1、检查 Vue Router 安装

没有安装请在根目录下面执行命令：npm install vue-router

1. 在components目录下，新建自定义组件：Bar.vue和Foo.vue，以及Hello.vue。内容如下：

（1）// src/components/Bar.vue

<template>

<div>{{msg}}</div>

</template>

<script>

export default {

name: 'bar',

data () {

return {

msg: 'bar page'

}

}

}

</script>

1. // src/components/Foo.vue

<template>

<div>{{msg}}</div>

</template>

<script>

export default {

name: 'foo',

data () {

return {

msg: 'foo page'

}

}

}

</script>

1. // src/components/Hello.vue

<template>

Hello，my First Router Test~

</template>

<script>

</script>

<style>

</style>

3、配置 Vue Router

在你的 src 目录下创建一个 router 文件夹，并在其中创建一个 index.js 文件。内容大致如下：

javascript

// src/router/index.js

import { createRouter, createWebHistory } from 'vue-router';

import Home from '../components/Home.vue'; // 示例组件路径，请根据实际情况调整

import Bar from '../components/Bar.vue'; // 示例组件路径，请根据实际情况调整

import Foo from '../components/Foo.vue'; // 示例组件路径，请根据实际情况调整

const routes = [

{

path: '/',

name: 'Home',

component: Home

},

{

path: '/bar',

name: 'Bar',

component: Bar

},

{

path: '/foo',

name: 'Foo',

component: Foo

}

// 其他路由配置

];

const router = createRouter({

history: createWebHistory(process.env.BASE\_URL),

routes

});

export default router;

4、在主入口文件中引入并使用 Vue Router

打开 src/main.js 文件（或者 src/main.ts 如果你使用的是 TypeScript），并确保你已经引入了刚刚配置好的 router：

javascript

// src/main.js

import { createApp } from 'vue';

import App from './App.vue';

import router from './router';

const app = createApp(App);

app.use(router); // 确保已经使用 router

app.mount('#app');

5、确保在模板中使用 <router-view>

打开你的根组件 App.vue，确保在模板中包含 <router-view> 标签，以便 Vue Router 可以正确地渲染匹配到的组件：

<!-- src/App.vue -->

<template>

<div id="app">

<router-link to="/bar">Bar</router-link>|

<router-link to="/foo">Foo</router-link>

<router-view></router-view> <!-- 确保包含这个标签 -->

</div>

</template>

<script>

export default {

name: 'App',

components: {

}

}

</script>

<style>

/\* 样式部分 \*/

</style>

检查 Vue 版本

确保你使用的是 Vue 3 和相应的 Vue Router 版本。如果你使用的是 Vue 2，相关的配置可能会有所不同。

自定义组件排除（如果使用自定义组建）

如果你确实有自定义组建，并且想让 Vue 忽略它们的格式标准化，那么你可以通过 lintOnSave来进行设置。例如，在 vue.config.js 文件中：

javascript

// vue.config.js

const { defineConfig } = require('@vue/cli-service')

module.exports = defineConfig({

transpileDependencies: true,

lintOnSave:false

})

通过以上步骤，基本可以解决 error Component name "bar" should always be multi-word 问题。如果问题仍然存在，请检查项目中的其他配置或错误日志，进一步排查问题。