Vue Router 手把手教你搭 Vue3 路由 - 卡拉云

作者：HiJiangChuan
原文链接：https://ld246.com/article/1645829901856
来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="Vue Router 手把手教你搭 Vu
3 路由" data-src="https://b3logfile.com/file/2022/02/89ff0e71729c45ec81806ba751f56ac0.jp
g?imageView2/2/interlace/1/format/jpg">
本文首发：《<a href="https://ld246.com/forward?goto=https%3A%2F%2Fkalacloud.co
%2Fblog%2Fvue-router-tutorial-for-vue-3%2F" target="_blank" rel="nofollow ugc">Vue Rou
er 手把手教你搭 Vue3 路由 - 卡拉云》
Vue 的单页面应用是基于路由 + 组件的形式，路由用于设定访问路径，并将路径与组件映射起
。这种形式相对于 a 标签超链来说不会重新加载页面，而是在同一个页面中进行路由跳转。
本教程在 Vue3 中手把手教你搭建 Router ，并详细讲解其中的路由原理以及当用户输入错误时
如何路由到 404 页面等方法。请打开你的 Terminal 跟随本教程一起学习。
如果你正在搭建后台管理工具，又不想处理前端问题，推荐使用卡拉云，卡拉云是新一代低代码
发工具，可一键接入常见数据库及 API ，无需懂前端，仅需拖拽即可快速搭建属于你自己的后台管理
具，一周工作量缩减至一天，详见本文文末。
<h2 id="从安装-Vue3-开始">从安装 Vue3 开始</h2>
我们先来安装 Vue3 ，一步一步来：
<pre><code class="highlight-chroma"><span class="highlight
cl">npm install -g @vue/cli
</code></pre>
然后我们将创建我们的基本 Vue 3 应用程序。
<pre><code class="highlight-chroma"><span class="highlight
cl">vue create kalacloud-vue-router
</code></pre>
<img src="https://ld246.com/images/img-loading.svg" alt="vue3-setup" data-src="https
//b3logfile.com/file/2022/02/f296e19c1d3b49ae86108dabdaa93bd9.png?imageView2/2/inter
ace/1/format/jpg">
选择 Vue 3 ，然后会自动帮你创建基于 Vue 3 的项目。
完成后，我们 <code>cd</code> 到项目目录，接下来我们本教程所有操作都在
个目录下完成。
<pre><code class="highlight-chroma"><span class="highlight
cl">cd kalacloud-vue-router
</code></pre>
我们先运行一下看看效果：
<pre><code class="highlight-chroma"><span class="highlight
cl">npm run serve
</code></pre>
在浏览器输入 <code>http://localhost:8080</code> ，可以看到 Vue3 已经运
起来了。
扩展阅读：《<a href="https://ld246.com/forward?goto=https%3A%2F%2Fkalacloud.co
%2Fblog%2Fbest-drag-drop%2F" target="_blank" rel="nofollow ugc">最好用的 6 款 Vue 拖
组件库推荐》
<h2 id="安装-Vue-Router-路由">安装 Vue Router 路由</h2>
我们先 Vue Router 安装到刚刚创建的 Vue3 项目中。
执行以下代码安装 Vue Router：
<pre><code class="highlight-chroma"><span class="highlight
cl">npm i vue-router@4
</code></pre>
<h2 id="配置并建立-Vue-Router-路由">配置并建立 Vue Router 路由</h2>
接着我们配置 Router ，添加配置文件。在 <code>/src</code> 目录下新建&nb
p;<code>router</code> 文件夹，然后新建 Router 配置文件 <code>index.js</co
e>
文件位置：/src/router/index.js
<pre><code class="highlight-chroma"><span class="highlight
cl">import { createWebHistory, createRouter } from "vue-router";
import Home fro

```
 "@/views/Home.vue";
</span></span><span class="highlight-line"><span class="highlight-cl">import About fro
