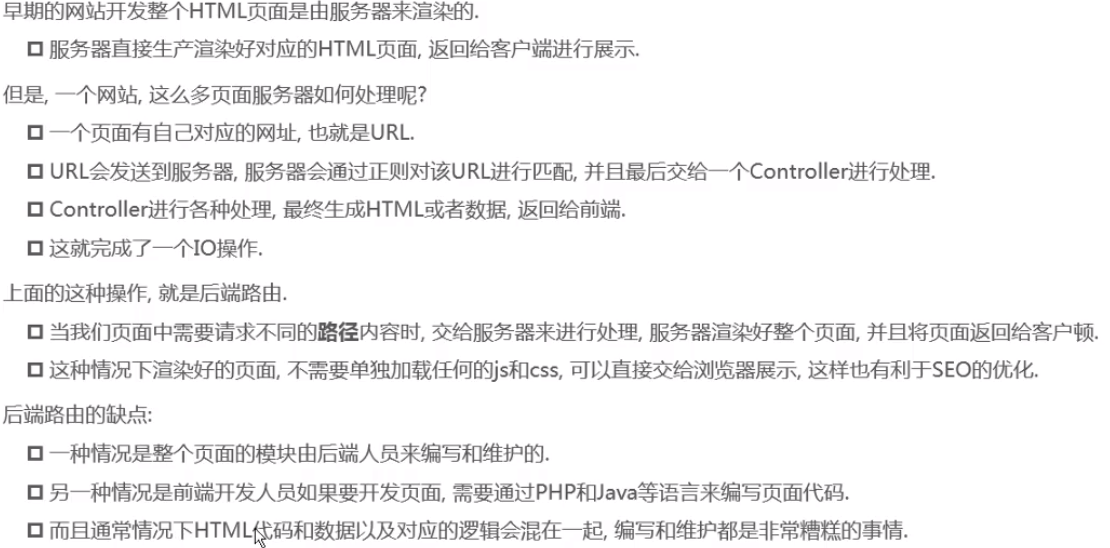
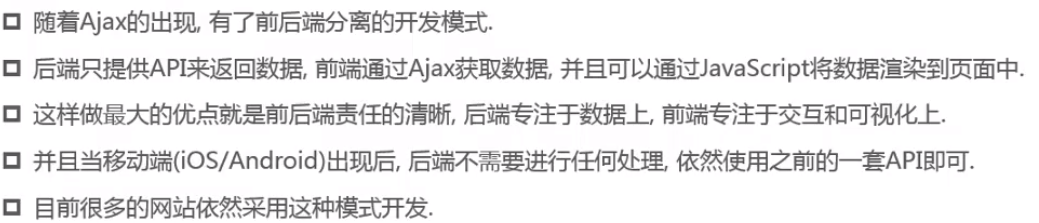
1、前端发展阶段

后端渲染阶段：

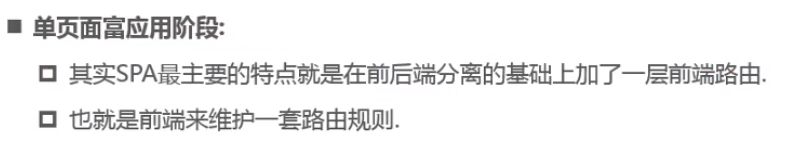


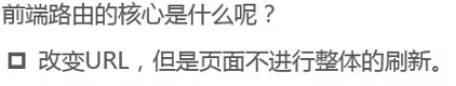
前后端分离，前端渲染阶段：



SPA(单页面富应用)页面阶段：由前端路由来进行管理各个页面

当url改变时前端路由会自动的去选择展示哪些组件，每个url有其各自的对应关系





2、改变url而不发生刷新

（1）通过改变location对象的hash属性来实现

location.hash = “test”

