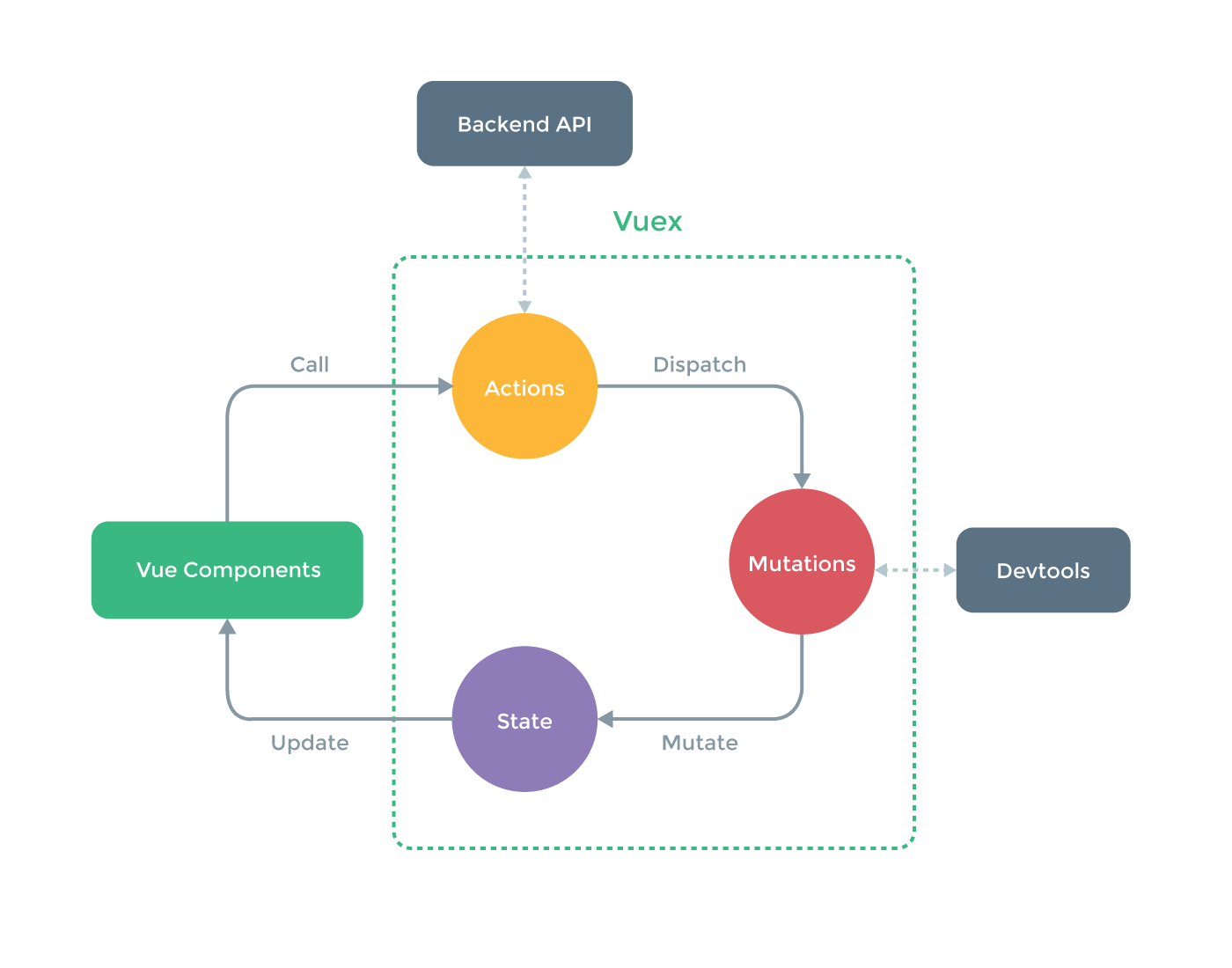
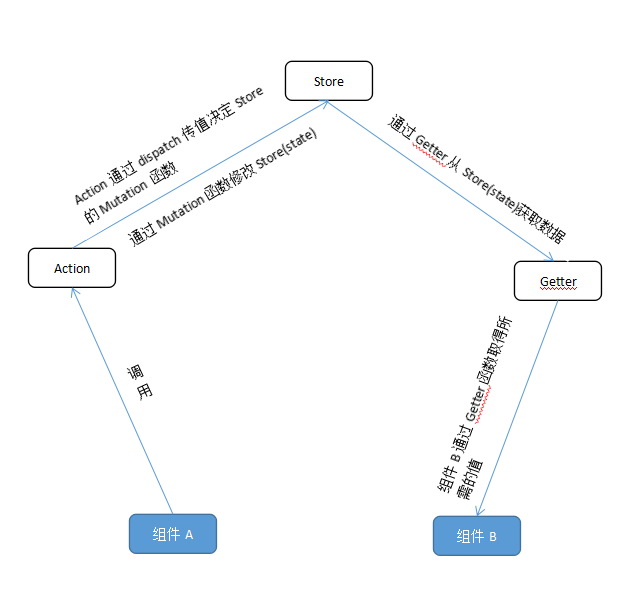