 "@/views/About.vue";
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">const routes = [
</span></span><span class="highlight-line"><span class="highlight-cl">  {
</span></span><span class="highlight-line"><span class="highlight-cl">    path: "/",
</span></span><span class="highlight-line"><span class="highlight-cl">    name: "Home",
</span></span><span class="highlight-line"><span class="highlight-cl">    component: H
me,
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">  {
</span></span><span class="highlight-line"><span class="highlight-cl">    path: "/about",
</span></span><span class="highlight-line"><span class="highlight-cl">    name: "About",
</span></span><span class="highlight-line"><span class="highlight-cl">    component: Ab
ut,
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">];
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">const router = cre
teRouter({
</span></span><span class="highlight-line"><span class="highlight-cl">  history: createW
bHistory(),
</span></span><span class="highlight-line"><span class="highlight-cl">  routes,
</span></span><span class="highlight-line"><span class="highlight-cl">});
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">export default rou
er;
</span></span></code></pre>
<p>在 Router 配置文件中，我们使用一个数组来写 Router 指向的每一个分页面。</p>
<ul>
<li><code>path</code>&amp;nbsp;路由分配的 URL</li>
<li><code>name</code>&amp;nbsp;当路由指向此页面时显示的名字</li>
<li><code>component</code>&amp;nbsp;路由调用这个页面时加载的组件名</li>
</ul>
<p>这段代码的最后是 Router 的主体。我们创建了一个路由，使用了&amp;nbsp;<code>history</cod
>&amp;nbsp;模式。</p>
<p><code>history</code>&amp;nbsp;模式是&amp;nbsp;<code>hash</code>&amp;nbsp;模式的升级版，
要区别在浏览器链接的显示的不同</p>
<ul>
<li>hash 模式：把前端路由路径用 # 号拼接在真实 URL 后面的模式。当 # 后面的路径发生变化时
浏览器不会重新发起请求，而是出发 hashchange 事件。hash 模式链接样式：<code>http://localh
st:8080/#/home</code></li>
<li>history 模式：history API 是 HTML5 的新特性，允许开发者直接更改前端路由。history 模式
接样式：<code>http://localhost:8080/home</code></li>
</ul>
<p>扩展阅读：《<a href="https://ld246.com/forward?goto=https%3A%2F%2Fkalacloud.co
%2Fblog%2Fvue-draggable-tutorial%2F" target="_blank" rel="nofollow ugc">vue.draggable
入门指南 - 手把手教你开发任务看板</a>》</p>
<h2 id="在-main-js-引入-Router">在 main.js 引入 Router</h2>
<p>接下来，我们要在 main.js 文件里引入我们刚刚设置的路由</p>
<p>文件位置：/src/main.js</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">import { createApp } from 'vue'
```

