



Vue-Router

路由系统



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



目录

Contents

- ◆ Vue路由简介和基础使用
- ◆ 声明式导航
- ◆ 重定向和模式
- ◆ 编程式导航
- ◆ 嵌套和守卫

学习目标

Learning Objectives

1. 能够说出单页面概念和优缺点
2. 能够掌握vue-router路由系统
3. 能够掌握声明式导航和程式化导航
4. 能够掌握路由嵌套和守卫



目录

Contents

- ◆ Vue路由简介和基础使用
- ◆ 声明式导航
- ◆ 重定向和模式
- ◆ 编程式导航
- ◆ 嵌套和守卫



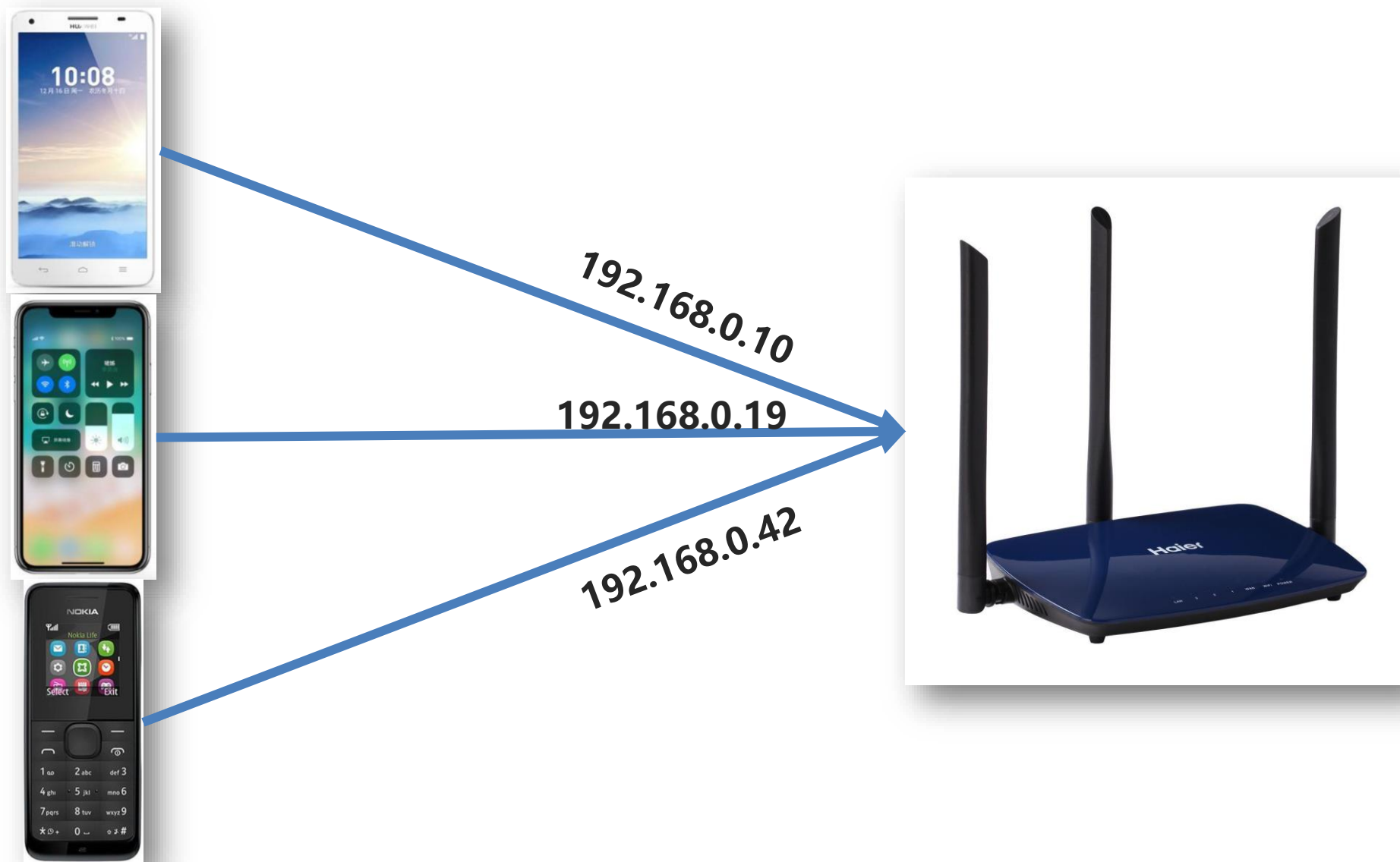
Vue路由简介和基础使用



问题: 什么是路由呢?

1.0 生活中的路由

目标：设备和ip的**映射**关系



1.0 nodejs路由

目标：接口和服务的映射关系

请求服务: 添加书籍

请求服务: 获取书籍

请求服务: 删除书籍

```
router.post("/addbook", (req, res) => { ...  
}))
```

```
router.get("/getbooks", (req, res) => { ...  
}))
```

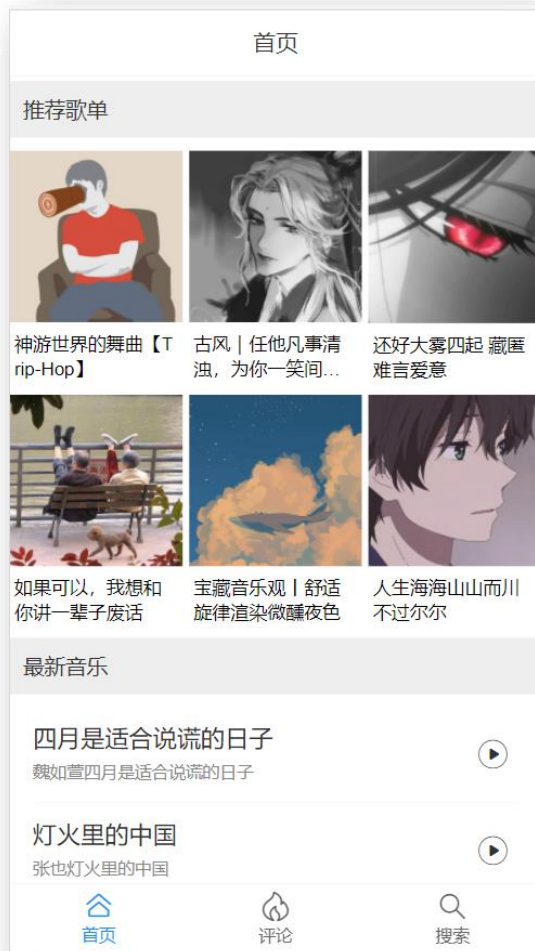
```
router.delete("/deletebook/:id", (req, res) => { ...  
}))
```

```
router.delete("/deletebook/:id", (req, res) => { ...  
}))
```