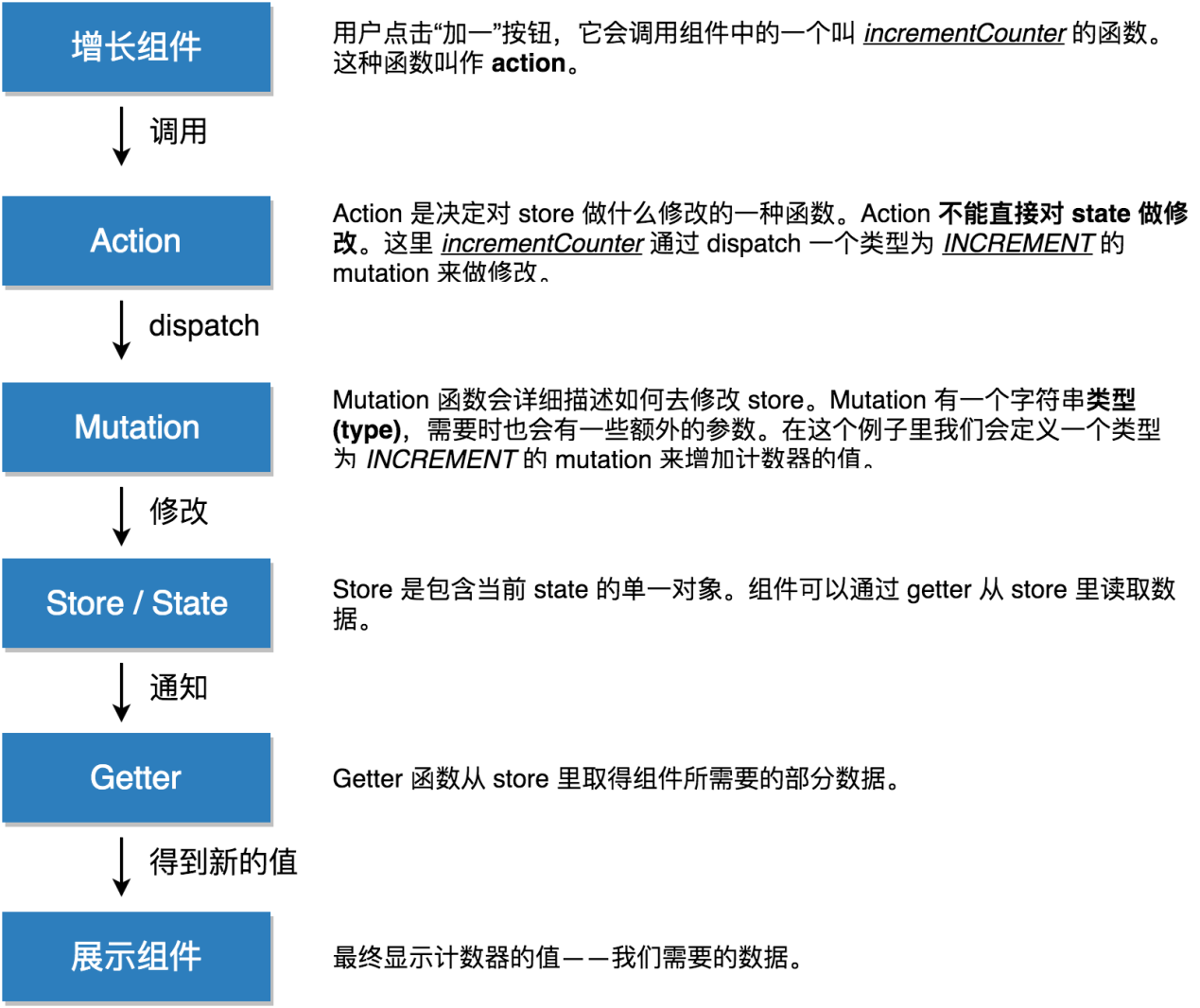
总思路：







