使用缓存时获取数据的流程

接收请求后，判断缓存中是否有数据，如果有直接返回、没有则查询数据库获取并更新缓存，然后返回结果。

**缓存穿透**

概念

查询一个并不存在的key，缓存不会命中，导致每次请求都直接落到数据库上，数据库压力过大。如请求一个id在数据库中不存在的数据（负数或很大的值）

解决方案

1、接口校验，用户鉴权，参数验证。

2、如果一个查询返回结果为空，将这个空结果进行缓存，并设置一个较短的过期时间

3、过滤非法请求，布隆过滤器（Bloom Filter）

**缓存击穿**

概念

大量并发请求查询某个值时该值缓存刚好过期，导致大量请求直接落到数据库上。如秒杀过程中，大量用户同时请求数据，而缓存刚好过期，请求就会同时到达数据库，造成数据库压力瞬间增大（一个key过期）

解决方案

1. 设置缓存永不过期

2、使用互斥锁 （使用setnx命令为key设置一个随机值，设置成功相当于获取锁然后执行数据库操作更新缓存并删除该值释放锁，设置失败usleep几百微秒，然后去获取缓存）

**缓存雪崩**

概念

在高并发下，大量缓存key在某个时间点同时失效，导致大量请求直接落到数据库上（多个key同时过期）

解决方案

1. 设置热点数据永不过期
2. 缓存数据过期时间添加随机值，防止同一时间点过期
3. 双缓存策略，设置一个备用缓存失效时间长点，缓存失效时访问备用缓存（为什么不直接设置一个时间长点带随机数的）还有一种方法记录过期标记，过期会出发通知另外线程更新缓存，为什么不直接永不过期呢？（个人觉得双缓存加自动更新组合使用比较靠谱，过期时间交替向前更新）