

@push.rocks/smartyaml

A module for smart handling of YAML data with support for converting between YAML strings and JavaScript objects.

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

readme.md for @push.rocks/smartyaml

Handle YAML in smart ways.

Install

To use @push.rocks/smartyaml in your project, you need to install it using npm or yarn. Run the following command in your project directory:

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

or if you prefer yarn:

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

This will add @push.rocks/smartyaml to your project's dependencies and enable you to start using the library to handle YAML data in a smart way.

Usage

To utilize @push.rocks/smartyaml in your TypeScript project, simply import the desired functionality from the package. This library streamlines YAML data handling by providing straightforward functions for converting YAML strings to objects and vice versa. Let's dive into how you can leverage these capabilities in your applications.

Basic Conversion: YAML String to JavaScript Object

Imagine you're building an application that needs to process configuration files stored in YAML format. With @push.rocks/smartyaml, converting these YAML strings into JavaScript objects for easy manipulation is a breeze.

First, here's how you import the function to convert a YAML string to a JavaScript object:

```
import { yamlStringToObject } from '@push.rocks/smartyaml';
```

Assuming you have a YAML string like this:

```
name: John Doe
age: 30
address:
  street: 1234 Somewhere Street
  city: Anytown
  zip: '12345'
```

You can convert it to a JavaScript object as follows:

```
const yamlExample = `
name: John Doe
age: 30
address:
  street: 1234 Somewhere Street
  city: Anytown
  zip: '12345'
`;

async function convertYamlToObject() {
  try {
    const yamlObject = await yamlStringToObject(yamlExample);
    console.log(yamlObject);
    /*
    * Output:
    * {
    *   name: 'John Doe',
    *   age: 30,
    *   address: { street: '1234 Somewhere Street', city: 'Anytown', zip: '12345' }
    * }
    */
  } catch (error) {
    console.error(`Error converting YAML to object: ${error}`);
  }
}
```

```
convertYamlToObject();
```

Converting JavaScript Objects back into YAML Strings

Let's say you've now manipulated your configuration data, and you want to save these changes back in a YAML file. [@push.rocks/smartyaml](#) provides a handy function for that:

```
import { objectToYamlString } from '@push.rocks/smartyaml';
```

Considering you have a JavaScript object like this:

```
const jsObject = {
  name: 'John Doe',
  age: 30,
  address: {
    street: '1234 Somewhere Street',
    city: 'Anytown',
    zip: '12345'
  }
};
```

You can convert it to a YAML string as follows:

```
async function convertObjectToYaml() {
  try {
    const yamlString = await objectToYamlString(jsObject);
    console.log(yamlString);
    /*
     * Output:
     * "name: John Doe\nage: 30\naddress:\n  street: 1234 Somewhere Street\n  city:
Anytown\n  zip: '12345'\n"
     */
  } catch (error) {
    console.error(`Error converting object to YAML: ${error}`);
  }
}
```

```
convertObjectToYaml();
```

These simple examples illustrate how [@push.rocks/smartyaml](https://github.com/pushrocks/smartyaml) greatly simplifies the process of converting between YAML strings and JavaScript objects, making it a valuable tool for managing configuration files or any data stored in YAML format in your TypeScript project.

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.