**app开发文档**

**项目代码获取：**

公司git仓库下载，地址可自主查询，目前外网git下载，没有git工具请先行下载，一个类似svn的多人协同开发代码管理工具，是目前使用最广泛的代码管工具之一。

**环境搭建及前端项目启动：**

打开项目目录，阅读readme.md 获取相关信息。

此处推荐使用visual studio code 作为前端项目编辑器，有几处优点：

1. 清晰可辨的语法高亮提示，可随意在扩展项里更换主题。
2. 源代码管理自带git管理，与本机git用户相连，亦可使用svn代替，代替方法自行百度。
3. 具有自带debugger功能，方便代码调试。
4. 丰富的扩展插件（extensions），使得coding更轻松，列举几个可能对app项目开发有用的插件：

**Chinese (Simplified) Language**：中文语言包

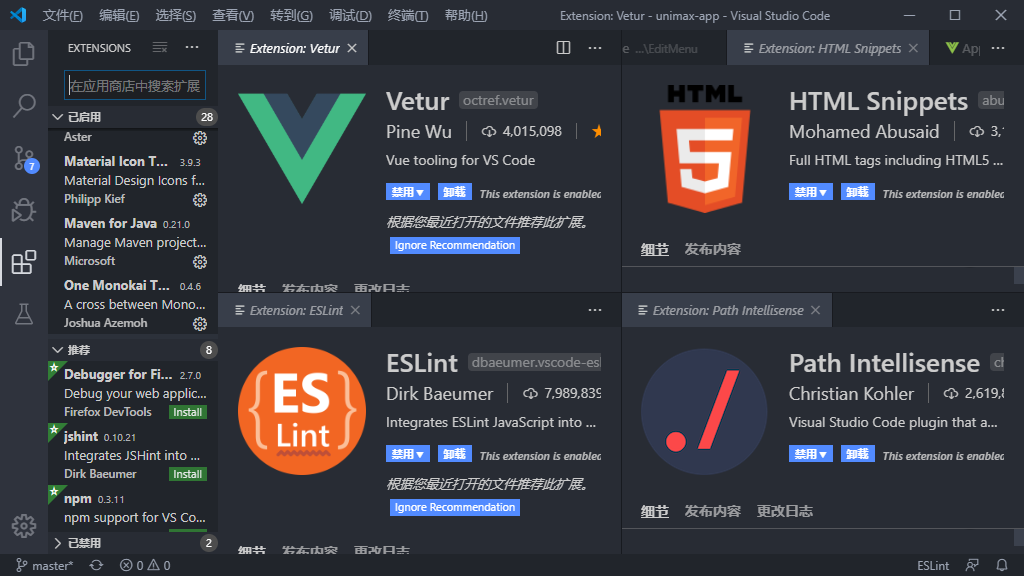
**Vetur** ： vue开发必备，大量vue相关语法提示，模版指令提示等等；

