作为一名后端开发，这个有点复杂

**1.我们执行如下命令**

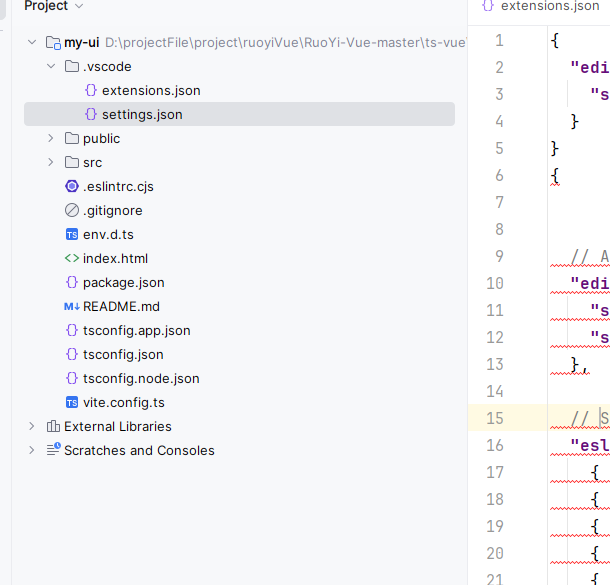
npm create [vue@latest](http://vue@latest)

选择ts，pinia和eslint以及vue router

所以我们前端采用

ts + pinia + eslint +vue router做单页面开发

**2.前端目录**



1.其中.vscode一看即使vscode的配置文件，我采用idea开发，直接删除

2.public -- 放置静态文件的

3..eslintrc.cjs --

4..gitignore -- git的忽略文件

5.env.d.ts -- 让ts支持各个格式的文件本来ts是不支持或者无法识别其它格式的文件的，例如.txt、.css、.jpg等，它这一句<reference types="vite/client" />就可以支持了，因为它在client.d.ts里面配置了对各个文件类型的支持。不过我们不用，在src下建立一个types文件夹，将改文件移入

6.index.html -- 项目文件入口 ，只做一件事，就是把src的main.ts引入，main.ts支撑着整个Vue3应用。

7、package.json --  依赖包的管理信息。

8.README.md --  项目说明文件，可以删除

9.3个tsconfig的json文件 -- ts的配置文件。但是核心是tsconfig.json这里面指定了其他配置文件如tsconfig.app.json和tsconfig.node.json，所以用一个就够了，删除这两个保留tsconfig.json，整合一下两个文件的内容，删除了一些多余的

{

"extends": ["@tsconfig/node20/tsconfig.json","@vue/tsconfig/tsconfig.dom.json"],

"compilerOptions": {

"noEmit": true, // 设置为 true 表示 TypeScript 编译器不会生成输出文件，这通常用于开发环境，防止每次保存时都生成新的 JavaScript 文件。

"module": "ESNext", //设置为 ESNext，表明输出的模块系统是 ES 模块的最新版本。

"moduleResolution": "Bundler",

"types": ["node"],

"baseUrl": ".", // 用于解析非相对模块名称的基目录

"paths": { // 模块名到基于 baseUrl 的路径映射的列表

"@/\*": ["./src/\*"]

}

},

"include": ["src/\*\*/\*", "src/\*\*/\*.vue", "vite.config.ts"], // 指定被编译文件所在的目录

"exclude": ["node\_modules", "dist"] // 指定不需要被编译的目录

}

10.vite.config.ts --  整个工程的配置文件，它可以修改输出目录、配置代理服务器、添加自定义插件等