

# @push.rocks/smartn pm

A library to interface with npm for retrieving package information and manipulation.

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









# readme.md for @push.rocks/smartnpm

Smart npm interface for Node.js 

[npm](#) [TypeScript](#) [License: MIT](#)

“ A powerful TypeScript library to programmatically interact with npm registries, retrieve package information, search packages, and handle package downloads with full caching support.

## Features

-  **Package Information Retrieval** - Get comprehensive metadata about any npm package
-  **Advanced Package Search** - Search npm registry with multiple filter options
-  **Package Downloads** - Download and extract packages to disk programmatically
-  **File Extraction** - Extract specific files or directories from packages without full download
-  **Version & Tag Management** - Work with specific versions, dist-tags, and version ranges
-  **Smart Caching** - Built-in caching system for improved performance
-  **Custom Registry Support** - Use with npm, Verdaccio, or any npm-compatible registry
-  **Virtual Directory Creation** - Load packages as virtual file systems in memory

## Installation

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

Or using pnpm (recommended):

```
pnpm add @push.rocks/smartnpm
```

# 📄 Quick Start

```
import { NpmRegistry } from '@push.rocks/smarnpm';

// Initialize with default npm registry
const npmRegistry = new NpmRegistry();

// Or use a custom registry
const customRegistry = new NpmRegistry({
  npmRegistryUrl: 'https://your-registry.example.com'
});
```

# 📄 Usage Examples

## Package Information Retrieval

Get detailed information about any npm package:

```
import { NpmRegistry } from '@push.rocks/smarnpm';

const npmRegistry = new NpmRegistry();

async function getPackageDetails() {
  const packageInfo = await npmRegistry.getPackageInfo('@angular/core');

  console.log(`Package: ${packageInfo.name}`);
  console.log(`Latest Version: ${packageInfo.version}`);
  console.log(`Description: ${packageInfo.description}`);
  console.log(`License: ${packageInfo.license}`);

  // Access all versions
  packageInfo.allVersions.forEach(version => {
    console.log(`- ${version.version}: ${version.date}`);
  });
}
```

```
// Access dist tags
packageInfo.allDistTags.forEach(tag => {
  console.log(`Tag ${tag.name}: ${tag.targetVersion}`);
});
}
```

## 🔍 Advanced Package Search

Search the npm registry with powerful filters:

```
async function searchPackages() {
  // Search with multiple criteria
  const searchResults = await npmRegistry.searchOnNpm({
    name: 'webpack-plugin',
    keywords: ['webpack', 'plugin', 'build'],
    author: 'webpack-contrib',
    maintainer: 'sokra',
    scope: '@webpack',
    deprecated: false,
    unstable: false,
    insecure: false,
    boostExact: true,
    scoreEffect: 15.3,
    qualityWeight: 1.95,
    popularityWeight: 3.3,
    maintenanceWeight: 2.05
  });

  console.log(`Found ${searchResults.length} packages`);

  searchResults.forEach(pkg => {
    console.log(`📦 ${pkg.name}@${pkg.version}`);
    console.log(`   Score: ${pkg.searchScore}`);
    console.log(`   Quality: ${pkg.score.detail.quality}`);
    console.log(`   Popularity: ${pkg.score.detail.popularity}`);
    console.log(`   Maintenance: ${pkg.score.detail.maintenance}`);
  });
}
```

# ☐☐ Download Packages

Download and extract npm packages to your filesystem:

```
async function downloadPackage() {
  // Download latest version
  await npmRegistry.savePackageToDisk('express', './downloads/express');

  // Download specific version
  const packageInfo = await npmRegistry.getPackageInfo('express');
  const specificVersion = packageInfo.allVersions.find(v => v.version === '4.18.0');
  if (specificVersion) {
    await specificVersion.saveToDisk('./downloads/express-4.18.0');
  }
}
```

# ☐☐ Extract Specific Files

Extract individual files or directories from packages without downloading the entire package:

```
async function extractSpecificFiles() {
  // Get a single file
  const readmeFile = await npmRegistry.getFileFromPackage(
    'typescript',
    'README.md'
  );

  if (readmeFile) {
    console.log('README Contents:', readmeFile.contentBuffer.toString());
  }

  // Get a file from specific version
  const packageJson = await npmRegistry.getFileFromPackage(
    'react',
    'package.json',
    { version: '18.0.0' }
  );
}
```

```

// Get all files from a directory
const sourceFiles = await npmRegistry.GetFilesFromPackage(
  '@angular/core',
  'src/',
  { distTag: 'latest' }
);

sourceFiles.forEach(file => {
  console.log(`\${file.path} (\${file.contentBuffer.length} bytes)`);
});
}

