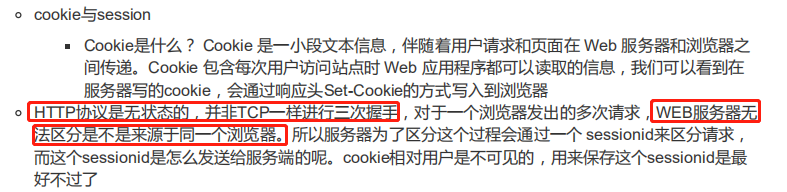
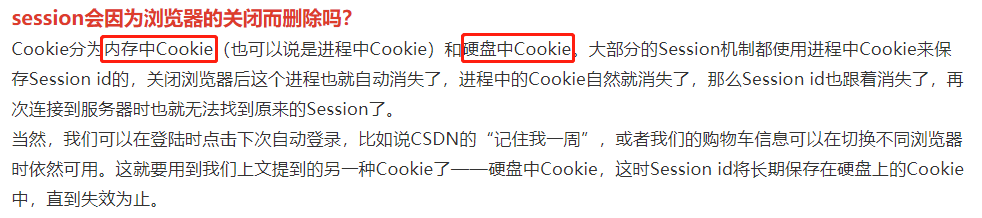


HTTP连接是无状态的

Session是客户端和服务器的会话，cookie是使这个会话功能变得更强大

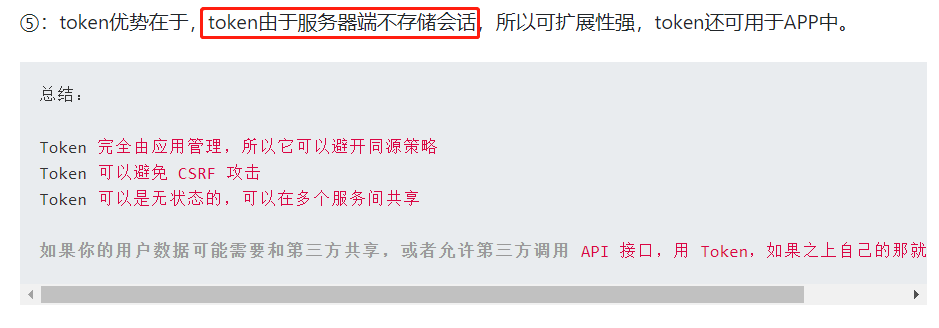


**session是解决http协议无状态问题的服务端解决方案，它能让客户端和服务端一系列交互动作变成一个完整的事务，能使网站变成一个真正意义上的软件**





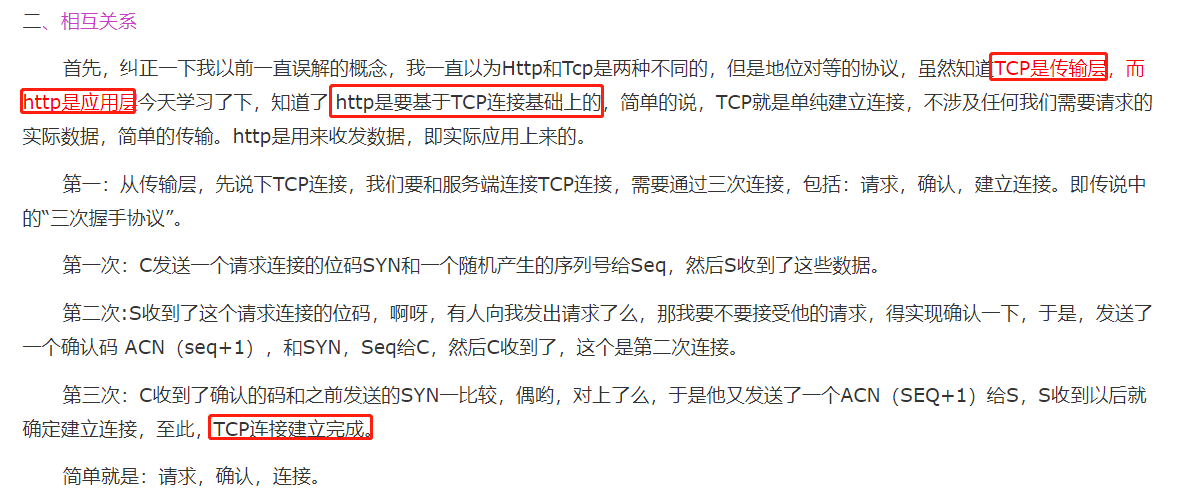
<https://blog.csdn.net/java_faep/article/details/78082802>

