

# @git.zone/tools

Documentation for @git.zone/tools

- [readme.md for @git.zone/tools](#)
- [changelog.md for @git.zone/tools](#)

# readme.md for @git.zone/tools

A powerful CLI utility for managing your `@git.zone` development toolchain. Scan, update, and install all official `@git.zone` packages across multiple package managers — from a single command.

## Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit [community.foss.global/](https://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/](https://code.foss.global/) account to submit Pull Requests directly.

## Installation

Install globally with your preferred package manager:

```
# npm
npm install -g @git.zone/tools

# pnpm (recommended)
pnpm add -g @git.zone/tools

# yarn
yarn global add @git.zone/tools
```

This gives you the `gtools` binary.

## Usage

# Overview

```
gtools          # Show available commands
gtools update   # Check & update installed @git.zone packages
gtools install  # Interactively install missing @git.zone packages
```

## gtools update — Keep Everything Fresh

Scans all available package managers (npm, pnpm, yarn) on your system, detects globally installed `@git.zone` packages, and checks them against the latest published versions.

```
gtools update          # Interactive – prompts before updating
gtools update -y       # Auto-approve all updates (great for CI)
gtools update --verbose # Show package manager detection diagnostics
```

### What it does:

1. `gtools` Detects which package managers are available on your system
2. `gtools` Shows a version table for each detected package manager
3. `gtools` Checks for a **self-update** of `gtools` itself first
4. `gtools` Lists all globally installed `@git.zone` packages with current vs. latest versions
5. `gtools` Prompts you to update any outdated packages (or auto-updates with `-y`)

### Example output:

Package managers:

Name	Current	Latest	Status
pnpm	10.28.2	10.28.2	✓ Up to date

Installed @git.zone packages:

Package	Current	Latest	PM	Status
@git.zone/cli	1.10.0	1.10.0	pnpm	✓ Up to date
@git.zone/tsbuild	4.3.0	4.3.0	pnpm	✓ Up to date
@git.zone/tstest	3.5.1	3.5.1	pnpm	✓ Up to date

All packages are up to date!

# gtools install — Bootstrap Your Toolchain

Detects which official `@git.zone` packages are **not** installed and lets you pick which ones to add.

```
gtools install          # Interactive selection
gtools install -y       # Install all missing packages automatically
gtools install --verbose # Show detection diagnostics
```

## Managed packages:

Package	Description
<code>@git.zone/cli</code>	The main gitzone CLI ( <code>gitzone</code> command)
<code>@git.zone/tsbuild</code>	TypeScript compiler wrapper
<code>@git.zone/tstest</code>	Test runner for TypeScript projects
<code>@git.zone/tsbundle</code>	Bundle TypeScript for browsers
<code>@git.zone/tswatch</code>	File watcher with live reload
<code>@git.zone/tspublish</code>	Publish packages to npm registries
<code>@git.zone/tsdoc</code>	Documentation generator
<code>@git.zone/tsdocker</code>	Docker integration for TS projects
<code>@git.zone/tsview</code>	Component viewer
<code>@git.zone/tsrust</code>	Rust cross-compilation support

## Smart Package Manager Detection

`gtools` doesn't assume which package manager you use. It probes for `npm`, `pnpm`, and `yarn` using multiple detection strategies:

- `which` **command** — checks the system PATH
- `--version` **fallback** — directly invokes the PM if `which` fails

When installing, it recommends the package manager that already has the most `@git.zone` packages, so your toolchain stays consistent.

# Programmatic API

You can also use `@git.zone/tools` as a library:

```
import { runCli } from '@git.zone/tools';

// Launch the CLI programmatically
await runCli();
```

Or use the `PackageManagerUtil` class directly for custom tooling:

```
import { PackageManagerUtil } from
 '@git.zone/tools/dist_ts/mod_update/classes.packagemanager.js';

const pmUtil = new PackageManagerUtil();

// Check if pnpm is available
const pnpmInfo = await pmUtil.detectPackageManager('pnpm');
console.log(pnpmInfo.available); // true/false

// Get globally installed @git.zone packages
const packages = await pmUtil.getInstalledPackages('pnpm');

// Get latest version from registry (checks private registry first, then npm)
const latest = await pmUtil.getLatestVersion('@git.zone/tsbuild');

// Compare versions
pmUtil.isNewerVersion('1.0.0', '2.0.0'); // true
```

