

生命周期 + 路由 + 组件缓存







- 1. 知道 组件生命周期,掌握组件生命周期函数
- 2. 知道 路由基础知识, 搭建单页应用程序
- 3. 知道 组件keep-alive,实现组件缓存





- ◆ 生命周期
- ◆ 路由基础
- ◆ 综合案例 面经H5



案例效果:







- ◆ 生命周期
- ◆ 路由基础
- ◆ 综合案例 面经H5



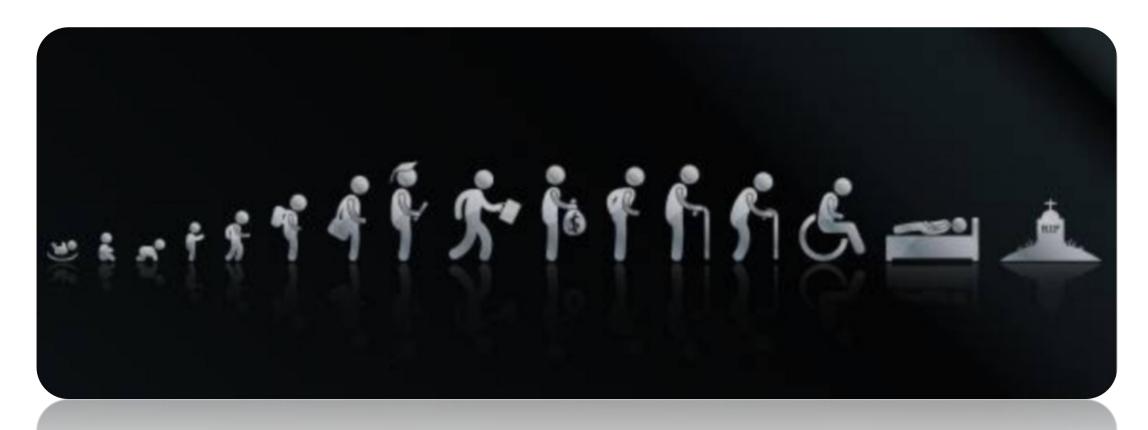
问题1: 什么是生命周期?

问题2: Vue所谓的生命周期又是什么?



生命周期基本认知

一个人的生命周期:一个人从出生到逝去的过程





生命周期基本认知

vue组件生命周期: 从创建 到 销毁 的整个过程就是 - Vue实例的 - 生命周期







Vue的生命周期是什么?

是Vue组件实例,从创建到销毁的全过程





Vue生命周期有哪些阶段呢?如何知道到了哪些阶段呢?



生命周期-钩子函数概述

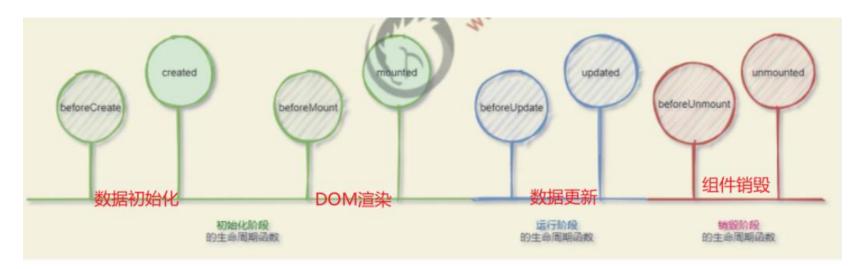
生命周期函数:是由 vue 框架提供的内置函数,会伴随着组件的生命周期,自动按次序执行。

● 作用:特定的时间点,执行特定的操作

● 比如:组件创建完毕后,可以在created 生命周期函数中发起Ajax 请求,从而初始化 data 数据

● 分类:三大阶段,8个方法

- 组件`初始化阶段`的生命周期函数
- 组件`运行阶段`的生命周期函数
- 组件`销毁阶段`的生命周期函数







什么是vue的生命周期函数?

vue 框架提供的内置函数。

自动在生命周期对应阶段,调用执行。

整体生命周期分为几大阶段?

初始化阶段 (准备数据+dom渲染)

运行阶段

组件销毁阶段



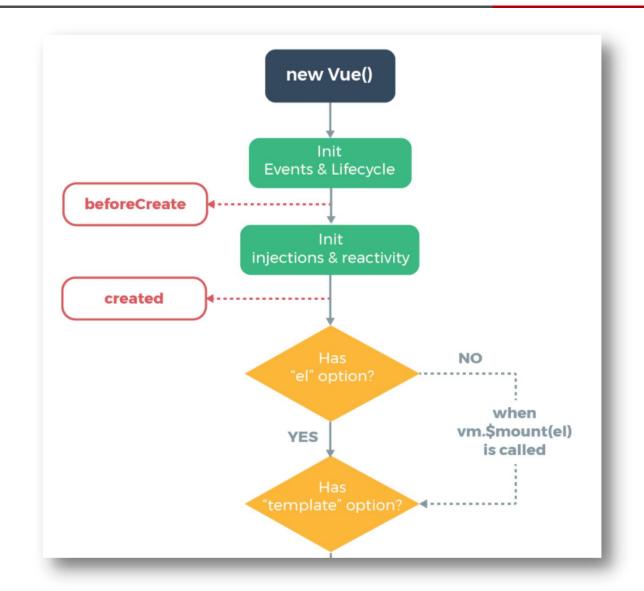


那 vue 组件 整个生命周期到底是如何进行的呢?



