**Redux**

中间件

1. redux-logger:能够在控制台清晰的展示出当前redux操作的流程和信息（有有状态、派发信息、修改后的状态信息）
2. Redux-thunk(瑞达斯-栈可)：处理异步的dispatch派发
3. Redux-promise:在dispatch派发的时候支持promise操作

(一般读数据的时候，派发完了再读)

**let** store=createStore(reducer,applyMiddleware(reduxLogger,reduxThunk,reduxPromise));

**thunk**

使用语法：在指定执行派发任务的时候，等待300ms后再派发

**return** dispatch=>{

**//dispatch都传给我们了，我们想什么时候传递，自己搞定即可**   
 setTimeout(()=>{  
 dispatch({  
 type:TYPES.CUSTOM\_CREATE,  
 payload  
 })  
 });  
}

**Promise**

**return**{  
 type:TYPES.CUSTOM\_CREATE,  
 //传递给reducer的payload需要等待promise成功，把成功的结果传递过去  
 playload:**new** Promise(resolve => {  
 setTimeout(()=>{  
 resolve(payload);  
 },3000)  
 })  
}