## 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](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 @git.zone/tools

## 2026-03-24 - 3.6.0 - feat(package)

refresh package metadata, update dependency versions, and document CLI update and install workflows

- rename npmextra.json to .smartconfig.json and update the published package file list
- replace placeholder package description with CLI-specific metadata in package.json and readme
- upgrade @git.zone/tsbuild, @git.zone/tstest, @push.rocks/smartshell, and @types/node
- rename the bundled license file reference from LICENSE to license

## 2026-02-09 - 3.5.0 - feat(install)

add interactive install command and module to detect and install missing @git.zone packages

- Add ts/mod\_install/index.ts: implements interactive/non-interactive flow to detect package managers, collect installed @git.zone packages, prompt user (via smartinteract) and install selected packages via PackageManagerUtil.
- Add ts/mod\_install/mod.plugins.ts: re-export smartinteract and smartshell for the installer.
- Update ts/tools.cli.ts: register new 'install' command and add help text for install flags.
- Update ts/mod\_update/index.ts: export GITZONE\_PACKAGES and print a summary of managed packages that are not installed with latest versions and a suggestion to run 'gtools install'.

## 2026-02-09 - 3.4.1 - fix(tools)

no changes to commit

- No files changed in this diff
- Current package version is 3.4.0 — no version bump required

# 2026-02-09 - 3.4.0 - feat(mod\_update)

add @git.zone/tsrust to supported modules list

- ts/mod\_update/index.ts: added '@git.zone/tsrust' to the modules array

# 2026-02-06 - 3.3.0 - feat(mod\_update)

add self-update flow, package name parser, dynamic CLI version, and tests

- Add a self-update check in mod\_update to detect and optionally update @git.zone/tools (prompts the user or uses --yes).
- Introduce PackageManagerUtil.parseYarnPackageName to correctly parse scoped and unscoped yarn package strings and use it when collecting installed packages.
- Add comprehensive unit tests for PackageManagerUtil.isNewerVersion and parseYarnPackageName.
- Use commitinfo.version for CLI reported version instead of a hardcoded value.
- Remove automatic invocation of runCli() from ts/index.ts to avoid immediate execution on import.

# 2026-02-05 - 3.2.0 - feat(update)

enhance package manager detection, version reporting, and add verbose option

- Add IPackageManagerInfo interface and detectPackageManager() to robustly detect npm/yarn/pnpm via 'which' and '--version' fallbacks
- Make isAvailable() delegate to detectPackageManager() and return structured detection info
- Add getPackageManagerVersion() to obtain current and latest versions (parses local --version and queries npm registry)
- Update run() to support a verbose flag, show a package-manager status table, and collect detectedPMs with version/update status
- Update CLI help and command handling to accept --verbose/-v and pass it through to mod\_update.run()

# 2026-02-03 - 3.1.3 - fix(mod\_update)

try private registry (verdaccio.lossless.digital) first when fetching package versions; fall back to public npm; handle unknown latest versions gracefully in output

- getLatestVersion now attempts a direct API request to <https://verdaccio.lossless.digital/> and parses dist-tags.latest
- Falls back to npm view when the private registry request fails
- Scoped package names are URL-encoded (replaces '/' with '%2f') before querying the private registry
- Packages with no resolvable latest version are included with latestVersion set to 'unknown' and displayed as '? Version unknown'
- needsUpdate is set to false when latest version is unknown

# 2026-02-03 - 3.1.2 - fix(scripts)

make test script output verbose by using --verbose instead of --web

- package.json: change npm "test" script from "(tstest test/ --web)" to "(tstest test/ --verbose)" to enable verbose test output

# 2026-02-03 - 3.1.1 - fix(tools)

no changes detected

- No files were modified in this diff
- No code or documentation changes to include in a commit message

# 2026-02-03 - 3.1.0 - feat(cli)

add update command to check and update globally installed @git.zone packages

- Adds a new mod\_update module and PackageManagerUtil to detect installed @git.zone packages and fetch latest versions

