（登录、注册、修改用户信息）流程：（小程序）

1. 注册：前台传wxCode、用户信息
   1. 根据wxCode、AppId、AppSecret、服务器url获取openId（用户的唯一标识）-调用微信的接口
   2. 查询数据库中有没有此用户的openId，有--已经注册 没有--添加数据
2. 登录：前台传wxCode
   1. 根据wxCode调用微信接口，获取openId
   2. 根据OpenId（唯一）查询用户信息
   3. 返回用户信息，token（userid加密的字符串）
3. 修改用户信息：
   1. 首先返回给前台当前用户的信息
   2. 其次获取前台传过来的修改用户信息之后的信息
   3. 然后根据信息修改数据
4. 登录校验：
   1. 根据请求头的token，解析出userid，根据userid查询当前用户
   2. Token加密方式：
      1. 异或加密方式（字节数组） 加密和解密的密钥要一样（当作一个参数，再后台固定写死）
      2. 异或的解密就是（用相同的密钥再次进行加密）
      3. 涉及到三个要素：明文，密钥，密文
         1. 明文和密钥进行异或运算可以得到密文
         2. 密文和密钥进行异或运算可以得到明文
      4. 相对于其他加密算法：
         1. 算法简单，对于高级语言很容易就能实现
         2. 速度快，可以再任何时候，任何地方使用
         3. 对任何字符都是有效的，不像有些简易的加密算法，只对西文字符有效，对中文加密后再解密，无法还原出原来的字符串。

秒杀流程：----redis（1.获取缓存2.lua对redis操作3.对数据库操作）

1. 前台传入当前秒杀产品的id
2. 将数据库中的秒杀信息写入redis中（秒杀信息/秒杀成功的userid）
3. 验证
4. 在redis中lua脚本中判断（原子性），是否还有库存，是否已经购买，减去库存量，把当前抢购成功的用户id写入redis中
5. 是否已经购买，是否还有库存，减去库存量，把当前抢购成功的用户写在秒杀成功记录表中
6. 将秒杀成功的这个用户记录加入缓存中，有多少条数据就有多少次点击率（秒杀率）

登录流程：（前后端分离）

1. 注册：
   1. 前台传入账号、密码、用户基本信息
   2. 校验数据格式和非空，校验账号是否存在（是否已经注册）
   3. MD5加密 存入数据库
   4. 返回bool true注册成功
2. 登录：
   1. 前台传入账号和密码
   2. 校验账号是否存在，密码是否正确，以及一些对用户状态的校验
   3. 加密token，且将当前用户的基本信息和token返回前台
3. 修改密码：
   1. 传入账号 新旧密码 再一次输入密码
   2. 校验账号是否存在 校验密码是否正确（校验信息不能为空）
   3. 取出数据库的盐值，md5加密之后和数据库的md5加密的密码比对
   4. 返回bool true 修改成功
4. 修改用户信息：
   1. 传入用户的基本信息和账号
   2. 校验账号是否存在，校验信息不能为空，校验账号不能重复，校验手机号电话
   3. 返回bool true 修改成功
5. 登录校验
   1. 前台请求头传入token，根据token解析出userid，查询当前用户

登录流程：（后台）

1. 登录：
   1. 前台传入账号和密码
   2. 校验账号密码不能为空
   3. 根据账号唯一，查询当前账号是否存在
      1. 如果存在 根据当前用户的salt 加密比对 校验密码是否正确
      2. 校验用户状态（逻辑删除、停用）
   4. 存放cookie
      1. 查询cookie是否为空
         1. 空 不管
         2. 非空 清除
      2. 将当前用户的id 存入cookie
      3. 登录成功，清除当前用户下的所有模块的缓存（redis）
      4. 清除记住账户
2. 退出账号：
   1. 清除缓存
   2. 查询当前（admin）cookie是否存在，存在删除（通过设置过期时间的方法）
3. 登录校验
   1. 解析cookie，得出userid，查询userid（当前用户）