@push.rocks/smartformat

A module that formats bytes and milliseconds into human-readable strings.

readme.md for @push.rocks/smartformat

format things

Install

To install `@push.rocks/smartformat`, use npm (or yarn, or pnpm) by running the following command in your terminal:

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

This will add `@push.rocks/smartformat` to your project dependencies, allowing you to use it in your project.

Usage

`@push.rocks/smartformat` is a TypeScript module designed to help with formatting things, specifically bytes and milliseconds into a more human-friendly format. It leverages the power of `pretty-bytes` and `pretty-ms` packages to do so. This guide will help you get started with `@push.rocks/smartformat` by providing a series of examples and use cases.

Importing in TypeScript

Firstly, ensure that your project is set up to support TypeScript and ES modules. Then, you can import `@push.rocks/smartformat` into your project as follows:

```
import * as smartformat from '@push.rocks/smartformat';
```

Alternatively, if you only need specific functionalities, you can import them directly:

```
import { prettyBytes, prettyMs } from '@push.rocks/smartformat';
```

Formatting Bytes

When working with file sizes or data amounts, it can be helpful to present these figures in a format that is easier for humans to understand. This is where `prettyBytes` comes into play.

Basic Usage

```
// Convert bytes to a human-readable string:
const fileSize = smartformat.prettyBytes(132480239);
console.log(fileSize); // Outputs: "132.5 MB"
```

Handling Negative Values

`prettyBytes` also gracefully handles negative values, which can be useful in certain contexts like showing the difference between file sizes.

```
const sizeDifference = smartformat.prettyBytes(-827391);
console.log(sizeDifference); // Outputs: "-827.4 kB"
```

Formatting Milliseconds

In many applications, representing time durations in a human-friendly manner is essential. `prettyMs` simplifies this task.

Basic Usage

```
// Convert milliseconds to a human-readable string:
const duration = smartformat.prettyMs(65799000);
console.log(duration); // Outputs: "18h 19m 59s"
```

More Options

`prettyMs` offers several options to customize the output. For example, you can include milliseconds in the output or opt for a compact representation.

```
// Including milliseconds in the output:
const preciseDuration = smartformat.prettyMs(65799450, { verbose: true });
console.log(preciseDuration); // Outputs: "18 hours 19 minutes 59.5 seconds"

// Compact representation:
```

```
const compactDuration = smartformat.prettyMs(12345678, { compact: true });
console.log(compactDuration); // Outputs: "3h 25m"
```

Conclusion

Using `@push.rocks/smartformat`, you've seen how to format bytes and milliseconds into more readable strings. These examples just scratch the surface of what's possible. Both `prettyBytes` and `prettyMs` come with additional options and capabilities worth exploring. Remember, formatting data effectively can significantly improve the user experience of your applications.

Note: The given examples use the ES Module (ESM) syntax and TypeScript, ensuring you're working with the latest standards for JavaScript development.

License and Legal Information

Trademarks

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.

Capital GmbH of any derivative works.