（2）通过history对象的pushState()方法来进行修改

history.pushState({},’’,’home’)，该方法类似于栈结构。

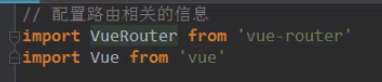
history.replaceState({},’’,’home’)，跟上面的push方法的主要区别是该方法为直接替换url因此浏览器无法进行返回等操作。

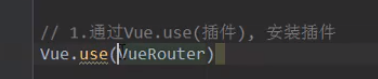
history.back()等于histroy.go(-1)，history.forward()等于history.go(1)，分别就相当于浏览器的前进与后退。

3、vue-router

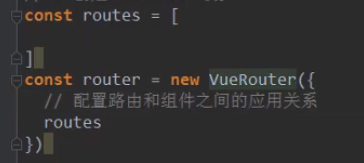
**（1）下述为CLI2环境下vue-router安装和基本配置**

新建工程的时候安装vue-router，在src下的router文件夹中新建index.js文件。先导入vue模块和router模块



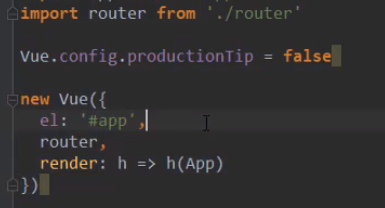


创建vue-router对象并配置映射关系，这里为了结构更清晰讲routes熟悉单独抽离了出来，它是一个数组的形式



在Vue实例中挂载vue-router对象，当然需要先导出然后在入口文件导入

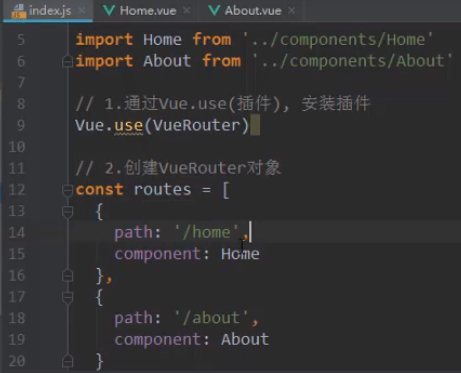




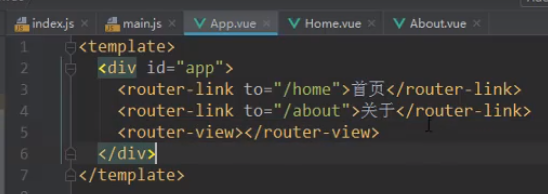
**（2）路由映射配置和呈现**

一个映射关系对应一个对象

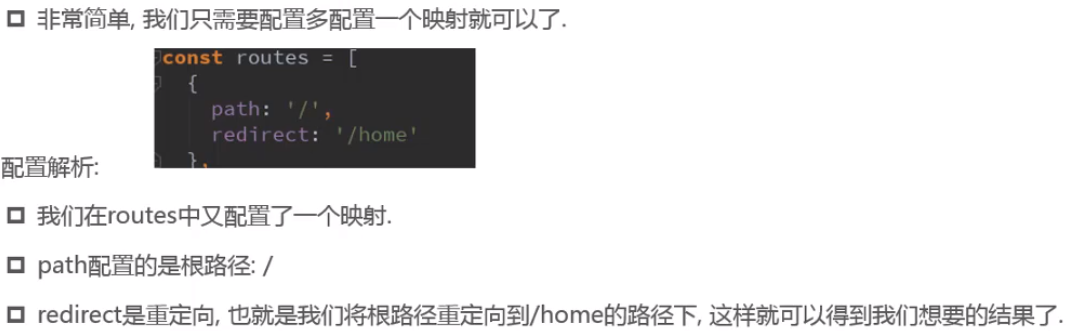
先导入需要的组件对象，然后配置映射关系



在app.vue的模板中添加router-link与router-view标签，这里的to属性的值与配置的映射关系中的path保持一致，router-link类似于a标签，router-view负责展示需要展示的组件，在路由切换时切换的是router-view挂载的组件，不会影响其他内容

