

- 利用小程序提供了websocket连接后台服务器，一旦断线就会设置定时器进行重连。
- 调用 wx.login() 获取 临时登录凭证code，并传到后台服务器
- 后台服务器接收到code后，调用 code2Session 接口，换取 用户唯一标识 OpenID 和 sessionKey和 session_key。

code2Session是一个 HTTPS 接口，通过http请求报文获得解密信息

openID:用户唯一标识

sessionKey:本次登录的会话密钥

unionId:用户在开放平台的唯一标识符(公众号，小程序等)

- 后台利用openID作为数据库主键将用户数据存储在Sqlite中。
- 其中加密sessionKey中保存有用户的个人信息，利用微信官方提供的解密代码解出个人信息，并将用户所需要的信息传回。
- 后台通过利用包装不同的头部信息来让客户端知道传来的是什么信息.此处头部为"user_Info"。客户端接受到后更新用户数据。这样，客户端的初始化完成。

user_Info, balance, carID等头部

后台发来数据均是json，客户端容易随机访问。

- 用户要停车时扫二维码发送carID给后台，后台比对车位上采集到的信息，如果车位空且未被其他人占用，则允许停车，并发回carID，这时后用户才算是占用该carID.并启动计时器，计算停车时间。

用户请求websocket发送，形式是"请求：数据"

如何采集？就是arm板采集霍尔元件发送来的车位占用信息，通过lora模块发送到lora网关，然后网关在送到服务器

计时器怎么写的？拉的轮子

如何同步计时器？开始计时时，其实是客户端和后台同时计时。如果用户关后台。用户再次登录时，后台会发送另一个数据包，里面包含已经停车的时间。客户端从此时间开始继续计时。

怎么判断车被占了?其实数据包里有curState，保存当前用户状态。通过返回不同的码来识别用户状态。【-1被占，-2余额不足，0代表历史已完成订单和无状态，1代表下了订单但未停车，2代表正在停车,3代表订单已完成，但消息未发送给wx端】

- 当用户把车开走的时候，arm板检测到车辆开走，发送给后台。后台计算停车费存入数据库后，并发送给客户端。客户端接受到信息后。显示账单并更新数据。