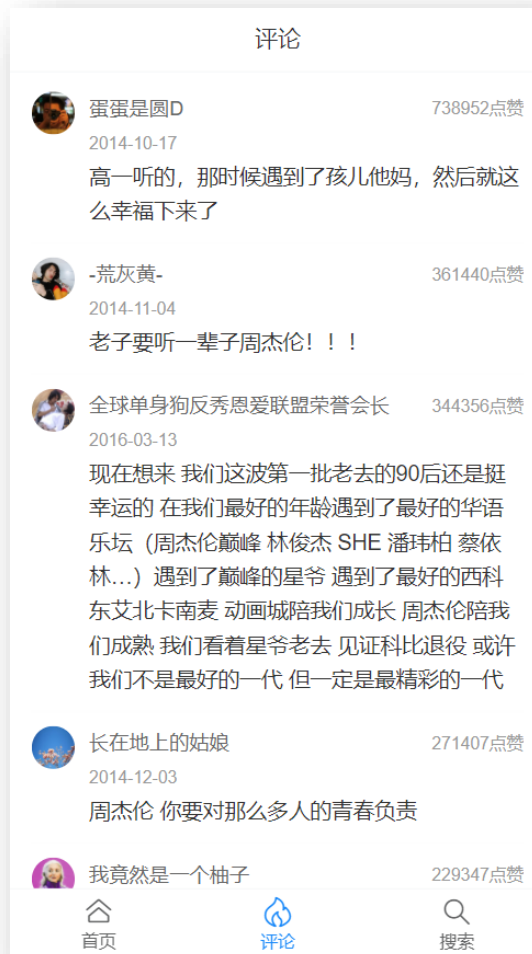

1.0 前端路由

目标：路径和组件的映射关系

http://localhost:8080/#/home



http://localhost:8080/#/comment



http://localhost:8080/#/search





1. 路由是什么呢?

路由是一种映射关系

2. Vue中的路由是什么?

路径和组件的映射关系



问题1: 为什么要使用路由呢?

问题2: 一个网页可以局部刷新吗?

1.1 为何使用路由

目标：了解为何使用路由

- 具体使用示例: 网易云音乐 <https://music.163.com/>
- 单页面应用(SPA): 所有功能在一个html页面上实现
- 前端路由作用: 实现业务场景切换
- 优点:
 - 整体不刷新页面, 用户体验更好
 - 数据传递容易, 开发效率高
- 缺点:
 - 开发成本高(需要学习专门知识)
 - 首次加载会比较慢一点。不利于seo



小结

1. 什么是单页面应用?

所有的业务都在一个页面编写, 只有一个html

2. 单页面应用好处?

开发效率高, 用户体验好

3. 单页面如何切换场景?

依赖路由切换显示



问题: 如何在Vue中使用路由呢?

1.2 vue-router

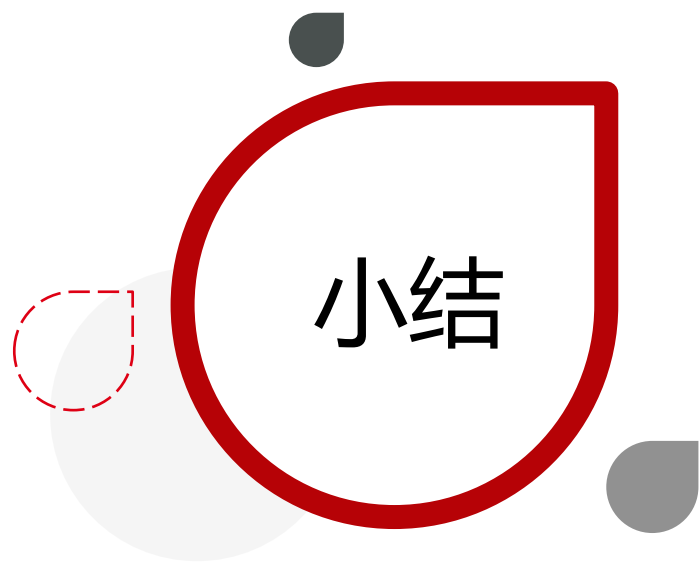
目标： vue-router本质是一个第三方包

- 官网: <https://router.vuejs.org/zh/>
- vue-router模块包
 - 它和 Vue.js 深度集成
 - 可以定义 - 视图表(映射规则)
 - 模块化的
 - 提供2个内置全局组件
 - 声明式导航自动激活的 CSS class 的连接
 -



Vue Router

Vue.js 官方的路由管理器。



以后Vue中如何实现路由？

集成vue-router模块包

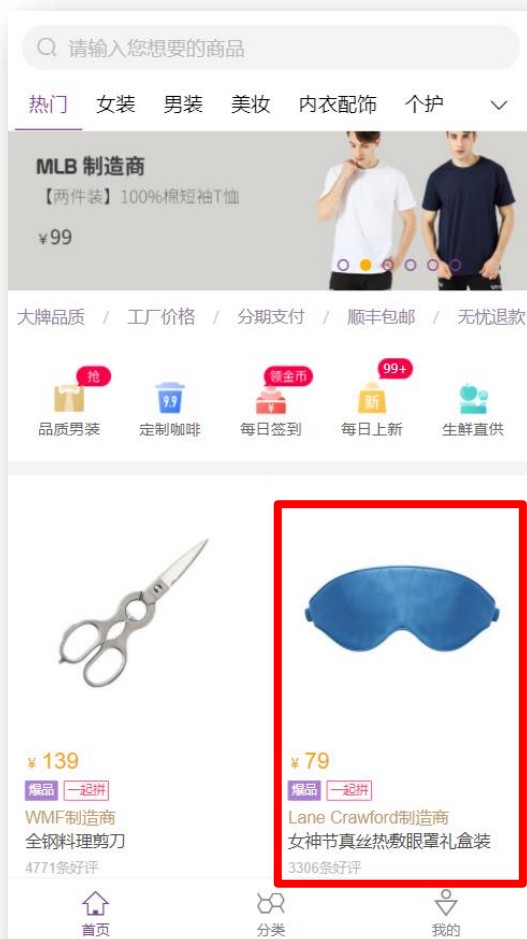


问题: 组件可以分为哪两类呢?

