安装webpack-dev-server

主要修改两个文件

package.json和webpack.config.js

package.json

**"scripts"**: {

**"build"**: **"cross-env NODE\_ENV=production webpack --config webpack.config.js"**,

**"dev"**: **"cross-env NODE\_ENV=development webpack-dev-server** --entry ./src/main.js --output-filename ./dist/bundle.js**”**

},

webpack.config.js

**const** path=require(**"path"**);*//是nodejs的一个基本包，用来处理路径的，在打包项目时用绝对路径，防止出错*

module.exports={

**target**:"web",*//webpack定义的目标是web平台*

**entry**:path.join(\_\_dirname,**"src/main.js"**),*//path.join就是把dirname和后面的路径拼接起来，形成绝对路径*

**output**:{

**path**:path.join(\_\_dirname,**"dist"**),

**filename**:**"bundle.js"**

},

**module**:{

**rules**:[

            {

**"test"**:/\.vue$/,

**"loader"**:**"vue-loader"**

},

            {

**"test"**:/\.css$/,

**"loader"**:**"style-loader!css-loader"**

},

            {

**"test"**:/\.(jpg|png|gif)$/,

**"loader"**:**"url-loader"**

},

**{**

**"test":/\.js$/,**

**"exclude":** **/node\_modules/,**

**"loader":"babel-loader",**

**"options":{**

**"presets": [“es2015”],**

**"plugins": ['transform-runtime']**

**}**

**}**

        ]

    }

};

//在这里要根据不同的环境做一个判断,

//因为这个配置文件同时需要用在开发环境和生产环境

//如何知道是在哪个环境,需要在package.json中运行npm的scripts时,设置一个环境变量,来标识到底是开发环境还是生产环境

//设置这个变量需安装一个依赖:cnpm install cross-evn

//因为mac电脑和windows电脑上设置这个环境变量的方式不同,

//在mac上用NODE\_ENV=production就能读取环境变量,而windows上要用set NODE\_ENV=production这种方式

//为了避免为不同的平台写不同的命令,就可以通过cross-env来实现

// "scripts": {

//     "build": "cross-env NODE\_ENV=production webpack --config webpack.config.js",

//     "dev": "cross-env NODE\_ENV=development webpack-dev-server --entry ./src/main.js --output-filename ./dist/bundle.js”

// }

webpack.config.js补充

**const** **path=require("path");**

**const** **webpack=require("webpack");**

**const** **isDev=process.env.NODE\_ENV=="development";***//在package.json里定义的环境变量，都可以通过process.env这个对象来得到*

**const** **config={**

**target:"web",***//webpack定义的目标是web平台*

**entry:path.join(\_\_dirname,"src/main.js"),**

**output:{**

**path:path.join(\_\_dirname,"dist"),**

**filename:"bundle.js"**

**},**

**module:{**

**rules:[**

**{**

**"test":/\.vue$/,**

**"loader":"vue-loader"**

**},**

**{**

**"test":/\.css/,**

**"loader":"style-loader!css-loader"**

**},**

**{**

**"test":/\.(jpg|png|gif)$/,**

**"loader":"url-loader"**

**},**

**{**

**"test":/\.js$/,**

**"exclude":** **/node\_modules/,**

**"loader":"babel-loader",**

**"options":{**

**"presets": [“es2015”],**

**"plugins": ['transform-runtime']**

**}**

**}**

**]**

**},**

**plugins:[**

**]**

**};**

**if(isDev){**

**config.devServer={**

**port:"9900",**

**host:"0.0.0.0",***//设置成0.0.0.0的好处是除了可以通过localhost和127.0.0.1访问,还能通过内网ip地址访问*

**overlay:{**

**errors:true***//当webpack编译过程中出现任何错误,让它显示在网页上面*

**},**

**open:true,***//启动webpack编译的时候自动打开浏览器*

**hot:true***//只刷新当前组件的代码不刷新整个页面*

**};**

*//设置hot后需要加相关插件*

**config.plugins.push(**

*//启动hot功能的插件*

**new** **webpack.HotModuleReplacementPlugin(),**

**new** **webpack.NoEmitOnErrorsPlugin()**

**)**

**}**

**module.exports=config;**

webpack.config.js其他补充

**const** **path=require("path");**

**const** **webpack=require("webpack");**

**const** **isDev=process.env.NODE\_ENV=="development";***//在package.json里定义的环境变量，都可以通过process.env这个对象来得到*

**const** **config={**

**target:"web",***//webpack定义的目标是web平台*

**entry:path.join(\_\_dirname,"src/main.js"),**

**output:{**

**path:path.join(\_\_dirname,"dist"),**

**filename:"bundle.js"**

**},**

**module:{**

**rules:[**

**{**

**"test":/\.vue$/,**

**"loader":"vue-loader"**

**},**

**{**

**"test":/\.css/,**

**"loader":"style-loader!css-loader"**

**},**

**{**

**"test":/\.(jpg|png|gif)$/,**

**"loader":"url-loader"**

**},**

**{**

**"test":/\.js$/,**

**"exclude":** **/node\_modules/,**

**"loader":"babel-loader",**

**"options":{**

**"presets": [“es2015”],**

**"plugins": ['transform-runtime']**

**}**

**}**

**]**

**},**

**plugins:[**

*//在使用vue框架的时候需要用一个webpack插件*

*//vue项目在打包时会区分不同的环境打包*

*//在这里定义了,在写的js代码中也可以引用process.evn这个对象*

**new** webpack.DefinePlugin({

**"process.env"**:{

**NODE\_ENV**:isDev?**'"development"'**:**'"production"'**

}

        })

**]**

**};**

**if(isDev){**

*//帮助在页面上调试代码,在项目里使用的是.vue的文件开发模式,使用的也是es6代码*

*//这些代码在浏览器里是不能直接运行的,如果直接去调试浏览器代码是经过编译的*

*//所以需要使用source-map去进行代码的映射,当在浏览器调试的时候还是自己编写的源码*

*//这样更容易定位到错误,webpack提供了devtool*

**config.devtool="#cheap-module-eval-source-map";**

**config.devServer={**

**port:"9900",**

**host:"0.0.0.0",***//设置成0.0.0.0的好处是除了可以通过localhost和127.0.0.1访问,还能通过内网ip地址访问*

**overlay:{**

**errors:true***//当webpack编译过程中出现任何错误,让它显示在网页上面*

**},**

**open:true,***//启动webpack编译的时候自动打开浏览器*

**hot:true***//只刷新当前组件的代码不刷新整个页面*

**};**

*//设置hot后需要加相关插件*

**config.plugins.push(**

*//启动hot功能的插件*

**new** **webpack.HotModuleReplacementPlugin(),**

**new** **webpack.NoEmitOnErrorsPlugin()**

**)**

**}**

**module.exports=config;**