十八、Vue中监听属性值发生变化

当监听的是单个属性时

data: function() {

return {

counter: 0,

};

}

watch: {

counter: function(newV, oldV) {

console.log('counter change to %d from %d', newV, oldV);

},

}

当监听的是一个object对象时，object中属性值发生变化时不会触发回调函数，因为object指向的地址没有发生变化，所以监听对象发生变化时有以下三种方法

data: function() {

return {

counter: {

up: 0,

down: 0,

},

};

}

方法1：给object重新赋值，可以直接监听object

watch: {

counter: function(newV, oldV) {

console.log('counter change to %o from %o', newV, oldV);

},

}

方法2：使用deep参数

使用deep需要使用watch的完整形式：handler是监听回调函数，deep: true指定了不仅仅监听counter的变化，也监听其内部属性的变化，所以当counter.up或counter.down变化时才能出发handler回调。

watch: {

counter: {

handler: function(newV, oldV) {

console.log('counter change to %o from %o', newV, oldV);

},

deep: true,

}

}

比较方法1和方法2的优缺点

使用deep参数会为数据每一层都添加监听，当层级较深时比较耗费性能，而且Vue不能监听到属性的添加或删除。所以一般来说使用重新赋值的方式是较优的方案，但如果只是想监听内部嵌套数据的话，重新赋值就比较重了，

方法3：监听object对象中的某个属性

watch: {

'counter.up': function(newV, oldV) {

console.log('counter.up change to %d from %d', newV, oldV);

},

'counter.down': function(newV, oldV) {

console.log('counter.down change to %d from %d', newV, oldV);

},

}

1. angular中数据监听的方法。

angular中监听单个属性的变化。（https://segmentfault.com/q/1010000022139047/）

1. 有两种方法
2. 方法一：使用钩子函数ngDoCheck(此钩子函数在发生脏值检测时就会被调用，因此使用此方法效率低，随时都会触发此钩子函数。)
3. 方法二：使用angular官方引入了rxJS来解决属性监听问题
4. 如：在commonService中
5. public language = new BehaviorSubject<string>("tc"); //此处表示默认语言是"tc"
6. 当在头部组件中点击切换语言成英文时：
7. this.commonService.language.next("en") //此时表示将commonService的属性language设置成"en"
8. 点击切换语言之后，在home页面监听language的变化来切换语言:
9. ngOnInit(){
10. this.commonService.language.subscribe((data)=>{
11. //返回的值为切换后的值。
12. //此处执行切换语言操作。
13. });
14. }