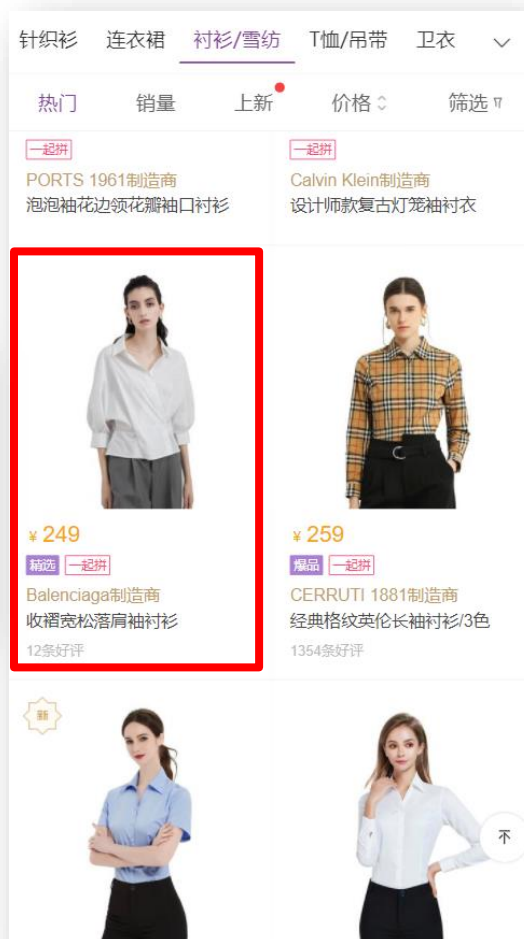
1.3 组件分类

目标：.vue文件分2类, 一个是页面组件, 一个是复用组件

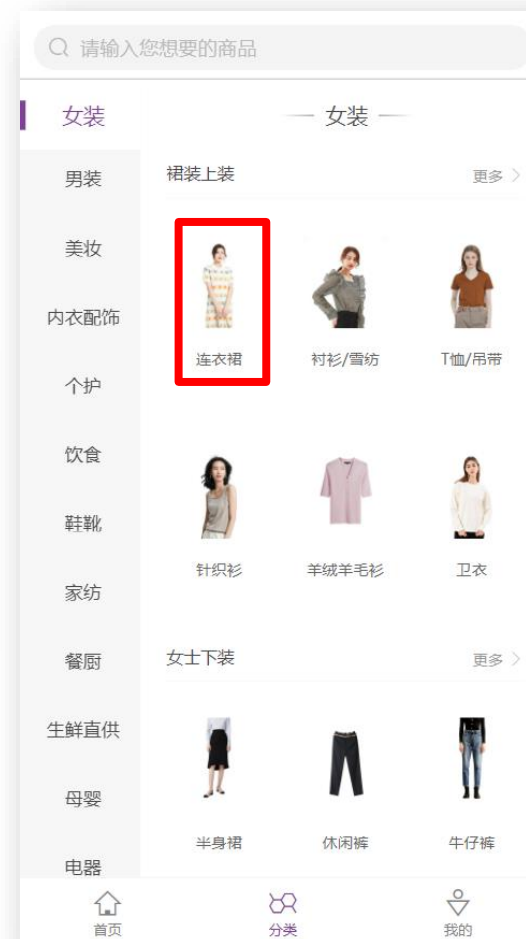
页面组件



页面组件



页面组件

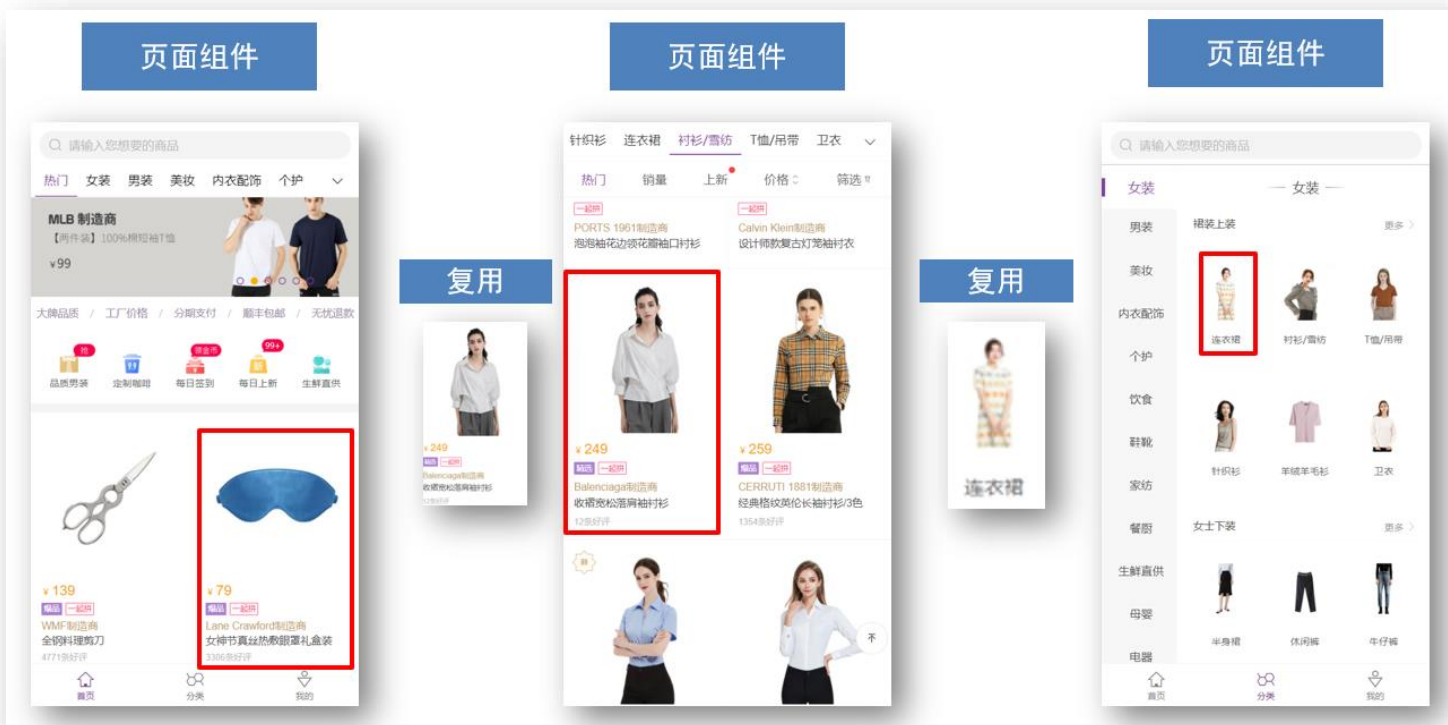


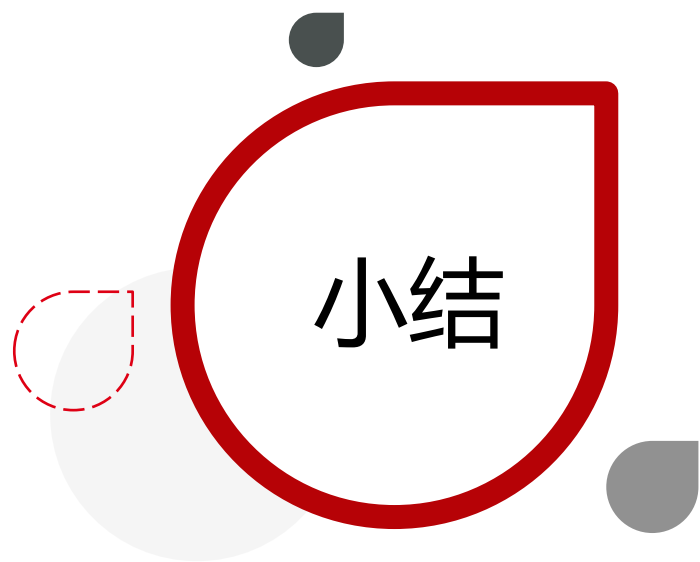
复用

复用

1.3 组件分类

- .vue文件本质无区别, 方便大家学习和理解, 总结的一个经验
- src/views文件夹
 - 页面组件 - 页面展示 - 配合路由用
- src/components文件夹
 - 复用组件 - 展示数据/常用于复用





1. 为何把.vue文件分类?

方便理解和使用

2. 页面组件用在哪里?

配合路由, 切换页面

3. 复用组件用在哪里?

页面组件, 重复渲染结构一样的标签



问题: vue-router如何使用呢?

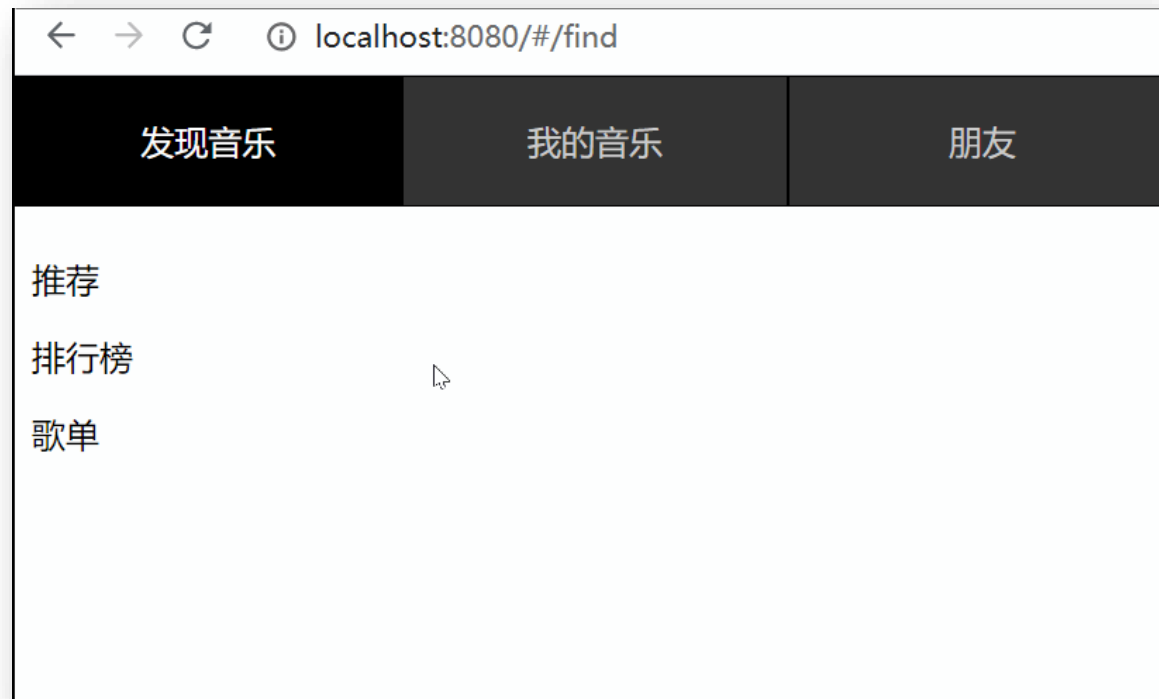
1.4 vue-router模块

