


---

## update-notifier

Update notifications for your CLI app



```
Update available 0.9.2 → 1.2.0
Run npm i -g public-ip to update
```

Inform users of your package of updates in a non-intrusive way.

### Contents

- Install
- Usage
- How
- API
- About
- Users

### Install

```
1 npm install update-notifier
```

### Usage

#### Simple

```
1 import updateNotifier from 'update-notifier';
2 import packageJson from './package.json' assert {type: 'json'};
3
4 updateNotifier({pkg: packageJson}).notify();
```

---

## Comprehensive

```
1 import updateNotifier from 'update-notifier';
2 import packageJson from './package.json' assert {type: 'json'};
3
4 // Checks for available update and returns an instance
5 const notifier = updateNotifier({pkg: packageJson});
6
7 // Notify using the built-in convenience method
8 notifier.notify();
9
10 // `notifier.update` contains some useful info about the update
11 console.log(notifier.update);
12 /*
13 {
14   latest: '1.0.1',
15   current: '1.0.0',
16   type: 'patch', // Possible values: latest, major, minor, patch,
17                 prerelease, build
18   name: 'pageres'
19 }
20 */
```

## Options and custom message

```
1 const notifier = updateNotifier({
2   pkg,
3   updateCheckInterval: 1000 * 60 * 60 * 24 * 7 // 1 week
4 });
5
6 if (notifier.update) {
7   console.log(`Update available: ${notifier.update.latest}`);
8 }
```

## How

Whenever you initiate the update notifier and it's not within the interval threshold, it will asynchronously check with npm in the background for available updates, then persist the result. The next time the notifier is initiated, the result will be loaded into the `.update` property. This prevents any impact on your package startup performance. The update check is done in a unref'ed child process. This means that if you call `process.exit`, the check will still be performed in its own process.

The first time the user runs your app, it will check for an update, and even if an update is available, it will wait the specified `updateCheckInterval` before notifying the user. This is done to not be

---

annoying to the user, but might surprise you as an implementer if you're testing whether it works. Check out [example.js](#) to quickly test out [update-notifier](#) and see how you can test that it works in your app.

## API

### **notifier = updateNotifier(options)**

Checks if there is an available update. Accepts options defined below. Returns an instance with an [.update](#) property if there is an available update, otherwise [undefined](#).

### **options**

Type: [object](#)

**pkg** Type: [object](#)

**name** *Required*  
Type: [string](#)

**version** *Required*  
Type: [string](#)

**updateCheckInterval** Type: [number](#)  
Default: `1000 * 60 * 60 * 24` (1 day)  
How often to check for updates.

**shouldNotifyInNpmScript** Type: [boolean](#)  
Default: [false](#)  
Allows notification to be shown when running as an npm script.

**distTag** Type: [string](#)  
Default: ['latest'](#)  
Which dist-tag to use to find the latest version.

---

## **notifier.fetchInfo()**

Check update information.

Returns an **object** with:

- **latest** (*string*) - Latest version.
- **current** (*string*) - Current version.
- **type** (*string*) - Type of current update. Possible values: **latest**, **major**, **minor**, **patch**, **prerelease**, **build**.
- **name** (*string*) - Package name.

## **notifier.notify(options?)**

Convenience method to display a notification message. (See *screenshot*)

Only notifies if there is an update and the process is TTY.

**options**   Type: **object**

**defer**    Type: **boolean**

Default: **true**

Defer showing the notification to after the process has exited.

**message**   Type: **string**

Default: See above screenshot

Message that will be shown when an update is available.

Available placeholders:

- **{packageName}** - Package name.
- **{currentVersion}** - Current version.
- **{latestVersion}** - Latest version.
- **{updateCommand}** - Update command.

```
1 notifier.notify({message: 'Run `{updateCommand}` to update.'});
2
3 // Output:
4 // Run `npm install update-notifier-tester@1.0.0` to update.
```

---

**isGlobal** Type: **boolean**

Default: Auto-detect

Include the `-g` argument in the default message's `npm i` recommendation. You may want to change this if your CLI package can be installed as a dependency of another project, and don't want to recommend a global installation. This option is ignored if you supply your own `message` (see above).

**boxenOptions** Type: **object**

Default: `{padding: 1, margin: 1, textAlign: 'center', borderColor: 'yellow', borderStyle: 'round'}` (See *screenshot*)

Options object that will be passed to `boxen`.

## User settings

Users of your module have the ability to opt-out of the update notifier by changing the `optOut` property to `true` in `~/.config/configstore/update-notifier-[your-module-name].json`. The path is available in `notifier.config.path`.

Users can also opt-out by setting the environment variable `NO_UPDATE_NOTIFIER` with any value or by using the `--no-update-notifier` flag on a per run basis.

The check is also skipped automatically: - in CI - in unit tests (when the `NODE_ENV` environment variable is `test`)

## About

The idea for this module came from the desire to apply the browser update strategy to CLI tools, where everyone is always on the latest version. We first tried automatic updating, which we discovered wasn't popular. This is the second iteration of that idea, but limited to just update notifications.

## Users

There are a bunch of projects using it:

- npm - Package manager for JavaScript
- Yeoman - Modern workflows for modern webapps
- AVA - Simple concurrent test runner
- XO - JavaScript happiness style linter
- Node GH - GitHub command line tool

---

And 2700+ more...