



Vue-Router 设计思路分析

VueRouter 是 Vue 官方的路由管理器。它和 Vue 深度集成，让我们构建单页面应用变得易如反掌。

分析

以上是 `VueRouter` 文档上的介绍，我们想要实现自己的 `mini-router`，先来看看它都完成了什么任务：

- 将我们的组件映射到路由上
- 让 VueRouter 知道在哪里渲染它们

体验

所以我们的代码会像下面这样写：

- `router-link` 负责路由跳转
- `router-view` 告诉知道在哪渲染组件

```
<!-- App.vue -->
<p>
  <!--使用 router-link 组件进行导航 -->
  <!--`<router-link>` 将呈现一个带有正确 `href` 属性的 `<a>` 标签-->
  <router-link to="/">Go to Home</router-link> |
  <router-link to="/about">Go to About</router-link>
</p>
<!-- 路由出口 -->
<!-- 路由匹配到的组件将渲染在这里 -->
<router-view></router-view>
```

Vue

还有个问题是上面的 url 如何映射到组件，于是有了创建路由时的映射表：

```
// router/index.js
// 1. 引入组件
import Home from "../views/home.vue";
import About from "../views/about.vue";

// 2. 定义一些路由：每个路由都需要映射到一个组件。
const routes = [
  { path: "/", component: Home },
  { path: "/about", component: About },
];

// 3. 创建路由实例并传递 `routes` 配置
const router = createRouter({
  // 4. 内部提供了 history 模式的实现。为了简单起见，我们在这里使用 hash 模式。
  history: createWebHashHistory(),
  routes, // `routes: routes` 的缩写
});

export default router;
```

Vue

最后是如何引入到 vue 中：

```
// main.js
import router from './router'

createApp(App).use(router).mount('#app')
```

Vue

效果如下：

mini-router

[Go to Home](#) | [Go to About](#)

home page

思路

经过上面分析，我们大致有了思路：

- 定义一个 Vue 插件作为载体
- 实现两个组件 `router_view` 和 `router_link`