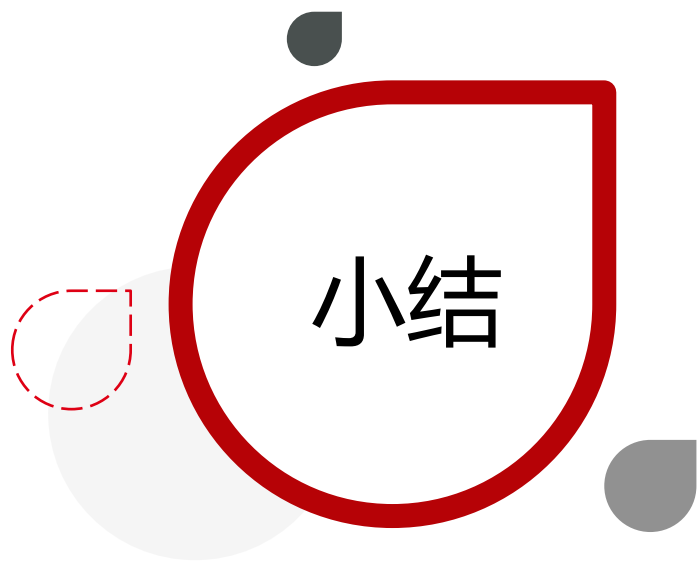
目标：学会vue-router路由系统使用

- 步骤

1. 下载vue-router模块到当前工程
2. 在main.js中引入VueRouter函数
3. 添加到Vue.use()身上 – 注册全局RouterLink和RouterView组件
4. 创建路由规则数组 – 路径和组件名对应关系
5. 用规则生成路由对象
6. 把路由对象注入到new Vue实例中
7. 用router-view作为挂载点, 切换不同的路由页面

注意：一切都要以url上hash值为准





1. vue-router本质是什么?

第三方包, 下载后集成到vue项目中

2. vue-router如何使用?

下包/引入/注册/规则/路由对象/注入/挂载点

3. 规则如何生效?

切换url上hash值, 开始匹配规则, 对应组件展示到
router-view位置



目录

Contents

- ◆ Vue路由简介和基础使用
- ◆ 声明式导航
- ◆ 重定向和模式
- ◆ 程式化导航
- ◆ 嵌套和守卫



声明式导航



问题1: 以前如何实现导航高亮效果?

问题2: 有没有更简单的方式呢?

