required, such as caching, storing user preferences, or maintaining application state across restarts.

```
Node-persist

const storage = require('node-persist');
// Initialize storage
storage.initSync();
// Set user preferences
storage.setItemSync('theme', 'dark');
storage.setItemSync('language', 'en');
// Retrieve user preferences
const theme = storage.getItemSync('theme');
console.log('Selected Theme:', theme);
```

96. DOTENV-FLOW

- **Description:** Dotenv-flow is an extension of the `dotenv` module for managing environment variables in Node.js. It provides advanced features like loading multiple `.env` files and ensuring the presence of required variables.
- **Use Case:** Dotenv-flow is helpful for organizing and managing environment-specific configuration files in different environments (e.g., development, testing, production) while enforcing the use of essential variables.



Dotenv-flow

```
require('dotenv-flow').config();  
  
console.log(process.env.MY_VARIABLE);
```


97. JSONFILE

- **Description:** Jsonfile is a Node.js library that simplifies reading and writing JSON files. It offers a straightforward way to interact with structured data stored in JSON format.
- **Use Case:** Jsonfile is commonly used for managing application configuration files, storing application data, or reading and writing JSON-based configuration settings.

```
Jsonfile

const jsonfile = require('jsonfile');
const file = 'config.json';

// Read JSON data from a file
const dataRead = jsonfile.readFileSync(file);
console.log('Read Data:', dataRead);

// Write JSON data to a file
const dataToWrite = { key: 'new value' };
jsonfile.writeFileSync(file, dataToWrite);
```

98. NODEMAILER-SENDGRID-TRANSPORT


- **Description:** Nodemailer-sendgrid-transport is a transport plugin for Nodemailer, a popular Node.js library for sending emails. It enables sending email messages through the SendGrid email service.
- **Use Case:** When you need to send transactional or marketing emails from your Node.js application, this module simplifies the integration with SendGrid, which offers features like email analytics and template rendering.

```
Nodemailer-sendgrid-transport

const nodemailer = require('nodemailer');
const sendgridTransport = require('nodemailer-sendgrid-transport');
const transporter = nodemailer.createTransport(
  sendgridTransport({ auth: { api_key: 'your-sendgrid-api-key' } })
);
const mailOptions = {
  from: 'you@your-domain.com', to: 'recipient@example.com',
  subject: 'HTML Email Example',
  html: '<h1>Hello!</h1><p>This is an HTML email.</p>'
};
transporter.sendMail(mailOptions, (error, info) => {
  if (error) console.log(error);
  else console.log('Email sent:', info.response);
});
```

99. SHELLJS

- **Description:** Shelljs is a cross-platform shell scripting library for Node.js. It allows you to execute shell commands and interact with the file system, making it a versatile tool for automating tasks.
- **Use Case:** Shelljs is valuable for creating scripts to automate common development and system administration tasks, like file manipulation, building, testing, and deployment.



```
const shell = require('shelljs');

// Create a new directory
shell.mkdir('-p', 'new-directory');

// Move files to the new directory
shell.mv('*.*txt', 'new-directory/');
```

100. DOTENV-SAFE

- **Description:** Dotenv-safe is an extension of the `dotenv` module, enhancing it with safety features. It ensures that required environment variables are defined, preventing your application from starting if critical variables are missing.
- **Use Case:** This module is crucial for securing your application by enforcing the presence of essential environment variables, reducing the risk of misconfiguration and potential security vulnerabilities.

```
Dotenv-safe

const dotenvSafe = require('dotenv-safe');

dotenvSafe.config({
  path: '.env',
  example: '.env.example',
  allowEmptyValues: false,
});

console.log(process.env.MY_VARIABLE);
```

<https://www.linkedin.com/in/call-me-milad/>



<https://www.linkedin.com/in/call-me-milad/>