

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

npm 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

`npm` 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.

Revision #5

Created 2026-03-28 10:49:29 UTC by foss.global Team

Updated 2026-03-28 12:15:26 UTC by foss.global Team