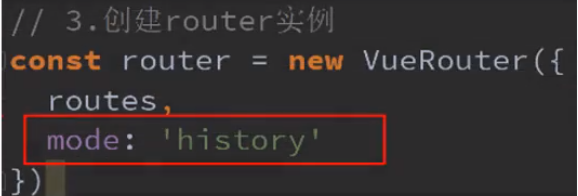


**（3）路由默认路径的设置**

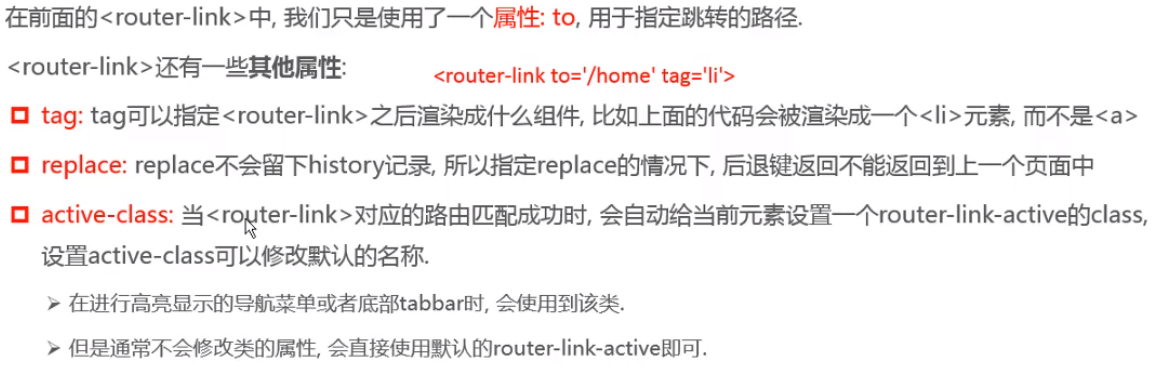


**（4）将默认的哈希路径模式改为history模式**

在创建vue-router对象时再添加一个mode属性，值设为history即可

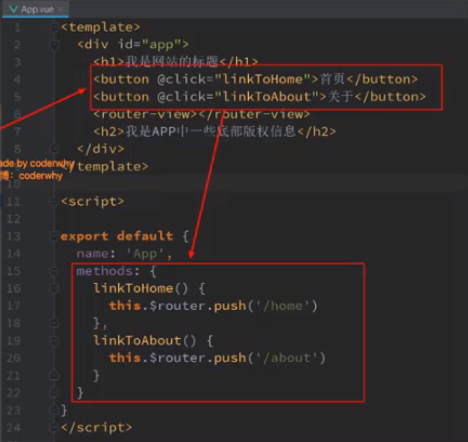


**（5）router-link标签补充**

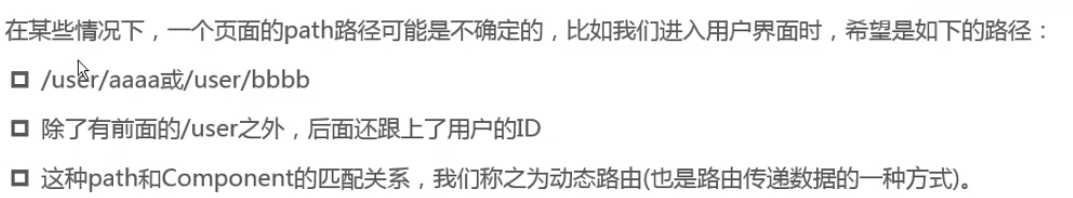


**（6）通过代码的方式跳转路由**

如果我们想不通过router-link标签实现跳转路由，可以自定义元素并为其添加click事件，代码如下，当然也有个相似的replace方法不多介绍



**（7）动态路由的使用**



这个时候需要在基础的映射关系的path上可以加入:加变量名的后缀

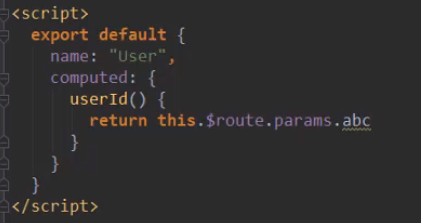


这样在使用router-link这种to路径时也需要加上相应的id名，当然正常是通过v-bind去绑定该属性并传递动态的值过去，这样点击用户的话路径就会展示带有用户名后缀的路径

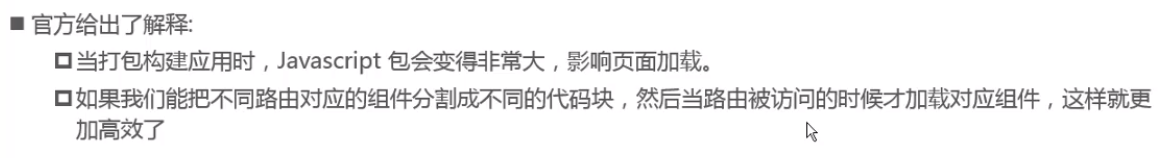


如果再想获取该后缀，可以在相应的组件中使用$route该对象，该对象获取的是当前处于活跃状态的路由，即被点击后的路由

