

路由

<router-link>

<!-- 字符串 -->

```
<router-link to="home">Home</router-link>
```

<!-- 渲染结果 -->

```
<a href="home">Home</a>
```

<!-- 使用 v-bind 的 JS 表达式 -->

```
<router-link v-bind:to="'home'">Home</router-link>
```

<!-- 不写 v-bind 也可以，就像绑定别的属性一样 -->

```
<router-link :to="'home'">Home</router-link>
```

<!-- 同上 -->

```
<router-link :to="{ path: 'home' }">Home</router-link>
```

<!-- 命名的路由 -->

```
<router-link :to="{ name: 'user', params: { userId: 123 } }">User</router-link>
```

<!-- 带查询参数，下面的结果为 /register?plan=private -->

```
<router-link :to="{ path: 'register', query: { plan: 'private' } }">Register</router-link>
```

路由信息对象

1. \$route.path

字符串，对应当前路由的路径，总是解析为绝对路径，如 `"/foo/bar"`。

2. \$route.params

一个 key/value 对象，包含了 动态片段 和 全匹配片段，如果没有路由参数，就是一个空对象。

3. \$route.query

一个 key/value 对象，表示 URL 查询参数。例如，对于路径 `/foo?user=1`，则有 `$route.query.user == 1`，如果没有查询参数，则是个空对象。

4. \$route.hash

当前路由的 hash 值（不带 #），如果没有 hash 值，则为空字符串。

5. \$route.fullPath

完成解析后的 URL，包含查询参数和 hash 的完整路径。

6. \$route.matched

一个数组，包含当前路由的所有嵌套路径片段的 路由记录 。路由记录就是 routes 配置数组中的对象副本（还有在 children 数组）。

综合上述，一个包含重定向、嵌套路由、懒加载的main.js如下：

懒加载

```
const routes = [
  { path: '/index', component: resolve => require(['./index.vue'], resolve) },
  { path: '/hello', component: resolve => require(['./hello.vue'], resolve) },
]
```

eg:

```
import Vue from 'vue'
import VueRouter from 'vue-router'
import App from './App'
Vue.use(VueRouter)
const router = new VueRouter({
  routes:[
    { path: '/', redirect: '/index' },
    { path: '/index', component: resolve => require(['./components/index.vue'],
resolve),
      children:[
        { path: 'info', component: resolve => require(['./components/info.vue'],
resolve) }
      ]
    },
    { path: '/hello', component: resolve => require(['./components/hello.vue'],
resolve) },
  ]
})
new Vue({
  router,
  render: h => h(App)
}).$mount('#app')
```