

## Build tools

- [Browserify](#)
- [Duo](#)
- [Grunt](#)
- [Gulp](#)
- [Jspm](#)
- [Webpack](#)
- [MSBuild](#)
- [NuGet](#)

# Browserify

## 安装

```
npm install tsify
```

## 使用命令行交互

```
browserify main.ts -p [ tsify --noImplicitAny ] > bundle.js
```

## 使用API

```
var browserify = require("browserify");
var tsify = require("tsify");

browserify()
  .add('main.ts')
  .plugin('tsify', { noImplicitAny: true })
  .bundle()
  .pipe(process.stdout);
```

更多详细信息: [smrq/tsify](#)

# Duo

## 安装

```
npm install duo-typescript
```

## 使用命令行交互

```
duo --use duo-typescript entry.ts
```

## 使用API

```
var Duo = require('duo');
var fs = require('fs')
```

```
var path = require('path')
var typescript = require('duo-typescript');

var out = path.join(__dirname, "output.js")

Duo(__dirname)
  .entry('entry.ts')
  .use(typescript())
  .run(function (err, results) {
    if (err) throw err;
    // Write compiled result to output file
    fs.writeFileSync(out, results.code);
  });
```

更多详细信息: [frankwallis/duo-typescript](https://github.com/frankwallis/duo-typescript)

# Grunt

## 安装

```
npm install grunt-ts
```

## 基本Gruntfile.js

```
module.exports = function(grunt) {
  grunt.initConfig({
    ts: {
      default : {
        src: ["**/*.ts", "!node_modules/**/*.ts"]
      }
    }
  });
  grunt.loadNpmTasks("grunt-ts");
  grunt.registerTask("default", ["ts"]);
};
```

更多详细信息: [TypeStrong/grunt-ts](https://github.com/TypeStrong/grunt-ts)

# Gulp

## 安装

```
npm install gulp-typescript
```

## 基本gulpfile.js

```
var gulp = require("gulp");
var ts = require("gulp-typescript");

gulp.task("default", function () {
  var tsResult = gulp.src("src/*.ts")
    .pipe(ts({
      noImplicitAny: true,
```

```
        out: "output.js"
    }));
    return tsResult.js.pipe(gulp.dest('built/local'));
});
```

更多详细信息: [ivogabe/gulp-typescript](https://github.com/ivogabe/gulp-typescript)

## Jspm

### 安装

```
npm install -g jspm@beta
```

*注意: 目前jspm的0.16beta版本支持TypeScript*

更多详细信息: [TypeScriptSamples/jspm](https://github.com/TypeScriptSamples/jspm)

## Webpack

### 安装

```
npm install ts-loader --save-dev
```

### 基本webpack.config.js

```
module.exports = {
  entry: './src/index.tsx',
  output: {
    filename: 'bundle.js'
  },
  resolve: {
    // Add '.ts' and '.tsx' as a resolvable extension.
    extensions: ['', '.webpack.js', '.web.js', '.ts', '.tsx', '.js']
  },
  module: {
    loaders: [
      // all files with a '.ts' or '.tsx' extension will be handled by 'ts-loader'
      { test: /\.tsx?$/, loader: "ts-loader" }
    ]
  }
};
```

查看[更多关于ts-loader的详细信息](#)

或者

- [awesome-typescript-loader](https://github.com/ivogabe/gulp-typescript)

## MSBuild

更新工程文件，包含本地安装的Microsoft.TypeScript.Default.props（在顶端）和Microsoft.TypeScript.targets（在底部）文件：

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="4.0" DefaultTargets="Build" xmlns="http://schemas.microsoft.com/build/2009"
  <!-- Include default props at the bottom -->
  <Import
    Project="$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v$(VisualStudioVersion)\Microsoft.TypeScript.Default.props"
    Condition="Exists('$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v$(VisualStudioVersion)\Microsoft.TypeScript.Default.props')"/>

  <!-- TypeScript configurations go here -->
  <PropertyGroup Condition="'$(Configuration)' == 'Debug'">
    <TypeScriptRemoveComments>false</TypeScriptRemoveComments>
    <TypeScriptSourceMap>true</TypeScriptSourceMap>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)' == 'Release'">
    <TypeScriptRemoveComments>true</TypeScriptRemoveComments>
    <TypeScriptSourceMap>false</TypeScriptSourceMap>
  </PropertyGroup>

  <!-- Include default targets at the bottom -->
  <Import
    Project="$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v$(VisualStudioVersion)\Microsoft.TypeScript.targets"
    Condition="Exists('$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v$(VisualStudioVersion)\Microsoft.TypeScript.targets')"/>
</Project>
```

关于配置MSBuild编译器选项的更多详细信息，请参考：[\[在MSBuild里使用编译选项\]](#)  
(./Compiler Options in MSBuild.md)

## NuGet

- 右键点击 -> Manage NuGet Packages
- 查找Microsoft.TypeScript.MSBuild
- 点击Install
- 安装完成后，Rebuild。

更多详细信息请参考[Package Manager Dialog](#)和[using nightly builds with NuGet](#)