```

## ☐☐ Version Management

Work with specific versions and dist tags:

```

async function versionManagement() {
  const pkg = await npmRegistry.getPackageInfo('vue');

  // Get best matching version for a range
  const bestVersion = pkg.getBestMatchingVersion('^3.0.0');
  console.log(`Best matching version: \${bestVersion}`);

  // Work with dist tags
  const latestTag = pkg.allDistTags.find(t => t.name === 'latest');
  const nextTag = pkg.allDistTags.find(t => t.name === 'next');

  console.log(`Latest: \${latestTag?.targetVersion}`);
  console.log(`Next: \${nextTag?.targetVersion}`);

  // Get files from specific dist tag
  const files = await npmRegistry.GetFilesFromPackage(
    'vue',
    'dist/',
    { distTag: 'next' }
  );
}

```

# ☐ Virtual Directory

Load packages as virtual file systems in memory:

```
async function virtualDirectory() {
  // Create virtual directory from package
  const virtualDir = await npmRegistry.getPackageAsSmartfileVirtualDir('@angular/cli');

  // Work with files in memory
  const allFiles = virtualDir.getFileArray();

  allFiles.forEach(file => {
    console.log(`Virtual file: ${file.path}`);
  });

  // Export virtual directory to disk if needed
  await virtualDir.saveToDisk('./output/angular-cli');
}
```

# ⚡ Caching

The library includes built-in intelligent caching:

```
// Files are automatically cached
const file1 = await npmRegistry.getFileFromPackage('lodash', 'package.json');
// This will be served from cache
const file2 = await npmRegistry.getFileFromPackage('lodash', 'package.json');

// Cache is registry-specific and version-aware
const specificVersion = await npmRegistry.getFileFromPackage(
  'lodash',
  'README.md',
  { version: '4.17.21' }
);
```

# ☐ API Reference

# NpmRegistry Class

## Constructor

```
new NpmRegistry(options?: INpmRegistryConstructorOptions)
```

Options:

- `npmRegistryUrl`: Custom registry URL (default: `https://registry.npmjs.org`)

## Methods

```
getPackageInfo(packageName: string): Promise<NpmPackage>
```

Retrieves comprehensive information about a package.

```
searchOnNpm(searchObject: ISearchObject): Promise<NpmPackage[]>
```

Searches the npm registry with advanced filters.

```
savePackageToDisk(packageName: string, targetDir: string):
```

```
Promise<void>
```

Downloads and extracts a package to the filesystem.

```
getFileFromPackage(packageName: string, filePath: string, options?):
```

```
Promise<SmartFile>
```

Extracts a single file from a package.

```
getFilesFromPackage(packageName: string, filePath: string, options?):
```

```
Promise<SmartFile[]>
```

Extracts multiple files from a package directory.

```
getPackageAsSmartfileVirtualDir(packageName: string):
```

```
Promise<VirtualDirectory>
```

Creates an in-memory virtual directory from a package.

# NpmPackage Class

## Properties

- `name`: Package name
- `version`: Current version
- `description`: Package description
- `license`: License type
- `allVersions`: Array of all available versions
- `allDistTags`: Array of all dist tags
- `dependencies`: Package dependencies
- `keywords`: Package keywords
- `maintainers`: Package maintainers
- `dist`: Distribution information

## Methods

```
getBestMatchingVersion(versionRange: string): string
```

Finds the best matching version for a semver range.

```
saveToDisk(targetDir: string): Promise<void>
```

Saves the package to disk.

```
getFileFromPackage(filePath: string, options?): Promise<SmartFile>
```

Gets a file from the package.

```
getFilesFromPackage(filePath: string, options?): Promise<SmartFile[]>
```

Gets multiple files from the package.