import App from '
/App.vue'
import router fro
'./router'

createApp(App).u
e(router).mount('#app')
</code></pre>
（特别提示：请使用以上代码全部替换 main.js 内代码，避免与本教程代码不一致）
扩展阅读：《<a href="https://ld246.com/forward?goto=https%3A%2F%2Fkalacloud.co
%2Fblog%2Fbest-vue-date-timepicker-components%2F" target="_blank" rel="nofollow ugc"
最好用的 12 款 Vue Timepicker 时间日期选择器测评推荐》
<h2 id="在-App-vue-中使用和">在 App.vue 中使用 <code><router-view></code> 和 <c
de><router-link></code></h2>
App.vue 是项目的主组件，可以理解为项目的入口页面，所有页面都在 App.vue 页之下进行切
。接下来我们要来修改 App.vue ，以适应我们添加的 Router 功能。
文件位置：/src/App.vue
<pre><code class="highlight-chroma"><span class="highlight
cl"><template>
  <div id="nav"
gt;
    <router-link
o="/"> 首页 </router-link> |
    <router-link
o="/about"> 关于 </router-link>
  </div>
  <router-view
>
</template>
</code></pre>
（特别提示：请使用以上代码全部替换 App.vue 内代码，避免与本教程代码不一致）

<code><router-view /></code>： 这个 <code>/</code> 是承接自路
的容器，所有一级路由都在 <code>/</code> 之后。比如前文我们写的两个页面，/H
me 和 /About
<code><router-link></code>：在 history 模式下会拦截点击，不让浏览器重新加载页
。

扩展阅读：《<a href="https://ld246.com/forward?goto=https%3A%2F%2Fkalacloud.co
%2Fblog%2Fdelivery-tracking-case-study%2F" target="_blank" rel="nofollow ugc">订单管理
统(OMS)搭建实战 - 低代码拖拽定制订单管理系统》
<h2 id="创建页面组件">创建页面组件</h2>
接着，我们来创建「首页」和「关于」两个页面组件，这也是本教程中使用路由进行分配的两个
件。我们不把这两个组件放在 components 文件夹中，我们来增加一些难度，新建一个 views 文件
，然后在这里创建 Home.vue 和 About.vue
文件位置：/src/views/Home.vue
<pre><code class="highlight-chroma"><span class="highlight
cl"><template>
  <h1>卡拉
是什么？</h1>
  <a>卡拉云 -
低代码开发工具。
  <a>无需懂
端，仅需拖拽即可快速搭建属于你的后台管理工具，<a href="http://kalacloud.com" t

rget="_blank">立即试用卡拉云
</template>
</code></pre>
文件位置：/src/views/About.vue
<pre><code class="highlight-chroma"><span class="highlight
cl"><template>
  <h1>关于
拉云</h1>
<a>卡拉云 -
低代码开发工具。
  <a>可接入
见的数据库及 API，快速搭建属于你的后台管理工具，一周工作量缩短至数小时，<a hre
="http://kalacloud.com" target="_blank">立即试用卡拉云
</template>
</code></pre>
好，现在我们把项目运行起来看看效果。
<pre><code class="highlight-chroma"><span class="highlight
cl">npm run serve
</code></pre>
<img src="https://ld246.com/images/img-loading.svg" alt="router" data-src="https://b3
ogfile.com/file/2022/02/995450649fd348058d2237d256277c60.gif">
在浏览器输入 <code>http://localhost:8080</code> 页面正常访问并且无刷新
转页面。
扩展阅读：《<a href="https://ld246.com/forward?goto=https%3A%2F%2Fkalacloud.co
%2Fblog%2Fbest-vue-tree-view%2F" target="_blank" rel="nofollow ugc">最好用的 7 个 Vue T
ee select 树形组件》
<h2 id="创建-404-页面">创建 404 页面</h2>
404 页面比较特殊，它不是用户输入的 URL 所指页面，而是当用户输入的 URL 在路由里完全没
匹配的页面时，才会路由到 404 页面。
我们修改 Router 配置文件，当没有找到匹配的页面时，路由指向 404 页。
在 index.js 插入以下代码，文件位置：/src/router/index.js
<pre><code class="highlight-chroma"><span class="highlight
cl">{
  path: "/:catchAll(.
)",
  component: Not
ound,
},
</code></pre>

<code>(.*)</code> 是一个正则表达式，如果用户输入的 URL 没有跟任何 Router 中的
nbsp;<code>path</code> 匹配，那么就会与 <code>(.*)</code> 匹配，Router
会把用户带到 404 页。

