快速搭建项目总结

目录

**[1. 准备工作](#_Toc2885_WPSOffice_Level1)** **[1](#_Toc2885_WPSOffice_Level1)**

[(1) 创建一个版本仓库](#_Toc5112_WPSOffice_Level2) [1](#_Toc5112_WPSOffice_Level2)

[(2) 本地拉取github项目仓库 git clone](#_Toc18334_WPSOffice_Level2) [1](#_Toc18334_WPSOffice_Level2)

**[2. 创建项目](#_Toc5112_WPSOffice_Level1)** **[1](#_Toc5112_WPSOffice_Level1)**

[(1) 安装vue脚手架 yarn global add vue-cli](#_Toc21818_WPSOffice_Level2) [1](#_Toc21818_WPSOffice_Level2)

[(2) 新建vue项目 vue init webpack [项目名称]](#_Toc15978_WPSOffice_Level2) [1](#_Toc15978_WPSOffice_Level2)

**[3. 安装常用依赖包及配置](#_Toc18334_WPSOffice_Level1)** **[1](#_Toc18334_WPSOffice_Level1)**

[(1) Less](#_Toc27073_WPSOffice_Level2) [1](#_Toc27073_WPSOffice_Level2)

[(2) Axios](#_Toc7220_WPSOffice_Level2) [2](#_Toc7220_WPSOffice_Level2)

[(3) 安装路由](#_Toc27625_WPSOffice_Level2) [2](#_Toc27625_WPSOffice_Level2)

[(4) Vuex](#_Toc31315_WPSOffice_Level2) [2](#_Toc31315_WPSOffice_Level2)

[(5) Ts](#_Toc32360_WPSOffice_Level2) [2](#_Toc32360_WPSOffice_Level2)

[(6) 配置build](#_Toc11355_WPSOffice_Level2) [3](#_Toc11355_WPSOffice_Level2)

**[4. 新建文件夹（新建在src文件目录下）](#_Toc21818_WPSOffice_Level1)** **[3](#_Toc21818_WPSOffice_Level1)**

[(1) 新建service--服务端口](#_Toc23385_WPSOffice_Level2) [3](#_Toc23385_WPSOffice_Level2)

[(2) 新建store--状态管理](#_Toc11660_WPSOffice_Level2) [3](#_Toc11660_WPSOffice_Level2)

[(3) 新建config--api](#_Toc15454_WPSOffice_Level2) [3](#_Toc15454_WPSOffice_Level2)

[(4) 新建Style--公共样式](#_Toc22229_WPSOffice_Level2) [3](#_Toc22229_WPSOffice_Level2)

**[Ts配置文件内容](#_Toc15978_WPSOffice_Level1)** **[4](#_Toc15978_WPSOffice_Level1)**

**[目录大概结构](#_Toc27073_WPSOffice_Level1)** **[4](#_Toc27073_WPSOffice_Level1)**

总结：刘圆圆

日期：2019/1/9

1. 准备工作
   1. 创建一个版本仓库
   2. 本地拉取github项目仓库 git clone
2. 创建项目
   1. 安装vue脚手架 yarn global add vue-cli
   2. 新建vue项目 vue init webpack [项目名称]
3. 安装常用依赖包及配置
   1. Less
      1. yarn add less less-loader --dev

配置：**less{**

**test: /\.less$/,**

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

**}**

* + 1. 使用方法<style scoped lang='less'>
  1. Axios
     1. Yarn add axios
     2. Axios低版本浏览器补丁 yarn add es6-promise
  2. 安装路由
     1. Yarn add vue-router
  3. Vuex
     1. Yarn add vuex
  4. Ts
     1. Yarn add typescript ts-loader --dev
     2. 在main.ts同级新建vue-shims.d.ts文件

declare module "\*.vue" {

import Vue from "vue";

export default Vue;

}

* + 1. 在根目录新建tsconfig.json文件
       1. Tsc --init（创建方法）；
    2. 配置webpack

{

   test: /\.tsx?$/,

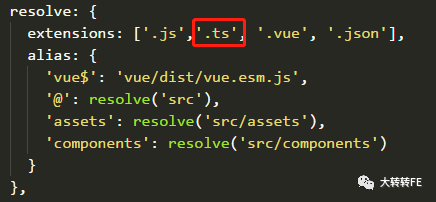
   loader: 'ts-loader',    exclude: /node\_modules/,      options: {

     appendTsSuffixTo: [/\.vue$/],

   }

 }

最后把 .ts 后缀添加上就OK了，在webpack.base.conf.js文件下



使用配置entry: {

app: './src/main.ts'},

将ts-loader从4.0降低到3.1.1解决问题。是由于webpack 和ts-loader版本不兼容造成的。

* 1. 配置build
     1. 打开webpack.dev.conf.js文件，找到publicPath: config.dev.assetsPublicPath,按Ctrl点击，跳转到index.js文件
     2. 其中dev是开发环境，build是构建版本，找到build下面的assetsPublicPath: '/',然后修改为assetsPublicPath: './',即“/”前加点。
     3. Yarn run build

1. 新建文件夹（新建在src文件目录下）
   1. 新建service-->服务端口.ts
   2. 新建store-->状态管理.ts（actions,getter,index,moutation）
   3. 新建config-->api.ts
   4. 新建Style-->公共样式.less

Ts配置文件内容

{

"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",

"scripthost"

],

"sourceMap": true,

"pretty": true,

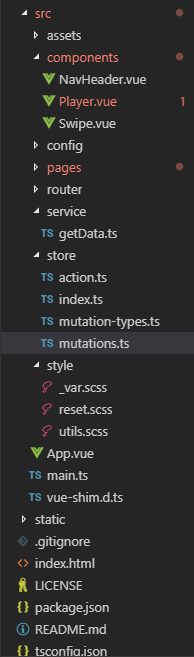
"strictFunctionTypes": false,

"importHelpers": true

}

}

目录大概结构



引入字体font-awesome

Yarn add font-awesome

Main.ts import "font-awesome/css/font-awesome.min.css";

引用echarts

父子通信https://www.cnblogs.com/hai-cheng/p/8021982.html