

VUE-ROUTER

- Vue Router 是 <u>Vue.is</u> 官方的路由管理器。
- · 它和 Vue.js 的核心深度集成,让构建单页面应用变得易如反掌。
- 使用 Vue.js , 我们已经可以通过组合组件来组成应用程序, 当你要把 Vue Router 添加进来, 我们需要做的是, 将组件 (components) 映射到路由 (routes), 然后告诉 Vue Router 在哪里渲染它们。

安装

npm install vue-router

```
import Vue from 'vue';
import VueRouter from 'vue-router';
Vue.use(VueRouter);
```

导航用法

- 使用 router-link 组件来导航
 - <router-link to="/foo">Go to Foo</router-link>
 - 通过传入 `to` 属性指定链接.
 - <router-link> 默认会被渲染成一个 `<a>` 标签
- this.\$router.push("/foo")
- 回退方法
 - this.\$router.go(n)

动态路由匹配

- 我们经常需要把某种模式匹配到的所有路由,全都映射到同个组件。

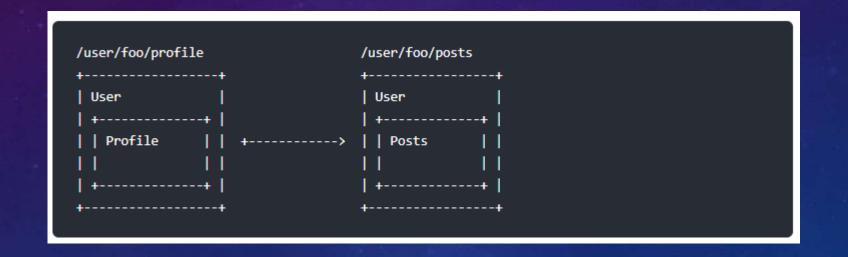
路径参数

• "路径参数"使用冒号:标记。当匹配到一个路由时,参数值会被设置到 this.\$route.params,可以 在每个组件内使用。

模式	匹配路径	\$route.params
/user/:username	/user/evan	{ username: 'evan' }
/user/:username/post/:post_id	/user/evan/post/123	{ username: 'evan', post_id: 123 }

嵌套路由

• 实际生活中的应用界面,通常由多层嵌套的组件组合而成。同样地,URL 中各段动态路径也按某种结构对应嵌套的各层组件,例如:



嵌套路由

```
const router = new VueRouter({
 routes: [
   { path: '/user/:id', component: User,
     children: [
         // 当 /user/:id/profile 匹配成功,
        // UserProfile 会被渲染在 User 的 <router-view> 中
         path: 'profile',
         component: UserProfile
         // UserPosts 会被渲染在 User 的 <router-view> 中
         path: 'posts',
         component: UserPosts
```

编程式的导航

- 除了使用 < router-link > 创建 a 标签来定义导航链接,我们还可以借助 router 的实例方法,通过 编写代码来实现。
 - router.push(location, onComplete?, onAbort?)
 - router.replace(location, onComplete?, onAbort?)
 - router.go(n)

命名路由

- 有时候,通过一个名称来标识一个路由显得更方便一些,特别是在链接一个路由,或者是执行— 些跳转的时候。你可以在创建 Router 实例的时候,在 routes 配置中给某个路由设置名称。
 - router.push({ name: 'user', params: { userId: 123 }})

命名视图

• 有时候想同时(同级)展示多个视图,而不是嵌套展示。

```
html
crouter-view class="view one">
crouter-view class="view two" name="a">
crouter-view class="view three" name="b">
crouter-view>
```

重定向

别名

路由组件传参

- 布尔模式
 - 如果 props 被设置为 true, route.params 将会被设置为组件属性。

```
const User = {
  props: ['id'],
  template: '<div>User {{ id }}</div>'
}

const router = new VueRouter({
  routes: [
    { path: '/user/:id', component: User, props: true },

    // 对于包含命名视图的路由,你必须分别为每个命名视图添加 `props` 选项:
    {
      path: '/user/:id',
      components: { default: User, sidebar: Sidebar },
      props: { default: true, sidebar: false }
    }
    ]
})
```

路由组件传参

- 对象模式
 - 如果 props 是一个对象,它会被按原样设置为组件属性。当 props 是静态的时候有用。

路由组件传参

• 函数模式

动态传参

```
// 字符串
router.push('home')

// 对象
router.push({ path: 'home' })

// 命名的路由
router.push({ name: 'user', params: { userId: 123 }})

// 带查询参数, 变成 /register?plan=private
router.push({ path: 'register', query: { plan: 'private' }})
```