

Vue Router (路由)

主讲人：和凌志

主要内容

- Vue Router 介绍
- Vue Router 应用
- Vue Router 动手实验



Vue Router

Vue Router 官网

- <https://router.vuejs.org>

简单路由应用

- 页面布局中，添加路由出口，路由到的页面显示到该区域。
- 通过使用 `<router-view>` 标签定义路由出口。
- 通过使用 `<router-link>` 标签，添加导航。`<router-link>` 默认在页面中会渲染为一个 `<a>` 标签组件，可通过 `tag` 属性 控制渲染成何中组件

Vue Router 介绍

- 对于大多数单页面应用，都推荐使用官方支持的 vue-router 库。更多细节可以移步 vue-router 文档。
- 使用 Vue 创建单页面应用，需要依赖 vue-router 来实现页面间的路由跳转。当使用 vue-router 时，需要配置组件和路由映射，vue-router 会根据当前的路由地址，显示相应的组件。

程式化导航 vue-router

- 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页面的跳转。
- `router.push(location)`
- 想要导航到不同的 URL，可以使用 `router.push` 方法。这个方法会向 history 栈添加一个新的记录，所以，当用户点击浏览器后退按钮时，则返回到之前的 URL。该方法的参数可以是一个字符串路径，或者一个描述地址的对象。

router.push ()

// 字符串

```
router.push('home')
```

// 对象

```
router.push({ path: 'home' })
```

// 带参数的路由

```
router.push({ name: 'user', params: { userId: 123 } })
```

// 带查询的路由

```
router.push({ path: 'login', query: { userName: '123', pwd: '1234' } })
```


router.push ()

// 字符串

```
router.push('home')
```

// 对象

```
router.push({ path: 'home' })
```

// 带参数的路由

```
router.push({ name: 'user', params: { userId: 123 } })
```

// 带查询的路由

```
router.push({ path: 'login', query: { userName: '123', pwd: '1234' } })
```

router.replace(location)

- 跟 router.push 很像，唯一的不同就是，它不会向 history 添加新记录，而是跟它的方法名一样，替换掉当前 history 记录。
- 声明式： `<router-link :to= '...' replace>`
- 编程式： `router.replace(...)`

router.go(n)

- 这个方法的参数是一个整数，意思是在 history 记录中向前或者向后退多少步，类似 window.history.go(n) 如下：

```
// 在浏览器记录中前进一步， 等同于 history.forward()  
router.go(1)  
  
// 后退一步记录， 等同于 history.back()  
router.go(-1)  
  
// 前进 3 步记录  
router.go(3)  
  
// 返回当前 history 记录的上一页  
router.back()
```

router.go(n)

// 在浏览器记录中前进一步， 等同于 history.forward()

- router.go(1)

// 后退一步记录， 等同于 history.back()

- router.go(-1)

// 前进 3 步记录

- router.go(3)

// 返回当前 history 记录的上一页

- router.back()

嵌套路由

- 实际的应用界面，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组组件，借助 `vue-router`，使用嵌套路由配置，就可以很简单地表达这种关系。
- 在 App.vue 中，`<router-view>` 是最顶层的路由出口，渲染最高级路由匹配到的组件。同样地，一个被渲染组件同样包含自己的嵌套 `<router-view>`
- 要在嵌套的出口中渲染组件，需要在 VueRouter 的参数中使用 `children` 配置

路由注意事项

- 要注意，以 / 开头的嵌套路径会被当作根路径。这让你充分地使用嵌套组件而无须设置嵌套的路径。



动手实验

路由的应用
