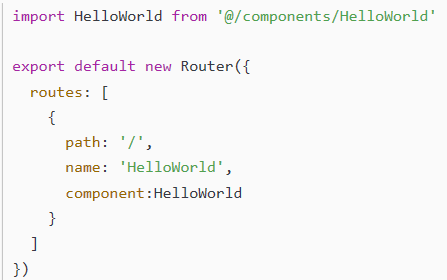
# Uwei 解决vue 首页加载慢的问题

路由懒加载

将引入方式改写

未使用懒加载



使用es6中的import进行懒加载（推荐）

const HW=function(){

return import(“路径”);

}



使用Vue中的异步组件进行懒加载



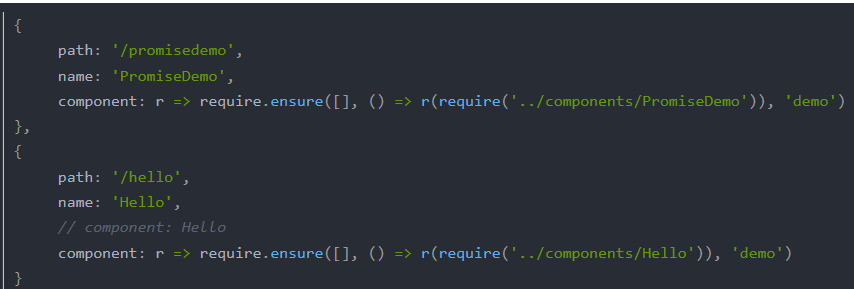
路由懒加载：return import()

组件懒加载：component:function(resolve){ return require([“组件懒加载”],resolve) }

按需加载

#### 1. require.ensure()

webpack提供的require.ensure(),这样可以实现按需加载，并且你可以将多个相同类的组件打包成一个文件，只要给他们指定相同的chunkName即可，如示例中的demo将会打包成一个文件



vue图片懒加载lazy-load

1. 安装插件

npm i vue-lazyload -S

第二步引入

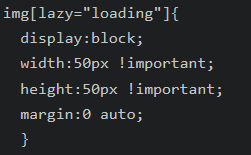
import VueLazyload from 'vue-lazyload'

第三步简单配置参数（两种图片路径方法）

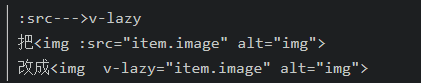


因为src中的文件会被webpack编译，assets文件夹中的图片地址，会在编译过程中重命名。vue-lazyload是在main.js文件中引入，不会被webpack进行编译，因此vue-lazyload无法获得正确的图片地址，所以直接写相对地址就无法获取到图片正确地址

第四步写loading图片的样式（具体怎么写样式根据情况来，也可以不写属性选择器）



第五步使用



# localStorage和sessionStorage的区别:

1.相同的使用方法,API基本是一样的

(1) localStorage API:

a.使用setItem设置储存内容:

localStorage.setItem("name","123")

localStorage.name = "123";

localStorage['name'] = "123"

b.使用getItem获取储存内容:

localStorage.getItem("name")

localStorage.name

c.使用removeItem删除储存内容(删除指定key):

localStorage.removeItem("name")

d.使用clear方法清除储存内容(清除所有key):

localStorage.clear();

e.使用length属性,获取储存的key个数

localStorage.length

(2) sessionStorage API:基本和localStorage差不多

2.不同的储存时效

localStorage 储存时间无限,不同浏览器无法共享localStorage或sessionStorage中的信息。相同浏览器的不同页面间可以共享相同的 localStorage

sessionStorage 会在网页会话结束时失效(刷新不会丢失,在网页关闭即失效)

3.容量不同:

localStorage容量一般 2-5MB

sessionStorage储存容量不一,部分浏览器不设限

# 关于Session和Cookie的区别

1.位置不同：Session是位于服务器上面；Cookie是位于浏览器上面客户端

2.大小不同：Cookie的大小相对而言要小一些Cookie有大小限制以及浏览器在存cookie的个数也有限制，只允许4KB，而session是无限量的

3.安全性：Cookie在浏览器上，容易被修改，安全性相对于Session安全性较低。

4.时间设置:Cookie时间可以使Cookie过期。但是使用Session-destory（），销毁会话

5、Session是保存在服务器端上会存在一段时间才会消失，如果session过多会增加服务器的压力。

# 总结

1、 localstorage和sessionstorage

大小

作用范围、时间有效期

2、session与cookie区别

大小

有效期

位置

安全性

3、解决vue 首页加载慢的问题

路由懒加载

组件懒加载

图片、js、html等资源webapck 优化配置

cdn