2.0 声明式导航

目标：可用组件router-link来替代a标签

- 1. vue-router提供了一个全局组件 router-link
- 2. router-link实质上最终会渲染成a链接 to属性等价于提供 href属性(to无需#)
- 3. router-link提供了声明式导航高亮的功能(自带类名)

```
<template>
  <div>
    <div class="footer_wrap">
      <router-link to="/find">发现音乐</router-link>
      <router-link to="/my">我的音乐</router-link>
      <router-link to="/part">朋友</router-link>
    </div>
    <div class="top">
      <router-view></router-view>
    </div>
  </div>
</template>
```

```
<\f6wbJ9f6>
<\qT^>
<\qT^>
```

```
<template>
  <div>
    <div class="footer_wrap">
      <a href="#/find">发现音乐</a>
      <a href="#/my">我的音乐</a>
      <a href="#/part">朋友</a>
    </div>
    <div class="top">
      <router-view></router-view>
    </div>
  </div>
</template>
```

```
<\f6wbJ9f6>
<\qT^>
<\qT^>
```



1. router-link是什么?

VueRouter在全局注册的组件, 本质就是a标签

2. router-link怎么用?

当标签使用, 必须传入to属性, 指定路由路径值

3. router-link好处?

自带激活时的类名, 可以做高亮



问题: 点击导航时, 能否给路由页面传值?

2.1 声明式导航 - 跳转传参

目标：在跳转路由时, 可以给路由对应的组件内传值

- 在router-link上的to属性传值, 语法格式如下
 - `/path?参数名=值`
 - `/path/值` – 需要路由对象提前配置 `path: "/path/参数名"`
- 对应页面组件接收传递过来的值
 - `$route.query.参数名`
 - `$route.params.参数名`





小结

1. 声明式导航跳转时, 如何传值给路由页面?

`to = "/path?参数名=值"`

`to = "/path/值"` (需在路由规则里配置 `/path/:参数名`)

2. 如何接收路由传值?

`$route.query.参数名`

`$route.params.参数名`



目录

Contents

- ◆ Vue路由简介和基础使用
- ◆ 声明式导航
- ◆ 重定向和模式
- ◆ 程式化导航
- ◆ 嵌套和守卫



重定向和模式



问题1: 网页第一次打开没有默认的面页?

问题2: 如何解决呢?

3.0 路由 - 重定向

目标：匹配path后, 强制跳转path路径

- 网页打开url默认hash值是/路径
- redirect是设置要重定向到哪个路由路径

```
const routes = [  
  {  
    path: "/",  
    redirect: "/find"  
  },  
  {  
    path: "/find",  
    component: Find  
  },  
  {  
    path: "/my",  
    component: My  
  },  
  {  
    path: "/part",  
    component: Part  
  }  
]
```



小结

1. 如何监测默认路由?

规则里定义path: '/'

2. 如何重定向路由路径?

redirect配置项, 值为要强制切换的路由路径



思考

问题1: 当你访问不存在的页面会显示什么?

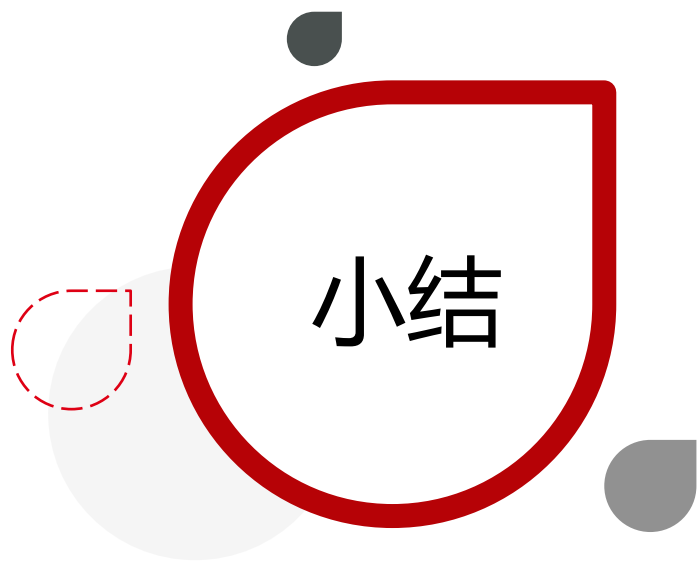
问题2: 当我们访问的路由路径不存在应该怎么办?

3.1 路由 - 404

目标：找不到路径给个提示页面

- 路由最后, path匹配*(任意路径) – 前面不匹配就命中最后这个

```
import NotFound from '@views/NotFound'
const routes = [
  {
    path: "/",
    redirect: "/find"
  },
  { ...
  },
  { ...
  },
  { ...
  },
  {
    path: "*",
    component: NotFound
  }
]
```



如何给路由体系里设置404页面?

在数组最后一个位置, 插入匹配*的规则, 展示404页面



问题1: 路由的路径看起来不自然, 能否切成真正路径形式?

问题2: 如何切换路由模式呢?

3.2 路由 - 模式设置

目标：修改路由在地址栏的模式

- hash路由例如: http://localhost:8080/#/home
- history路由例如: http://localhost:8080/home (以后上线需要服务器端支持, 否则找的是文件夹)

```
const router = new VueRouter({  
  routes,  
  mode: "history"  
})
```



如何修改路由模式呢？

在实例化路由对象时, 传入mode选项和值修改



目录

Contents

- ◆ Vue路由简介和基础使用
- ◆ 声明式导航
- ◆ 重定向和模式
- ◆ 程式化导航
- ◆ 嵌套和守卫



编程式导航



问题: a标签可以切换路由路径, 用JS方式可以切换吗?

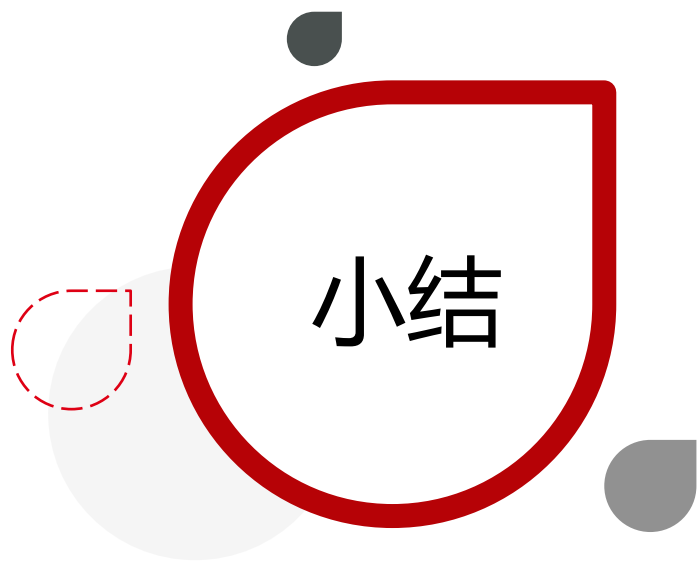
4.0 程式导航 - 基础使用

目标：用JS代码来进行跳转

- 语法: path或者name任选一个

```
●●●  
this.$router.push({  
  path: "路由路径", // 都去 router/index.js定义  
  name: "路由名"  
})|
```

```
}|
```



JS如何切换路由路径?