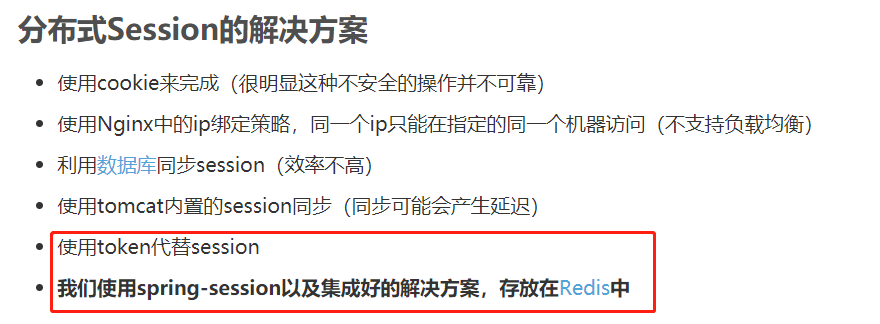




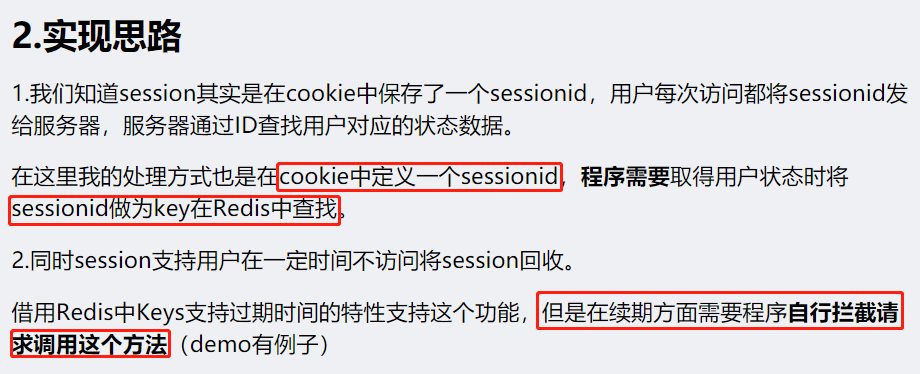
Session依赖于会话，服务端会保存session对象，服务端http通信是通过cookie来判断是哪个客户端发出的请求并给与回应，cookie被劫持之后可能造成CSRF跨站伪造请求攻击，而且每个用户的session对象服务器都要存储一份，直到过期时间过去后才删除，加大内存的损耗

Token可以存储于cookie和localstorage,放在localsrorage时可以通过js获取，不用每次请求都携带。Token不用在服务端保存一份对象，通过加密和解密获取用户信息来判断时候已经登陆





