

readme.md for @git.zone/tsbundle

A powerful multi-bundler tool supporting **esbuild**, **rolldown**, and **rspack** for painless bundling of web projects.

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.

Installation

```
# Global installation for CLI usage
pnpm add -g @git.zone/tsbundle

# Local installation for project usage
pnpm add --save-dev @git.zone/tsbundle
```

Quick Start

Interactive Setup

The easiest way to get started is with the interactive wizard:

```
tsbundle init
```

This guides you through setting up your bundle configuration with preset options:

- **element** - Web component / element bundle (`./ts_web/index.ts` -> `./dist_bundle/bundle.js`)
- **website** - Full website with HTML and assets (`./ts_web/index.ts` -> `./dist_serve/bundle.js`)
- **npm** - NPM package bundle (`./ts/index.ts` -> `./dist_bundle/bundle.js`)
- **custom** - Configure everything manually

Build Your Bundles

Once configured, simply run:

```
tsbundle
```

Your bundles will be built according to your `smartconfig.json` configuration.

CLI Commands

Command	Description
<code>tsbundle</code>	Build all bundles from <code>smartconfig.json</code> configuration
<code>tsbundle custom</code>	Same as above (explicit)
<code>tsbundle init</code>	Interactive wizard to create/update bundle configuration

Configuration

tsbundle uses `smartconfig.json` for configuration. Here's an example:

```
{
  "@git.zone/tsbundle": {
    "bundles": [
      {
        "from": "./ts_web/index.ts",
        "to": "./dist_bundle/bundle.js",
        "outputMode": "bundle",
        "bundler": "esbuild",

```

```

    "production": false
  },
  {
    "from": "./ts_web/index.ts",
    "to": "./dist_serve/bundle.js",
    "outputMode": "bundle",
    "bundler": "esbuild",
    "includeFiles": ["/html/**/*.html", "/assets/**/*."]
  }
]
}
}

```

Bundle Configuration Options

Option	Type	Default	Description
<code>from</code>	<code>string</code>	-	Entry point TypeScript file
<code>to</code>	<code>string</code>	-	Output file path
<code>outputMode</code>	<code>"bundle" "base64ts"</code>	<code>"bundle"</code>	Output format (see below)
<code>bundler</code>	<code>"esbuild" "rolldown" "rspack"</code>	<code>"esbuild"</code>	Which bundler to use
<code>production</code>	<code>boolean</code>	<code>false</code>	Enable minification
<code>includeFiles</code>	<code>string[]</code>	<code>[]</code>	Glob patterns for additional files (HTML, assets)
<code>maxLineLength</code>	<code>number</code>	<code>0</code> (unlimited)	For <code>base64ts</code> mode: max chars per line in output

Output Modes

`bundle` (default)

Standard JavaScript bundle output. Additional files specified in `includeFiles` are copied to the output directory.

`base64ts`

Generates a TypeScript file with base64-encoded content - perfect for **Deno compile** scenarios where you need everything embedded in a single executable:

```
// Auto-generated by tsbundle
export const files: { path: string; contentBase64: string }[] = [
  { path: "bundle.js", contentBase64: "Y29uc3QgaGVsbG8gPSAid29ybGQi..." },
  { path: "index.html", contentBase64: "PCFET0NUWVBFIGh0bWw+..." },
];
```

If you're working with AI tools that have line length limitations, set `maxLength` (e.g., `200`) to split long base64 strings across multiple lines.

Available Bundlers

tsbundle supports three modern bundlers, each with different strengths:

Bundler	Speed	Bundle Size	Best For
esbuild	Fastest	Medium	Development, quick iterations
rolldown	Fast	Smallest	Production builds, tree-shaking
rspack	Fast	Largest (webpack runtime)	Webpack compatibility

API Usage

TsBundle Class

The core bundling class, usable programmatically:

```
import { TsBundle } from '@git.zone/tsbundle';

const bundler = new TsBundle();

await bundler.build(
  process.cwd(),           // Working directory
  './src/index.ts',       // Entry point
  './dist/bundle.js',     // Output path
  {
    bundler: 'esbuild',    // 'esbuild' | 'rolldown' | 'rspack'
```

```
    production: true
  }
);
```

Each bundler runs in a separate child process via `smartspawn.ThreadSimple`, keeping the main process clean and isolated from bundler-specific dependencies.

HtmlHandler Class

Process and optionally minify HTML files:

```
import { HtmlHandler } from '@git.zone/tsbundle';

const htmlHandler = new HtmlHandler();

await htmlHandler.processHtml({
  from: './html/index.html',
  to: './dist/index.html',
  minify: true
});
```

AssetsHandler Class

Copy static assets between directories:

```
import { AssetsHandler } from '@git.zone/tsbundle';

const assetsHandler = new AssetsHandler();

await assetsHandler.processAssets({
  from: './assets',
  to: './dist/assets'
});
```

Base64TsOutput Class

Generate TypeScript files with base64-encoded content for embedding:

```
import { Base64TsOutput } from '@git.zone/tsbundle';

const output = new Base64TsOutput(process.cwd());
output.addFile('bundle.js', bundleBuffer);
await output.addFilesFromGlob('./html/**/*.html');
await output.writeToFile('./ts/embedded-bundle.ts', 200); // optional maxLineLength
```

CustomBundleHandler Class

Process multiple bundle configurations from `smartconfig.json`:

```
import { CustomBundleHandler } from '@git.zone/tsbundle';

const handler = new CustomBundleHandler(process.cwd());
const hasConfig = await handler.loadConfig();
if (hasConfig) {
  await handler.processAllBundles();
}
```

Embedding for Deno Compile

For single-executable scenarios with Deno:

```
tsbundle init
# Select "custom", set outputMode to "base64ts"
```

Config:

```
{
  "@git.zone/tsbundle": {
    "bundles": [
      {
        "from": "./ts_web/index.ts",
        "to": "./ts/embedded-bundle.ts",
        "outputMode": "base64ts",
        "bundler": "esbuild",
        "production": true,
        "includeFiles": [".html/index.html"],
```

```
    "maxLength": 200
  }
]
}
}
```

Then in your Deno app:

```
import { files } from './ts/embedded-bundle.ts';

// Decode and serve your embedded files
const bundle = files.find(f => f.path === 'bundle.js');
const html = files.find(f => f.path === 'html/index.html');

const bundleContent = atob(bundle.contentBase64);
const htmlContent = atob(html.contentBase64);
```

Project Structure Recommendations

```
your-project/
├─ ts_web/           # Web bundle entry points
│  └─ index.ts
├─ ts/              # Library/node entry points
│  └─ index.ts
├─ html/            # HTML templates
│  └─ index.html
├─ assets/          # Static assets (images, fonts, etc.)
├─ dist_bundle/     # Output for element/npm bundles
├─ dist_serve/      # Output for website bundles
└─ smartconfig.json # tsbundle configuration
```

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.

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.

Revision #5

Created 2026-03-28 10:49:27 UTC by foss.global Team

Updated 2026-03-28 12:15:24 UTC by foss.global Team