

# @push.rocks/smartupdate

A library designed to facilitate smarter update notifications and checking for Node.js projects.

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

# readme.md for @push.rocks/smartupdate

update your tools in a smart way

## Install

To get started with `@push.rocks/smartupdate`, you need to install it via npm. Run the following command in your terminal:

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

This will add `@push.rocks/smartupdate` as a dependency to your project and download it to your `node_modules` folder.

## Usage

The `@push.rocks/smartupdate` module offers a smart way to notify users of your CLI tools or libraries about new versions available on npm. Let's take a closer look at how you can utilize it in your project using TypeScript.

## Importing and Instantiating `SmartUpdate`

First, you need to import the `SmartUpdate` class from the `@push.rocks/smartupdate` package and create an instance of it. You can optionally pass settings to the constructor to customize its behavior:

```
import { SmartUpdate } from '@push.rocks/smartupdate';  
  
const smartUpdate = new SmartUpdate();
```

## Checking for Updates

To check for an update for a specific npm package, you can use the `check` method. This method requires the npm package name, the current local version of the package you are checking against, and optionally, a URL to the changelog. If a newer version is found, it will log a message to the console and, if a changelog URL is provided and not running in a CI environment, attempt to open the changelog in the default web browser.

Here's an example:

```
// Async function to demonstrate usage
async function checkForUpdates() {
  const npmPackageName = 'some-npm-package';
  const currentVersion = '1.0.0';
  const changelogUrl = 'https://example.com/changelog';

  // Check for an update for 'some-npm-package'
  const hasNewerVersion = await smartUpdate.check(npmPackageName, currentVersion,
changelogUrl);

  if (hasNewerVersion) {
    console.log('A newer version is available. Please consider updating.');
```

## Checking for CLI Updates

For CLI applications, `@push.rocks/smartupdate` offers a more tailored method called `checkForCli`. This method works similarly to `check` but is optimized for CLI tools, taking into account factors like avoiding frequent checks (e.g., not more than once per hour).

Example usage:

```
// Async function to demonstrate CLI update checks
async function checkCliUpdates() {
  const cliPackageName = 'your-cli-tool';
  const currentCliVersion = '0.1.0';
  const changelogUrl = 'https://example.com/cli-changelog';
```

```
const needsUpdate = await smartUpdate.checkForCli(cliPackageName, currentCliVersion,
changelogUrl);

if (needsUpdate) {
  console.log(`A newer version of ${cliPackageName} is available. Visit ${changelogUrl} for
more information.`);
} else {
  console.log(`You are using the latest version of ${cliPackageName}.`);
}
}

// Run the CLI update check
checkCliUpdates().catch(console.error);
```

By integrating `@push.rocks/smartupdate` into your project, you ensure users are always informed about the latest updates, encouraging them to keep their installations current and benefiting from new features, performance improvements, and bug fixes.

## 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](mailto: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.