# @git.zone/tscoverage

Documentation for @git.zone/tscoverage

# readme.md for @git.zone/tscoverage

A CLI tool for collecting and reporting test coverage information for TypeScript projects in the gitzone ecosystem.

## 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 globally for CLI usage:

```
pnpm install -g @git.zone/tscoverage
```

Or as a dev dependency in your project:

```
pnpm install --save-dev @git.zone/tscoverage
```

## □□ Usage

### CLI

Run `tscoverage` in your gitzone TypeScript project directory to execute tests and generate a coverage report:

```
tscoverage
```

This will run your project's test suite and produce a coverage report, giving you visibility into which parts of your codebase are exercised by tests.

## Programmatic

You can also import `tscoverage` in your own scripts:

```
import * as tscoverage from '@git.zone/tscoverage';
```

## □□ Development

This project uses the standard `@git.zone` toolchain:

```
# Install dependencies
pnpm install

# Build the project
pnpm run build

# Run tests
pnpm test
```

# License and Legal Information



## 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.

# docs/00footer.md for @git.zone/tscoverage

Legal Info Privacy Policy /// Git.Zone tools for a seamless dev workflow

Lossless GmbH the company behind git.zone and npmts ///

# docs/changelog.md for @git.zone/tscoverage

## 2017-07-30: Version 7.x.x -> 8.x.x

Testfiles in ./test/ can now import files directly from the ts dir:

```
// ./test/test.ts
import * as myModule from '../ts/index
```

# docs/config.md for @git.zone/tscoverage

## name: config

# Configuration

npmts can be configured to your needs.

## npmextra.json

the npmts section in npmextra.json can be used to configure npmts.

**Default**

> Note: When you are using `"mode":"default"` it'll cause npmts to override any other settings you may have made except for tsOptions (ES target etc.) with default behaviour.

```
{
  "npmts": {
    "mode": "default"
  }
}
```

**Custom settings**

```
{
  "mode": "custom",
```

```
  "test": true,
  "npmts": {
    "ts": {
      "./customdir/*.ts": "./"
    },
    "tsOptions": {
      "declaration": false,
      "target": "ES6"
    },
    "cli": true
  }
}
```

| key | default value | description |
|---|---|---|
| `"mode"` | `"default"` | "default" will do default stuff and override , "custom" only does what you specify, "merge" will merge default options with whatever you specify on your own |
| `"test"` | `true` | test your module |
| `"ts"` | `{"./ts/*.ts":"./","./test/test.ts":"./test/"}` | allows you to define multiple ts portions |
| `"tsOptions"` | `{"target":"ES5", "declaration":"true"}` | specify options for tsc |
| `"cli"` | `"false"` | some modules are designed to be used from cli. If set to true NPMTS will create a cli.js that wires you dist files up for cli use. |
| `"testConfig"` | `{ parallel: true, coverage: true }` | allows you to control test behaviour. `"parallel"` controls wether testfiles are run sequentially or in parallel, and `"coverage` wether to create coverage reports |

# TypeScript

by default npmts looks for `./ts/*.ts` and `./test/test.ts` that will compile to `./dist/*.js` and `./test/test.js`

Use commonjs module system for wiring up files.

# Declaration files

**npmts** also creates declaration files like `./dist/index.d.ts` by default. You can reference it in your package.json like this.

```
"main": "dist/index.js",
"typings": ".dist/index.d.ts",
```

This is in line with the latest TypeScript best practices. You can then import plugins via the TypeScript `import` Syntax and tsc will pick up the declaration file automatically.

# Some notes:

## Typings for third party modules that do not bundle declaration files

NPMTS no longer supports typings.json. Instead use the new TypeScript 2.x approach to typings using the @types/ npm scope.

## Instrumentalize Code

npmts instrumentalizes (using istanbul) the created JavaScript code to create a coverage report.

## Tests

Any errors will be shown with reference to their originating source in TypeScript thanks to autogenerated source maps.

# docs/default.md for @git.zone/tscoverage

## name: Default Behaviour

## Default Behaviour

when you don't configure it otherwise.

1. **Config:** Check config in ./npmextra.json (Check out npmextra)
2. **Clean:** Clean up from any previous builds (old js files)
3. **Check:** Check project for typings declaration in package.json, unused dependencies and missing dependencies
4. **Transpile:** Transpile TypeScript with **inline sourcemaps** and **declaration files** to ES target
5. **Test:** transpile TypeScript of module to ES5 for tests (so it can be instrumentalized) and pipe it to tapbuffer. All this happens in memory.

