今天学习的内容

1:复习昨天重点

练习: 9:20--9:30

(1)创建二个组件 main(父组件)/son(子组件)

(2)main 显示一个按钮

(3)son 显示 一个p标签

(4)main 定义一个变量count初始化为0

将count传递给son组件,在son子组件接收数据并且在p标签显示

(5)main 点击按钮时，实现count自增操作

(6)要求 son组件显示count对应变化

(7)son在接到count之后计算,在count大于10时候,将main按钮禁用

提示:

<button v-bind:disabled="isValid">clickMe</button>

总结:

1:传递到子组件的数据，父组件中数据发生变化子组件一并更新

2:子组件中updated方法，会随着数据改变执行

3:修改父元素数据 this.$parent

2:今天的目标

2.1: Vue Router 路由

2.2 SPA(single page application)

SPA(single page application)单页面应用程序，在一个完成的应用或

者站点中，只有一个完整的HTML页面，这个页面有一个容器，

可以把需要加载的代码插入到该容器中.

2.3SPA的工作原理

http://127.0.0.1/index.html#/start

(1)根据地址栏 中url解析完的页面index.html 加载index.html

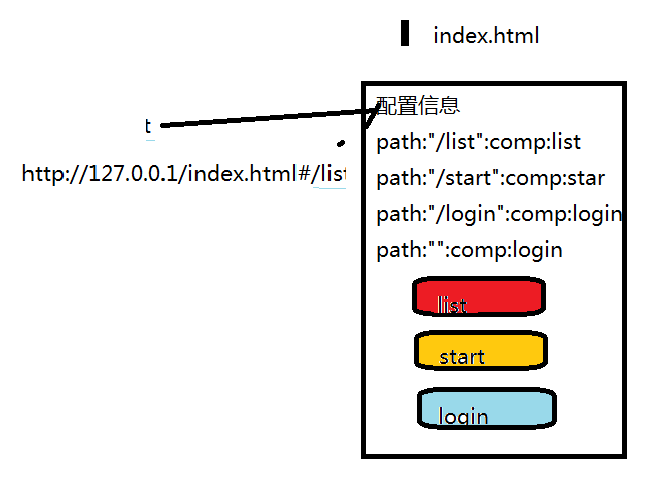
(2)根据地址栏 解件#后的**路由地址**:start

根据路由地下，去当前应用的配置中找路由地址配置对象去查

找路由地址，所对应的模块页面地址

发起异步请求加载该页面地址

(3)把请求来的数据加载到指定容器中



2.4 SPA的实现步骤

(1)引入对应 vue-router.js 文件

(2)显示组件 前:<div id="app"><parent></parent></div>

显示组件 现: <div id="app">

<router-view></router-view> 渲染不同组件

</div>

(3)创建各个组件

(4)配置信息(路由词典)

每一个路由地址配置对象(组件)

const myRoutes=[

{path:"/start",component:myStart},

{path:"/list",component:myList},

];

(5)创建路由表对象(语法标签)

const myRouter = new VueRouter({

routes: myRoutes

});

(6)在Vue对象添加现在应用路由表对象

new Vue({

router: myRouter

});

(7)测试

在地址中输入对应的不同路由地址中，显示不同组件

练习:11:25--11:35

创建一个SPA,这个SPA有3个组件，分别对

path /collect -->收藏页组件

path /detail -->详情页组件

path /order -->订单页组件

2.5 SPA 通过VueRouter来实现组件之间跳转

提供了3种方式实现跳转

(1)直接修改地址栏中路由地址

(2)通过router-link 实现跳转

<router-link to="/order">订单</router-link>

(3)通过js的编程方式来实现

jumpToLogin:function(){

this.$router.push("/login");

}

2.x: Vue Resource http协议插件

2.x: Vue cli 脚手架