原文地址：<https://segmentfault.com/a/1190000011744210#articleHeader12>

1. **引入ts**

npm i ts-loader typescript tslint tslint-loader tslint-config-standard --save-dev

1. **添加 tsconfig.json**

tsconfig.json是ts的编译配置文件

{

    "include": [

      "src/\*\*/\*"

    ],

    "exclude": [

      "node\_modules"

    ],

    "compilerOptions": {

      "allowSyntheticDefaultImports": true,

      "experimentalDecorators": true,

      "allowJs": true,

      "module": "esnext",

      "target": "es5",

      "moduleResolution": "node",

      "isolatedModules": true,

      "lib": [

        "dom",

        "es5",

        "es2015.promise"

      ],

      "sourceMap": true,

      "pretty": true

    }

  }

1. **tslint.json**

Tslint.json是ts的书写规范配置文件

{

    "extends": "tslint-config-standard",

    "globals": {

      "require": true

    }

}

1. **src/vue-shim.d.ts**

vue-shim.d.ts是让ts识别vue，如果不配置这个文件，那么如下语句会报错

import App from './App.vue'

vue-shim.d.ts文件配置：

declare module "\*.vue" {

    import Vue from "vue";

    export default Vue;

}

1. **修改webpack文件**

入口修改

    entry: {

        Index: \_\_dirname + "/src/Index.ts"

    },

规则设置

            {

                test: /\.ts$/,

                exclude: /node\_modules/,

                enforce: 'pre',

                loader: 'tslint-loader'

            },

            {

                test: /\.tsx?$/,

                loader: 'ts-loader',

                exclude: /node\_modules/,

                options: {

                    appendTsSuffixTo: [/\.vue$/],

                }

            },

添加后缀解析

    resolve: {

        extensions: ['.vue', '.js', ".ts", "tsx", ".css", '\*'],

        modules: [libPath, nodeModPath]     // 模块路径

    }

整体配置如下：

var path = require("path"); // 获取环境路径模块

var libPath = path.resolve('./src/lib');    // 获取保存第三方库（即模块）的路径

var nodeModPath = path.resolve(\_\_dirname, './node\_modules');    // 获取模块路径

const HtmlWebpackPlugin = require('html-webpack-plugin');

const VueLoaderPlugin = require('vue-loader/lib/plugin');

module.exports = {

    // 开发环境

    devtool: 'source-map',

    mode: "development",

    entry: {

        Index: \_\_dirname + "/src/Index.ts"

    },

    output: {

        path: \_\_dirname + "/build",

        filename: "[name].bundle.js",

        chunkFilename: "[id].chunk.js"

    },

    plugins: [       //插件

        new HtmlWebpackPlugin({

            filename: \_\_dirname + '/build/Index.html',      //html生成路径,

            template: \_\_dirname + '/src/Index.html',        // html模板路径,

            inject: 'body',   //打包之后的js插入的位置，true/'head'/'body'/false,

            chunks: ['Index']    //需要引入的js

        }),

        new VueLoaderPlugin()

    ],

    devServer: {

        contentBase: './build',

        port: 9000, //把默认端口号8080修改成9000

    },

    module: {

        rules: [

            {

                test: /\.js$/, //用正则匹配文件，用require或者import引入的都会匹配到

                loader: "babel-loader", //加载器名，就是上一步安装的loader

                exclude: /node\_modules/ //排除node\_modules目录，我们不加载node模块中的js哦~

            },

            {

                test: /\.ts$/,

                exclude: /node\_modules/,

                enforce: 'pre',

                loader: 'tslint-loader'

            },

            {

                test: /\.tsx?$/,

                loader: 'ts-loader',

                exclude: /node\_modules/,

                options: {

                    appendTsSuffixTo: [/\.vue$/],

                }

            },

            {

                test: /\.css$/,

                use: [

                    'style-loader',

                    'css-loader',

                    // 'postcss-loader'

                ]

            },

            {

                test: /\.(png|jpe?j|gif|svg)(\?.\*)?$/,

                loader: 'url-loader',

                options: {

                    limit: 10000,

                    name: 'img/[name].[ext]?[hash]'

                }

            },

            {

                test: /\.(woff2?|eot|ttf|otf)(\?.\*)?$/,

                loader: "url-loader",

                options: {

                    limit: 10000,

                    name: 'fonts/[name].[ext]?[hash]'

                }

            },

            {

                // vue-loader会把vue单文件直接转成js。

                test: /\.vue$/,

                loader: 'vue-loader'

            },

        ]

    },

    resolve: {

        extensions: ['.vue', '.js', ".ts", "tsx", ".css", '\*'],

        modules: [libPath, nodeModPath]     // 模块路径

    }

}

**示例：**

Index.ts

import Vue from 'vue/dist/vue.esm.js'

import router from './Router'             // 引入我们的路由文件

import App from './App.vue'

var vm = new Vue({

  router,         // 应用路由

  el: "#container",

  template: '<App />',

  components: { App }

})

App.vue

<template>

  <div id="app">

    <h3>App</h3>

    <!-- 跳转按钮 -->

    <router-link to="/">Go to Index</router-link>

    <router-link to="/About">Go to About</router-link>

    <!-- 用于放置匹配到的组件 -->

    <router-view></router-view>

  </div>

</template>

<script>

import Vue from "vue";

export default {

  name: "app",

  data() {

    return {

    };

  }

};

</script>