生命周期-图解

生命周期详细图解:地址





整体生命周期分为几大阶段?

初始化阶段 (准备数据+dom渲染)

运行阶段 (数据更新阶段)

组件销毁阶段



生命周期钩子-详解

八大生命周期钩子函数:

1. beforeCreate: data数据初始化之前,组件还没有数据

2. created: data数据初始化之后,可以获取到组件的数据

3. beforeMount: DOM渲染之前, DOM还没渲染

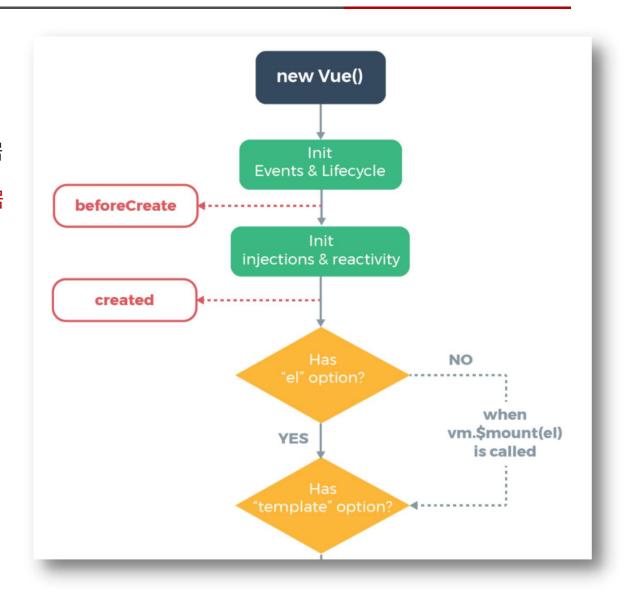
4. mounted: DOM渲染之后,可以操作DOM了

5. beforeUpdate: 数据更新, DOM更新前

6. updated: 数据更新, DOM更新后

7. beforeDestroy: 组件销毁前

8. destroyed: 组件销毁后







八个生命周期钩子函数中,最常用的两个是什么?

created (可以开始发送ajax请求,获取数据)

mounted (可以开始操作dom了)





- ◆ 生命周期
- ◆ 路由基础
- ◆ 综合案例 面经H5





思考: 什么是单页应用程序?



单页应用程序: SPA - Single Page Application

- 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 (多页面应用程序MPA)
- 优点:
 - **不整个刷新页面**,每次请求仅获取需要的部分,**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 路由)
 - 首次加载会比较慢一点。不利于seo





1. 什么是单页面应用程序?

所有功能在一个html页面上实现

2. 单页面应用好处?

不整个刷新页面,开发效率高,用户体验好





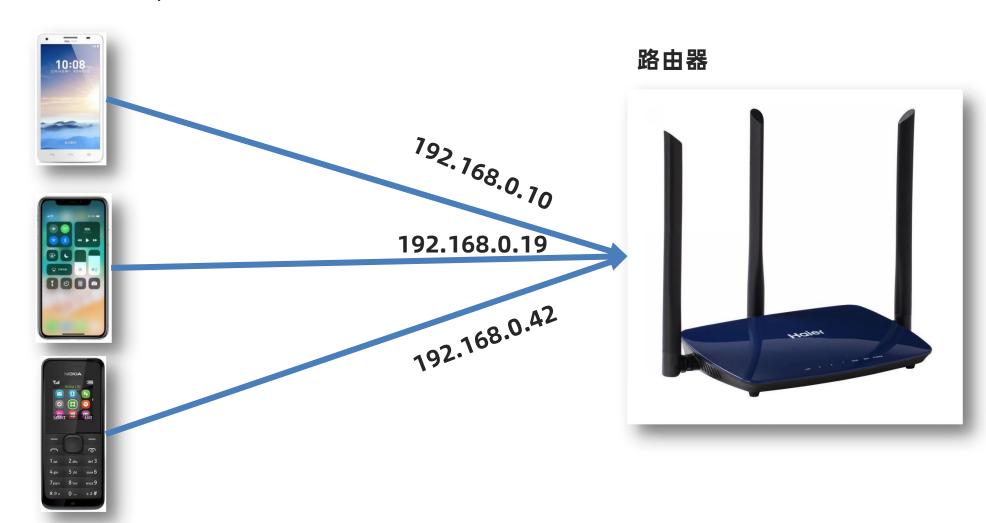
学习单页应用程序,需要学习路由。

那么,什么是路由呢?



