

@push.rocks/smartv alidator

A wrapper for performing validations using the validator package.

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

readme.md for

@push.rocks/smartvalidator

a wrapper for the validator package to perform validations

Install

To use `@push.rocks/smartvalidator` in your project, installation via npm or yarn is straightforward. Run the following command in your project directory:

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

or if you prefer using yarn:

```
yarn add @push.rocks/smartvalidator
```

This will download the package and add it to your `node_modules` folder as well as list it as a dependency in your `package.json` file.

Usage

`@push.rocks/smartvalidator` serves as a convenient wrapper around the popular `validator` library, providing you with additional utilities and simplifications for validation tasks. Below is a comprehensive guide on how to leverage `@push.rocks/smartvalidator` effectively in your TypeScript project.

First, ensure that your module system supports ECMAScript Modules (ESM) syntax and that you have TypeScript configured in your project.

Getting Started

To begin, import `SmartValidator` from the package:

```
import { SmartValidator } from '@push.rocks/smartvalidator';
```

Create an instance of `SmartValidator`:

```
const validator = new SmartValidator();
```

Validating an IBAN

Validating an International Bank Account Number (IBAN) is straightforward:

```
const iban = 'AT95 0100 0000 0555 4915';
if (validator.isIban(iban)) {
  console.log('The IBAN is valid.');
```

```
} else {
  console.log('The IBAN is invalid.');
```

```
}
```

Checking URLs

To check if a string is a valid URL:

```
const url = 'https://example.com';
if (validator.isUrl(url)) {
  console.log('The URL is valid.');
```

```
} else {
```

```
  console.log('The URL is invalid.');
```

```
}
```

Escaping Strings

When you need to escape strings, for instance, to sanitize input data:

```
const userInput = '<script>alert("xss")</script>';
const escapedInput = validator.escape(userInput);
console.log(escapedInput); // Output will have HTML special characters escaped
```

Checking CSV Strings

You can check if a string could potentially be interpreted as CSV data:

```
const csvString = 'name,age,city\nJohn Doe,29,New York';
if (validator.isCsv(csvString)) {
  console.log('The string is likely a CSV data.');
```

```
} else {
  console.log('The string is unlikely to be CSV data.');
```

```
}
```

Underlying Validator Access

The underlying `validator` library can be accessed directly if you need functionalities not directly exposed by `SmartValidator`:

```
const email = 'example@example.com';
if (SmartValidator.validator.isEmail(email)) {
  console.log('The email address is valid.');
```

```
} else {
  console.log('The email address is invalid.');
```

```
}
```

Additional Notes

- Keep in mind that while `@push.rocks/smartvalidator` simplifies certain tasks, not all functionalities of the `validator` library are wrapped. Access the original validator object for advanced use cases.
- For contributions or issues, please refer to the project's GitHub repository.

Note: The usage examples shown above cover basic usage and common validation tasks. For more complex use cases, feel free to explore the original `validator` library's [documentation](#) as `@push.rocks/smartvalidator` simply extends its capabilities.

By integrating `@push.rocks/smartvalidator` into your project, you leverage a streamlined, typed interface for data validation, enhancing both development experience and code maintainability.

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.