今天项目做到了基于redis的单点登录登出设计，所以有必要记录一下。

**重点知识，面试。**

**涉及到的知识点：token，UUID，userID（模仿微信的扫码登录），cookie，redis等**

## 什么是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。它包括可以将这次主要的登录映射到其他应用中用于同一个用户的登录的机制。它是目前比较流行的企业业务整合的解决方案之一。

## 原来的登录逻辑实现



### 问题

单台tomcat，以上实现是没有任何问题的，但是我们现在是集群的tomcat，就会存在session共享问题。

只要解决session共享问题，登录问题即可解决。

订单系统

用户中心

protal

session

session

session

每个系统都有自己的session，不能统一。

### 解决session共享问方案

1. tomcat的session复制

优点：不需要额外开发，只需要搭建tomcat集群即可。

缺点：tomcat 是全局session复制，集群内每个tomcat的session完全同步（也就是任何时候都完全一样的) 在大规模应用的时候，用户过多，集群内tomcat数量过多，session的全局复制会导致集群性能下降， 因此，tomcat的数量不能太多，5个以下为好。

1. 实现单点登录系统，提供服务接口。把session数据存放在redis。

Redis可以设置key的生存时间、访问速度快效率高。

优点：redis存取速度快，不会出现多个节点session复制的问题。效率高。

缺点：需要程序员开发。

订单系统

用户中心

protal

redis

session

## 单点登录系统的流程



**那么登出就是和登录相反的逻辑。先网页request，然后从 request遍历所有cookie，看是否有一个名为token的cookie，有则取出value值(即token)，然后删掉，然后去redis把这个token&value键值删掉即可。从而实现登出功能。**

**那如果要单点登录同步登出呢？比如李4去别的地方登录，要踢掉张三已经登录的号，那么需要用到以下的方法：**

**redis.put(token,userId);//存储一条对应用户信息的token**

**redis,put("your setting",token);//再存储一条对应userId的唯一标识，your setting可以写成loginSys+userId**

为什么要这么存，正常的思路是只存上面一条就行了，稍后解答。

我们把token返回，response.addCookie(new Cookie("my cookie",token));//**这里的my cookie可以写成loingSysCookie**

流程实现是这样的,需要设置过期时间等等参数，这里就不写了。

这时候只要是同一浏览器，就会带着这个cookie，访问该项目下的子项目(父级域名下的子级域名,比如www.baidu.com和mp3.baidu.com的关系)，

进入登录系统，拿到token，进到redis中比对，是否有用户就可以了，如果有，就不需要再登录了。

下面说一下上面的疑问：

当我再一次登录系统的时候，**先去找redis中的redis,get("your setting")是否为空，已经在别处登录过的用户,可以取到他的token**

**redis.del(token),这样就实现了单点登录，同步登出。**

之后再设置redis和cookie，方法同上。

**很多没有redis中放第二条，就无法实现单点登录同步登出，虽然设置了过期时间，但用户体验和安全性还是差一点。**