路由的介绍

生活中的路由:设备和ip的映射关系





路由的介绍

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. 前端Vue中的路由是什么?

路径 和 组件的映射关系





路径和组件的切换匹配,怎么实现呢?



vue-router的介绍 和 组件的分类说明

vue-router: vue-router本质是 vue 官方的一个路由插件,是一个第三方包

- 官网: https://router.vuejs.org/zh/
- vue-router集成了路径和组件的切换匹配处理,我们只需要配置规则即可。

在具体学习 vue-router 如何使用之前,我们需要认知 组件的两种分类



Vue Router

Vue.js 官方的路由管理器。



vue-router的介绍 和 组件的分类说明

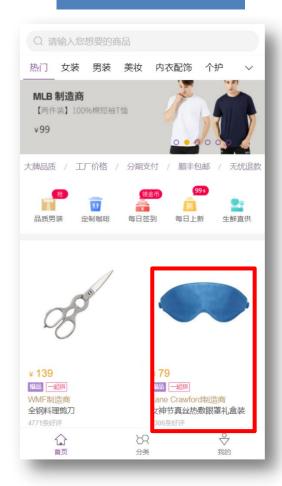
组件分类: .vue文件分2类,一个是页面组件,一个是复用组件

复用

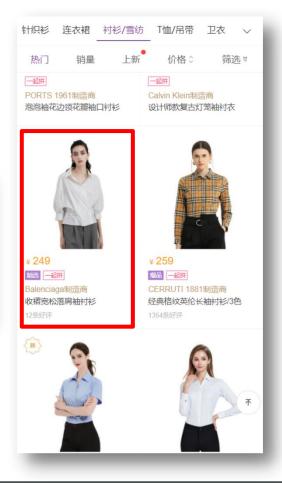
¥ 249

総法 一起料

页面组件



页面组件



页面组件



复用

连衣裙



vue-router的介绍 和 组件的分类说明

- .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





vue-router 是什么?

是一个vue官方提供的一个插件,用于处理路由

组件分类有哪两类?放在什么文件夹?作用分别是什么?

页面组件 和 复用组件

页面组件 => views文件夹 => 页面展示,配合路由

复用组件 => components文件夹 => 封装复用



问题: vue-router如何使用呢?



vue-router的基本使用(5 + 2)

- 5个基础步骤
- 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 帮你注册全局RouterLink和RouterView组件
- 4. 创建路由对象
- 5. 将路由对象注入到new Vue实例中

2个核心步骤:

- 1. 配置路由规则
- 2. 指定路由出口 router-view



D localhost:8080/#/



vue-router的使用基本步骤? (5+2)

下包/引入/use注册/路由对象/注入Vue实例

规则/路由出口





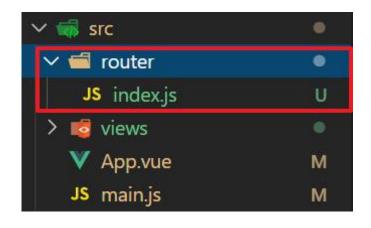
所有的路由配置都堆在main.js中合适么?

应该做抽离!



路由的封装抽离

目标:将路由模块抽离出来。好处:拆分模块,利于维护



```
import router from './router'

new Vue({
    // 5. 关联路由对象和vue实例 data methods
    render: (h) => h(App),
    router,
}).$mount('#app')
```





路由模块的封装抽离的好处是什么?

拆分模块, 利于维护





问题:点击导航时,希望高亮怎么办?



声明式导航 - 导航链接

组件router-link 替代 a标签,能跳转,能高亮

- 1. vue-router提供了一个全局组件 router-link
- 2. router-link<mark>实质上</mark>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>
```

```
<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>
```





1. router-link是什么?

VueRouter提供的全局组件,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: router-link 会自动给当前导航添加两个类名,

这两个类名有区别么?



声明式导航 - 两个类名

说明: router-link 会自动给当前导航添加两个类名

● router-link-active: 激活的导航链接 模糊匹配

to="/my" 可以匹配 /my /my/a /my/b

● router-link-exact-active: 激活的导航链接 精确匹配

to="/my" 仅可以匹配 /my

可以修改默认高亮的类名

