Vue框架==》一套项目的解决方案

Jquery核心则是操作dom节点

Vue核心则是展示数据（数据从后台调取）

Mvvm model指的是js中的数据

View 视图 html中展示的内容

后面两个字母代表 view-model帮助咱们从js向html中绑定数据 帮助咱们获取html获取数据

Vue2.x 不兼容i8及以下版本 因为 Vue 使用了 IE8 无法模拟的 ECMAScript 5 特性

因为vue核心是双向绑定，使用es5实现, es5的特性是object.defineProperty

{{}}差值表达式 作用：1.把vue管理的数据显示到html中

2.调用函数{{a()}}

3.执行js表达式{{text+url}} 但是不能执行++ --

能用的是+-\*/等

指令：封装好的dom操作 如：

v-text==>innertext 把标签的内容渲染成字符串 <div v-text="text"></div><div v-text="htmlstr"></div>

v-html==>innerhtml 可以把标签的内容中字符串类似于字节的内容渲染成真正的节点 还能执行method中的方法

也可以执行js表达式 如果直接写字符串 要用引号引起来

<div v-html="text"></div> <div v-html="htmlstr"></div>

<div v-html="a()"></div> <div v-html="text+'滚你大爷的'"></div>

<div v-html="'滚你大爷的'"></div>-->

V-model 双向数据绑定 可以将input中的value值和data中数据进行绑定

，<div id=’app’>

<input type=’text’ v-model=’text’>

<p>{{text}}</p>

</div>

V-model绑定select的值 option有value绑定其value值 没有value那么久绑定option的内容

V-bind给dom节点绑定属性 缩写 不写v-bind 直接：...

<img v-bind:src=’url’>

V-for v-for循环中的v-for=”school in arr” school在本标签里面可以随意使用代表的是arr中的值

<li v-for=”school in arr” v-text=”school”></li>

指令就是封装好的dom操作 这点要知道

绑定class的三种方式

5.16

V-for=”（item,index）in arr” index代表了item在arr中的索引（下标）这时候通过索引可以做到v-bind:class=”’bg’；index%2==0?false:true”

奇数偶数背景变换

Forin循环对象 第一种 =”value in obj “value代表的就是obj里面的属性值 注意是属性值 不是key属性

第二种 =”(value,key) in obj”value是属性值 key是属性

第三种 =”(value,key,index) in obj”value是属性值 key是属性 index是下标

V-show 显示和隐藏 通过控制display来操作 v-show=”true/false”

V-on：click=‘’v-on添加事件

<button v-on:click=’a’></button>

这个a可以是函数相当于点击时执行函数a 而且可以传参 参数可以是data中的属性值 也可以是for循环里面新产生的那个

5-17

钩子函数 也叫生命周期 按照一定顺序执行不同时期的函数 首先是bind

Vue-directive(‘redtext’,{

Bind:function(el){//指令第一次和元素绑定 el指挂载到的函数 和通过getelment获取的节点一样

El.style.width=’200px’;

}，

Inserted:function(el){

//注意这里是元素插入到页面中的时候触发 假如把el.focus放在上面bind中 它是没有效果的 因为 bind相当于页面渲染 insert是渲染过后的事件触发

El.focus();

}

})参数一是指令名称 参数二是对象即钩子函数 相当于自定义函数

自定义指令在任意标签中都可以使用 即使不在指定id的标签中

指令放在vue中就变成了私有指令 而且它可以为复数

New Vue({

El:’#app’,

Directives:{

‘red’:{

Bind:function(el){...},

Inserted:function(el){...}

}

}

})

简单写法 直接名称加函数

Red:function(el){el.style.width=’20px’}

但是它只能在bind和update中起作用 也就是说他只能在一开始渲染和之后标签更新的时候使用

This.$options.自定义函数名（）自定义函数要使用data中的参数时。要用具体的vue对象来使用 var vm=new Vue(..

Data:{

Text=’尼玛’

}

.)

