Cart谷粒购物车

# 1 介绍

|  |
| --- |
| 1 购物车在不登陆的情况下，也可以使用  需要引入对浏览器cookie的操作  2 购物车在登录的情况下，需要使用mysql和redis来存储数据  Redis作为购物车的缓存  3 在缓存情况下，或者用户已经添加购物车后，允许购物车中的数据和原始商品数据的不一致性  4 购物车同步问题  什么时候同步(结算、登录)  同步购物车后，是否需要删除cookie数据  5 用户在不同的客户端同时登录  如何处理购物车数据 |

用户

Cart

Db

Pms\_sku\_info

# 2 购物车添加功能

|  |
| --- |
| 1 传递参数(商品skuid，添加数量)  2 根据skuid调用skuService查询商品的详细信息  3 将商品详细信息封装成购物车信息  4 判断用户是否登录  5 根据用户登录决定走cookie的分支还是db  **if**(StringUtils.*isBlank*(memberId)){  *// 用户没有登录*  Cookie：response.addCookie(cookie) request.getCookies();  Cookie：的跨域问题,setDomain(),getDomain() item.gmall.com🡪.gmall.com}**else**{  *// 用户已经登录*  Db+cache}  6 购物车的类型  Db ： cartListDb（有主键和用户id）  Cookie : cartListCookie（没有主键和用户id）  Redis : cartListCache（有主键和用户id） |

Mysql

商品

Redis

添加购物车

用户已登录

用户未登录

同左侧

Cookie有值

Cookie无值

更新

添加

同步缓存

覆盖cookie

服务器

Cookie

Cookie副本

覆盖浏览器

服务器

地址

Session



# 3 购物车的缓存结构

|  |
| --- |
| 1 存储的是购物车集合  2 键：用户id  3 购物车缓存中的某一个购物车数据的更新  X如果用set kv 取出json，转化成集合，从集合中取出对象，修改对象，放回集合，集合放回缓存  V使用hash进行存储(方便查询和修改用户购物车集合中的某一个单独的购物车对象)  Mapkey  Key value  Key value  userId  skuId cart  skuId cart    User:memberId:cart |

# 4 购物车列表

|  |
| --- |
| 1 购物车列表数据从缓存中取  2 购物车页面展示    3 修改选中状态  A ajax异步+内嵌页  B修改数据库的购物车状态  C数据库修改后同步缓存 |

服务器

列表：选中状态

内嵌页面

# 5 价格的运算

