1. 未登录（已登录）状态下是如果保存用户信息的（例如：未登录下的购物车信息）
   1. 未登录：同一台设备的相同浏览器，页面刷新，购物车信息也不会消失
      1. 只往redux中存储不行：因为redux在页面刷新的时候会清空里面的内容
      2. 方案一：基于本地存储解决（cookie localStorage）(兼容问题、大小问题、周期、杀毒问题、cookie不是严格的本地存储，当服务器创建一个session后，服务器会在响应头，把当前客户端和。。。)
         1. 点击加入的时候，把商品信息，存放到本地存储中（移动端一般都用localStorage.setItem,get,remove）(cookie是document.cookies)
         2. 需要展示或者验证的时候，只要从本地获取到这些信息即可，存储到本地的信息都是字符串，使用的时候注意内容格式转换
         3. **弊端：不安全、不稳定容易被手动修改**
      3. 方案二：基于服务器的session存储
         1. 点击加入的时候，基于api向服务器发送请求，把增加的商品信息传递给服务器，服务器接收到请求后把当前商品信息使用session存储起来（和客户端创建了SID）。。。
         2. 如果后期想获取购物车的信息，直接再通过api从服务器获取即可
         3. **弊端：对服务器操作频繁，性能不是最好的**
         4. **方案二的优化：基于redux来增加上述操作的性能**
            1. **创建redux容器，容器中存储购物的信息**
            2. 点击加入的时候，除了向服务器发送请求让服务器存储之外，同时执行一个dispatch派发，把当前加入购物车的信息向redux中也存储一份（也可以从服务器获取最新的购物信息放到redux中，这种模式宝恒服务器和redux存储的信息实时同步）
            3. 从购物车移除也要从服务器和redux中各自移除
            4. 页面不刷新，都是获取redux中的信息展示校验
            5. 页面刷新，首先从服务器获取到这些信息，然后再次dispatch派发存储到redux中一份。。。
   2. 已登录：同一个账号不同平台下，信息也能共享“数据库存储（和用户账号是关联的）”
      1. 未登录下存储的购物车信息，在登录的时候，也要整体迁移到数据库中存储
         1. 未登录下存储的购物车信息，在登录的时候，也要整体迁移到数据库中存储
            1. 如果客户端是基于本地存储实现未登录信息存储，则登录成功后，需要客户端基于api把本地存储的信息发送给服务器，服务器存储到数据库中
            2. 如果是基于session处理的未登录，服务器则自己在登录成功后，把之前session存储的购物车信息放到数据库中
      2. 剩下的加入购物车等具体操作和session方案一样，客户端不要调整什么，还是向服务器发一份，向redux发一份，页面刷新从服务器获取最新信息存储到redux中
   3. 登录后存储个人信息或者登录态（验证是否登录），也可以参照购物车模式：

方案一：登录成功后从服务器获取个人信息，把信息存储到本地（本地存储方案解决），以后需要验证是否登录或者获取个人信息，值需要从本地获取验证即可

方案二：服务器session存储，登录成功后，后态把基本信息等存储到session找中，客户端想要获取或者验证是否登录，向服务器发送api完成校验即可（弊端：对服务器操作频繁）

优化：登录（注册）成功后，把redux中存储一份个人信息和是否已登录信息（同时服务器sessin中也要存），特殊注意，如果是退出登录了，我们也要把服务器和redux中的信息清空掉才可以，以后再验证是否登录，直接redux获取即可，页面刷新从服务器获取信息

**考察的知识点（本地存储、性能优化、redux一套：基础知识和原理）**

1. 基于react做移动端的webapp项目，是如何调取原生接口实现一些底层功能（例如调取微信的api）（Hbride）

JS-SDK接口js文件不支持ES6Module和CommonJS模块规范，不能直接在react中导入进来，需要我们这样处理：public/

1. 把js文件在public/index.html中单独的导入
2. –public
3. -js
4. -jwexin-1.2.0.js
5. -index.html

在html的head中导入这个js：<script src=”js/weixin-1.2.0.js”></script>

在src的某一个组件中基于window.wx获取到这个对象，调取微信的一些方法实现某些功能

=>Hybrid

=>AMD/CMD/Commonjs/ES6Module模块规范