A:function(){console.log(vm.text)}

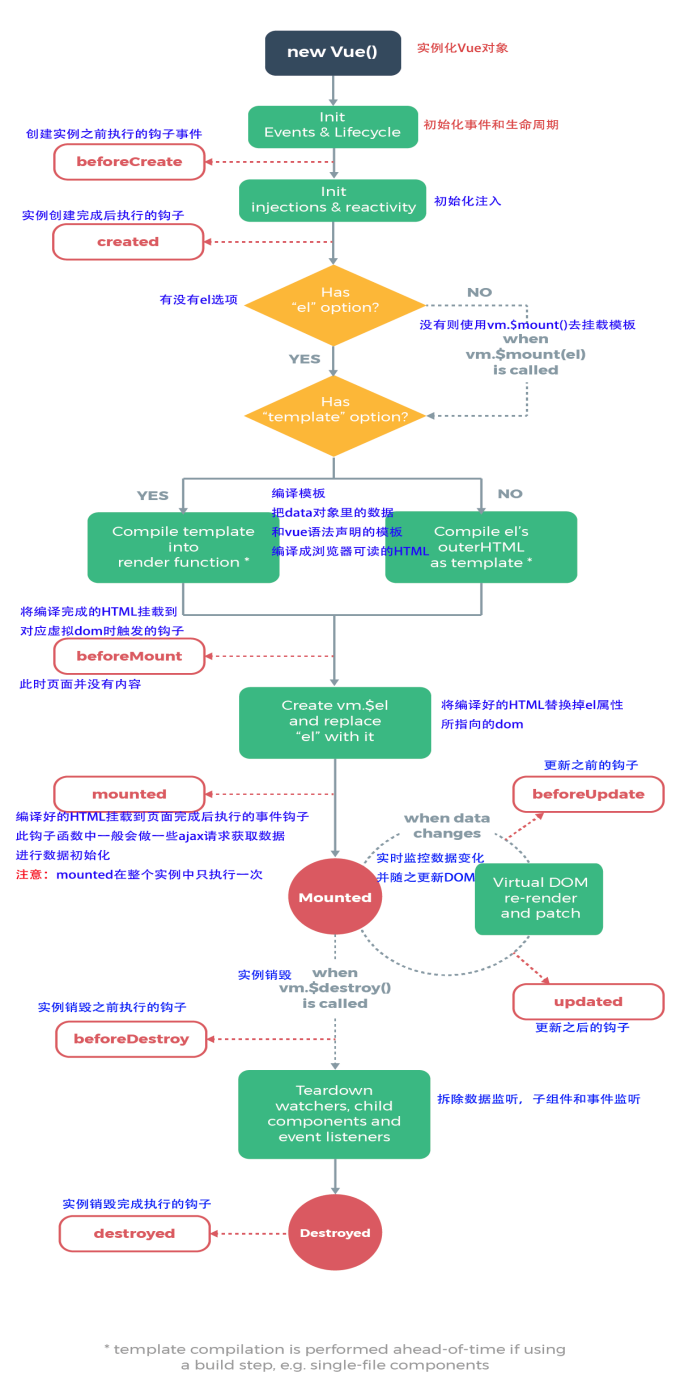
$mount(‘#app’)可以手动把vue挂载到一个实例中

Vm.$destroy()销毁实例 销毁数据 方法指令 与视图的直接联系 就是在操作板里面无法直接操作数据 即使操作数据 与原数据绑定的标签内容也不会改变 就是双向绑定消失了

computed

不过只能用在methods的function中

Html不区分大小写 js区分大小写 js中是aB 在html中要写成a-b -b表示大写B



5-18

组件 参数一 新标签名称 参数二新标签内容

<div id=”app”><nihao></nihao></div>

Vue.component('nihao',{

template:"<divclass='haha'>houhou</div>"

})

new Vue({

el:'#app'})