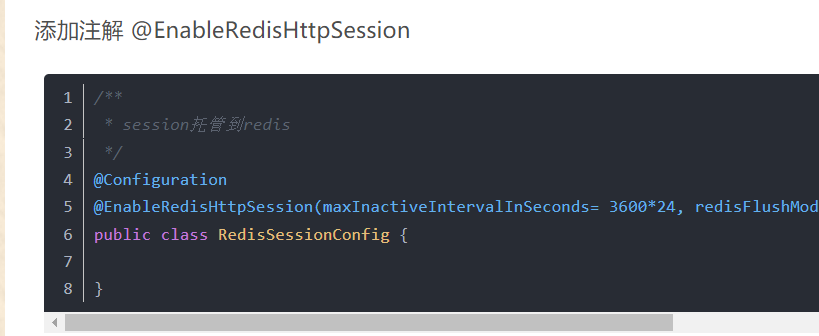


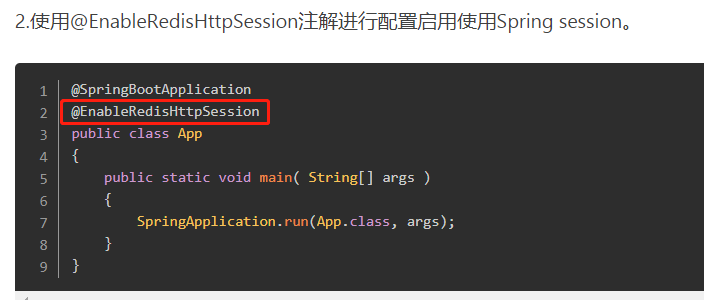




<https://www.cnblogs.com/54chensongxia/p/12096493.html>

即为把服务器的session对象保存到redis中，实现共享，过期则自动删除





public Long zRank(String key, Object value) {

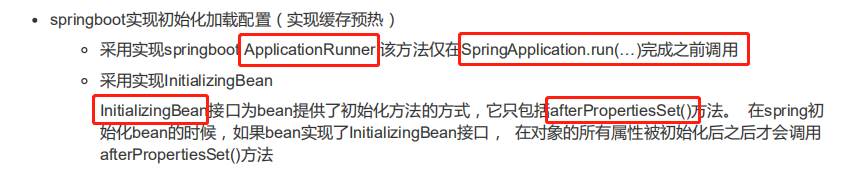
ZSetOperations<String, Object> zset = redisTemplate.opsForZSet();

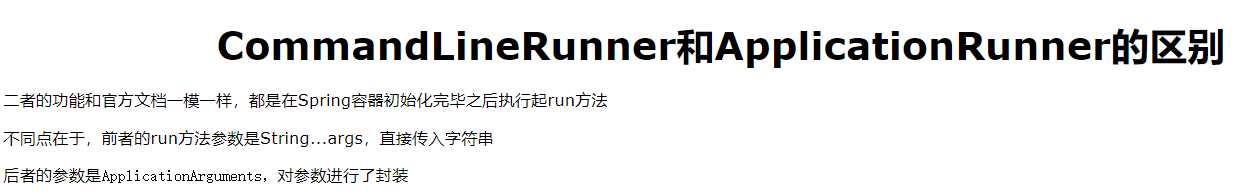
return zset.rank(key,value);

