

readme.md for @push.rocks/projectinfo

A tool for gathering project information, supporting npm, git, and more.

Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit community.foss.global/. This is the central community hub for all issue reporting. Developers who sign and comply with our contribution agreement and go through identification can also get a code.foss.global/ account to submit Pull Requests directly.

Install

Install via npm:

```
npm install @push.rocks/projectinfo
```

Or with pnpm:

```
pnpm install @push.rocks/projectinfo
```

This module is ESM-only and requires Node.js 18+.

Usage

`@push.rocks/projectinfo` provides an async API for extracting npm and git metadata from any project directory. It uses `@push.rocks/smartfs` under the hood for filesystem operations.

All classes use async factory methods — no synchronous constructors.

Quick Start

```
import { ProjectInfo, ProjectinfoNpm, getNpmNameForDir } from '@push.rocks/projectinfo';
```

Get Full Project Info

The `ProjectInfo` class is the main entry point. It gathers both npm and git info from a given directory:

```
import { ProjectInfo } from '@push.rocks/projectinfo';

const info = await ProjectInfo.create('./my-project');

// npm info
console.log(info.npm.name);           // e.g. "@push.rocks/projectinfo"
console.log(info.npm.version);       // e.g. "5.0.2"
console.log(info.npm.license);       // e.g. "MIT"
console.log(info.npm.isNpm);         // true if package.json exists

// git info (parsed from directory path)
console.log(info.git.gitrepo);       // e.g. "projectinfo"
console.log(info.git.gituser);       // e.g. "push.rocks"

// project type
console.log(info.type);               // "npm" if package.json found, otherwise "git"
```

NPM Info Only

Use `ProjectinfoNpm` directly when you only need package.json data:

```
import { ProjectinfoNpm } from '@push.rocks/projectinfo';

const npm = await ProjectinfoNpm.create('/path/to/project');

if (npm.isNpm) {
  console.log(npm.name);              // package name
  console.log(npm.version);           // package version
  console.log(npm.license);           // license field
}
```

```
console.log(npm.packageJson); // full parsed package.json object
}
```

With Git Access Token

If your `package.json` has a `repository` field, `ProjectinfoNpm` will parse it into a `GitRepo` object. You can provide an access token for authenticated URLs:

```
const npm = await ProjectinfoNpm.create('/path/to/project', {
  gitAccessToken: 'ghp_your_token_here',
});

if (npm.git) {
  console.log(npm.git.httpsUrl); // e.g.
  "https://ghp_your_token_here@github.com/user/repo.git"
}
```

Git Info Only

Use `ProjectinfoGit` for path-based git metadata extraction:

```
import { ProjectinfoGit } from '@push.rocks/projectinfo';

// Parses the directory path to extract git user and repo
const git = new ProjectinfoGit('/home/user/repos/myorg/myrepo');

console.log(git.gituser); // "myorg"
console.log(git.gitrepo); // "myrepo"
```

You can also parse a remote URL:

```
git.getGitInfoFromRemote('git+https://github.com/someuser/somerepo.git');
```

Quick Helper: Get Package Name

A convenience function to quickly get the npm package name from a directory:

```
import { getNpmNameForDir } from '@push.rocks/projectinfo';
```

```
const name = await getNpmNameForDir('./my-project');
console.log(name); // e.g. "@push.rocks/projectinfo"
```

Returns `undefined` if no valid `package.json` is found.

API Reference

Class / Function	Description
<code>ProjectInfo.create(cwd)</code>	Async factory — gathers both npm + git info
<code>ProjectinfoNpm.create(cwd, opts?)</code>	Async factory — parses <code>package.json</code>
<code>ProjectinfoGit(cwd)</code>	Sync constructor — extracts git info from path
<code>getNpmNameForDir(cwd)</code>	Async helper — returns the npm package name

Types

```
type TProjectType = 'git' | 'npm';
```

License and Legal Information

This repository contains open-source code licensed under the MIT License. A copy of the license can be found in the [LICENSE](#) file.

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 or third parties, 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 or the guidelines of the respective third-party owners, and any usage must be approved in writing. Third-party trademarks used herein are the property of their respective owners and used only in a descriptive manner, e.g. for an implementation of an API or similar.

Company Information

Task Venture Capital GmbH Registered at District Court Bremen HRB 35230 HB, Germany

For any legal inquiries or 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 11:09:56 UTC by foss.global Team

Updated 2026-03-28 12:16:33 UTC by foss.global Team