index.js 完整代码如下，请全部替换 index.js 内代码，避免与本教程代码不一致。
<pre><code class="highlight-chroma"><span class="highlight
cl">import { createWebHistory, createRouter } from "vue-router";
import Home fro
 "@/views/Home.vue";
import About fro
 "@/views/About.vue";
import NotFound
rom "@/views/NotFound.vue";

```
</span></span><span class="highlight-line"><span class="highlight-cl">const routes = [
</span></span><span class="highlight-line"><span class="highlight-cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  path: "/:catchAll(.
)",
</span></span><span class="highlight-line"><span class="highlight-cl">  component: Not
ound,
</span></span><span class="highlight-line"><span class="highlight-cl">},
</span></span><span class="highlight-line"><span class="highlight-cl">  {
</span></span><span class="highlight-line"><span class="highlight-cl">    path: "/",
</span></span><span class="highlight-line"><span class="highlight-cl">    name: "Home",
</span></span><span class="highlight-line"><span class="highlight-cl">    component: H
me,
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">  {
</span></span><span class="highlight-line"><span class="highlight-cl">    path: "/about",
</span></span><span class="highlight-line"><span class="highlight-cl">    name: "About",
</span></span><span class="highlight-line"><span class="highlight-cl">    component: Ab
ut,
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">];
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">const router = cre
teRouter({
</span></span><span class="highlight-line"><span class="highlight-cl">  history: createW
bHistory(),
</span></span><span class="highlight-line"><span class="highlight-cl">  routes,
</span></span><span class="highlight-line"><span class="highlight-cl">});
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">export default rou
er;
</span></span></code></pre>
<p>建立好 404 的路由后，我们再来创建 404 错误页面，在 views 文件夹中创建 NotFound.vue</
>
<p>文件位置：/src/views/NotFound.vue</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">&lt;template&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">  &lt;h1&gt; 啊，4
4 了，没有找到你想要的页面。&lt;/h1&gt;
</span></span><span class="highlight-line"><span class="highlight-cl">&lt;/template&gt;
</span></span></code></pre>
<p>好，接着我们运行项目，看看效果：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">npm run serve
</span></span></code></pre>
<p><img src="https://ld246.com/images/img-loading.svg" alt="404" data-src="https://b3lo
file.com/file/2022/02/254b29540b92401ab04f848c78d33eda.jpeg?imageView2/2/interlace/1/
ormat/jpg"></p>
<p>在浏览器中随便输入一个 URL，类似&amp;nbsp;<code>http://localhost:8080/asdfasdfasdf</co
e>&amp;nbsp;浏览器指向我们刚刚写的 404 页面。</p>
<p>扩展阅读：《<a href="https://ld246.com/forward?goto=https%3A%2F%2Fkalacloud.co
%2Fblog%2Fvue-upload-image-axios%2F" target="_blank" rel="nofollow ugc">Vue 搭建带
览的「上传图片」管理后台</a>》</p>
<h2 id="总结">总结</h2>
<p>本文手把手带领大家搭建了一套简单的 Router 单页面路由。如果你学习前端仅仅是为了接上自
```

的数据库，快速搭一套根据自己工作流设计的后台管理系统的话，推荐使用卡拉云，卡拉云是一套低码开发工具，完全不用懂前端，鼠标拖拽，简单直观。

上图为使用卡拉云搭建的销售 SaaS Demo，卡拉云内置标签容器，让你轻松在多个页面中切换无需任何前端技术，直接拖拽组件即可实现。
卡拉云是新一代低代码开发工具，免安装部署，可一键接入包括 MySQL 在内的常见数据库及 AP。可根据自己的工作流，定制开发。无需繁琐的前端开发，只需要简单拖拽，即可快速搭建企业内部具。原来三天的开发工作量，使用卡拉云后可缩减至 1 小时，欢迎免费试用卡拉云。
扩展阅读：

<a href="https://ld246.com/forward?goto=https%3A%2F%2Fkalacloud.com%2Fblog%2Fest-translation-api%2F" target="_blank" rel="nofollow ugc">最好的 5 款翻译 API 接口对比测评/a>
<a href="https://ld246.com/forward?goto=https%3A%2F%2Fkalacloud.com%2Fblog%2Fpi-testing-tools%2F" target="_blank" rel="nofollow ugc">最好用的七大顶级 API 接口测试工具<a>
最好用的 5 款 React 富文本编辑器
Postman 使用教程 - 手把手教你 API 接口测试
最好的 6 个免费天气 API 接口对比测
<a href="https://ld246.com/forward?goto=https%3A%2F%2Fkalacloud.com%2Fblog%2Faw-tutorial%2F" target="_blank" rel="nofollow ugc">PAW 使用教程 - 手把手教你 API 接口测试/a>
