

# @push.rocks/smartp npm

A library for integrating pnpm package management capabilities within your code.

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

# readme.md for @push.rocks/smartpnp

use pnpm in your code

## Install

To add `@push.rocks/smartpnp` to your project, run the following command:

```
pnpm add @push.rocks/smartpnp --save
```

Ensure that you have `pnpm` installed globally on your system. If not, you can install it via npm with:

```
npm install -g pnpm
```

## Usage

This TypeScript guide assumes you have basic knowledge of TypeScript and its ecosystem. To get the most out of `@push.rocks/smartpnp`, ensure your environment is set up for TypeScript execution.

## Getting Started

First, import `SmartPnpm` from `@push.rocks/smartpnp` in your TypeScript file:

```
import { SmartPnpm } from '@push.rocks/smartpnp';
```

Instantiate `SmartPnpm` with the directory path of the project you want to manage with pnpm. This is typically the root directory of your project:

```
const smartPnpm = new SmartPnpm('/path/to/your/project');
```

# Retrieving Dependency License Information

One of the core features of `@push.rocks/smartnpm` is the ability to retrieve license information for all dependencies within a project. This can be especially useful for license compliance and audit purposes.

## Get Dependency License JSON

To get an overview of the licenses used by your project's dependencies in JSON format, use the `getDependencyLicenseJson` method:

```
async function printDependencyLicenses() {
  const licenseInfo = await smartPnpm.getDependencyLicenseJson();
  console.log(licenseInfo);
}

printDependencyLicenses();
```

This method returns a JSON object where each key is a license type, and the value is an array of packages licensed under that type.

## Print Dependency License Summary

To print a summary of the license usage directly to the console, the `printDependencyLicenseSummary` method can be utilized:

```
async function printLicenseSummary() {
  await smartPnpm.printDependencyLicenseSummary();
}

printLicenseSummary();
```

This method prints a concise summary, making it easy to see how many packages are using each license type.

## Get License Information as Flat Array

If you prefer to work with a flat array of all dependencies, regardless of their license type, you can use the `getDependencyLicenseFlatArray` method:

```
async function getLicensesFlatArray() {
  const licenseArray = await smartPnpm.getDependencyLicenseFlatArray();
  console.log(licenseArray);
}

getLicensesFlatArray();
```

This method returns an array where each element is an object containing the license information of a single package.

## Retrieve All License Types

To get an array of all the license types used by the dependencies of your project, you can use the `getDependencyLicenseArray` method:

```
async function getLicenseTypes() {
  const licenseTypes = await smartPnpm.getDependencyLicenseArray();
  console.log(licenseTypes);
}

getLicenseTypes();
```

This method provides a simple list of all unique license types found in your project's dependencies.

## Advanced Use Cases

`@push.rocks/smartpnpm` offers flexibility for integrating pnpm functionality into your code. Here are advanced scenarios you might consider:

- **Automating License Checks:** Integrate `@push.rocks/smartpnpm` into your CI/CD pipeline to ensure that only compatible licenses are used in your project.
- **Custom Reporting:** Use the provided methods to generate custom reports or dashboards showcasing your project's compliance with licensing requirements.
- **Dependency Management Automation:** Beyond license checks, you can extend the functionality to automate other dependency management tasks, combining the power of `@push.rocks/smartshell` with `@push.rocks/smartpnpm`.

## Conclusion

`@push.rocks/smartpnpm` facilitates direct interaction with pnpm, enhancing your project's dependency management and compliance processes within a TypeScript environment. By

leveraging its capabilities, you can better manage your project's dependencies, ensure compliance with license requirements, and automate your workflows effectively.

For specific use cases or troubleshooting, refer to the official `pnpm` documentation alongside this guide to address your project's particular needs.

# 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.