通过在对象里this.$route.params.id就可以拿到当前id，params就是参数列表的意思



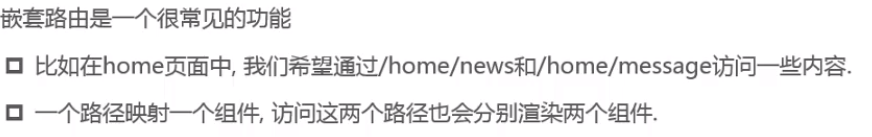
**（8）vue-router懒加载的使用**



有以下三种方式，现在一般都是使用第三种



**(9)嵌套路由的使用**



它的根本思路还是在一个组件中继续抽取几个不同的组件根据选择进行显示，所以应首先新建几个需要的子组件并在index.js文件中导入，然后在index.js其父组件中添加children属性配置映射关系

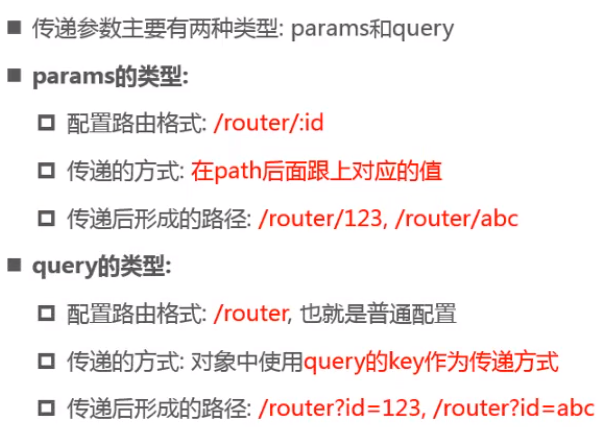


注意子模块的path中不能写/，最后在需要展示的父组件中通过router-link与router-view来进行选择性展示，注意这里的路径要写完整



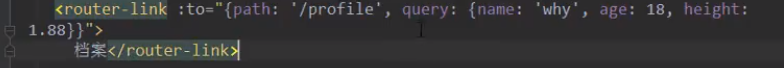
**（10）vue-router参数传递**

一般数据比较多的话都是使用query这种模式



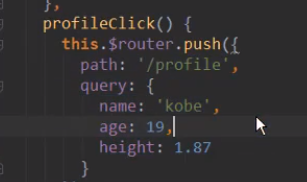
Params类型动态路由里提过不多说。第二种query类型的要点如下

<router-link v-bind:to=”{path:’路径’,query : {name:’xxx’,age:xx}}”>测试</router-link>这样之后通过$route.query就可以获得参数对象，通过$route.query.name就能拿到具体属性





