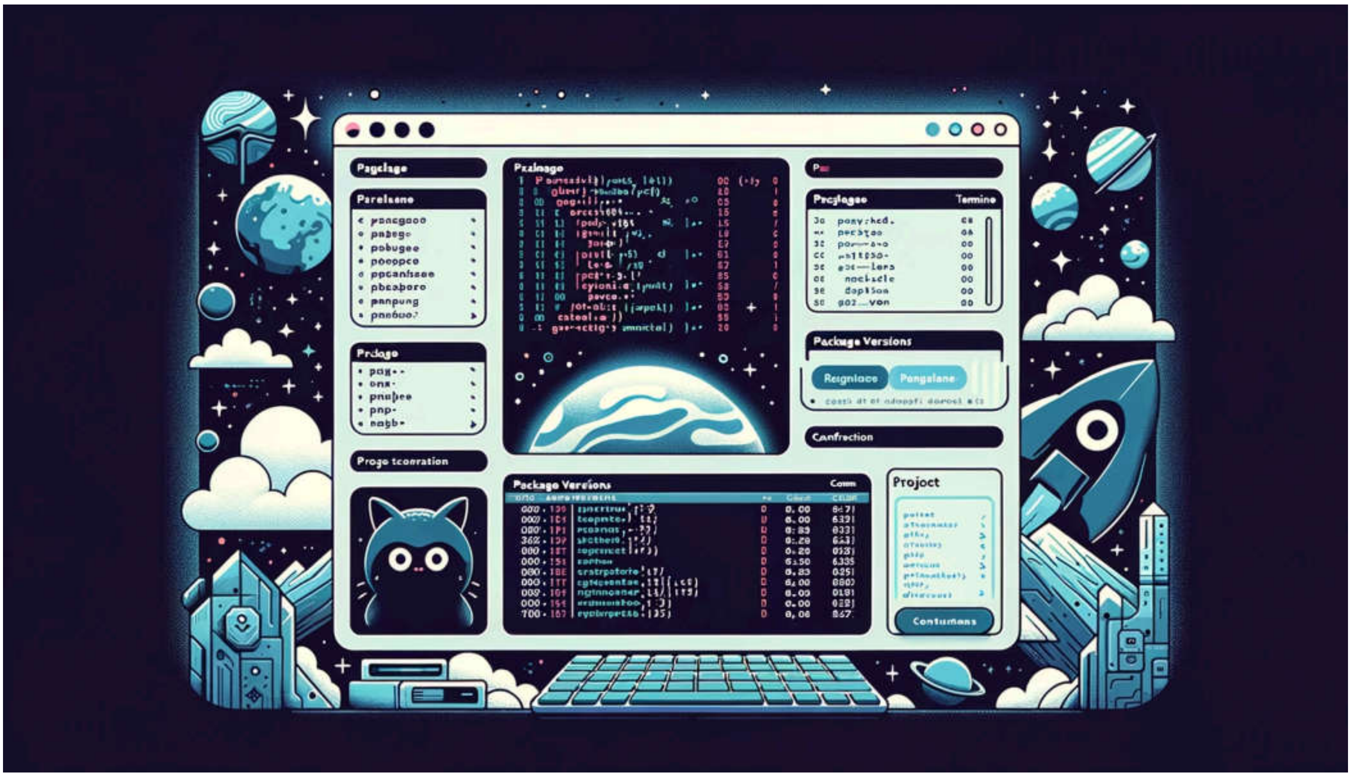
What Is Pnpm? The Fast And Efficient Nodejs Package Manager.

什么是 Pnpm？快速高效的 Nodejs 包管理器。

OCTOBER 16, 2023 10月 16， 2023

Photo of utho

WRITTEN BY TRISTAN 作者 TRISTAN



Pnpm Is A Super-Fast And Disk Space Efficient Alternative To Npm And Yarn For Node.Js. Learn How Pnpm Works And Why It's Better For Modern JavaScript Projects.