}

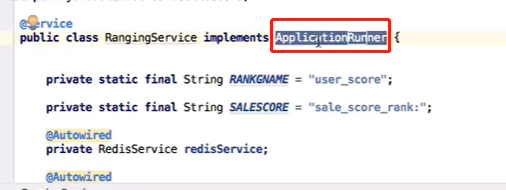
Key是索引，value是该数据在索引中的排名

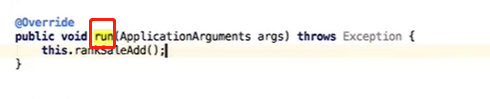
时间是要加索引的





实现applicationRunner接口在run方法中预热数据

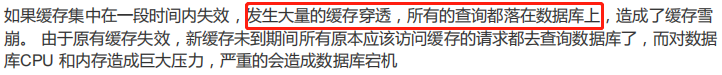


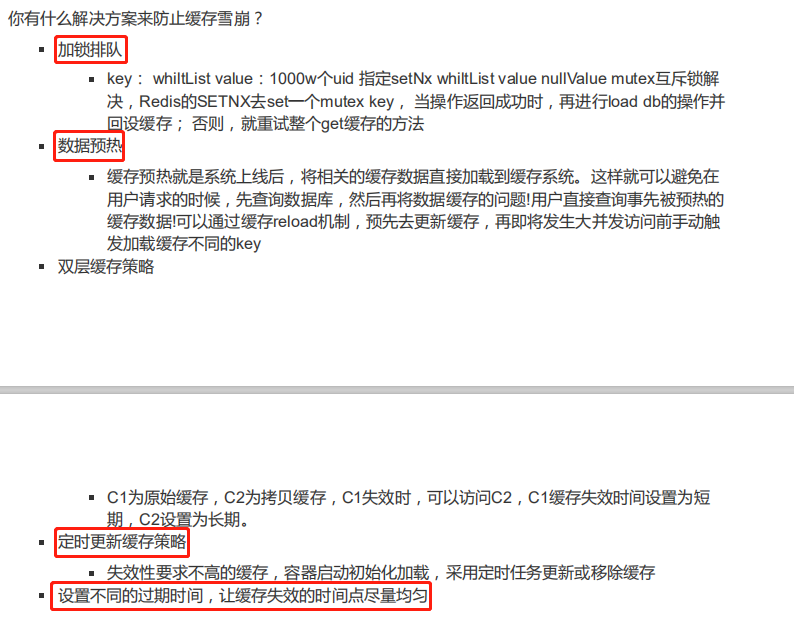


InitializingBean则为afterPropertiesSet

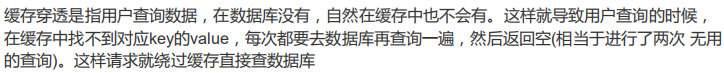


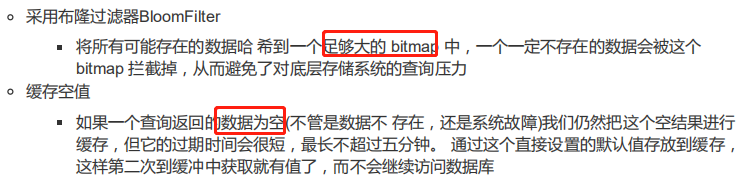




