Gmall

单点登录&购物车

# 单点登录（SSO）

SSO英文全称Single Sign On，单点登录；

SSO是在多个应用系统中，用户只需要登录一次就可以访问所有相互信任的应用系统。

1）、任何系统都必须去登陆服务器进行登录

2）、服务器就记住了登录状态

3）、其他系统访问受保护资源，需要再次登录，跳转到sso\_server登录的时候，服务器告诉客户端，已经登录过，无须登录。登录过得信息

Sso\_server

8080

video

8082

news

8081

## 1）、单点登录框架调试

<https://gitee.com/xuxueli0323/xxl-sso>

<http://www.xuxueli.com/xxl-sso/#/>

XXL-SSO 是一个分布式单点登录框架。只需要登录一次就可以访问所有相互信任的应用系统。 拥有"轻量级、分布式、跨域、Cookie+Token均支持、Web+APP均支持"等特性。现已开放源代码，开箱即用。

### 1）、下载框架代码

默认规则：

单点登录服务器的地址应该是：

<http://xxlssoserver.com:8080/xxl-sso-server> 服务器

/8081/xxl-sso-web-sample-springboot web系统1

/8082/xxl-sso-web-sample-springboot web系统2

### 2）、修改各个项目中redis地址信息

### 3）、为系统添加以下域名映射。

域名只要不同：cookie不能共享，跨域还需配置。

127.0.0.1 xxlssoserver.com

127.0.0.1 xxlssoclient1.com

127.0.0.1 xxlssoclient2.com

## 4）、启动三个服务

java -jar xxl-sso-server-1.1.1-SNAPSHOT.jar

java -jar xxl-sso-web-sample-springboot-1.1.1-SNAPSHOT.jar

java -jar xxl-sso-web-sample-springboot-1.1.1-SNAPSHOT.jar –server.port=8082

## 5）、访问服务测试

访问 <http://xxlssoclient1.com:8081/xxl-sso-web-sample-springboot/> 需要登录

访问<http://xxlssoclient2.com:8082/xxl-sso-web-sample-springboot/> 需要登录

但是他们都重定向到了 <http://xxlssoserver.com:8080/xxl-sso-server/> 统一认证中心

只要有一个人登录

## 6）、观察核心

每一个应用下都有一个相同的cookie。单点登录的核心就是不同系统之间同步cookie即可。

## 7）、手动编写一个SSO-server

# 购物车

购物车：

在线购物车；

离线购物车；

离线购物车：

只要每次看购物车。都有一个**user-key=uuid**；如果这个user-key丢了，购物车就没了。

如果我们要看购物车肯定会带上这个user-key的cookie；如果有就用自带的，没有就新建一个以后用这个。

登录以后离线的和在线的合并了，并且清空了离线购物车里面的数据。

京东怎么做？

购物车有一个cookie对应的key关联，即使没有登录也有，登录以后进行购物车合并，合并完成后删除购物车数据；

以后我的系统也设计离线在线购物车。

离线购物车：永远用 cart-key=uuid；购物车数据都在redis中提升系统的性能。cart:temp: uuid；

在线购物车：cart:1={};放在redis中；cart:user:1；

1）、购物车数据保存在redis中，使用分布式集合；【redisson.getMap】

2）、用户对于购物车的所有操作，都需要传入cart-key，如果用户登录了，还需要传入token

购物车需要提供的所有方法；

1）、添加到购物车

2）、修改购物项

3）、删除购物项

4）、选中/选不中购物项

5）、返回整个购物车

6）、点击去结算-获取购物车中需要结算的数据

技术：

1）、分布式集合

2）、JSON.parseObject(str, new TypeReference<Set<String>>() { });

3）、RPC隐式传参，透传用户令牌给其他服务。一般用于给其他服务传递一些增量数据（这些数据其他服务不一定会用到）。