**Path Intellisense** ：路径引用提示，只需记得大致位置，一路提示选择，避免路径写错；

**HTML Snippets** ：html的emmet，html简写提示；

**Auto Rename Tag**：自动闭合标签，标头修改自动改标尾；

**ESLint**：app产品包含eslint配置，开启eslint插件后，随时检查会出现问题的代码语句，尽量遵守，既避免低级语法bug，又规范格式，整洁美观；



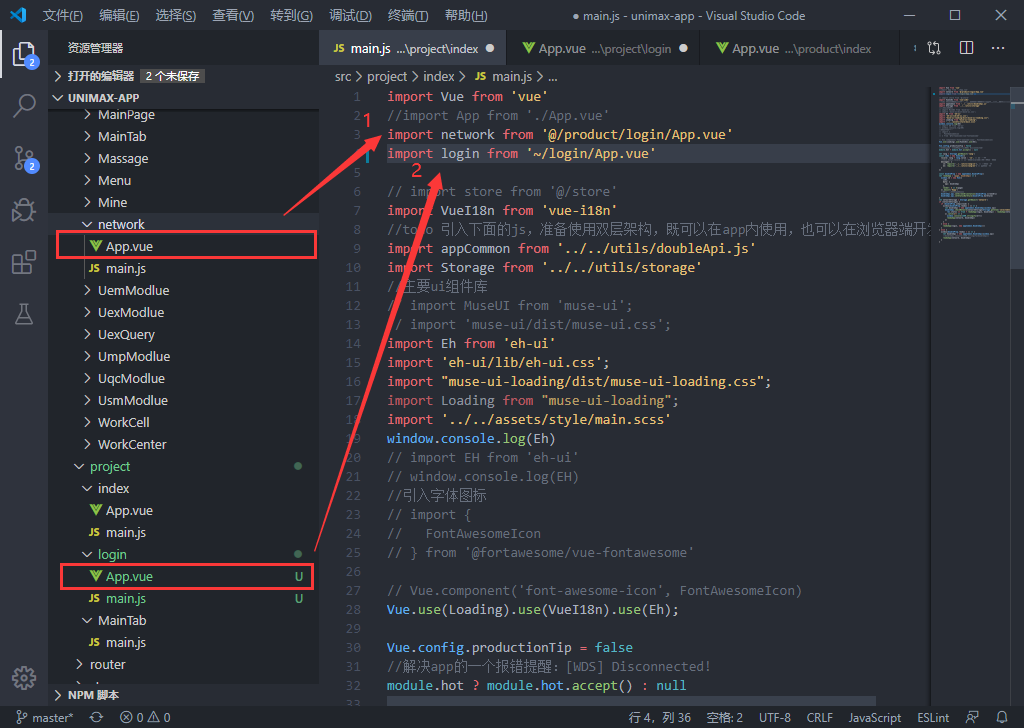
**项目结构一览：**





**项目开发流程：**

打开项目目录 src / project，此目录存放项目二开的文件，首先必须有index文件夹，其中包含main.js和 App.vue，index文件夹打包之后形成index.html，是作为默认的主页。

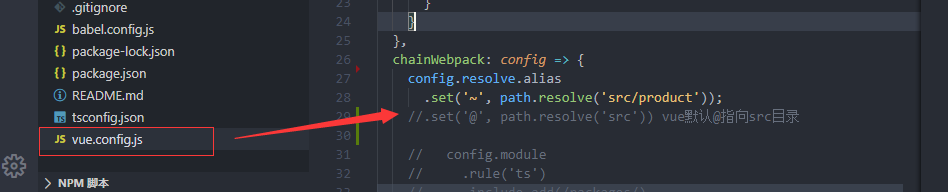


如上图所示，在index文件夹中，main.js作为一个页面承载配置信息的js，包含了所需要引入的页面（模版）的内容。

第1个箭头，从product文件夹下引入network/App.vue，即产品提供的网络设置页模版；

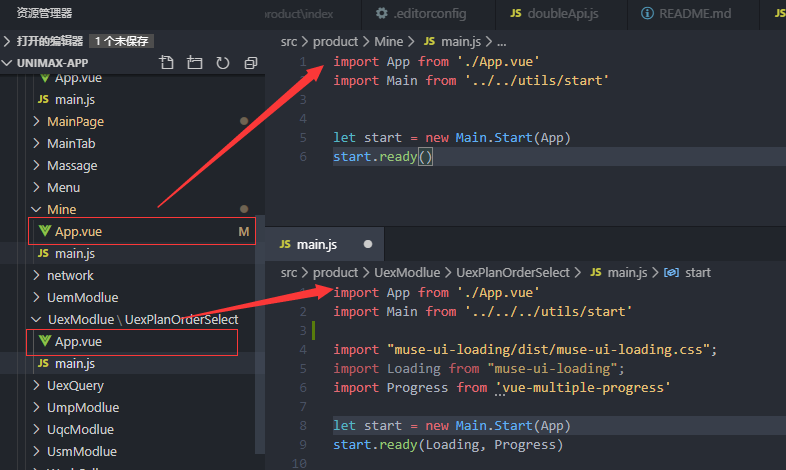
第2个箭头，从project文件夹下引入login/App.vue，即项目可修改的登录页模版；

在此需要说明的是，这个路径缩略符号是可以修改的，在vue.config.js中：



不过不建议再次修改，如果需要引入其他路径的模块文件，使用相对路径一样可以做到，例如在上图index/main.js中，第18行引入的样式文件。

在除index文件夹的其他页面目录中，main.js一般可以使用通用的格式，例如：



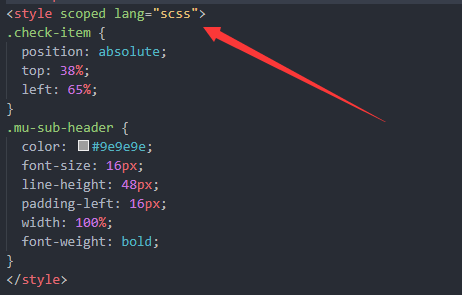
在上图中，有上下两个页面的mian.js文件。

上一个引入了页面需要加载的vue模版组件，和工具库中的start.js，将vue模版组件传入构建方法中调用ready，即可完成页面加载。

下一个main.js中，由于所引入的vue模版组件用到了俩个vue插件，loading和process，所以只需要将他们导入后，依次传入ready方法中，即可完成页面包括插件的加载。

在vue单组件页面中，组件写法需要大家熟悉vue的相关知识才能做好，其余并无太多特别需要说明的地方，使用**Vetur**、**Eslint** 扩展插件，在编写代码时会避免很多由于语法不熟悉导致的错误。

在组件内样式编写时，推荐添加属性lang="scss"，即开启sass预编译器，对css语法完全覆盖，提供更多高级写法，可通过学习sass了解更多。在编写有可能与其他组件内容相似的页面时，推荐在style标签增加scope属性，对css增加作用域，让该标签内的样式只作用于当前文件下。



**产品有一些可使用的工具库：**

对照工具库文档查看