1. 需要将userid换成merchantcode，因为userid是自身得到的自增主键，而merchantcode才是有意义的字符。
2. 与促销引擎相关的字符暂时无需存，因为插入之后会自动提交到促销引擎
3. 暂时将劵的面额信息放入template表中
4. 查询返回的信息如果有具体内容则需要加密
5. 使用templatecode即用户自身输入的信息和merchantcode来进行查询，由于不同的商户下可能templatecode相同，所以把这两个作为联合搜索。再添加劵模版时，需要检测templatecode和merchantcode的unique，如果不唯一需要提醒用户templatecode重复请重新输入。
6. 删除template时，需要检测是否已经发放（即查看Distributednum）如果不是0，则告诉客户该类型优惠券已经发放，无法删除
7. 发放劵需要添加张数，目前num就是代表它直接的意思，不存在循环和非循环，2张即分配2张，其中发放、修改、删除、都要检测该优惠券是否已经过期，如果已经过期，对其进行的任何措施都会失效。
8. 目前统一usercode就是用户的手机号
9. 优惠券发放错误中统一失败的status就是false，msg就是中文的错误信息
10. 使用coupon要检测的内容有：1、merchantcode和deptcode是否在该劵的使用范围内，2、usercode检测该用户是否能使用该优惠券3、由于使用劵是一个数组，所以要对其进行事务处理。只要一张使用错误，就回滚，必须所有都插入成功。
11. 当前dept和brand是平行关系。
12. 撤销使用需要检测：1、该用户是否已经使用该劵2、劵是否过期 3、劵是否属于该用户
13. mechantcode都需要检测，对劵和模版的接口调用一定要是对应的商户
14. merchantcode不能为空，如果dept为true，brand为002，则代表该优惠劵可以运用于该商户的品牌，不分门店。如果dept为02011，brand为true，则代表02011分店的任何专柜都可使用该优惠券。如果全场都可使用该优惠券则可设置dept和brand都为true（由于template\_usescope的主键不能为空）所以将null该为true

/////////////////////////////////////////////////////////////////////

新修改：

有多个品牌的品牌商把它算作商户，具有一个商户帐号，发劵模版时，它的agenttype是1.，2和3一定是一个门店和一个子品牌。

Usercode无法得知属于哪个商户，所以对于任何用户都可以查看

不同的商户是否具有相同的couponid