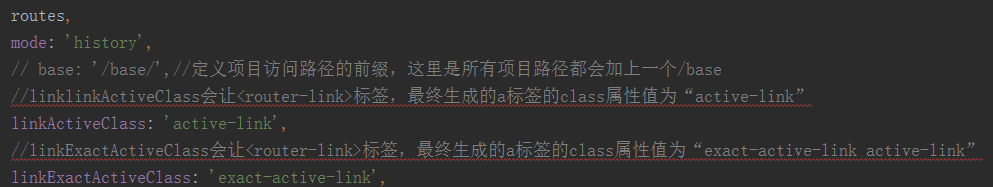
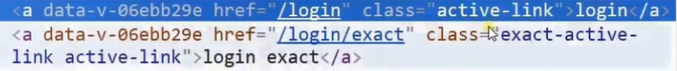
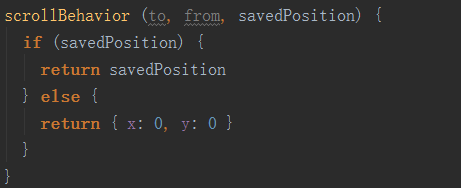
## 路由配置（全局）—配置文件



上面图中的linkActiveClass和linkExactActiveClass配置后，生成的a标签的class属性值如下图所示：



**scrollBehavior函数**

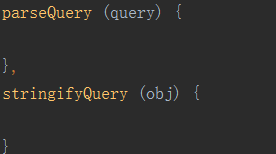


to：目标路径

from：源路径

savePosition：保存过的一个路径

x:0,y:0：表示如果页面没有一个保存的路径，就会把页面滚动到最上面和最左边



**注意：这两个方法也可以在单个路由配置中使用**

上面两个方法是处理请求路径中的查询参数的方法，类似这种：xxx?参数1=值1&参数2=值2。

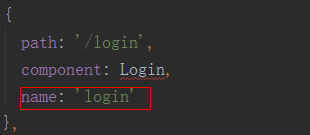
parseQuery方法是将查询参数和它的值组成的字符串转为json对象的方法（vue-router为默认将查询参数转为json对象），其参数就是查询参数和值组成的字符串

stringifyQuery方法则和parseQuery方法想法，是将查询参数和它的值组成的json对象转为字符串，其参数就是查询参数和它的值组成的json对象

## 开发配置（单个路由）

当配置的路由匹配到时，就会去占用<router-view>标签所在的位置，生成对应的页面，所以在对应的位置必须有一个<router-view>标签

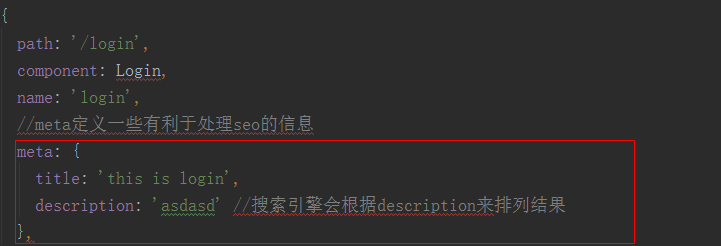
### 给路由命名



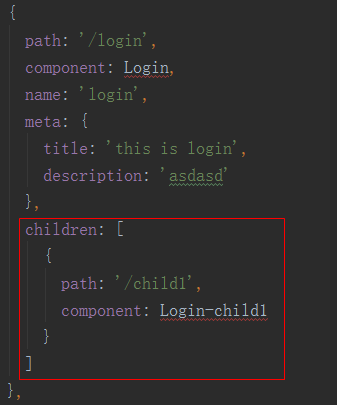


标签中name就对应在陪路由配置中定义的name

### 处理seo相关

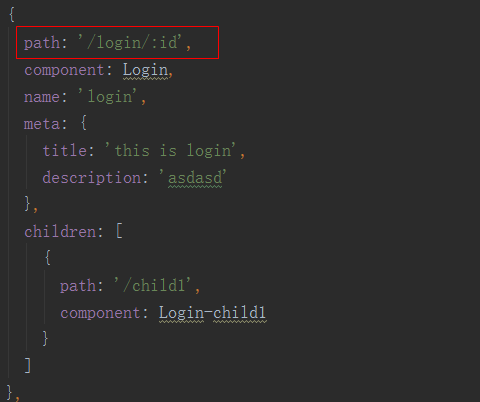


### 配置子路由



那么子路由最终的路径会是这样（以上面图片中例子为例）：xxx/login/child1

### 路由的:xx写法





那么这个路由中，id就会为1，注意，这里的:id不是查询参数。假如有一个需求根据商品的id显示商品的详情页面，这个时候，这种用法就派上用场了。这种方式生成的路由最终会是这种形式（以上面图片中为例）：域名/login/1

在上面图片中路由配置项里新增一项：props：true，那么在路由上的:id这个id就会自动的传入到Login这个组件中的props属性中去，就不需要再通过$route（$route在同一个路由下，包含的内容是一样的，不以在哪个组件中而不同，因为都是在同一个路由下）来获取了。只需要在Login组件中的props属性里面声明一个id，就可以通过这个id获取到路由上id的值了。

props可以有下面这几种写法：

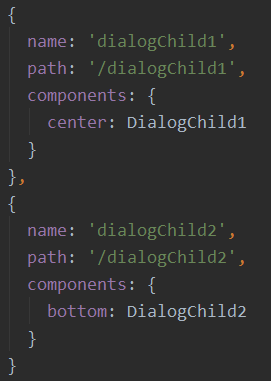
1. 就是上面这种直接props:true
2. props:{id:2}，那么此时id的值就是2了
3. props:(route)=>({id:route.query.cs1})，设置id的值为查询参数—》cs1的值

## 命名视图---<view-router>

<view-router>标签是用于渲染匹配的组件，简单可以理解为一个占位容器，组件到时候就被渲染到此位置。此标签还有一个name属性，用于定位，具体用法看下面一个例子：



路径配置：



dialogChild1组件：



dialogChild2组件：



最终效果是-----》view-router的name属性效果.gif