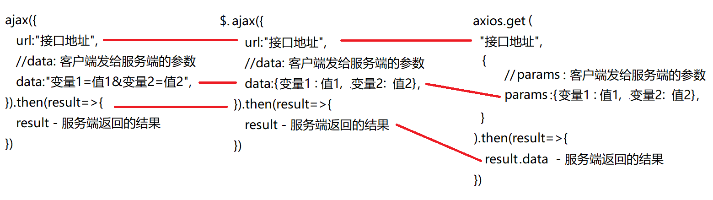
1. axios懵逼?

答:



2. 自定义组件的标签，可不可以写在new Vue({el:"#app"})所监控的div外部？

答：不行。因为<div id="app"外的部分，new Vue是扫描不到的。new Vue只能扫描<div id="app"内的内容。所以，new Vue就无法认出组件的标签。浏览器也不认识。

3. Error: Cannot find module 'C:\xampp\htdocs\dd\PROJECT\xz\routes\app.js

答: 路径错了。node app.js指令必须运行在xz根目录下。也就是必须保证输入node app.js前的路径为: "...\xz>"。要向不出错，都用右键直接单击app.js，选在终端中运行。再输入node app.js。

4. new Vue中发送axios请求时，如何获得组件component内的数据？

答：每个组件都自带created()，methods:{}，所以，组件自己要发的axios请求，应该写在每个组件内的created()或methods中，不应该写在外部。这样，在组件内的created()或methods中，就可以轻松使用this.变量，获得自己组件内的变量值，作为参数发给服务端了。没有障碍。

5. Vue如何跟踪调试变量值的变化:

答: watch。在watch中，添加对某个变量的监控函数。在监控函数中，输出变量值: console.log(this.xxx)

只要程序中变量发生变化，都会输出调试信息。

6. 组件的引入顺序

答: 子组件必须在父组件之前引入。

7. 组件中的data中的变量，methods中的函数等等也是自动升级为当前new Vue()平级吗？输出vm对象，看不见组件中的成员。

答: 每个组件都是一个独立的对象。它内部的数据和函数，都是专属的。外部不可能访问到！

子组件的内容，只能在子组件中用watch跟踪。在子组件外，不能用console.log方式输出子组件对象，因为找不到！

8. Vue中绑定事件时，什么时候加()，什么时候不加()

答: 其实Vue的事件绑定支持两种绑定方式:

1. 只用函数名绑定: @click="add"

对应的处理函数应该也不带参数！

methods:{

add(){ ... }

}

2. 其实也支持用js语句方式绑定:

@click="add(-1)" @click="add(+1)"

其中: add(-1) 和add(+1)是一条js语句而已！

对应的处理函数，也必须带参数:

methods:{

add(i){ ... }

}

而且事件处理函数以js语句方式绑定时，也支持变量绑定: @mouseover="changei(i)"

实际绑定时，这次i是0，changei(i)就会被临时换成changei(0)。如果下次i是1， changei(i)就会被临时换成changei(1)。

总结: 如果事件处理函数不需要传入参数时就不加！

如果事件处理函数需要传入参数时，就加()，()中写参数或变量！

9. 笔试题:

父组件中用到了一个子组件：

Vue.component("parent",{

template:`<div>

<h1>父组件</h1>

<child v-for="(item,i) of arr"

:key="i" ></child>

</div>`,

data(){

return { arr:[1,2,3] }

},

methods:{

handlerTest(item,e){

console.log(item);

console.log(e);

}

},

components:{

child

}

})

子组件中通过触发自定义事件向父组件传递一个数据:

var child={

template:`<div>

<h2>子组件</h2>

<button @click="handler">click</button>

</div>`,

methods:{

handler(){

this.$emit("test","something");

}

}

}

要求: 父组件的函数handlerTest中如何即输出item，又输出子组件传入的"something"字符串？

答: <child ... @test="handlerTest(item, $event)">

首先，子组件给父组件传参，需要用到自定义事件

2步:

1. 父元素在自己的<template>中的子元素标签<child>上用@自定义事件名="父组件处理函数"方式，将父组件的处理函数绑定在子组件的自定义事件上。

2. 子组件在单击按钮时，通过this.$emit("自定义事件名",参数值)方式，触发父组件绑在自己身上的自定义事件，并将参数值沿着事件触发的路线，顺便传递给父组件提前绑定的事件处理函数。

结果: 即使父组件中为子组件绑定自定义事件的处理函数后没有加()，子组件传的参数值，也会默认放在VUE框架自动生成的$event关键字中，作为事件处理函数的第一个参数默认传入父组件处理函数内。

父组件处理函数只需要在定义时，添加一个形参等着接就行了。

比如: 子组件中: 触发名为test的自定义事件，自动传递数据"something"到父组件处理函数

this.$emit("test", "something")

父组件中:

<child @test="handlerTest">

methods:{

handlerTest(val){

console.log(val)

}

}

结果: handlerTest中的形参val自动接到的就是VUE在事件发生时自动创建的$event关键字，其中保存的就是子组件传来的"something"字符串。

问题: 这个题: 事件处理函数需要两个参数，第一个参数是来自于v-for的item变量，第二个参数才是子组件传来的事件参数值。

解决: 因为默认$event是作为第一个参数传入的。如果第一个参数不是$event，就不能用默认方式传值。而应该在绑定自定义事件时，显式写明每个参数的来源，比如:

@test="handlerTest(item,$event)"

这就是说，当事件发生时，处理函数的两个参数分别来自于item变量和VUE内置对象$event，而$event会自动收到子组件触发事件时，传来的参数值。