Pnpm 是 Npm 和 Yarn For Node.Js 的超快速且节省空间的替代方案。了解 Pnpm 的工作原理以及为什么它更适合现代 JavaScript 项目。

Introduction 介绍

JavaScript developers today have access to the largest collection of open source packages in history via npm, the default package manager for Node.js. However, as JavaScript projects grow in complexity, managing dependencies and dealing with disk space bloat is becoming a pain point. This is where pnpm comes in – a fast, disk space efficient alternative package manager designed specifically for modern JavaScript applications.

如今，JavaScript 开发人员可以通过 npm（Node.js 的默认包管理器）访问历史上最大的开源包集合。然而，随着 JavaScript 项目复杂性的增加，管理依赖项和处理磁盘空间膨胀正在成为一个痛点。这就是 pnpm 的用武之地 - 一个快速、磁盘空间高效的替代包管理器，专为现代 JavaScript 应用程序设计。

pnpm (pronounced: p-npm) offers game-changing improvements in speed, disk usage, and security over both npm and Yarn. While relatively new, it has quickly become a favorite among JavaScript developers working on large projects.

pnpm（发音：p-npm）在速度、磁盘使用和安全性方面比 npm 和 Yarn 都有了改变游戏规则的改进。虽然相对较新，但它已迅速成为从事大型项目的 JavaScript 开发人员的最爱。

In this post, we will understand what pnpm is, how it works, its key benefits, and why JavaScript developers should consider switching to pnpm as their package manager of choice.

在这篇文章中，我们将了解什么是 pnpm、它是如何工作的、它的主要好处，以及为什么 JavaScript 开发人员应该考虑切换到 pnpm 作为他们选择的包管理器。

What Is Pnpm? 什么是 Pnpm?

pnpm stands for “performant npm”. It is a Node.js package manager alternative to npm and Yarn that installs project dependencies using a content-addressable file system.

pnpm 代表 “performant npm”。它是 npm 和 Yarn 的Node.js包管理器替代方案，它使用内容可寻址文件系统安装项目依赖项。

Instead of copying packages into `node_modules` like npm does, pnpm symlinks packages from a global store on your disk. This avoids duplicating files across projects, saving huge amounts of disk space.

pnpm 不是像 npm 那样将包复制到 `node_modules` 中，而是从磁盘上的全局存储中对包进行符号链接。这避免了跨项目复制文件，从而节省了大量磁盘空间。

pnpm was created in 2017 by Zoltan Kochan, motivated by the disk space and speed problems of `npm install`. It has since become one of the fastest growing and most efficient package managers for JavaScript.

pnpm 由 Zoltan Kochan 于 2017 年创建，其动机是 `npm install` 的磁盘空间和速度问题。此后，它已成为增长最快、最高效的 JavaScript 包管理器之一。

Key Features And Benefits

主要特点和优势

Here are some of the standout features and benefits of using pnpm over npm or Yarn:

以下是使用 pnpm 而不是 npm 或 Yarn 的一些突出功能和优势：

- **Fast installation**: pnpm installs dependencies in parallel using hard links. This makes installation up to 3x faster.

  快速安装：pnpm 使用硬链接并行安装依赖项。这使得安装速度提高了 3 倍。

- **Lower disk usage**: By symlinking from a global store instead of copying, pnpm saves over 90% in disk space.

  降低磁盘使用率：通过从全局存储进行符号链接而不是复制，pnpm 节省了 90% 以上的磁盘空间。

- **Built-in dependency optimization**: pnpm dedupes and hoists common dependencies for faster installs.

  内置依赖项优化：pnpm 删除和提升常见依赖项以加快安装速度。

- **Strictness**: pnpm uses checksums and a content-addressed file system to avoid dependency conflicts.

  严格性：pnpm 使用校验和和内容寻址文件系统来避免依赖关系冲突。

- **Scales to large projects**: pnpm remains performant and efficient even for projects with hundreds of dependencies.

  扩展到大型项目：即使对于具有数百个依赖项的项目，pnpm 也能保持高性能和高效。

- **Backward compatibility**: pnpm ensures compatibility with npm registries and the wider Node.js ecosystem.

  向后兼容性：pnpm 确保与 npm 注册表和更广泛的 Node.js 生态系统兼容。

How Pnpm Works Pnpm 的工作原理

pnpm manages dependencies in a fundamentally different way from npm and Yarn under the hood. Here’s a quick look at how it works:

pnpm 在后台以与 npm 和 Yarn 截然不同的方式管理依赖项。以下是它的工作原理：

- Packages are stored in a single, global `store` folder (default: `~/.pnpm-store`).

