**Promise相关的demo**

1. vuex中的action.js中用到

**组合 Actions**

1．Action 通常是异步的，那么如何知道 action 什么时候结束呢？更重要的是，我们如何才能组合多个 action，以处理更加复杂的异步流程？

2. 首先，你需要明白 store.dispatch 可以处理被触发的action的回调函数返回的Promise，并且store.dispatch仍旧返回Promise：

1. actions: {
2. actionA ({ commit }) {
3. return new Promise((resolve, reject) => {
4. setTimeout(() => {
5. commit('someMutation')
6. resolve()
7. }, 1000)
8. })
9. }
10. }

你可以：

1. store.dispatch('actionA').then(() => {
2. // ...

})

在另外一个 action 中也可以：

actions: {

// ...

actionB ({ dispatch, commit }) {

return dispatch('actionA').then(() => {

commit('someOtherMutation')

})

}

}

最后，如果我们利用 [**async / await**](https://tc39.github.io/ecmascript-asyncawait/) 这个 JavaScript 即将到来的新特性，我们可以像这样组合 action：

// 假设 getData() 和 getOtherData() 返回的是 Promise

actions: {

async actionA ({ commit }) {

commit('gotData', await getData())

},

async actionB ({ dispatch, commit }) {

await dispatch('actionA') // 等待 actionA 完成

commit('gotOtherData', await getOtherData())

}

}

二

promise中套一个promise的时候是否需要，还会执行将两个promise执行完，再次继续执行，是否执行完取决于reslove()的位置

例子在SDK的entrance.js中



wcm