# docs/examples.md for @git.zone/tscoverage

## name: Examples

# Examples

modules that use npmts in theis development workflow

| Module Name | Description |
| --- | --- |
| gitzone | fast npm module prototyping |
| gulp-browser | browserify for gulp |
| npmdocker | dockerized npm development |
| smartcli | easy cli tool creation |

> There are tons more... We will add them here over time.

## Tips and tricks:

- Use npmts-g to use globally installed npmts and install npmts locally if no global npmts is available.
- Use npmpage to create a webpage from coverage reports and TypeDoc for the module
- Use hosttoday/ht-docker-node:npmci for speedy CI builds
- Use npmdocker for running tests consistently with docker.

# docs/getstarted.md for @git.zone/tscoverage

# name: Get Started description: learn how to quickly write npm TypeScript modules

# Get Started with NPMTS

and learn how to quickly write npm TypeScript modules

## Step1: Install the tools

To use npmts install it using npm or yarn:

```
npm install -g npmts # install with npm
yarn global add npmts # install with yarn
```

For the purpose of getting started quickly please also install **gitzone**. It'll proovide awesome scaffolding for new npmts maintained modules and also updates them later on.

```
npm install -g gitzone # install with npm
yarn global add gitzone # install with yarn
```

You can make sure npmts and gitzone are installed correctly by typing `npmts -v && gitzone -v`.

# Scaffold a new module

To scaffold a new module type

```
gitzone template npm
```

This will run you through a series of question to get gitzone to know the specifics of your module. Enter all information accordingly.

# Run NPMTS for the first time

# docs/index.md for @git.zone/tscoverage

## name: Index description: best practice npm TypeScript modules

## npmts

best practice npm TypeScript modules

## Availabililty

npm git git docs

## Status for master

build status coverage report npm downloads per month Dependency Status bitHound Dependencies bitHound Code TypeScript node JavaScript Style Guide

## Quick Demo

asciicast

# Usage

NPMTS is your friend when writing, testing, publishing and documenting npm modules written in TypeScript.

npmts will

1. check your dependencies and package.json (unused, missing, updates, security)
2. transpile your code with tsc,
3. test your code with tap (supports the fancy stuff like Promises, Generators, async/await, sourcemaps, parallel test execution in child processes)
4. create coverage with istanbul (supports tracing of the originating TypeScript)

For more information on how tests are run check out the tapbuffer module.

This works on your machine and in CI. There is a prebuild docker image available that includes npmts to make CI a breeze:

hosttoday/ht-docker-node:npmts on Dockerhub

> MIT licensed | © Lossless GmbH | By using this npm module you agree to our privacy policy

repo-footer

# docs/install.md for @git.zone/tscoverage

# Install npmts

Get started with TypeScript awesomeness.

> *_npmts-g_ checks if the global version of npmts suffices the modules requirements. If not it installs npmts locally in the right version during npm install.

```
npm install npmts -g # installs npmts globally
npm install npmts-g --save-dev # installs npmts-g checking tool as devDependency
```

Then add it to your package.json's script section to trigger a build:

```
"scripts": {
    "test": "(npmts)"
}
```

# docs/structure.md for @git.zone/tscoverage

# name: npmts project structure description: how npmts projects are structured

# npmts - Project Structure

**locally**

```
projectroot
|- .nogit/           # contains files that should not be checked into git - NOgit
|- dist/             # contains compiled js files and their corresponding typings - git
|- node_modules/     # contains the installed node modules - NOgit
|- test/             # contains the test files - git
|- ts/               # contains the source TypeScript files - git
|
|- .gitignore        # the normal gitignore file
|- .gitlab-ci.yml    # the gitlab ci yml file
|- npmextra.json     # npmextra.json
|- package.json      # the standard npm module package.json file
|- readme.md         # the standard project readme
|- tslint.json       # the standard tslint.json for TypeScript
|- yarn.lock         # yarn.lock - the standard yarn.lock file
```

**in git**

# changelog.md for @git.zone/tscoverage

## 2026-03-24 - 10.0.0 - BREAKING CHANGE(package)

migrate package to the @git.zone scope and ESM-based tooling

- rename the published package from @gitzone/tscoverage to @git.zone/tscoverage
- switch CLI entrypoints from CommonJS require() to ESM dynamic imports and set package type to module
- update build, test, and runtime tooling to the @git.zone toolchain and replace npmextra.json with .smartconfig.json

## 2020-06-03 - 9.0.1 - core

Applied a core fix update.