- Supports npm, yarn and pnpm: detection, listing, version comparison and executing updates
- Interactive confirmation via @push.rocks/smartinteract with a -y flag to skip prompt
- Uses @push.rocks/smartshell to run shell commands for listing and updating packages
- Wires the new 'update' command into the CLI (tools.cli) and updates CLI version to 3.0.0
- Adds CLI helper files (cli.child.js, adjusts cli.js and cli.ts.js) to run TypeScript CLI via tsrun/runPath
- Updates package.json to include new dependencies and npmextra.json with release registries and @git.zone/cli metadata
- Updates .gitignore to exclude local AI tool dirs (.claude, .serena)

## 2026-02-03 - 3.0.0 - BREAKING CHANGE(tools)

replace install functionality with a minimal placeholder CLI; remove installer, logging, path utilities, and related assets

- Removed installer implementation (ts/tools.install.ts) and package library asset (assets/package\_library.json).
- Removed logging and path utilities (ts/tools.logging.ts, ts/tools.paths.ts) and simplified plugins (ts/tools.plugins.ts) to only export smartcli.
- Reworked CLI (ts/tools.cli.ts) to a placeholder standard command that prints messages instead of running installs; internal CLI version set to 2.0.22.
- Module export/behavior changed: ts/index.ts now exports runCli and no longer auto-runs; tests updated to expect runCli.
- package.json updated: description changed, several dependencies removed (e.g. @push.rocks/smartlog, @push.rocks/smartshell), @types/node bumped to ^22.0.0, and assets removed from packaged files list.
- Removed project policy and metadata files: .snyk deleted and readme.md added with usage and legal information.

## 2026-02-03 - 2.0.22 - core

Core maintenance release.

- fix(core): update — minor core fixes and maintenance improvements.

# 2026-02-03 - 2.0.23 - release tag

Release tag only — no recorded changelog details.

- 2.0.23: release tag with no additional commit message.

# 2020-10-05 - 2.0.3 - 2.0.21 - core (maintenance)

Series of maintenance releases containing repeated small core fixes.

- Multiple commits "fix(core): update" applied across 2.0.3 through 2.0.21 addressing minor bugs and stability improvements.
- These releases are maintenance-focused; no user-facing feature additions recorded.

# 2019-06-17 - 2.0.2 - core

Maintenance update to core.

- fix(core): update — continued minor fixes and upkeep.

# 2018-06-07 - 2.0.0 - 2.0.1 - policy/package fixes

Initial 2.0.x releases with policy and package fixes.

- 2.0.0: fix(snyk policy): update — update to Snyk policy.
- 2.0.1: fix(package.json): fix private property — corrected package.json property.
- 2.0.2: release tag (later maintenance continued in subsequent 2.0.x).

# 2018-06-07 - 1.0.9 - core (feature)

New core scope added.

- feat(core): new tools scope — introduces a "tools" scope in core.

## 2017-06-04 - 1.0.8 - release tag

Release tag only — no recorded changelog details.

- 1.0.8: release tag with no additional commit message.

## 2017-06-04 - 1.0.7 - cli

CLI update.

- update cli — improvements/updates to the command-line interface.

## 2017-05-28 - 1.0.6 / 1.0.5 - docs & CLI

Documentation and CLI additions.

- 1.0.6: update README — documentation updates.
- 1.0.5: add cli.js — adds CLI entry point.

## 2017-05-28 - 1.0.4 - maintenance and cleanups

Collection of maintenance, cleanup, and packaging updates for 1.0.4 over several commits.

- update / cleanup — general code cleanups.
- Update packageLibrary.json — package metadata adjusted.
- added cli.js, added .npmignore — packaging and CLI support added.
- Consolidated multiple 1.0.4 changes across 2016-2017 into this release.

# 2016-03-03 - 1.0.3 - tests

Test fix.

- fix test.ts — test suite correction.

# 2016-03-03 - 1.0.2 - CI

Continuous integration fix.

- fix Travis — adjusts CI configuration.

# 2016-03-02 - 1.0.1 - install/tests

Installer and tests improvements.

- now installing things correctly — fixes installation behavior.
- create test / update — test additions and miscellaneous updates.

# 2016-03-02 - 1.0.0 - initial release

Project initial structure and first release.

- add structure / set up initial structure / cleanup — initial project layout and housekeeping.
- initial (2016-02-29 and 2016-02-24) — first commits establishing the repository.