Vuex状态管理

软通大学
ISOFTSTONE UNIVERSITY

本节目标

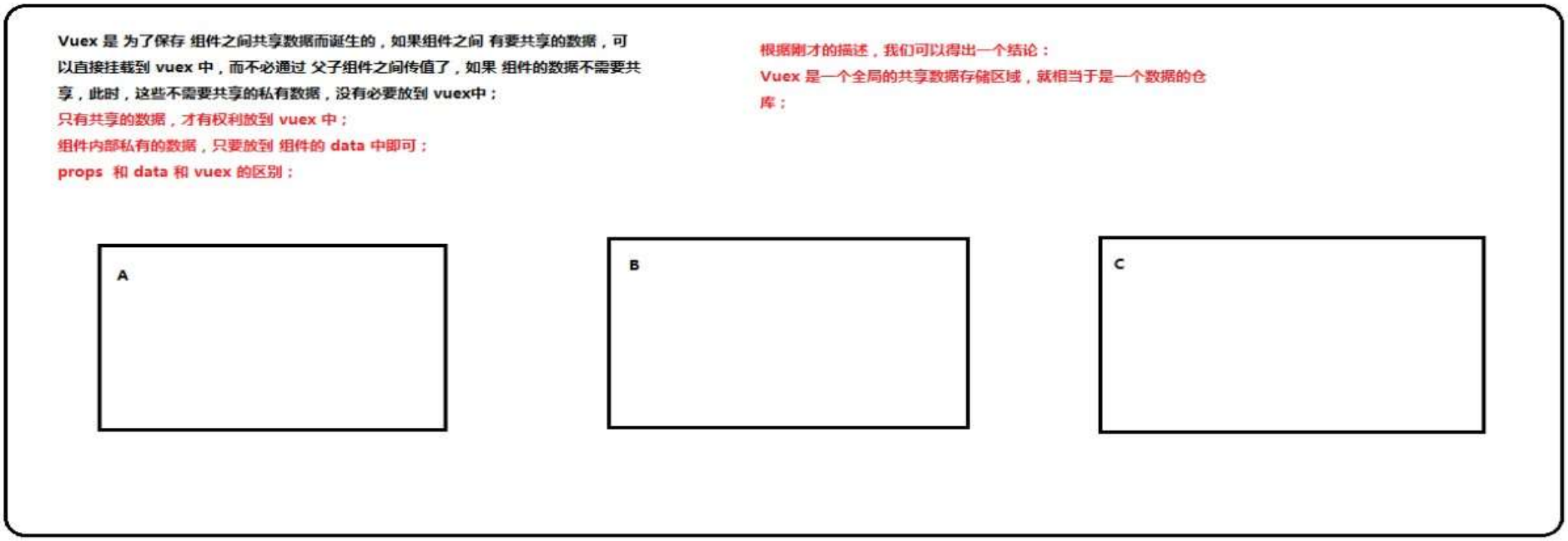
- 掌握Vuex的运用

目录 CONTENTS

01 Vuex介绍

Vuex介绍

vuex是Vue配套的公共数据管理工具，它可以把一些共享的数据，保存到 vuex 中，方便 整个程序中的任何组件直接获取或修改我们的公共数据;



02 Vuex的基本使用

Vuex的基本使用

Vuex介绍所述: 创建三个组件： App.vue， App.vue包含2个子组件B.vue,C.vue.

Vuex的基本使用:

1.安装vuex

cnpm i vuex –D

2.导入vuex并注册全局使用

import Vuex from "vuex“

Vue.use(Vuex)

3.创建new Vuex.Store() 创建数据仓储对象,并将其绑定到vm实例上

```
var store = new Vuex.Store({
  state:{ count:0 },
  mutations:{
  subtract(state, obj) {
      console.log(obj)
      state.count -= (obj.c + obj.d)
    }
  }
})
```

(创建store对象)

```
new Vue({
  render: h => h(App),
  //store对象挂载到vm实例中
  store
}).$mount('#app')
```

(挂载store对象到vm实例上)

Vuex的基本使用

Vuex的store对象:

- 1: state属性:需要共享的数据都定义在state属性对象中, 组件中，想要访问store中的数据：**this.$store.state.***,**比如count
- 2：mutations属性:如果要操作 store 中的 state 值，通过mutations 属性对象定义的方法，这是vuex的规范， 组件中想要调用 mutations 中的方法：**this.$store.commit('方法名'),**比如subtract.
- 3: mutations属性中定义的方法，**最多支持2个参数**，第一个参数永远是state属性；第二个参数由调用者传入,参数类型不限

03 store数据显示与mutations数据提交

store数据显示与mutations数据提交

store对象中的state属性定义的就是共享数据，组件中，通过this.$store.state.*访问**

```
<template>
  <div>
    <h3>B组件</h3>
    {{this.$store.state.count}}
  </div>
</template>
```