- Updated core behavior.

## 2020-06-03 - 9.0.2 - release

Version-only release with no additional relevant changes.

- Summarizes trivial release tagging after 9.0.1.

## 2020-06-03 - 9.0.0 - core

Applied a core fix update.

- Updated core behavior.

# 2020-06-03 - 1.0.1 - core

Applied a core fix update.

- Updated core behavior.

# 2020-06-03 - 1.0.2 - release

Version-only release with no additional relevant changes.

- Summarizes trivial release tagging after 1.0.1.

# 2018-05-03 - 8.0.35 - build

Improved project automation and dependency maintenance across recent 8.0.x releases.

- Updated dependencies in 8.0.36.
- Added a code quality CI step in 8.0.35.
- Updated build and CI configuration in 8.0.34.
- Updated CI configuration and security policy in 8.0.32-8.0.31.
- Split build and test commands to improve speed in 8.0.30.

# 2018-04-08 - 8.0.29 - maintenance

Applied packaging and security policy updates across 8.0.29-8.0.28.

- Fixed npmextra.json configuration in 8.0.29.
- Updated Snyk policy in 8.0.28.

# 2018-04-08 - 8.0.27 - packaging

Updated package naming and offline behavior across 8.0.27-8.0.26.

- Renamed the npmts package to the @gitzone scope in 8.0.27.
- Added offline runtime support in 8.0.26.

# 2017-11-28 - 8.0.25 - maintenance

Applied internal cleanup and dependency refreshes across 8.0.25-8.0.23.

- Refactored internals in 8.0.25.
- Updated dependencies in 8.0.24 and 8.0.23.

# 2017-10-05 - 8.0.22 - test

Adjusted test library inclusion and platform resilience across 8.0.22-8.0.20.

- Changed library inclusion for tests in 8.0.22.
- Updated dependencies and CI behavior in 8.0.21.
- Improved operation when npms.io is unavailable in 8.0.20.

# 2017-09-08 - 8.0.18 - infrastructure

Enhanced mirroring and tracking support across 8.0.18-8.0.17.

- Updated tracking domains in 8.0.18.
- Added a mirror stage in 8.0.17.

# 2017-08-16 - 8.0.7 - maintenance

Delivered a series of small maintenance improvements across 8.0.16-8.0.7.

- Updated documentation, CI, analytics, and dependencies.
- Improved update logging.
- Fixed smartupdate execution.
- Refreshed tooling and package integrations.

# 2017-07-30 - 8.0.3 - docs

Introduced documentation and usability updates across 8.0.3-8.0.0.

- Updated to the latest smartsystem in 8.0.3.
- Improved highlighting and removed the old changelog in 8.0.2.
- Updated docs and description in 8.0.1.
- Added docs in 8.0.0.

# 2017-07-28 - 7.2.10 - test

Improved test handling and coverage support across 7.2.10-7.2.4.

- Updated test file loading approach in 7.2.10.
- Fixed module test import recognition in 7.2.9.
- Added smart replacer in 7.2.8.
- Updated the --nocoverage option in 7.2.7.
- Added proper sourcemap tracing for coverage in 7.2.4.

# 2017-07-18 - 7.2.3 - dependencies

Dependency-only maintenance updates across 7.2.3-7.2.0.

- Updated and upgraded dependencies with no major functional changes.

# 2017-06-30 - 7.1.9 - test

Expanded test execution options and developer controls across 7.1.9-7.1.1.

- Added support for SHELL PATH distributions in tests in 7.1.9.
- Fixed the --nochecks option in 7.1.6.
- Added better test run configuration in 7.1.4.
- Added --nochecks and --nocoverage CLI options in 7.1.3.
- Added coverage and merge options in 7.1.2.
- Added smarterror in 7.1.1.

# 2017-05-13 - 7.1.0 - cleanup

Removed obsolete cleanup code.

- Simplified legacy cleanup behavior.

# 2017-05-04 - 7.0.18 - test

Improved test, coverage, and build workflows across 7.0.18-7.0.1.

- Added sourcemap support for tests in 7.0.18.
- Ensured coverage percentage is properly detected in 7.0.16.
- Fixed smartgulp and related pathing issues in 7.0.15-7.0.14.
- Replaced gulp in 7.0.13.
- Updated runtime standards and environment variable handling in 7.0.8-7.0.7.
- Fixed coverage calculation in 7.0.2.
- Added support for picking up all .ts files in .test/ in 7.0.1.

# 2017-03-26 - 7.0.0 - test

Corrected test execution behavior.

