

readme.md for @push.rocks/smartcsv

handle csv data | gitzone standard compliant

Install

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

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

Or, if you prefer using Yarn:

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

Usage

`@push.rocks/smartcsv` is a powerful library designed for handling CSV data efficiently and in a developer-friendly manner. This document will guide you through various use cases and demonstrate the flexibility and capabilities of `@push.rocks/smartcsv`.

Importing the Module

First, import `Csv` from the `@push.rocks/smartcsv` package into your TypeScript file:

```
import { Csv } from '@push.rocks/smartcsv';
```

Creating a CSV Instance from a String

You can create a `Csv` instance directly from a string. This is particularly useful when you have CSV data as a string from a file or an API response:

```
const csvString = `Name, Age, Occupation\nJohn Doe, 28, Software Developer\nJane Smith, 32,
Data Scientist`;

const csvOptions = {
  headers: true, // indicates the first row contains headers
  unquote: true, // unquote values if necessary
  trimSpace: true, // trim spaces around values
  removeBomMarkers: true // remove BOM markers if present
};

const csvInstance = await Csv.createCsvFromString(csvString, csvOptions);
```

Exporting CSV Data as JSON

Once you have a `Csv` instance, you can convert the CSV data into JSON objects for easier manipulation and access:

```
const jsonData = await csvInstance.exportAsObject();
console.log(jsonData);
```

This will output:

```
[
  { "Name": "John Doe", "Age": "28", "Occupation": "Software Developer" },
  { "Name": "Jane Smith", "Age": "32", "Occupation": "Data Scientist" }
]
```

Creating a CSV String from an Array of Objects

If you have an array of objects and wish to convert it into a CSV string, [@push.rocks/smartcsv](https://github.com/pushrocks/smartcsv) offers a straightforward approach:

```
const arrayOfObjects = [
  { Name: "John Doe", Age: 28, Occupation: "Software Developer" },
  { Name: "Jane Smith", Age: 32, Occupation: "Data Scientist" }
];
```

```
const csvString = await Csv.createCsvStringFromArray(arrayOfObjects);
console.log(csvString);
```

This will generate a CSV string that represents the array of objects provided.

Advanced Usage

Custom Separator Detection

By default, `@push.rocks/smartcsv` intelligently detects the separator used in the input CSV data (either a comma `,` or a semicolon `;`). You can rely on this automatic detection or specify preferred options for finer control over how your CSV data is handled.

Handling Quotes and BOM Markers

The library is capable of processing CSV files that include quoted fields, optionally removing quotes where necessary. Similarly, it can handle Byte Order Mark (BOM) markers at the start of UTF-8 encoded files, ensuring that the text is interpreted correctly.

Trimming Spaces

For data cleanliness, `@push.rocks/smartcsv` can trim leading and trailing spaces from values in the CSV, ensuring that the resulting data is neat and uniform.

Conclusion

`@push.rocks/smartcsv` offers a comprehensive set of features for working with CSV data in TypeScript, providing flexibility, ease of use, and powerful data manipulation capabilities. Whether you're importing CSV data from a file, converting JSON to CSV, or manipulating CSV strings directly in your application, `@push.rocks/smartcsv` has you covered.

For detailed API documentation, please refer to the [official documentation](#).

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.

Revision #3

Created 2026-03-28 11:10:30 UTC by foss.global Team

Updated 2026-03-28 12:17:15 UTC by foss.global Team