(所有组件都可以访问共享数据)

```
<div>
    <h3>C组件</h3>
    <input type="button" value="减少" @click="remove" />
    <input type="button" value="增加" @click="add" />
    <br />
    <input type="text" v-model="this.$store.state.count" />
 </div>
.....
methods:{
    add(){
        this.$store.commit("increment");
    }
    remove(){
      var obj = { c: 1, d: 1 };
      this.$store.commit("subtract", obj);
    }
}
```

1:操作state中的数据，不能直接赋值，必须通过mulations属性定义的方法
2:mutations属性的方法只能支持2个参数，第一个方法名，第二个调用传入，如果需要传递复杂的数据，可以用对象的形式作为参数传递

store数据显示与mutations数据提交

Vuex.store()对象的mutations定义

```
var store = new Vuex.Store({
  state:{
    count:0
  }
  mutations:{
    increment(state){
      state.count++
    },
    subtract(state, obj) {
      console.log(obj)
      state.count -= (obj.c + obj.d)
    }
  }
})
```

*: 由于更改的是共享数据，当C.vue中的add和remove方法提交后，会发现B.vue和C.vue中的数据同步更改了

03 getters提供包装数据

• getters提供包装数据

当访问state中的数据，不是直接显示，而是需要包装一下，比如加一些字符前缀提示，组件中显示如:”当前的count值是; 1”, 如果在每一个组件中都去添加这个字符串，无疑就显得很愚蠢了，store对象中，通过定义gettters方法，可以将state中的数据包装后对外提供，类似于Vue中的过滤器功能.

store对象定义gettters属性:

```
var store = new Vuex.Store({
  ...
 getters:{
   optCount:function(state){
      return '当前最新的count值是：' + state.count
   }
 }
})
```

1:getters只负责 对外提供数据，不负责修改数据，如果想要修改 state 中的数据,请使用mutations

2:访问getters方式:this.$store.getters.***

3:getters和computed 比较像， 只要 state 中的数据发生变化了，那么，如果 getters正好也引用了这个数据，就会立即触发getters的重新求值

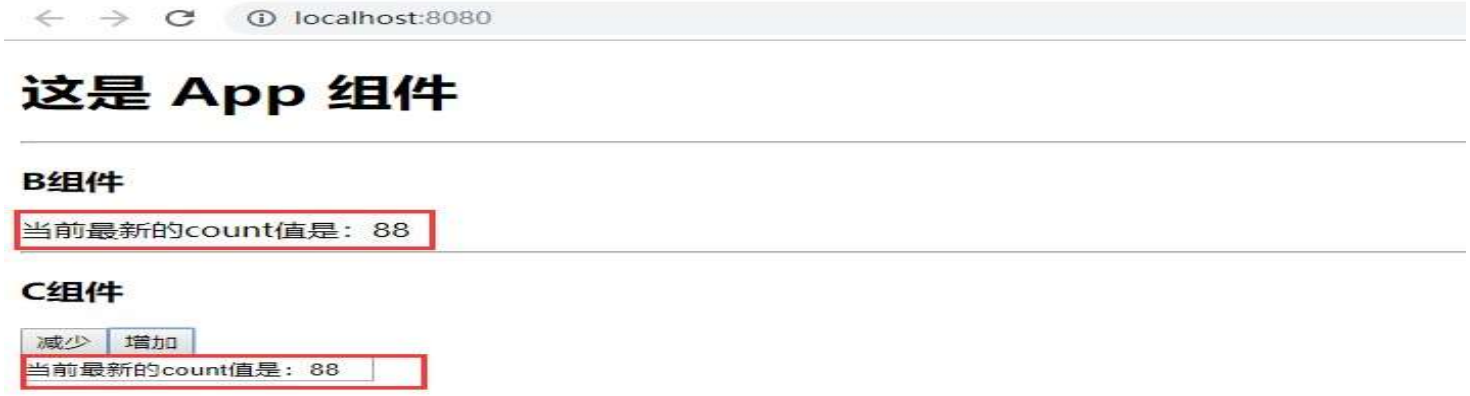
• getters提供包装数据

B.vue和C.vue中访问getters获取包装后的数据

```
<div>
  <h3>B组件</h3>
  {{this.$store.getters.optCount}}
 </div>
```

*：c组件中的add和remove方法提交,state中的数据发生了变化，而getters的optCount引用了count，所以B组件和C组件的数据会同步变化。

```
<div>
  <h3>C组件</h3>
  <input type="button" value="减少" @click="remove" />
  <input type="button" value="增加" @click="add" />
  <br />
  <input type="text" v-model="this.$store.getters.optCount" />
 </div>
```



本节总结

- Vuex的概念
- Vuex的数据共享与数据操作提价
- Vuex提供包装过滤数据

本节练习

- 课堂案例: 实现课堂案例中的A，B, C多组件共享数据？

软通大学
ISOFTSTONE UNIVERSITY