`this.$router.push()`配置path/name

要和路由规则数组里对应



问题: JS切换路由, 能否传值给路由页面?

4.1 程式导航 - 跳转传参

目标: JS跳转路由, 传参

- 语法: query或者params任选一个

```
this.$router.push({  
  path: "路由路径"  
  name: "路由名",  
  query: {  
    "参数名": 值  
  },  
  params: {  
    "参数名": 值  
  }  
})
```

注意: 使用path会忽略params

```
<template>  
  <div>  
    <p>关注明星</p>  
    <p>发现精彩</p>  
    <p>寻找伙伴</p>  
    <p>加入我们</p>  
    <p>人名: {{ $route.query.name }}  
    --  
    {{ $route.params.username }}  
  </p>  
  </div>  
</template>
```



小结

JS切换路由如何传值, 如何接收值?

query传, `$route.query`接

params传, `$route.params`接



目录

Contents

- ◆ Vue路由简介和基础使用
- ◆ 声明式导航
- ◆ 重定向和模式
- ◆ 编程式导航
- ◆ 嵌套和守卫



嵌套和守卫

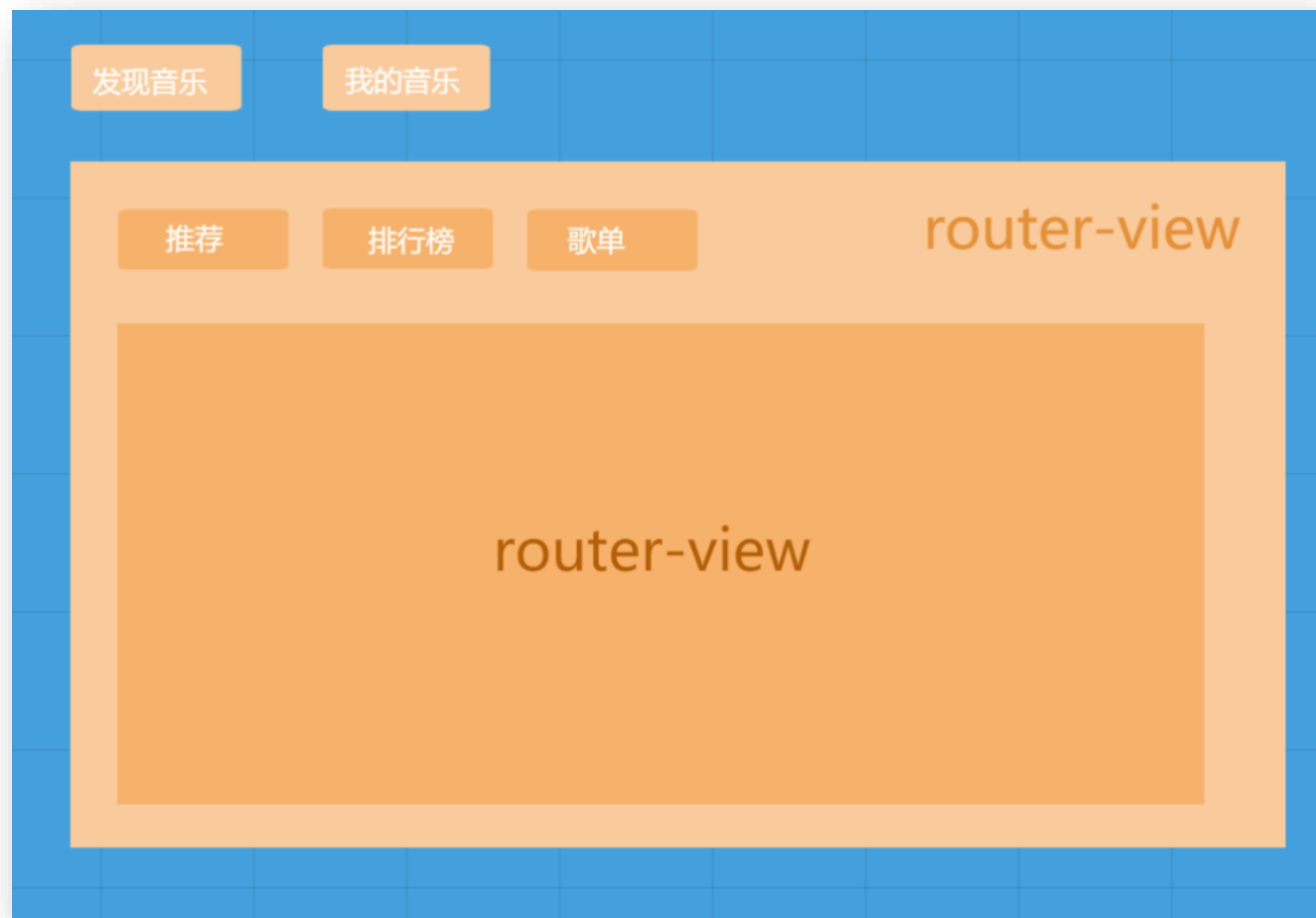


问题: 某路由页面, 可否再套入一套路由系统?