组件的本质还是vue实例vue实例上面支持的生命周期 计算属性等等组件也可以支持

注意 假如组件里面有两个标签只会显示第一个 可以用一个大标签包裹所有想显示的标签

第二种写法

Html5写组件模板的标签

<template id=’abc’>

<div>haha</div>

</template>

Vue.component('nihao',{

template:"#abc"

})

new Vue({

el:'#app'})

第三种写法

<script type=’text/template’ id=’’adc>

<div>哈哈哈</div>

</script>

template:"#abc"

})

new Vue({

el:'#app'})

三种写法都要注意根标签问题 但是第二种 和第三种无疑更好 因为这两个不用考虑引号问题

组件编写顺序 ：

1. 编写data中的数据 注意要放在function
2. 编写method方法来改变data中的数据 注意this指向是当前组件
3. 在模板中，通过指令绑定当前组件的数据和方法 它的使用范围在template标签内使用 和vue对象使用方法相同

组件template里边无法直接使用new Vue里的数据data 要定义一个接受数据

Props[‘text’，‘haha’] 通过v-bind:text=’字符串/new Vue里面data数据’

假如组件使用自己内部数据 改变也是使用自己内部方法改变数据 那么不同组件即使相同 但是 一个组件数据改变并不会影响其他组件 也就是说其他组件数据不变

computed:{

//在这里处理其他组件传递的信息

//处理后台调取的数据

//注意计算后的名称不要和data中的名称相同

num1:function(){

return this.num\*2

}

}

5-20

父传子 子级设变量 props:[”list”] v-bind:接收的数据名称（component中的变量）=传递的数据名称（父级Vue中的变量） <list-body v-bind:list="list"></list-body>

一般情况下 这两个名称写成一样

建议写tableData表示表格中的数据

子传父 监听子级触发子级用以触发父级 <list-body v-on:myevent='gettext'></list-body>

this.$emit(‘自定义事件’，传递的数据) v-on:自定义事件 触发父组件的方法

V-on：自定义事件=‘接收数据的父组件的方法’

在注意调用接收数据的方法时不能加（） 因为加括号代表了传参

mounted:function(){

bus.$on('myevent',(data)=>{

this.list.push(data)

})

}

这里要注意的是 放在mounted中 就是挂载过后执行 不能没挂载就直接接收数据 即使接收到了 也是旧的数据啊 还有 bus.$on里面的data函数function(data)指向是bus 而不是接收组件 所以要么在mounted中 bus.on这句话外面设置一个变量把this（代指接收组件）然后在bus.on中使用该变量 要么使用es6新写法 (data)=>{}里面没有this会出去寻找

Vue使用vuex不走

1. 写vuex实例，应用的数据 改变数据的方法，
2. 在vue实例或者bue组件中获取vuex中数据 触发vuex中改变数据的方法
3. 把vue实例或者vue组件中的数据或者方法通过指令绑定到dom节点上