```
const router = new VueRouter({
    linkActiveClass: 'aa',
    linkExactActiveClass: 'aa',
    // route: 一条规则
}
```





router-link 会自动给当前导航添加两个类名,有什么区别呢?

router-link-active 模糊匹配

router-link-exact-active 精确匹配

默认的类名太长了,可以自定义么?

linkActiveClass

linkExactActiveClass



问题:点击导航时,能否给路由页面传值?



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





声明式导航 - 跳转传参

直接query传值不就好了么?为啥要在路径上传值? => 利用url分层特性 暗示结构

提供合理的资源名称

制作出色的API是80%的艺术和20%的科学。创建表示合理资源的URL层次结构是艺术部分。拥有合理的资源名称(只是URL路径,例如/customers / 12345 / orders)可以提高给定请求的清晰度。

适当的资源名称为服务请求提供上下文,从而提高API的可理解性。通过URI名称对资源进行分层查看,为消费者提供友好,易于理解的资源层次结构,以便在其应用程序中使用。

以下是一些URL路径(资源名称)设计的快速规则:

• 在您的网址中使用标识符,而不是在查询字符串中使用。使用URL查询字符串参数非常适合过滤,但不适用于资源名称。

• 好: / users / 12345

差: / api? type = user&id = 23

- 利用URL的分层特性来暗示结构。
- 为您的客户而不是您的数据设计。





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



问题1: 我们的网页第一次打开,有没有默认的页面?

问题2: 如何解决呢?



vue路由 - 重定向

重定向: 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
```





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





问题: 当访问不存在的页面应该要显示什么呢?



vue路由 - 404

404: 当找不到路径匹配时, 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: 路由的路径看起来不自然, 有#, 能否切成真正路径形式?



vue路由-模式设置-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)

```
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



问题: a标签可以切换路由路径, 用JS方式可以切换吗?



编程式导航 - 基本跳转

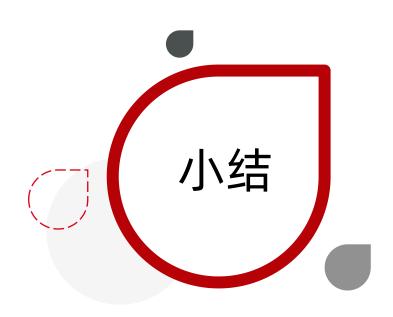
编程式导航:用JS代码来进行跳转

● 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
})

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





JS如何跳转路由路径?

this.\$router.push() 配置path 或 name



问题: JS切换路由, 能否传值给路由页面?



编程式导航 - 路由传参

● 语法: query或params都可以传参

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

query传: \$route.query.xxx 接收

params传: \$route.params.xxx 接收

区别:

- params传参:是在内存中传参,刷新会丢失
- query传参:是在地址栏传参,刷新还在

注意: 使用path会忽略params

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



编程式导航, 切换路由如何传值, 如何接收值?

query传, \$route.query接收

params传, \$route.params接收



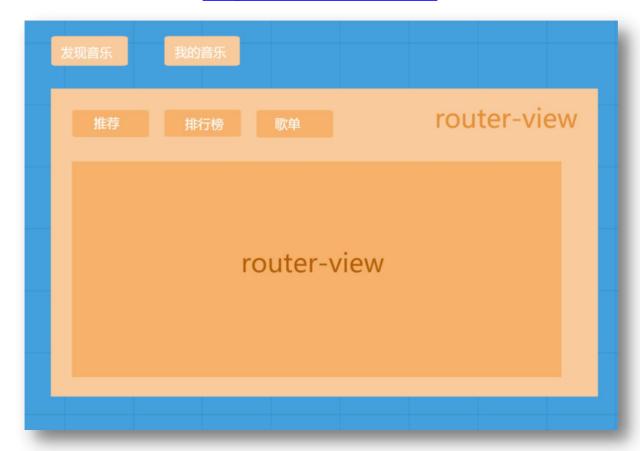
问题: 某路由页面, 能否再套入一套路由系统?



路由嵌套

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/





路由嵌套

- 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 (会自动拼接)
 - 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
```





二级嵌套路由如何配置?

创建需要的二级页面组件

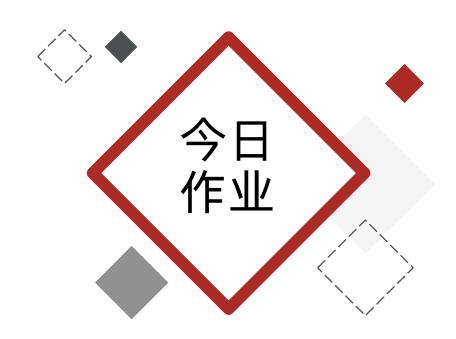
路由规则里children中,配置二级路由规则对象

一级页面中设置router-view, 切换显示二级路由页面





- ◆ 生命周期
- ◆ 路由基础
- ◆ 综合案例 面经H5



把课上例子,练习,案例做一遍

预习第二天内容



案例效果:





传智教育旗下高端IT教育品牌