



告别 hash 路由，迎接 history 路由

📌 知识总结 📅 2019-03-02 👁 483 💬 4

vue

java

🕒 15:29:45

三月来了，春天还会远吗、、

在这里，隆重宣布本博客告别 Vue 传统的 hash 路由，迎接好看而优雅的 history 路由~~

#映照官方说法

vue-router 默认 hash 模式 —— 使用 URL 的 hash 来模拟一个完整的 URL，于是当 URL 改变时，页面不会重新加载

如果不想要很丑的 hash，我们可以用路由的 history 模式，这种模式充分利用 history.pushState API 来完成 URL 跳转而无须重新加载页面

```
export default new Router({
  mode: 'history',
  routes: [...]
})
```

当使用 history 模式时，URL 就像正常的 url，例如 <https://ainyi.com/about>，好看又优雅！

不过这种模式要玩好，还需要后台配置支持。因为我们的应用是个单页客户端应用，如果后台没有正确的配置，当用户在浏览器直接访问 <https://ainyi.com/about> 就会返回 404，因为后端没有 /about 相应的拦截器，自然 404

所以，要在服务端增加一个覆盖所有情况的候选资源：如果 URL 匹配不到任何静态资源，则应该返回同一个 index.html 页面，这个页面就是 app 依赖的页面

#后端实现

本博客后端使用的是 java 的 ssm 框架，只做后端接口，只为前端提供接口

#刷新报服务端 404

对于诸如 /about 的 url，SpringMVC 的 @RequestMapping() 没有做映射
如果跳到该页面，手动刷新，后端自然是报 404 错误的

考虑到除了接口，其他访问的 history 全都是返回 404 页面

想到两个方法

1. 就是直接在后端做 404 页面的跳转转发，全部转发到 `index.html`，这就解决了 history 路由刷新 404



2. 后端除了接口，`@RequestMapping()` 做所有可能出现路由的匹配，全部转发到 `index.html`，不过这样

代码量多，后续如果增加更多路由，都要在后端做映射，不利于维护，但体验比第一种方法好



本项目目前采用第一种方法（**目前而已啦~**）

就是在 `web.xml` 中进行 404 页面的配置跳转，在 `web-app` 标签中配置 `error-page`

```
<web-app>
...
<!-- 未匹配到 url 的跳转页面，用于 vue history 路由，未匹配的路径自动转发到 index.html -->
<error-page>
  <error-code>404</error-code>
  <location>/index.html</location>
</error-page>
</web-app>
```

完美，

但又有一个问题出现了，点击跳转可以进入二级路由，例如进入了 <https://ainyi.com/tag/vue>
当点击刷新的时候，会报一个找不到静态资源的错误，也就是 js 静态资源没找到 `qaq`

这就要前端来解决这个问题

#刷新找不到静态资源

由于之前是使用 hash 路由，`npm run build` 打包之后，直接打开 `index.html` 会报找不到资源的情况，这我在之前写的博客记录解决方案：<https://ainyi.com/47>

当时是这么做的：

文件路径出错解决：在 `config` 的 `index.js` 下 `build` 的设置：`assetsPublicPath: './'`

— **划重点** —

而如今变成 history 路由，此字段应设置成 `assetsPublicPath: '/'`

原因如下：

```
'./' 是指用户所在的当前目录（相对路径）
'/' 是指根目录，也就是项目的根目录
对于 hash 模式，根路径是固定的，就是项目的根目录
但是 history 模式下，以 / 开头的嵌套路径会被当作根路径
所以当使用 './' 引入文件，就会找不到文件了
因为文件本身就是在项目根目录下的，并不在嵌套 history 的路径目录下
```

设置好之后，重新打包，测试，部署上线，完美解决~

#前端 404 情况

[🏠 首页](#)[📁 存档](#)[🔗 友链](#)[❤️ 关于](#)

后端做了 404 页面自动跳转到 index.html 之后，服务器就不再返回 404 错误页面，因为对于所有路径都会返回 index.html 文件

为了避免这种情况，应该在 vue-router 里面的 **最后面** 设置无法匹配正确路由的情况跳转到前端的 404 页面

记住是 **routes** 数组的最后一个数组元素做 404 匹配

```
export default new Router({
  mode: 'history',
  routes: [
    // TODO, 这里写所有路由，写完所有路由后，最后面写如下
    {
      path: '*',
      component: () => import('@components/NotFound'),
      name: 'NotFound',
      meta: {title: '404 - 找不到页面'},
    }
  ]
})
```

本文由 [Krry](#) 创作，转载请注明

最后编辑时间：2019-04-30 17:32:00



发表评论

昵称

邮箱

网址(http://)

留下你的足迹... (支持 Markdown)

Emoji | Preview

提交评论

厉害



lilye

2019-04-13

完美 ⚡



天地四方

2019-03-03

挺好的 😊



Krry

2019-04-13

@天地四方 🍇 给串葡萄吃，hash 就没了 😂