

1. vuex工作原理

2. 双向绑定和vuex是否冲突

3. vuex是什么？怎么用？使用场景是什么？

状态管理模式。集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化。 场景：多个组件共享数据或跨组件传递数据

vuex的流程： 页面通过mapAction异步提交事件到action， action通过commit把对应参数提交到mutation， mutation会修改state中的对应的值。最后通过getter把对应值抛出去。在页面的计算属性中通过mapGetter来动态获取state中的值

4. mutation和action区别

mutation: 提交更新数据的方法，必须是同步的，如果需要异步用action， 每个mutation都有一个字符串的事件类型type和回调函数 action 异步操作 提交的是mutation 不直接更改状态 mutations: { increment (state) { // 变更状态 state.count++ } } actions: { increment (context) { context.commit('increment') } }

提交方式不同， action 是用this.\$store.dispatch('ACTION_NAME',data)来提交。mutation是用this.\$store.commit('SET_NUMBER',10)来提交。 接收参数不同， mutation第一个参数是state， 而action第一个参数是context， 其包含了{ state, // 等同于 store.state， 若在模块中则为局部状态 rootState, // 等同于 store.state， 只存在于模块中 commit, // 等同于 store.commit dispatch, // 等同于 store.dispatch getters, // 等同于 store.getters rootGetters // 等同于 store.getters， 只存在于模块中 }

5. vue中的ajax请求应该放在组件的methods中还是vuex的actions中

如果请求来的数据不是被其他组件公用，仅仅在请求的组件中使用，则不放在vuex

如果复用则放在vuex

6.vuex数据持久化

vuex里面存放的数据，页面一经刷新就会丢失 解决方法：存放在localStorage或sessionStorage里面，进入页面判断是否丢失，如果丢失去本地存储里面找

7. vuex的5个核心属性

state getters mutations actions modules

8.vuex中状态是对象时，使用时要注意什么？

因为对象是引用类型，复制后改变属性还是会影响原始数据，这样会改变state里面的状态，是不允许，所以先用深度克隆复制对象，再修改。

9.怎么在组件中批量使用Vuex的state状态？

使用mapState辅助函数，利用对象展开运算符将state混入computed对象中 computed: {
...mapState('price','number')} }

10. Vuex中要从state派生一些状态出来，且多个组件使用它，该怎么做，？

使用getter属性 相当Vue中的计算属性computed，只有原状态改变派生状态才会改变。

11. 怎么通过getter来实现在组件内可以通过特定条件来获取state的状态？

通过让getter返回一个函数，来实现给getter传参。然后通过参数来进行判断从而获取state中满足要求的状态。然后在组件中可以用计算属性computed通过this.\$store.getters.getTodoById(2)这样来访问这些派生状态。

12 在Vuex中使用mutation要注意什么。

mutation 必须是同步函数

13 在组件中多次提交同一个mutation，怎么写使用更方便。

使用mapMutations辅助函数在组件中这么使用

import { mapMutations } from 'vuex' methods:{ ...mapMutations({ setNumber:'SET_NUMBER', }) } 然后调用this.setNumber(10)相当调用this.\$store.commit('SET_NUMBER',10)

14 Vuex中action和mutation有什么相同点？

第二参数都可以接收外部提交时传来的参数。 this.\$store.dispatch('ACTION_NAME',data)和this.\$store.commit('SET_NUMBER',10)

15 在组件中多次提交同一个action，怎么写使用更方便。

使用mapActions辅助函数,在组件中这么使用

methods:{ ...mapActions({ setNumber:'SET_NUMBER', }) } 然后调用this.setNumber(10)相当调用
this.\$store.dispatch('SET_NUMBER',10)

16 Vuex中action通常是异步的，那么如何知道action什么时候结束呢？

在action函数中返回Promise，然后再提交时候用then处理

17 在模块中，getter和mutation接收的第一个参数state，是全局的还是模块的？

第一个参数state是模块的state，也就是局部的state。

18. 在组件中怎么访问vuex模块中的getter和state，怎么提交mutation和action

直接通过this.\$store.getters和this.\$store.state来访问模块中的getter和state 直接通过
this.\$store.commit('mutationA',data)来提交模块中的mutation 直接通过this.\$store.dispatch('actionA',data)来提交模块中的action

19. 用过vuex命名空间吗？如何用？为什么用？

默认情况下，模块内部的action、mutation和getter是注册在全局命名空间，如果多个模块中action、mutation的命名是一样的，那么提交mutation，action时，将会触发所有模块中命名相同的mutation，action 所以可以添加namespaced:true 避免这种问题 export default{ namespaced: true, state, getters, mutations, actions }

20. 怎么在带命名空间的模块内提交全局的mutation和action

将 { root: true } 作为第三参数传给 dispatch 或 commit 即可。 this.\$store.dispatch('actionA', null, { root: true })
this.\$store.commit('mutationA', null, { root: true })

21怎么在带命名空间的模块内注册全局的action？

参考答案

```
actions: { actionA: { root: true, handler (context, data) { ... } } }
```

22 在组件中怎么提交modules中的命名空间的moduleA中的mutationA

```
this.$store.commit('moduleA/mutationA',data)
```

23 怎么使用mapState, mapGetters, mapActions和mapMutations这些函数来绑定带命名空间的模块?

首先使用createNamespacedHelpers创建基于某个命名空间辅助函数

```
import { createNamespacedHelpers } from 'vuex'; const { mapState, mapActions } = createNamespacedHelpers('moduleA'); export default { computed: { // 在 module/moduleA 中查找 ...mapState({ a: state => state.a, b: state => state.b }) }, methods: { // 在 module/moduleA 中查找 ...mapActions([ 'actionA', 'actionB' ]) } }
```

24.Vuex的严格模式是什么,有什么作用,怎么开启?

在严格模式下，无论何时发生了状态变更且不是由 mutation函数引起的，将会抛出错误。这能保证所有的状态变更都能被调试工具跟踪到。

```
const store = new Vuex.Store({ strict:true, })
```

25.在v-model上怎么用Vuex中state的值?

需要通过computed计算属性来转换。

```
computed: { message: { get () { return this.$store.state.message }, set (value) { this.$store.commit('updateMessage', value) } } }
```