

@push.rocks/smartlegal

A module for handling legal aspects of software projects programmatically, including license checking.

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

readme.md for @push.rocks/smartlegal

go legal programmatically

Install

You can install `@push.rocks/smartlegal` using npm:

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

Or using Yarn:

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

Ensure you have TypeScript and the necessary TypeScript types installed for optimal development experience:

```
npm install typescript @types/node --save-dev
```

Usage

`@push.rocks/smartlegal` offers a streamlined approach to handling legal operations programmatically. It is designed with ease-of-use in mind, allowing developers to integrate legal checks into their workflow. The primary feature as showcased in the current version is a license checker, which is invaluable for projects that need to ensure compliance with legal requirements regarding dependencies' licenses.

Getting Started

To use `@push.rocks/smartlegal`, first import the module in your TypeScript project:

```
import { createLicenseChecker } from '@push.rocks/smartlegal';
```

Checking Licenses

The module provides functions to either include or exclude specific licenses within your project's dependencies, making it easier to manage legal compliance.

Excluding Specific Licenses

If you want to exclude specific licenses, you may use the `excludeLicenseWithinPath` function. This function checks all dependencies and lists those not compliant with the specified licenses.

```
(async () => {
  const licenseChecker = await createLicenseChecker();
  const checkResult = await licenseChecker.excludeLicenseWithinPath('./', ['MIT']);
  console.log('Passing modules (compliant):', checkResult.passingModules);
  console.log('Failing modules (non-compliant):', checkResult.failingModules);
})();
```

Including Specific Licenses

Conversely, to ensure only dependencies with certain licenses are used, employ the `includeLicencesWithinPath` function.

```
(async () => {
  const licenseChecker = await createLicenseChecker();
  const checkResult = await licenseChecker.includeLicencesWithinPath('./', ['MIT']);
  console.log('Passing modules (compliant):', checkResult.passingModules);
  console.log('Failing modules (non-compliant):', checkResult.failingModules);
})();
```

Understanding the Results

Both `excludeLicenseWithinPath` and `includeLicencesWithinPath` methods return a `CheckResult` object containing two arrays: `passingModules` and `failingModules`.

- `passingModules` includes all dependencies that meet your specified criteria (either included or excluded licenses).
- `failingModules` lists those dependencies that do not meet the criteria, aiding in identifying potential legal compliance issues quickly.

Integration into CI/CD Pipelines

`@push.rocks/smartlegal` is designed to be easily integrated into CI/CD pipelines, enabling automated checks for license compliance. By incorporating the above examples into your build or deployment scripts, you can halt the pipeline if non-compliant dependencies are detected, ensuring legal compliance at each stage of development.

Conclusion

`@push.rocks/smartlegal` provides an essential tool for managing the legal aspects of your project's dependencies. By leveraging its functionality, developers can ensure compliance with legal requirements, minimizing risks associated with software licenses. As projects evolve and dependencies change, `@push.rocks/smartlegal` simplifies the ongoing maintenance of legal compliance, making it an invaluable addition to any 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.