1. 写接口要考虑到其中会发生的异常，并处理
2. 写接口文档，参数需要设置是否验证
3. 后台接收参数一定要检测，否则就会出异常
4. 降低成本 ？= 提高程序复杂度
5. 在使用完对象后，将对象置为空，方便虚拟机进行辣鸡回收
6. JSON可以检测是否有某个key，要不然直接get可能会报不存在此key的错误
7. JSON对象get时之前，要检测该key是否存在
8. 前端传的需要用到的值一定要检测，否则放在服务器上面就会疯狂空指针
9. 如果表的变化大，那么使用所有参数构造器注解的时候要注意了，每次都得改
10. 涉及分页做一个分页对象，页码和数量可以由参数指定

public class BaseFrom {

private int pageNum = 1;

private int pageSize = 10;

private Map<String,Object> cnd = new HashMap<>();

}

1. 详细信息进入页面再查，不要查出来废数据
2. 若是假删除，有必要将删除和更新放一起吗？
3. 抛出异常后，sql语句会回滚，事务机制（mmp）
4. 写接口时，最好能写出参数的类型（论接口管理工具的重要性）
5. 有些地方接口的参数设置的要轻量点，否则太多废数据
6. 购物车中修改数量，监听的是离开页面的瞬间，同事发送请求，否则点一次就请求一次太夸张了
7. 设置图片/视频数据时设置全路径，不要在前端拼接
8. map中get出来的值转String最好是toString转，这样不容易报错
9. btb系统中，拦截器中有设置用户id，记得打开
10. 开发接口

需要考虑用户的各种操作

登录操作

不登录操作

换位思考其会如何操作

需要考虑到数据库中无数据的情况

需要考虑参数的各种情况

为null

为""

正常

and so on 情况

1. 后台如果改接口，前台如果调用该接口的地方多了就不好修改