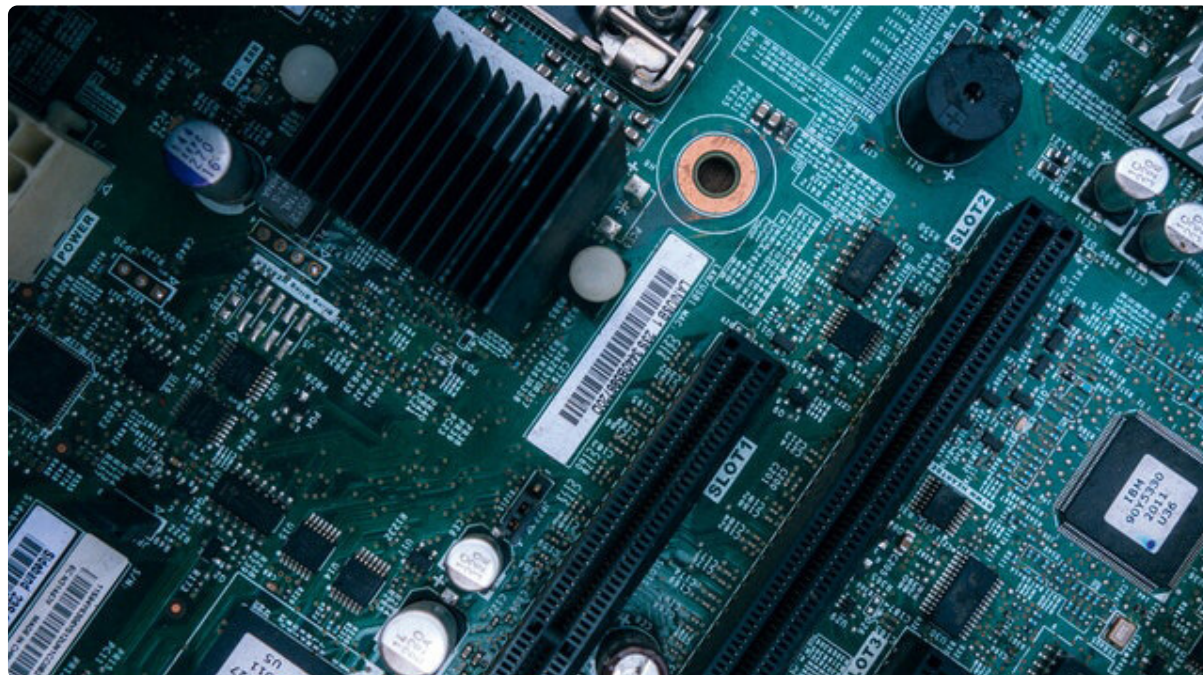


09 Webpack 中使用 TypeScript 开发项目

更新时间：2019-06-24 09:25:45



“上天赋予的生命，就是要为人类的繁荣和平和幸福而奉献。

——松下幸之助”

TypeScript 是微软公司提出来的一个 JavaScript 的超集语言，主要作用是给 JavaScript 增加静态类型检测系统和 ECMAScript 语法的扩展。TypeScript 代码可以编译为普通 JavaScript 代码在浏览器或者 Node.js 环境执行。本小节主要讨论在 Webpack 中集成 TypeScript 开发环境，不再做 TypeScript 的优劣比较和语法讲解，日常项目不涉及到 **TypeScript** 开发，则可以直接跳过本小节内容。

TypeScript 编译器和 tsconfig.json

TypeScript 代码不能直接在浏览器执行，所以需要编译器将 TypeScript 文件转换成可以在浏览器执行的 JavaScript。NPM 安装编译器方法：

```
npm i -g typescript
```

这时候我们新建一个 ts 文件（TypeScript 文件扩展名是 **ts**）：

```
// hello.ts
function sayHello(name: string) {
  return 'Hello, ' + name;
}

let webpack = 'Webpack';

console.log(sayHello(webpack));
```

然后使用 **tsc hello.ts**，执行编译后 **hello.ts** 被编译成了 **hello.js**：

```
function sayHello(name) {  
  return 'Hello, ' + name;  
}  
var webpack = 'Webpack';  
console.log(sayHello(webpack));
```

为了方便编译器和编辑器识别 TypeScript 项目，TypeScript 约定了 `tsconfig.json` 文件来存储项目配置，如果运行 `tsc` 时不指定输入文件，编译器则会查找项目目录中的这个文件，如果找不到则会依次向父级目录查找。比如这样：

```
{  
  "compilerOptions": {  
    "outFile": "dist/main.js",  
    "sourceMap": true  
  },  
  "files": ["src/index.ts", "src/source.ts"]  
}
```

关于 `tsconfig.json` 更多配置，可以继续浏览[官方文档](#)。

Webpack 集成 TypeScript

如果要想 Webpack 来识别 `ts` 文件，需要配合 TypeScript 的 loader，使用 `ts-loader` 可以方便地构建浏览器可以运行的 JS 代码。

安装 `ts-loader` 的命令为：`npm i ts-loader --save-dev`，

然后配置项目目录中的 `webpack.config.js`：

```
module.exports = {  
  entry: './src/app.ts',  
  output: {  
    filename: 'app.js',  
    path: './dist'  
  },  
  resolve: {  
    extensions: ['', '.webpack.js', '.web.js', '.ts', '.js']  
  },  
  module: {  
    loaders: [{test: /\.ts$/, loader: 'ts-loader'}]  
  }  
};
```

为了方便配置 TypeScript 项目，还可以在 Webpack 的项目根目录创建一个 `tsconfig.json` 文件。

小结

本小节主要介绍 TypeScript 的简单概念和 `tsconfig.json`，最后通过配置 `webpack.config.js` 来让 webpack 集成 TypeScript 编译功能。

专栏代码已经整理好给大家共享出来：

[点此下载 源码](#)

欢迎在这里发表留言，作者筛选后可公开显示



目前暂无任何讨论