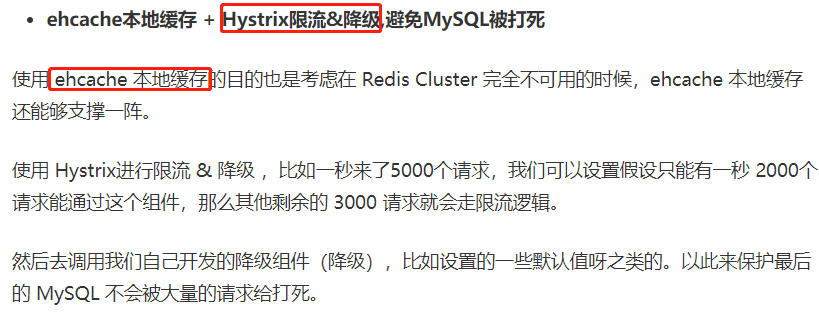


**加锁排队，第一个请求进来获得锁，其他请求休眠30毫秒**

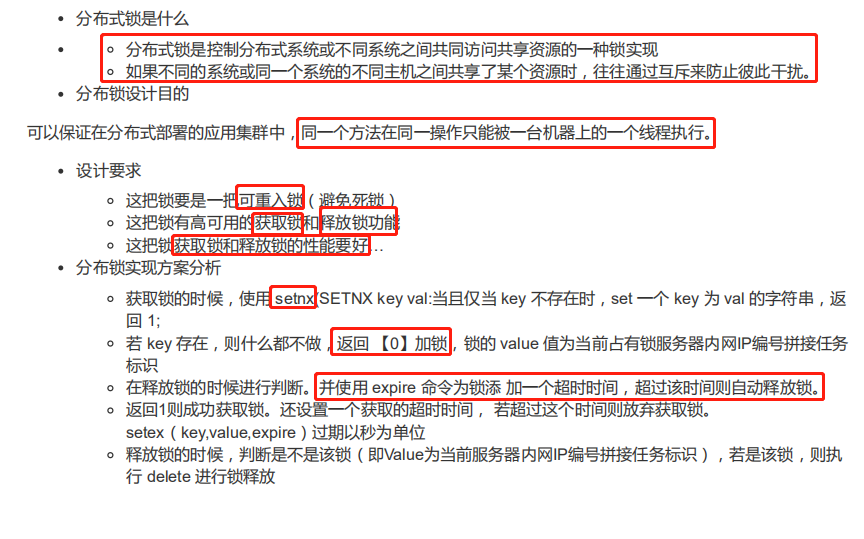


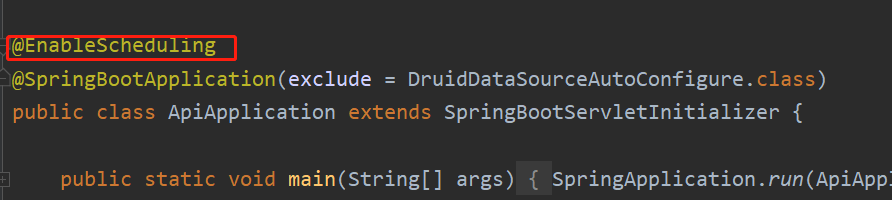


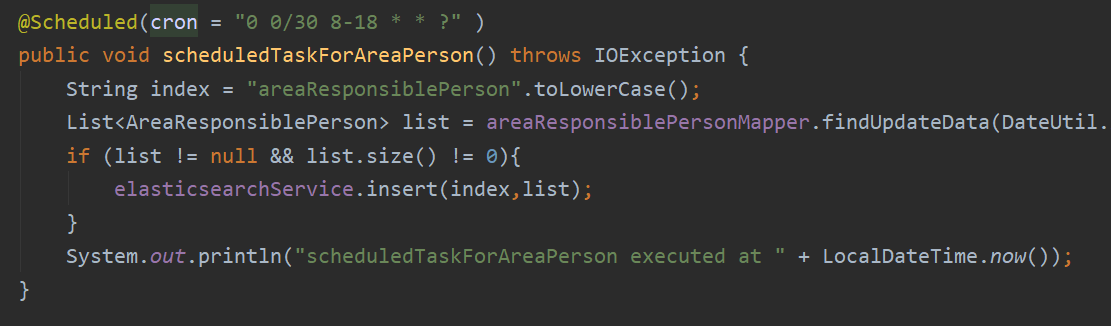
**空值也返回**

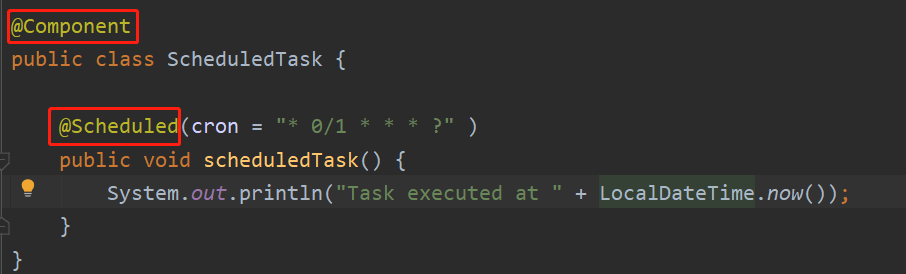


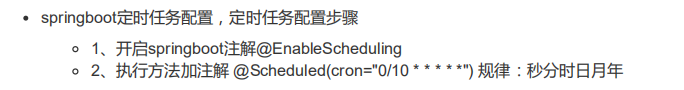
**多个线程同时访问同一资源可能找出业务逻辑的混乱，此时加锁来保证线程间不会彼此干扰**

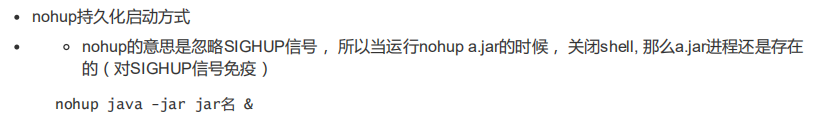










****