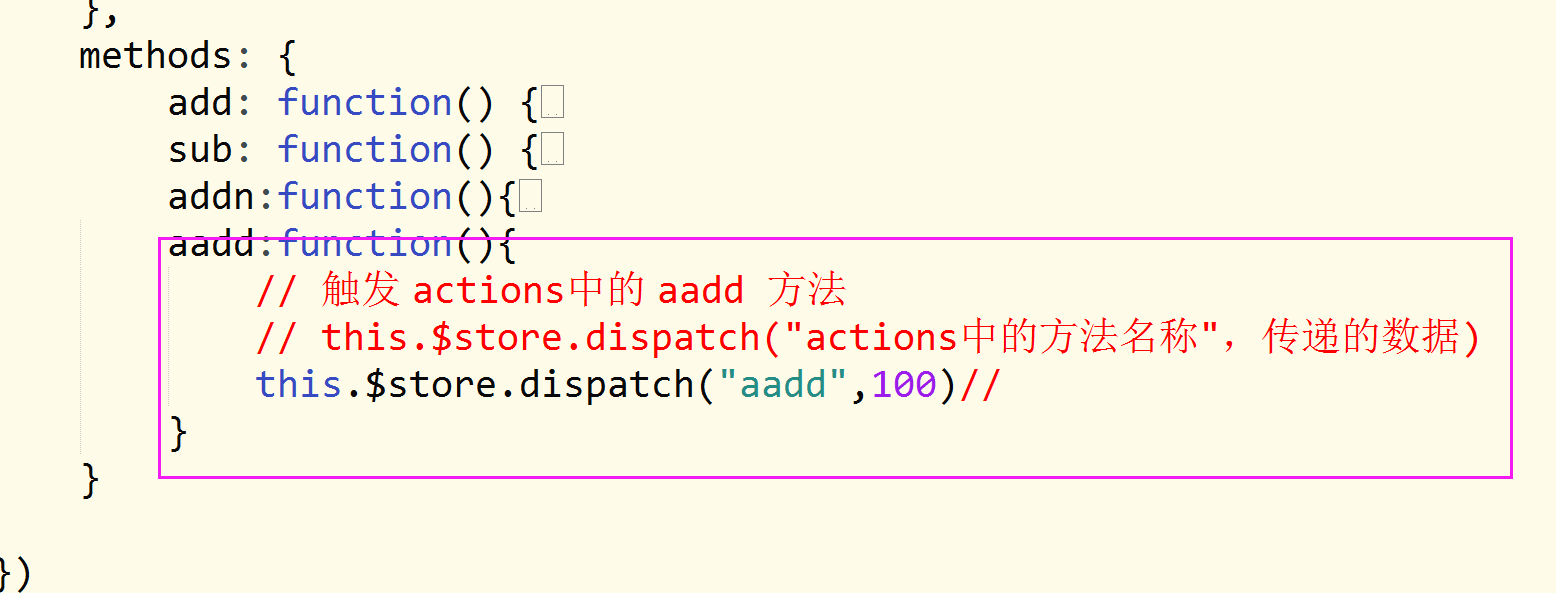
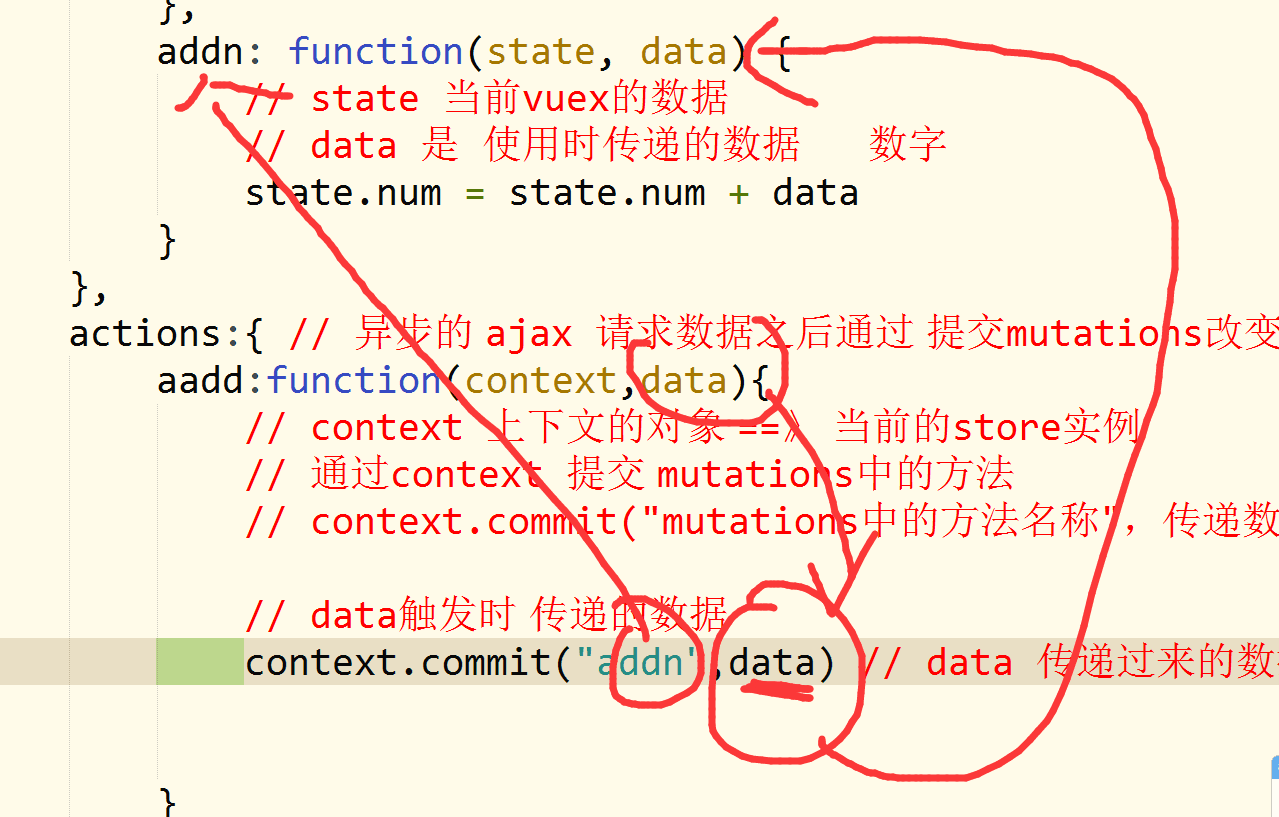
5-22

