

@push.rocks/smartunique

A library to generate unique identifiers using methods like UUID and nanoid.

- [readme.md for @push.rocks/smartunique](#)

readme.md for @push.rocks/smartunique

make things unique

Install

To install `@push.rocks/smartunique`, use the following npm command:

```
npm install @push.rocks/smartunique --save
```

This will add it to your project's dependencies. Make sure you have Node.js and npm installed in your development environment.

Usage

`@push.rocks/smartunique` is a TypeScript-powered module designed to help you generate unique identifiers such as short IDs, UUIDs, or custom unique strings. Its primary use is in scenarios where you need to ensure the uniqueness of elements or entities within your application, such as database keys, user identifiers, or session tokens.

Setting Up Your Project

Before diving into the examples, ensure your project is set up for TypeScript:

1. Initialize a new npm project if you haven't already:

```
npm init -y
```

2. Install TypeScript:

```
npm install typescript --save-dev
```

3. Initialize TypeScript in your project:

```
npx tsc --init
```

4. Ensure your `tsconfig.json` is configured to support ES Modules, as `@push.rocks/smartunique` is an ES Module package.

```
{
  "compilerOptions": {
    "target": "es6",
    "module": "ESNext",
    "strict": true,
    "esModuleInterop": true,
    "forceConsistentCasingInFileNames": true
  }
}
```

Generating Unique Identifiers

Let's explore how to generate different types of unique identifiers using `@push.rocks/smartunique`.

Short Unique Identifiers

Short Ids are handy when you need a concise, highly unique identifier.

```
import { shortId } from '@push.rocks/smartunique';

const myShortId = shortId();
console.log(`Generated shortId: ${myShortId}`);

// Specify a custom length for the shortId
const customLengthShortId = shortId(10);
console.log(`Generated shortId with custom length: ${customLengthShortId}`);
```

UUID Version 4 (Random)

UUIDs are universally unique identifiers that are widely used for ensuring uniqueness across distributed systems.

```
import { uuid4 } from '@push.rocks/smartunique';

const myUuid4 = uuid4();
console.log(`Generated UUID v4: ${myUuid4}`);
```

UUID Version 5 (Name-Based)

UUID v5 generates a unique identifier based on a namespace identifier and a name.

```
import { uuid5, uuid4 } from '@push.rocks/smartunique';

const namespaceUuid = uuid4(); // For example purposes, a random UUID serves as the namespace
const myUuid5 = uuid5('myUniqueName', namespaceUuid);
console.log(`Generated UUID v5: ${myUuid5}`);
```

Custom Unique Identifiers

For cases where you need a custom format for your unique identifiers, `@push.rocks/smartunique` offers `uni` and `uniSimple` functions.

```
import { uni, uniSimple } from '@push.rocks/smartunique';

const customUni = uni('prefix', 16);
console.log(`Generated custom unique identifier: ${customUni}`);

const simpleUni = uniSimple('simple', 4);
console.log(`Generated simple unique identifier: ${simpleUni}`);
```

Conclusion

Through these examples, you've seen how to generate various types of unique identifiers using `@push.rocks/smartunique`. Whether you need short IDs for concise references, UUIDs for global uniqueness, or custom unique identifiers for specific use cases, this package provides the tools necessary for your project's unique identifier needs.

Make sure to explore the package further to fully leverage its capabilities in your applications.

For any further assistance or contribution to the project, please refer to the repository on [GitLab](#) or its mirror on [GitHub](#). Contributions, whether in the form of feature requests, bug reports, or pull

requests, are always welcome.

License and Legal Information

This repository contains open-source code that is licensed under the MIT License. A copy of the MIT License can be found in the [license](#) file within this repository.

Please note: The MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as required for reasonable and customary use in describing the origin of the work and reproducing the content of the NOTICE file.

Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH and are not included within the scope of the MIT license granted herein. Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines, and any usage must be approved in writing by Task Venture Capital GmbH.

Company Information

Task Venture Capital GmbH

Registered at District court Bremen HRB 35230 HB, Germany

For any legal inquiries or if you require further information, please contact us via email at hello@task.vc.

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.