- Tests now execute correctly.
- Improved README documentation.

# 2017-03-04 - 6.1.15 - test

Modernized the test stack and project integrations across 6.1.15-6.1.0.

- Switched to tap in 6.1.15.
- Added smartanalytics in 6.1.15.
- Fixed README and project metadata in 6.1.13-6.1.12.
- Shifted branding and updated dependencies in 6.1.9-6.1.5.
- Fixed testing for rxjs in 6.1.4.
- Replaced q with smartq and added ES2015 iterable support in 6.1.2.
- Fixed lib inclusion for tests in 6.1.1.
- Improved ES5 transpilation behavior in 6.1.0.

# 2017-01-15 - 6.0.0 - migration

Migrated the project away from Babel to TypeScript.

- Removed Babel-based compilation.
- Adopted TypeScript as the main toolchain.

# 2016-12-18 - 5.5.12 - maintenance

Improved tooling, logging, and modularization across 5.5.12-5.5.0.

- Removed TypeDoc in 5.5.12.
- Added a new npmpage in 5.5.10.
- Reduced unnecessary logging and fixed transpilation edge cases in 5.5.9-5.5.8.
- Improved feedback messages and error catching in 5.5.6-5.5.5.
- Fixed minor issues and version display in 5.5.1-5.5.0.

# 2016-10-21 - 5.4.49 - architecture

Restructured the CLI and module architecture across 5.4.49-5.4.35.

- Modularized the project and moved the CLI into its own file in 5.4.49.
- Improved CLI path discovery and CLI fixes in 5.4.47-5.4.46.
- Added smartstream and updated npmextra integration in 5.4.37-5.4.36.
- Added a --watch option in 5.4.35.

# 2016-09-15 - 5.4.34 - compatibility

Improved TypeScript, documentation, and platform compatibility across 5.4.34-5.4.13.

- Upgraded tsn and cleaned up promise handling for TypeScript compilation.
- Added decorator and reflect metadata support via dependency updates.
- Fixed checks, typedoc integration, and type issues.
- Removed Travis/AppVeyor and added GitLab Pages support.
- Improved npmpage compatibility and project cleaning.
- Updated branding and project page generation.

# 2016-08-13 - 5.4.5 - cli

Expanded CLI and dependency analysis features across 5.4.5-5.3.27.

- Added --nodocs option in 5.4.5.
- Improved local Babel plugin resolution in 5.4.4.
- Added missing devDependency detection in 5.4.3.
- Added typings field checks and dependency checks in 5.4.0-5.3.28.
- Fixed compiler option parsing in 5.3.27.

# 2016-07-19 - 5.3.26 - coverage

Improved coverage, test, and documentation tooling across 5.3.26-5.3.11.

- Improved coverage reporting and output behavior.
- Fixed asset handling, dependency issues, tests, and Istanbul integration.
- Switched to npmextra and improved performance by skipping HTML coverage reports.
- Introduced TypeDoc and completed the move to ES6.
- Restructured internal project layout.

# 2016-07-11 - 5.3.8 - docs

Expanded documentation tooling and source map handling across 5.3.8-5.3.7.

- Switched to EsDoc in 5.3.8.
- Updated sourcemap handling in 5.3.7.

# 2016-07-01 - 5.3.6 - maintenance

Prepared the tool for broader usage across 5.3.6-5.2.0.

- Removed debug code and fixed npm variant issues.
- Integrated tsn and readied the project for wider adoption.
- Added --notest support and fixed it.
- Improved console output and CI setup.
- Added legacy test support and separate test compilation.

# 2016-05-31 - 5.1.19 - ci

Significant CI, packaging, and project maintenance updates across 5.1.19-5.1.15.

- Expanded and fixed GitLab CI configuration.
- Added pages support and improved GitLab compatibility.
- Added changelog and contribution guide.
- Fixed package.json and added .npmignore.
- Updated repository URLs and publishing behavior.

# 2016-05-25 - 5.1.14 - dependencies

Improved module integrations and CLI test behavior across 5.1.14-5.1.5.

- Updated smartstring and typings-global integration.
- Added and updated the early module.
- Improved CLI pipe clearing for tests.
- Improved log output and console support.
- Fixed promise chain behavior.
- Added AppVeyor and updated related CI files.
- Switched to beautylog.ora.

# 2016-04-30 - 5.1.4 - typescript

Improved TypeScript defaults and declaration handling across 5.1.4-5.0.2.

