@push.rocks/smartrobots

A module for parsing and making sense of robots.txt files.

readme.md for @push.rocks/smartrobots

@push.rocks/smartrobots

a module for making sense of robots.txt

Install

To install `@push.rocks/smartrobots`, you need to have Node.js installed on your system. Once Node.js is installed, you can install `@push.rocks/smartrobots` by running the following command in your terminal:

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

This command will download and install `@push.rocks/smartrobots` and its dependencies into your project's `node_modules` directory.

Usage

This tutorial will guide you through utilizing `@push.rocks/smartrobots`, a TypeScript module designed for interpreting `robots.txt` files of websites. By working through various scenarios, you'll learn how to efficiently parse and work with the data provided by `robots.txt`.

Prerequisites

Before diving into the examples, ensure you are familiar with basic TypeScript syntax and concepts. You also need a basic understanding of what `robots.txt` is and its significance in web development.

Basic Setup

First, ensure you have imported `Smartrobots` from the `@push.rocks/smartrobots` package in your TypeScript file:

```
import { Smartrobots } from '@push.rocks/smartrobots';
```

Instantiate the `Smartrobots` class to get started:

```
const mySmartrobots = new Smartrobots();
```

Parsing `robots.txt` from a URL

One common use case is to parse the `robots.txt` file directly from a website. The `Smartrobots` class provides an easy-to-use method to achieve this, as demonstrated below:

```
async function parseRobotsFromUrl() {
  const url = 'https://example.com/robots.txt'; // Replace with the URL to the desired
robots.txt
  try {
    const parsedData = await mySmartrobots.parseRobotsTxtFromUrl(url);
    console.log('Parsed robots.txt data:', parsedData);
  } catch (error) {
    console.error('Error parsing robots.txt from URL:', error);
  }
}

parseRobotsFromUrl();
```

This function asynchronously fetches the `robots.txt` file from the specified URL and logs the parsed content to the console.

Parsing a `robots.txt` String

If you already have the contents of a `robots.txt` file as a string, you can parse it directly using the `parseRobotsTxt` method. Here's how:

```
async function parseRobotsFromString(robotsTxtString: string) {
  try {
```

```
    const parsedData = await mySmartrobots.parseRobotsTxt(robotsTxtString);
    console.log('Parsed robots.txt data:', parsedData);
  } catch (error) {
    console.error('Error parsing robots.txt string:', error);
  }
}

// Example robots.txt string
const robotsTxtString = `
User-agent: *
Disallow: /secret-page
Sitemap: https://example.com/sitemap.xml
`;

parseRobotsFromString(robotsTxtString);
```

This function takes a string representation of a `robots.txt` file, parses it, and logs the results. In the example string, there are directives for user-agents and a sitemap URL.

Understanding the Parsed Data

The parsed data from `robots.txt` is returned as an object. In its current implementation, `@push.rocks/smartrobots` focuses on extracting sitemap URLs. Here's a sample output from parsing the example `robots.txt` string:

```
{
  "sitemaps": ["https://example.com/sitemap.xml"]
}
```

You can extend the parsing logic based on your requirements to handle more directives from `robots.txt`.

Conclusion

`@push.rocks/smartrobots` provides a straightforward and efficient way to interpret `robots.txt` files in TypeScript projects. Whether you're fetching and parsing `robots.txt` from a URL or working with its contents as a string, this module simplifies the process, allowing you to focus on utilizing the data rather than parsing intricacies.

Remember, `robots.txt` files are publicly accessible and should be used responsibly following web standards and etiquette.

For more advanced use cases, consider contributing to or extending the functionality of `@push.rocks/smartrobots` to cover a broader range of directives and scenarios.

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.