5.0 路由 - 路由嵌套

目标：在现有的一级路由下, 再嵌套二级路由

- 二级路由示例-网易云音乐-发现音乐下 <https://music.163.com/>

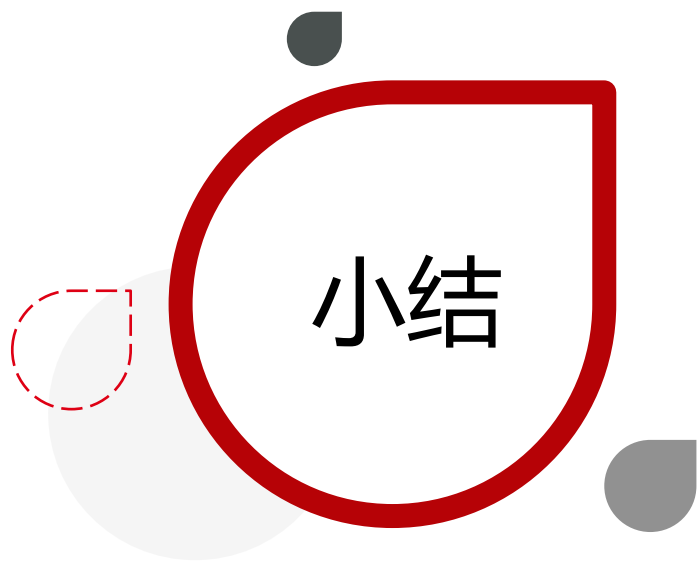


5.0 路由 - 路由嵌套

目标：在现有的一级路由下, 再嵌套二级路由

- 1. 创建需要用的所有组件
 - src/views/Find.vue -- 发现音乐页
 - src/views/My.vue -- 我的音乐页
 - src/views/Second/Recommend.vue -- 发现音乐页 / 推荐页面
 - src/views/Second/Ranking.vue -- 发现音乐页 / 排行榜页面
 - src/views/Second/SongList.vue -- 发现音乐页 / 歌单页面
- 2. main.js- 继续配置2级路由
 - 一级路由path从/开始定义
 - 二级路由往后path直接写名字, 无需/开头
 - 嵌套路由在上级路由的children数组里编写路由信息对象
- 3. 说明:
 - App.vue的router-view负责发现音乐和我的音乐页面, 切换
 - Find.vue的router-view负责发现音乐下的, 三个页面, 切换

```
{
  path: "/find",
  component: Find,
  children: [
    {
      path: "ranking",
      component: Ranking
    },
    {
      path: "recommend",
      component: Recommend
    },
    {
      path: "songlist",
      component: SongList
    }
  ]
},
```



1. 二级路由如何配置？

创建需要的二级页面组件

路由规则里children中配置二级路由规则对象

一级页面中设置router-view显示二级路由页面

2. 二级路由注意什么？

二级路由path一般不写根路径/

跳转时路径要从/开始写全

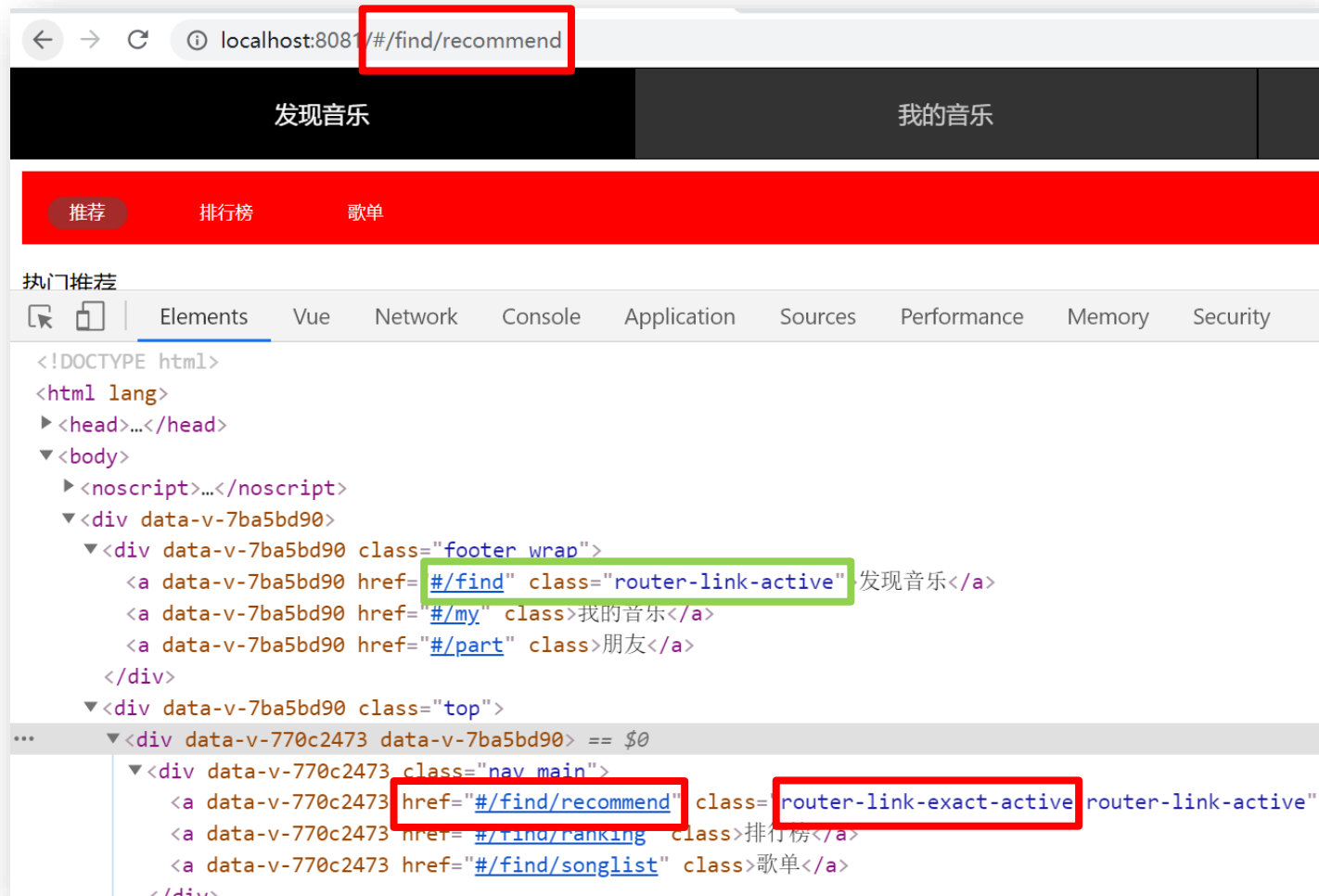


问题: router-link激活的2个类名有什么区别呢?

5.1 声明式导航 – 类名区别

- 观察路由嵌套导航的样式

- router-link-exact-active (精确匹配) url中hash值路径, 与href属性值完全相同, 设置此类名
- router-link-active (模糊匹配) url中hash值, 包含href属性值这个路径





小结

1. 自动添加的2个类名的区别?

router-link-exact-active – url的hash值和href完全匹配

router-link-active – url的hash值包含href路径值匹配



问题: 如何给路由添加一个权限判断?

5.2 全局前置守卫

目标：路由跳转之前, 会触发一个函数

- 例如: 登陆状态去<我的音乐>页面, 未登录弹窗提示
- 语法: `router.beforeEach((to, from, next) => {})`
 - 一定调`next()`, 才会跳转下一页

```
let isLogin = false; // 未登录
router.beforeEach((to, from, next) => {
  if (to.path === '/my' && isLogin === false) {
    alert("请登录")
    next(false) // 阻止路由跳转
  } else {
    next()
  }
})
```

```
})
```

```
}
```



小结

什么是路由守卫?

路由在真正跳转前, 会执行一次beforeEach函数, next调用则跳转, 也可以强制修改要跳转的路由



目录

Contents

◆ Vant组件库



Vant组件库



问题1: 什么是组件?

问题2: 什么是组件库?