- Brought dependencies up to date in 5.1.4.
- Fixed declaration file issues in 5.1.3.
- Made TypeScript modules fully typed by default in 5.1.0.
- Added tsconfig-based tsOptions and declaration file handling in 5.0.4.
- Improved compatibility for projects outside npmts conventions in 5.0.3.
- Restored working behavior in 5.0.2.

# 2016-04-04 - 5.0.1 - dependencies

Dependency-only maintenance across 5.0.1-4.0.2.

- Updated dependencies and internal descriptions with no major feature changes.

# 2016-04-02 - 4.0.1 - coverage

Improved coverage publishing and reporting across 4.0.1-3.6.8.

- Added a codecov badge and updated log messaging in 4.0.1.
- Enabled coverage publishing on every Travis run in 4.0.0.
- Switched from Coveralls to Codecov in 3.6.10.
- Improved coverage failure handling in 3.6.8.

# 2016-03-26 - 3.6.6 - logging

Refined logging, execution flow, and publishing behavior across 3.6.6-3.6.0.

- Added better test log identifiers in 3.6.6.
- Fixed a small promise error in 3.6.5.
- Improved execution order in 3.6.3.
- Fixed doPublish in 3.6.2.
- Made options handling easier in 3.6.1.
- Fixed a small error in 3.6.0.

# 2016-03-23 - 3.5.0 - release

Improved release detection, cleanup, and CLI handling across 3.5.0-3.3.0.

- Correctly determined release builds in 3.5.0.
- Added cleanup and visual polish in 3.5.0.
- Improved CLI option handling and added ship support in 3.4.1.
- Updated developer dependencies and cosmetics in 3.4.0.
- Switched to gulp-typings in 3.3.2.
- Added optional docs publishing in 3.3.1.
- Fixed CLI calls in 3.3.0.

# 2016-02-23 - 3.2.2 - build

Improved build setup, configuration handling, and test preparation across 3.2.2-3.0.0.

- Installed typings before compilation and hid git console output in 3.2.2.
- Fixed config travel, repo path, and environment variable handling in 3.2.1.
- Updated module system usage and added new test files in 3.1.2-3.1.0.
- Improved TypeScript compilation and sourcemap handling in 3.0.3-3.0.2.
- Disabled declaration files temporarily in 3.0.1.
- Fixed coverage issues in 3.0.0.

# 2016-02-17 - 2.4.1 - defaults

Improved defaults, module settings, and source map support across 2.4.1-2.2.0.

- Updated default behavior and tests in 2.4.1.
- Switched to CommonJS as the default module format in 2.4.0.
- Added sourcemap support in 2.3.2.
- Fixed compile script behavior in 2.3.1.
- Updated build target from ES3 to ES5 in 2.2.4.
- Added badges and greeting improvements in 2.2.3-2.2.2.
- Re-added test.ts to the default build process in 2.2.1.
- Fixed initial coveralls value in 2.2.0.

# 2016-02-09 - 2.1.10 - coverage

Expanded coverage and configuration support across 2.1.10-2.0.0.

- Fixed coverage path handling and added a coverage badge in 2.1.10.
- Fixed config travel and task name scoping in 2.1.9-2.1.8.
- Refined paths and coverage failure thresholds in 2.1.7-2.1.6.
- Added coveralls configuration and code coverage in 2.1.5.
- Improved recursive typings handling and path behavior in 2.1.1-2.1.0.
- Added config file support in 2.0.3.
- Fixed timing errors caused by missing stream returns in 2.0.1.
- Ensured mocha tests run properly in 2.0.0.

# 2016-01-31 - 1.0.12 - test

Major update to testing and project structure.

- Added mocha integration.
- Restructured the project.
- Included broader test workflow improvements.

# 2016-01-25 - 1.0.10 - ci

Improved CI detection and plugin handling across 1.0.10-1.0.7.

- Fixed CI detection in 1.0.10.
- Updated plugins in 1.0.11.
- Added tsd handling in 1.0.7.

# 2016-01-18 - 1.0.5 - docs

Early documentation and project setup updates across 1.0.9-1.0.0.

- Improved logging and general project information.
- Added Travis support.
- Added declaration file generation in 1.0.3.
- Restructured code and created development documentation in 1.0.1.
- Improved .gitignore in 1.0.0.

# 2016-01-14 - 0.0.7 - bootstrap

Initial project bootstrapping and early setup across 0.0.7-0.0.0.

- Added .gitignore and package.json.
- Added initial bin handling.
- Reached a working first implementation.
- Included assorted cleanup and update commits.