从获取消息列表数据来说明一下执行顺序

参考：

<http://www.tuicool.com/articles/vQZbYn7>

<http://zhangdiweb.com/2016/09/08/Vuex%E2%80%94%E2%80%94Vue%E5%8D%95%E9%A1%B5%E5%BA%94%E7%94%A8%E7%8A%B6%E6%80%81%E7%AE%A1%E7%90%86%E6%9E%B6%E6%9E%84/>

<http://www.html-js.com/article/3663>

1. store.js使用Vue.use(Vuex)之后export new Vuex.Store某个module如modules/messageList
2. messageList/messageList.js引入import store from '../../vuex/store'，然后在new Vue载入store，相当于注入根组件，使得子组件通过this.$store实现组件之间通讯
3. messageList.vue中created方法使用this.$store.dispatch('getMessageList')（mutations 也可以通过这个来触发）触发actions中的getMessageList
4. messageList/messageList.js中actions中getMessageList方法使用commit(types.MESSAGE\_GET\_LIST, dataStr)提交mutation
5. mutation中方法[types.MESSAGE\_GET\_LIST] (state, res) {}赋值给state中的banner
6. messageList.vue中使用mapGetters调用方法名为getMessageListGetstate的getters赋值后的新数据
   1. computed: {  
      mapGetters**(**{  
       **banner**: **'getMessageListGet'** }**)**},
7. 最后前端使用渲染返回的数据
   1. <**ul**>  
       <**li v-for='item in banner'**>{{item}}</**li**>  
      </**ul**>

附注computed加上更多方法的话可以这样加上

* + 1. computed: {  
        a **()** {  
        **return 'a'** },  
        ...mapGetters**(**{  
        **banner**: **'getMessageList'** }**)**},

完整代码：

vuex/module/messageList.js

// 主页  
**import** \* **as** types from **'../mutation-types'  
const** state = {  
 banner: **[]**,  
 banner1: **[]**}  
  
**const** actions = {  
 // 获取banner列表  
 getMessageList: **function (**{commit}**)** {  
 **let** dataStr = **[**{**'num'**: **'1'**}, {**'num'**: **'2'**}**]  
 console**.log**('b')** console.log**(**state**)** commit**(**types.MESSAGE\_GET\_LIST, dataStr**)**;  
 }  
}  
**const** getters = {  
 getMessageListGet: state => state.banner  
}  
  
**const** mutations = {  
 // es6使用变量作为方法名  
 **[**types.MESSAGE\_GET\_LIST**] (**state, res**)** {  
 console.log**('c')** console.log**(**res**)** state.banner = res  
 }  
}  
  
**export default** {  
 state,  
 actions,  
 getters,  
 mutations  
}

Vuex/store.js

**import** Vue **from 'vue'  
import** Vuex **from 'vuex'  
import** messageList **from './modules/messageList'**Vue.*use***(**Vuex**)  
  
export default new** Vuex.*Store***(**{  
 modules: {  
 messageList  
 },  
 strict: process.env.NODE\_ENV !== **'production'**// 线上环境关掉  
}**)**

Vuex/mutation-types.js

// 获取消息列表数据  
**export const** MESSAGE\_GET\_LIST = **'MESSAGE\_GET\_LIST'**

Views/messageList/ messageList.js

**import** { Vue, $, Common } from **'js/base'  
import** myHeader from **'../../components/my/my-header.vue'***require***('./scss/messageList.scss')  
import** myCenter from **'./messageList.vue'  
import** store from **'../../vuex/store'  
var** homeVue = **new** Vue**(**{  
 **el**: **'#messageList'**,  
 store,  
 **template**: **'<div class="pageview"><my-header></my-header><my></my></div>'**,  
 **components**: {  
 **'my-header'**: myHeader,  
 **'my'**: myCenter  
 }  
}**)**