如果想通过button那种形式传递query类型的话，代码如下

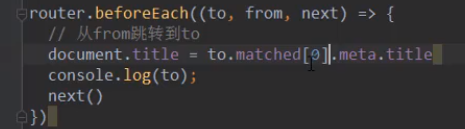


**（11）导航守卫**

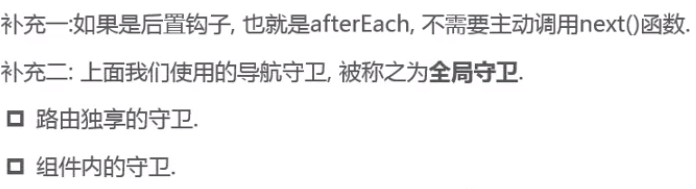
为了实现点击不同的例如首页，用户，关于而同时更改网页title可以借助导航守卫

先在映射关系中给相应的映射组件添加meta属性，再在index.js中调用router.beforeEach





导航守卫补充



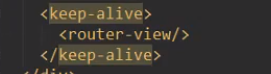
上述的beforeEach叫前置钩子，顾名思义是在路由跳转前调用，而后置钩子是跳转后调用。

4、keep-alive遇见vue-router

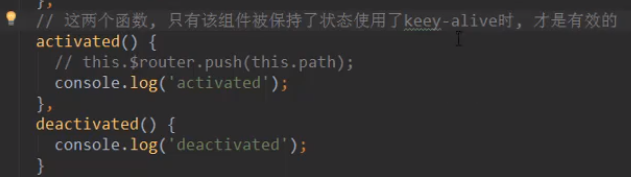




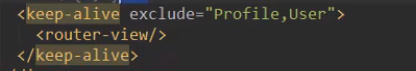
如果直接进行组件切换那么本质上是不断的销毁创建组件，这样的性能并不好，因此需要keep-alive，做法就如下图。

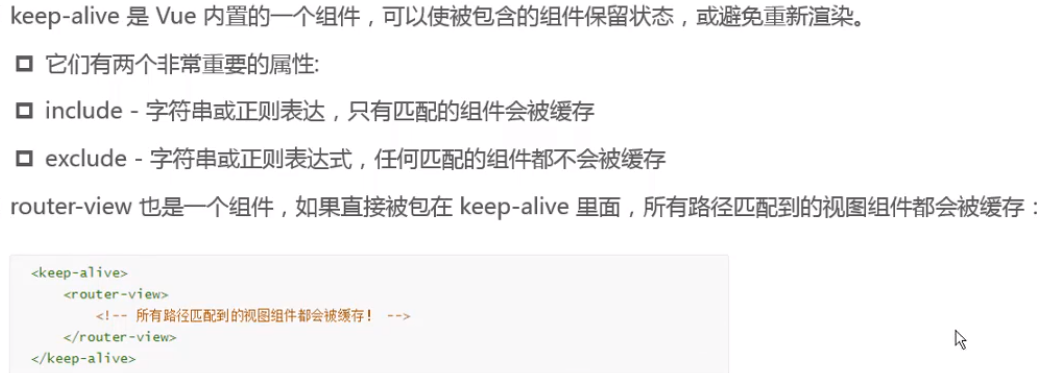


当然这两个函数就是字面上的意思当组件处于活跃时调用activated，否则调用deactivated



如果想要从一堆组件中选出几个不进行keep-alive也有办法，就是在keep-alive标签中添加exclude属性，例如下图，这里的exclude内容直接写的字符串因为组件name就是这个，多个的话用逗号隔开，注意逗号右边不要有空格





5、常用路由方法

this.$route.path可以拿到当前活跃路由的path。常用于this.$route.pah.indexOf来判断当前处于哪个路由。