1.0 Vant组件库

目标： Vant是一个轻量、可靠的移动端 Vue 组件库, 开箱即用

- Vant组件库: <https://vant-contrib.gitee.io/vant/#/zh-CN/>
- 特点
 - 提供 60 多个高质量组件, 覆盖移动端各类场景
 - 性能极佳, 组件平均体积不到 1kb
 - 完善的中英文文档和示例
 - 支持 Vue 2 & Vue 3
 - 支持按需引入和主题定制



Vant

轻量、可靠的移动端 Vue 组件库



什么是vant组件库?

封装了很多的组件, 支持移动项目中大多数使用场景



问题: 我的Vue项目中, 如何使用Vant组件库?

1.1 全部引入

目标：看官网文档, 下载, 引入Vant组件库

1. 全部引入, 快速开始: <https://vant-contrib.gitee.io/vant/#/zh-CN/quickstart>
2. 下载Vant组件库到当前项目中
3. 在main.js中全局导入所有组件,
4. 使用按钮组件 – 作为示范的例子

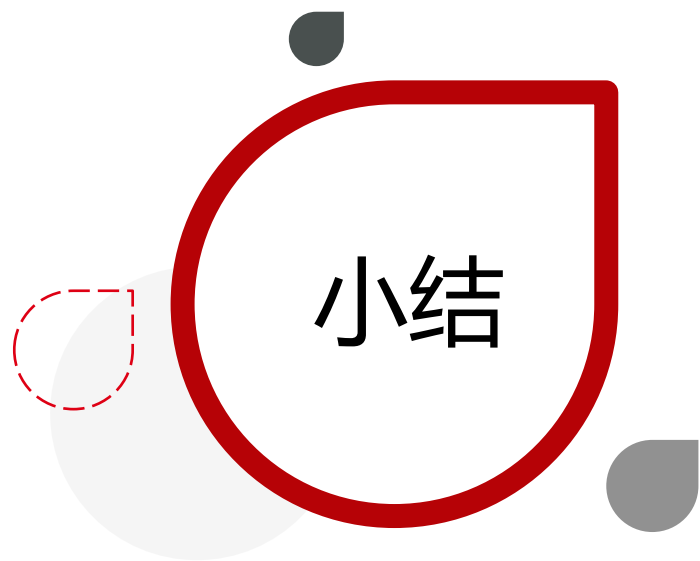
方式三. 导入所有组件

Vant 支持一次性导入所有组件, 引入所有组件会增加代码包体积, 因此不推荐这种做法。

```
import Vue from 'vue';  
import Vant from 'vant';  
import 'vant/lib/index.css';
```

```
Vue.use(Vant);
```

Tips: 配置按需引入后, 将不允许直接导入所有组件。



如何引入Vant组件库?

组件库是一个包, 先下载

按照文档指引, 在main.js全局注册

到某.vue内直接使用Vant组件名



问题1: 全部引入, 但是只用几个组件, 浪费项目体积?

问题2: 如何能让它按需引入呢?

1.2 按需引入

目标：手动引入使用的某个组件

1. 手动单独引入, 快速开始: <https://vant-contrib.gitee.io/vant/#/zh-CN/quickstart>

方式二. 手动按需引入组件

在不使用插件的情况下，可以手动引入需要的组件。

```
import Button from 'vant/lib/button';  
import 'vant/lib/button/style';
```




小结

Vant组件库如何手动按需引入使用？

import 从vant库里引出某个组件

import 单独引出样式

在当前页面注册此组件名即可



问题1: 手动引入单个, 每次都引入样式累不?

问题2: 如何能让它自动按需引入?

1.3 自动按需引入

目标：依赖插件自动按需引入

1. 自动按需引入, 快速开始: <https://vant-contrib.gitee.io/vant/#/zh-CN/quickstart>

方式一. 自动按需引入组件 (推荐)

`babel-plugin-import` 是一款 babel 插件, 它会在编译过程中将 `import` 的写法自动转换为按需引入的方式。

安装插件

```
npm i babel-plugin-import -D
```



小结

如何自动按需引入Vant组件库?

webpack依赖babel-plugin-import的插件

把import按需引入方式, 自动转成只引入某个组件方式



问题: 做个弹出框, Vant里有没有?

1.4 弹出框使用

目标：完成效果

- 弹出框地址: <https://vant-contrib.gitee.io/vant/#/zh-CN/dialog>





想使用某个UI界面时怎么办?

去文档查找相关例子

然后根据步骤和提示使用即可



问题: 想做个登录页面?

1.5 表单使用

目标: 使用Vant组件里的表单组件

- 具体 – 看表单组件地址: <https://vant-contrib.gitee.io/vant/#/zh-CN/form>



A screenshot of a Vant Form component. It features two input fields, each with a label '用户名' (Username) and a placeholder '用户名'. Below these is a second input field with a label '密码' (Password) and a placeholder '密码'. At the bottom of the form is a large blue button with the text '提交' (Submit).



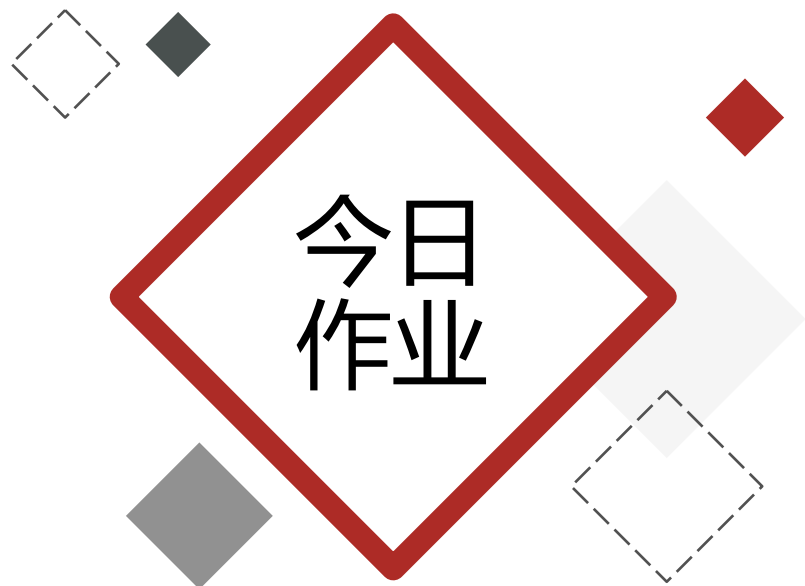
使用Vant组件库, 需要注意什么?

vant组件库里有很多标签, 仔细和快速的熟悉文档



思考

1. 什么是路由?
2. 单页面概念和优缺点?
3. vue-router路由模块如何使用?
4. 链接导航和程式化导航的区别?
5. 路由嵌套和守卫如何使用?
6. Vant组件库如何使用?



1. md笔记里有附加练习
2. 完成今晚三级路由嵌套作业



传智教育旗下高端IT教育品牌