Views/messageList/ messageList.vue

<**template**>  
 <**div class="container padding\_t\_122 messageList"**>  
 <**ul**>  
 <**li v-for='item in banner'**>{{item}}</**li**>  
 </**ul**>  
 <**ul class="bg\_white"**>  
 <**li v-for="list in listData"**>  
 <**a href="javascript:void(0)"**>  
 <**h1 class="top flex flex\_v\_center"**>  
 <**div class="title flex\_item"**><**span data-count="1"**>{{list.title}}</**span**></**div**>  
 <**div class="time"**><**time**>{{list.create\_time | toFriendlyTime}}</**time**></**div**>  
 </**h1**>  
 <**p class="detail"**>{{list.content}}</**p**>  
 </**a**>  
 </**li**>  
 </**ul**>  
 <**mugen-scroll :handler="fetchData" :should-handle="!loading"**>  
 <**list-bottom ref="listBottom"**></**list-bottom**>  
 </**mugen-scroll**>  
 </**div**>  
</**template**>  
<**script**>  
**import** { Vue, $, Common, wx } **from 'js/base'  
import** MugenScroll **from 'vue-mugen-scroll'  
import** listBottom **from '../../components/common/list-bottom.vue'  
import** filters **from '../../assets/js/filters'  
import** {mapGetters} **from 'vuex'  
export default**{  
 data **()** {  
 **return** {  
 **listData**: **[]**,  
 **loading**: **false**,  
 **currentPageCount**: 1,  
 **pageCount**: 0  
 }  
 },  
 mounted: **function ()** {  
 **this**.**$bus**.emit**('setCenterHeader'**, {**title**: **'消息中心'**, **left**: {**className**: **'back'**}, **'right'**: {**hide**: **true**}}**)** },  
 created: **function ()** {  
 **if (this**.**banner**.**length** === 0**)** {  
 **console**.log**('a')  
 this**.**$store**.dispatch**('getMessageList')** }  
 },  
 computed: {  
 a **()** {  
 **return 'a'** },  
 ...mapGetters**(**{  
 // 在 mapState 里面我们既可以调用 store 的 state ，也可以调用 store 的 getters  
 **banner**: **'getMessageListGet'** }**)** },  
 **components**: {  
 MugenScroll,  
 listBottom  
 },  
 **filters**: {  
 **toFriendlyTime**: filters.toFriendlyTime// 注册filters  
 },  
 methods: {  
 fetchData **()** {  
 **if (**!**this**.**loading)** {  
 **this**.**loading** = **true**;  
 **if (this**.**currentPageCount** > **this**.**pageCount** && **this**.**pageCount** !== 0**)** {  
 **this**.finishAction**()  
 return false** }  
 **this**.**$http**.get**('../../../static/api/center/getNotification.json')**.then**((**response**)** => {  
 console.log**(**response.**body)  
 if (**response.**body**.notification\_list.length === 0 || response.**body**.total\_index === 0**)** {  
 **this**.finishAction**()  
 return false** }  
 setTimeout**(()** => {  
 **for (let** i = 0; i < response.**data**.notification\_list.length; i++**)** {  
 **this**.**listData**.push**(**response.**data**.notification\_list**[**i**])** }  
 **this**.**loading** = **false** // 不明白为什么要提到外面,不提到这里会不断请求不止  
 **this**.**pageCount** = Common.index2PageCount**(**response.**data**.total\_index**)  
 this**.**currentPageCount**++  
 }, 3000**)** }, **(**response**)** => {  
 **console**.log**(**response**)** }**)** }  
 },  
 finishAction **()** {  
 **this**.**loading** = **true  
 this**.**$refs**.listBottom.**$el**.**children[**0**]**.**classList**.add**('finished')  
 this**.**$refs**.listBottom.**$el**.**children[**0**]**.**textContent** = **this**.**$refs**.listBottom.**$data**.finishText  
 }  
 }  
}  
</**script**>

注意：如果要在对应组件想获取对应更新后的数据的话，那么要在state要有一个存在的参数（即使为空也要定义）与my-header.vue对应，否者获取都是undefined

common.js

**const** state = {  
 headerTitle: '我的'  
}

messageList.vue(与my-header.vue为兄弟组件)

mounted: **function** () {  
 **this**.$store.dispatch('setPageInfo', {  
 headerTitle: '消息中心',  
 left: {className: 'back'},  
 'right': {hide: **true**}  
 })  
},

my-header.vue

computed: {  
 setLeftHeader: **function** () {  
 console.log(**this**.$store.getters.getPageInfo.left)// undefined  
 },  
 ...mapGetters({  
 pageInfo: 'getPageInfo'  
 })  
},

所以common.js要改为

**const** state = {  
 headerTitle: '我的',  
 left: '',  
 right: ''  
}

但是这会导致出现两次结果

一次为空，一次为你更新后的数据