  包存储在单个全局 `store` 文件夹（默认：`~/.pnpm-store`）中。

- Instead of copying packages into `node_modules`, pnpm creates symlinks from the store.

  pnpm 不是将包复制到 `node_modules` 中，而是从 store 创建符号链接。

- The store uses content-addressing – packages are saved by content hash, avoiding duplication.

  商店使用内容寻址 – 包按内容哈希保存，避免重复。

- A shrinkwrap file locks down the dependency tree. Adding a new package creates a new lockfile.

  shrinkwrap 文件会锁定依赖项树。添加新的软件包会创建一个新的 lockfile。

- Transitive dependency hoisting is used to flatten module locations.

  传递依赖项提升用于平展模块位置。

This optimized install strategy allows super fast installs and reduced disk usage. The store persists across projects to maximize efficiency.

这种优化的安装策略可实现超快速安装并减少磁盘使用量。该存储跨项目持续存在，以最大限度地提高效率。

Why Use Pnpm? 为什么使用 Pnpm?

For modern, complex JavaScript applications with hundreds of direct and indirect dependencies, pnpm is a no-brainer over npm.

对于具有数百个直接和间接依赖项的现代复杂 JavaScript 应用程序，pnpm 比 npm 更不费吹灰之力。

Here are some of the top reasons to switch to pnpm:
以下是切换到 pnpm 的一些主要原因：

- blazing fast installs – you get your dependencies 3x faster
  极快的安装 – 您获得依赖项的速度提高了 3 倍
- save over 90% in disk space for your projects
  为您的项目节省 90% 以上的磁盘空间
- avoid `node_modules` duplications across projects
  避免跨项目`node_modules`重复
- strictness and reproducibility with content-addressing
  内容寻址的严格性和可重复性
- efficient dependency optimization under the hood
  后台的高效依赖项优化
- excellent scaling even for massive projects
  即使对于大型项目也具有出色的扩展性
- easy migration from npm 从 npm 轻松迁移

For large JavaScript codebases and monorepos, pnpm is enabling faster CI builds, simplified deployment, and easier local development. The performance and disk benefits are especially noticeable compared to npm.

对于大型 JavaScript 代码库和 monorepo，pnpm 可实现更快的 CI 构建、简化的部署和更轻松的本地开发。与 npm 相比，性能和磁盘优势尤其明显。

Getting Started 开始

To start using pnpm, first install it globally:
要开始使用 pnpm，请先全局安装它：

```
npm install -g pnpm
```

Then install dependencies in your project:

然后在您的项目中安装依赖项：

```
pnpm install
```

That’s it! pnpm will now manage your packages from its global store.

就是这样！ pnpm 现在将从其 global store 管理您的软件包。

To learn more about migrating from npm or Yarn and using pnpm effectively, check out the excellent pnpm docs.

要了解有关从 npm 或 Yarn 迁移以及有效使用 pnpm 的更多信息，请查看优秀的 pnpm 文档。

Conclusion 结论

pnpm is a promising new package manager built specifically for the challenges of modern JavaScript dependency management.

pnpm 是一个很有前途的新包管理器，专为应对现代 JavaScript 依赖项管理的挑战而构建。

With its content-addressable store, symlinking strategy, and built-in optimization, pnpm unlocks game-changing benefits in install speed, disk usage, and strictness.

凭借其内容可寻址存储、符号链接策略和内置优化，pnpm 在安装速度、磁盘使用和严格性方面解锁了改变游戏规则的优势。

As JavaScript projects grow larger and more complex, pnpm is the most efficient package manager available today. For any serious JavaScript developer working on a complex codebase, it’s worth taking pnpm for a spin.

随着 JavaScript 项目变得越来越大和越来越复杂，pnpm 是当今可用的最高效的包管理器。对于任何处理复杂代码库的认真 JavaScript 开发人员来说，值得尝试一下 pnpm。

Technology

How to Overcome the 1500 Block Storage Limit on Notion

如何克服 Notion 的 1500 个区块存储限制