利用webpack 搭建vue3+ts项目

# 一、基本框架

1、全局安装ts--npm install typescript -g

2、npm init

3、tsc --init

4、创建文件夹及文件，比如src,public等

5、创建webpack.config.js

6、安装webpack webpack-cli webpack-dev-server html-webpack-plugin

7、简单配置webpack.config.js---实验是否能够打包成功

const {

  Configuration

} = require("webpack")

const path = require("path");

const HtmlWebpackPlugin = require("html-webpack-plugin")

/\*\*

 \* @type {Configuration} //注意这种写法，加完注解以后，写配置项的时候有提示

 \*/

const config = {

  mode: 'development',

  entry: "./src/main.ts",

  output: {

    filename: "app.[hash].js",

    path: path.resolve(\_\_dirname, "./dist")

  },

  plugins: [

    new HtmlWebpackPlugin({

      template: "./public/index.html"

    })

  ]

}

module.exports = config;

package.json中

scripts: {

“dev”: “webpack-dev-server”,

“build”: “webpack”

}

# 二、引入vue

1、安装vue

2、安装vue-loader@next @vue/compiler-sfc（用于解析vue结尾的文件）

3、创建声明文件--不然的话ts不识别vue文件，也就是在ts文件中引入.vue文件会报错。

declare module "\*.vue" {

import { DefineComponent } from "vue"

const component: DefineComponent<{}, {}, any>

export default component

}

4、webpack.config.js中增加代码

const {VueLoaderPlugin} = require("vue-loader")

module: {

    rules: [

      {

        test: /\.vue$/,

        use: "vue-loader"

      }

    ]

  },

plugins: [

    new HtmlWebpackPlugin({

      template: "./public/index.html"

    }),

    new VueLoaderPlugin()

  ]

# 三、配置别名和自动识别扩展

resolve: {

    alias: {

     "@": path.resolve(\_\_dirname, "./src"),

     "@pages":path.resolve(\_\_dirname, "./src/pages")

    },

    extensions: [".js", ".vue", ".ts"]

  },

# 四、配置解析css,less文件

1、安装css-loader less-loader style-loader -D

2、安装less -S

3、webpack.config.js中增加配置文件

{

        test: /\.less$/,

        use: ["style-loader", "css-loader", "less-loader"]

      },

      {

        test: /\.css$/,

        use: ["style-loader", "css-loader"]

      }

# 五、每次打包清除dist目录

1、安装插件npm install clean-webpack-plugin

2、webpack.config.js中增加配置项

const {CleanWebpackPlugin} = require("clean-webpack-plugin")

plugins: [

    new HtmlWebpackPlugin({

      template: "./public/index.html"

    }),

    new VueLoaderPlugin(),

    new CleanWebpackPlugin()

  ]

# 六、支持vue组件中ts的写法

1、下载安装typescript ts-loader

2、webpack.config.js增加配置

{

        test: /\.ts$/,  //解析ts

        loader: "ts-loader",

        options: {

            configFile: path.resolve(process.cwd(), 'tsconfig.json'),

            appendTsSuffixTo: [/\.vue$/]

        },

      }

# 七、美化webpack的输出样式

1、安装friendly-errors-webpack-plugin

2、webpack.config.js中

const FriendlyErrorsWebpackPlugin = require("friendly-errors-webpack-plugin")

stats: "errors-only",

plugins: [

    new FriendlyErrorsWebpackPlugin({

      compilationSuccessInfo:{ //美化样式

          messages:['You application is running here http://localhost:8080']

      }

    })

  ]

# 八、性能优化

1、webpack.config.js中增加externals配置

externals: {

    vue: "Vue" //CDN 引入

  },

作用：构建的时候，vue不会包含在构建完成的项目中，我们可以通过cdn引入的方式引入vue文件。