# 面试官：vue项目本地开发完成后部署到服务器后报404是什么原因呢？

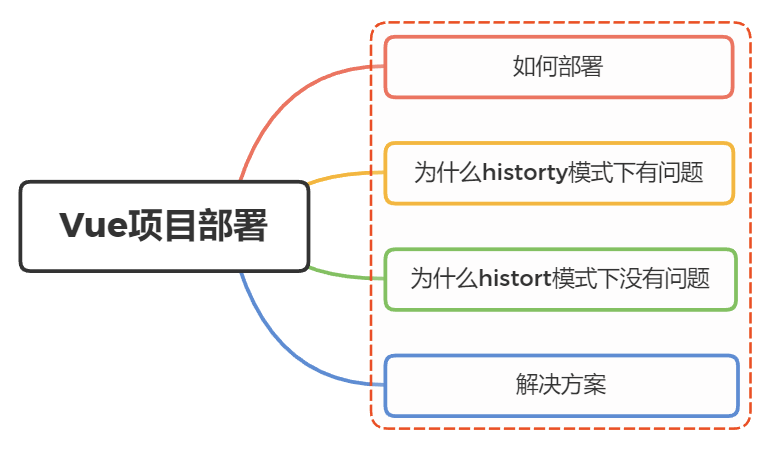


image.png

## 一、如何部署

前后端分离开发模式下，前后端是独立布署的，前端只需要将最后的构建物上传至目标服务器的web容器指定的静态目录下即可

我们知道vue项目在构建后，是生成一系列的静态文件

常规布署我们只需要将这个目录上传至目标服务器即可

// scp 上传 user为主机登录用户，host为主机外网ip, xx为web容器静态资源路径  
scp dist.zip user@host:/xx/xx/xx

让web容器跑起来，以nginx为例

server {  
 listen 80;  
 server\_name www.xxx.com;  
  
 location / {  
 index /data/dist/index.html;  
 }  
}

配置完成记得重启nginx

// 检查配置是否正确  
nginx -t   
  
// 平滑重启  
nginx -s reload

操作完后就可以在浏览器输入域名进行访问了

当然上面只是提到最简单也是最直接的一种布署方式

什么自动化，镜像，容器，流水线布署，本质也是将这套逻辑抽象，隔离，用程序来代替重复性的劳动，本文不展开

## 二、404问题

这是一个经典的问题，相信很多同学都有遇到过，那么你知道其真正的原因吗？

我们先还原一下场景：

* vue项目在本地时运行正常，但部署到服务器中，刷新页面，出现了404错误

先定位一下，HTTP 404 错误意味着链接指向的资源不存在

问题在于为什么不存在？且为什么只有history模式下会出现这个问题？

### 为什么history模式下有问题

Vue是属于单页应用（single-page application）

而SPA是一种网络应用程序或网站的模型，所有用户交互是通过动态重写当前页面，前面我们也看到了，不管我们应用有多少页面，构建物都只会产出一个index.html

现在，我们回头来看一下我们的nginx配置

server {  
 listen 80;  
 server\_name www.xxx.com;  
  
 location / {  
 index /data/dist/index.html;  
 }  
}

可以根据 nginx 配置得出，当我们在地址栏输入 www.xxx.com 时，这时会打开我们 dist 目录下的 index.html 文件，然后我们在跳转路由进入到 www.xxx.com/login

关键在这里，当我们在 website.com/login 页执行刷新操作，nginx location 是没有相关配置的，所以就会出现 404 的情况

### 为什么hash模式下没有问题

router hash 模式我们都知道是用符号#表示的，如 website.com/#/login, hash 的值为 #/login

它的特点在于：hash 虽然出现在 URL 中，但不会被包括在 HTTP 请求中，对服务端完全没有影响，因此改变 hash 不会重新加载页面

hash 模式下，仅 hash 符号之前的内容会被包含在请求中，如 website.com/#/login 只有 website.com 会被包含在请求中 ，因此对于服务端来说，即使没有配置location，也不会返回404错误

## 解决方案

看到这里我相信大部分同学都能想到怎么解决问题了，

产生问题的本质是因为我们的路由是通过JS来执行视图切换的，

当我们进入到子路由时刷新页面，web容器没有相对应的页面此时会出现404

所以我们只需要配置将任意页面都重定向到 index.html，把路由交由前端处理

对nginx配置文件.conf修改，添加try\_files $uri $uri/ /index.html;

server {  
 listen 80;  
 server\_name www.xxx.com;  
  
 location / {  
 index /data/dist/index.html;  
 try\_files $uri $uri/ /index.html;  
 }  
}

修改完配置文件后记得配置的更新

nginx -s reload

这么做以后，你的服务器就不再返回 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，你应该在 Vue 应用里面覆盖所有的路由情况，然后在给出一个 404 页面

const router = new VueRouter({  
 mode: 'history',  
 routes: [  
 { path: '\*', component: NotFoundComponent }  
 ]  
})

关于后端配置方案还有：Apache、nodejs等，思想是一致的，这里就不展开述说了

## 参考文献

* https://juejin.cn/post/6844903872637632525
* https://vue-js.com/topic/5f8cf91d96b2cb0032c385c0