@push.rocks/smarturl

A library for parsing URLs in a detailed and flexible way.

- readme.md for @push.rocks/smarturl
- changelog.md for @push.rocks/smarturl

readme.md for @push.rocks/smarturl

a url parsing lib

Install

To install `@push.rocks/smarturl`, you'll need Node.js and npm on your system. Once you have those set up, run the following command in your terminal:

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

This will add `@push.rocks/smarturl` to your project's dependencies.

Usage

`@push.rocks/smarturl` is a library designed to simplify the parsing and manipulation of URLs within your TypeScript projects. Below is an overview of how you can leverage this library effectively in various scenarios.

Basic URL Parsing

To start parsing URLs, you first need to import `@push.rocks/smarturl` into your TypeScript file:

```
import { Smarturl } from '@push.rocks/smarturl';
```

Now, you can parse a URL string and access its components:

```
const myUrl = Smarturl.createFromUrl('https://www.example.com:8080/path?query=string#hash');

console.log(myUrl.href); // https://www.example.com:8080/path?query=string#hash
console.log(myUrl.protocol); // https:
```

```
console.log(myUrl.host); // www.example.com:8080
console.log(myUrl.pathname); // /path
console.log(myUrl.search); // ?query=string
console.log(myUrl.hash); // #hash
console.log(myUrl.searchParams); // { query: 'string' }
```

Modifying Search Parameters

`@push.rocks/smarturl` makes it easy to modify search parameters of a URL:

```
// Creating a URL with initial search parameters
const myUrl = Smarturl.createFromUrl('https://www.example.com', {
  searchParams: {
    page: '1',
    filter: 'none',
  },
});

// Adding or updating search parameters
myUrl.searchParams['filter'] = 'newVal';
myUrl.searchParams['newParam'] = 'value';

console.log(myUrl.toString()); //
https://www.example.com:443/?page=1&filter=newVal&newParam=value
```

Complex URL Creation

Sometimes you may need to create a URL from parts:

```
let myUrl = new Smarturl();
myUrl.protocol = 'https';
myUrl.hostname = 'www.example.com';
myUrl.port = '3000';
myUrl.pathname = '/path/to/resource';
myUrl.searchParams = {
  key: 'value',
  another: 'parameter'
};
```

```
console.log(myUrl.toString()); // Prints the full URL
```

Handling Username and Password in URL

You can also include authentication details within the URL:

```
const myUrl = Smarturl.createFromUrl('https://user:password@www.example.com');
console.log(myUrl.username); // user
console.log(myUrl.password); // password
```

Working with URL Paths and Hashes

Manipulating paths and hashes is straightforward:

```
const myUrl = Smarturl.createFromUrl('https://www.example.com/path/to/resource');
myUrl.hash = 'section1';

console.log(myUrl.toString()); // https://www.example.com:443/path/to/resource#section1
```

URL Encoding and Decoding

When working with URLs, encoding and decoding of search parameters is handled automatically by the library. This ensures that the URL remains valid and interpretable by browsers and servers.

```
const myUrl = Smarturl.createFromUrl('https://www.example.com');
myUrl.searchParams['redirect_uri'] = 'https://www.redirect.com/path';

console.log(myUrl.toString()); // Automatically encodes the URI
```

In addition to basic parsing and manipulation, `@push.rocks/smarturl` provides methods for deep manipulation of URLs, aiding in scenarios where complex URL operations are needed.

By understanding and utilizing these features, you can handle most URL-related tasks within your TypeScript applications with ease, ensuring that URLs are constructed, interpreted, and modified in a consistent and error-free manner.

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.

changelog.md for @push.rocks/smarturl

2024-10-03 - 3.1.0 - feat(Smarturl)

Enhanced documentation and added method for property setting and chaining in Smarturl

- Added detailed comments to methods and interfaces for better code understanding.
- Introduced a method to set properties in a chainable manner.

2024-10-03 - 3.0.8 - fix(core)

Fix code formatting and indentation issues in smarturl.classes.smarturl.ts

- Improved code readability by fixing indentation and ensuring consistent formatting.

2022-07-27 - 3.0.0 to 3.0.7 - Core Updates

Maintenance and enhancement updates across multiple minor versions.

- **3.0.0**: Core updated with foundational changes.
- **3.0.1**: Further core updates with minor fixes.
- **3.0.2**: Continued enhancements and core updates.
- **3.0.3**: Additional core updates.
- **3.0.4**: Continued improvements in core functions.
- **3.0.5**: Ongoing core enhancements and updates.
- **3.0.6**: Maintenance improvements and stable updates.

2022-07-27 - 2.0.2 to 3.0.0 - Major Transition

Introduction of significant changes to the core.

- **2.0.2**: Introduced breaking changes with a switch to ESM (ECMAScript Modules).

2021-05-02 - 1.0.8 to 2.0.2 - Stability Updates

Updates focused on core stability and functionality.

- **1.0.8**: Breaking core updates were introduced to stabilize the platform.
- **2.0.0**: Core was updated for greater efficiency.
- **2.0.1**: Minor core improvements and updates.

2021-04-12 - 1.0.3 to 1.0.8 - Core Maintenance

Multiple updates for improving the core functionality.

- Incremental updates from 1.0.3 through 1.0.7 enhancing core stability.

2020-03-27 - 1.0.1 to 1.0.2 - Initial Core Enhancements

Early updates focusing on core aspects.

- Introduction of minor core improvements and fixes.

2015-10-21 - 0.0.1 to 0.0.4 - Project Initialization

Initial setup and foundational updates for the project.

- **0.0.1**: Project initiation and Travis integration.
- **0.0.3**: Creation of initial test infrastructure.
- **0.0.4**: Basic project setup completed.