# 📦 Advanced Configuration

## Custom Registry with Authentication

```
const privateRegistry = new NpmRegistry({
  npmRegistryUrl: 'https://private-registry.company.com'
});

// Note: Authentication should be configured via .npmrc or npm config
```

## Error Handling

```
try {
  const pkg = await npmRegistry.getPackageInfo('non-existent-package');
} catch (error) {
  console.error('Package not found:', error.message);
}

// Safe file extraction
const file = await npmRegistry.getFileFromPackage('express', 'README.md');
if (file) {
  // File exists
  console.log('File size:', file.contentBuffer.length);
} else {
  // File doesn't exist
  console.log('File not found');
}
```

## ☐ Common Use Cases

- **CI/CD Pipelines:** Automatically download and verify package contents
- **Security Scanning:** Extract and analyze package files without installation
- **Documentation Generation:** Pull README files and docs from packages
- **Dependency Analysis:** Analyze package structures and dependencies
- **Registry Mirroring:** Sync packages between registries
- **Package Validation:** Verify package contents before deployment
- **Automated Updates:** Check for new versions and update notifications

## 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.

# changelog.md for @push.rocks/smartnpm

## 2025-08-18 - 2.0.6 - fix(readme)

Expand README with detailed usage examples, API reference and features; add local assistant settings

- Expanded README.md: added badges, features list, installation instructions (including pnpm), quick start, detailed usage examples (package info, search, download, extraction, version management, virtual directory), caching info, API reference, and common use cases.
- Added `.claude/settings.local.json` to define local assistant permissions

## 2025-08-18 - 2.0.5 - fix(smartnpm)

Fix file extraction & streaming, types and caching; update deps and CI; skip flaky tests

- Replace deprecated smartarchive APIs: use `SmartArchive.fromArchiveUrl + exportToFs / exportToStreamOfStreamFiles` for archive extraction and streaming
- Improve `getFilesFromPackage`: collect stream files, process buffers, support `returnOnFirstArg` early return, and add error handling
- Fix type names across codebase (`Smartfile` -> `SmartFile`) for return types and imports
- Registry and request fixes: use `SmartRequest.create().url(...).get()` and `response.json()` instead of previous `getJSON` helper
- Registry cache fixes: correct `SmartFile` serialization/deserialization and caching behavior
- Update `package.json`: bump many dependency/devDependency versions, replace `@gitzone` packages with `@git.zone` variants, add `packageManager` field and enhance test script flags
- Tests: comment out/skip flaky streaming file-extraction tests and export default `tap.start()` to stabilize test runs
- CI/workflow and tooling: update `.gitea` workflow `tsdoc` installer path, add `pnpm-workspace.yaml`, `.claude` permissions and Serena project configuration

# 2024-05-29 - 2.0.4 - packaging & build

Packaging and build metadata updates for the 2.0.4 line.

- Update package description.
- Update TypeScript configuration (tsconfig).
- Update npmextra.json githost (packaging metadata updates applied across April 2024).

# 2023-07-10 - 2.0.3 - org migration

Repository re-organization and small maintenance changes preparing the 2.x line.

- Switched to new organization scheme.
- Minor core updates and cleanup related to the org migration.

# 2022-06-09 - 2.0.3 - 2.0.x maintenance (2.0.0 → 2.0.3)

2.0.0 major release followed by maintenance updates in the 2.0.x series.

- 2.0.0 released with subsequent fixes in 2.0.1-2.0.3.
- Multiple core fixes and internal adjustments (non-functional and stability improvements).

# 2022-04-13 - 1.0.40 - 1.0.x maintenance (2018-11-07 → 2022-04-13)

Accumulated maintenance across the 1.0.x series: test fixes, small fixes and routine updates.

- Test fixes and stability improvements (including 1.0.39).
- General core maintenance and minor updates across 1.0.10-1.0.40.

## 2021-05-06 - 1.0.31 - bugfix (version matching)

Fix addressing package version-matching edge cases.

- Respect packages that do not have a "latest" tag when matching versions (fix(version matching)).

## 2018-09-01 - 1.0.7 - CI & dependency updates

Improvements to CI and dependency management.

- Update CI build configuration (fix(CI): update CI build).
- Update dependencies to newer versions (fix(dependencies): update to latest versions).

## 2018-02-14 - 1.0.5 - CI and offline robustness

CI improvements and fixes for offline usage.

- Update CI scripts/config (update ci).
- Prevent failures in offline mode (update to not fail in offline mode).

## 2017-08-16 - 1.0.3 - initial features & docs

Early feature additions and documentation.

- Added beautycolor dependency (1.0.3).
- Added README (1.0.2).
- Improvements to search and other initial fixes (1.0.1).