

readme.md for @push.rocks/smartwhois

a package for dealing with whois requests

Install

To install `@push.rocks/smartwhois`, you can use npm or yarn. First, open your terminal, then execute one of the following commands in your project directory:

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

or

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

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

Usage

The `@push.rocks/smartwhois` module provides an intuitive TypeScript interface for dealing with WHOIS requests. It allows you to gather WHOIS information for domains, including registration details, registrar information, and more. For using this module, it is highly recommended to be familiar with TypeScript and async/await patterns for promise handling.

Getting Started

First, ensure that you import the necessary classes from the module. The primary class you will interact with is `SmartWhois`.

```
import { SmartWhois } from '@push.rocks/smartwhois';
```

```
// Create an instance of SmartWhois
const smartWhois = new SmartWhois();
```

Fetching WHOIS Information for a Domain

The most common operation is to fetch WHOIS information for a given domain. This can easily be done using the `getWhoisForDomain` method, which returns detailed WHOIS information.

```
async function getDomainWhois() {
  const domainInfo = await smartWhois.getWhoisForDomain('example.com');
  console.log(domainInfo);
}
getDomainWhois();
```

Determining Domain Delegation

To understand the structure of a domain or URL, you can use `getDomainDelegation`. This method provides insights into various components of a domain or URL, such as its TLD, domain name, etc.

```
async function checkDomainDelegation() {
  const delegationInfo = await smartWhois.getDomainDelegation('https://www.example.com/path');
  console.log(delegationInfo);
}
checkDomainDelegation();
```

Additional WHOIS Data

For some domains, additional WHOIS data beyond the basics might be useful. This includes HTTPS status, headers, and more. Retrieving additional data is straightforward:

```
async function fetchAdditionalData() {
  const additionalData = await smartWhois.getAdditionalWhoisDataForDomain('example.com');
  console.log(additionalData);
}
fetchAdditionalData();
```

Validating TLDs

When working with various top-level domains (TLDs), you might want to verify if a given TLD is valid. The `isValidTld` method helps check the validity of a TLD:

```
async function isTldValid() {
  const isValid = await smartWhois.isValidTld('.com');
  console.log(`Is .com a valid TLD? ${isValid}`);
}
isTldValid();
```

Handling Errors

When making requests, there is always a possibility of running into errors due to network issues, invalid domains, or other reasons. It's crucial to properly handle these cases:

```
async function safelyGetWhois() {
  try {
    const whoisInfo = await smartWhois.getWhoisForDomain('nonexistentdomain.xyz');
    console.log(whoisInfo);
  } catch (error) {
    console.error('Failed to fetch WHOIS info:', error);
  }
}
safelyGetWhois();
```

Integration with Other Systems

The retrieved WHOIS data can be easily integrated into other parts of your application or system. For instance, you could store the data in a database, use it to monitor domain registration changes, or create alerts for domain expiry dates.

Advanced Usage

While the above examples cover the basics, [@push.rocks/smartwhois](https://github.com/pushrocks/smartwhois) offers much more. Explore the source code and TypeScript definitions to uncover additional functionality, such as parsing complex WHOIS server responses, handling internationalized domain names (IDNs), and more.

Summary

`@push.rocks/smartwhois` is a powerful tool for querying WHOIS information, designed with modern development practices in mind. By leveraging TypeScript, it offers a robust and type-safe way to integrate WHOIS data into your applications, enriching your projects with valuable domain-related insights.

For further information and the latest updates, refer to the official documentation and GitHub repository.

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 11:12:43 UTC by foss.global Team

Updated 2026-03-28 12:19:30 UTC by foss.global Team