通过data获取vuex中的数据 只能获取一次 vuex中的数据改变 data中的值不变

Computed计算属性来获取数据 vuex中的数据改变computed中的值变化

Vuex中的methods中的函数data参数是commit传过来的数据 可以使用

Getters 相当于vuex中的计算属性（computed）getters:{numnew:function(state){return state.num\*10}} 在new Vue中的computed可以直接使用 和vuex中data数据使用相同



单页面如何展示多个页面的效果 通过vue-router实现

Vue-router组件和路径相互映射

控制激活路由样式 .router-link-exact-active{color:red;} 控制正在显示组件的router-link的 样式

5-28

Npm install axios --save

下载axios 方式 ；1.get

Axios({method:’get’,

Url:’请求的地址’,

Params:{

//Get方式发送的数据

Name:value

}

}).then(function(res){}//请求成功后的回调).catch(function(err){})//失败后的回调

Post

Axios({

Method:”post”,

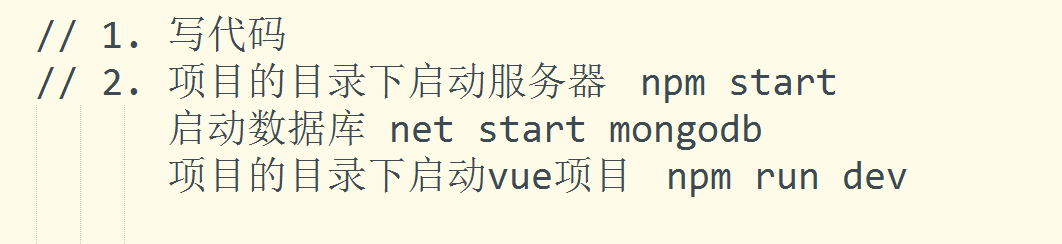
Url:””,

Data:{//发送的数据}

})

1. 由于不是vue提供 不能使用vue.use

有数据传递的时候的步骤



注意是在后台文件下面启动服务器 npm start（注意是后台文件）

是在cmd管理员身份下面启动数据库 net start mongodb

是在自己写的代码目录下面启动npm run dev 千万不要混淆 草！！！！

Sessionstorage存储用户信息

1. 在src下面的新建utils文件夹
2. 创建一个index.js用来写工具函数