不同方式获取用户信息:

* 简单显示当前用户信息: open-data组件
* wx.getUserInfo()，只有授权过的用户，调用后才能正常获取用户信息
* 推荐方法，写一个button, bind一个getuserinfo事件，这样会弹出授权框
* Openid是在当前帐号下小程序，公众号，等等每个都不一样，unionid是这个帐号下所有小程序公众号都一样

页面渲染:

* 注意小程序的背景图只能从外部引入或者转成base64格式, 这里推荐base64

播放历史本地存储:

1. 在全局的app.js中拿到用户的openid，判断当前存储中的openid是否存在，如果不存在，set存储openid为空数组
2. 在player中定义保存播放历史功能的方法:

拿到当前播放的歌曲

拿到openid

拿到当前本地存储的数组

判断：设个值代表本地存储的歌曲是否重复，默认为false, 如果当前这首歌和本地存储里的这首歌重复，那这值为true，然后跳出循环。 循环完毕后如果还是false, 证明历史里没有这首歌，就把这首歌头进推入本地存储的数组

最后重新set本地存储，然后在播放歌曲的时候调用这个方法

1. 前端渲染，在profile-playhistory里拿到当前存储的值，如果为空，提示框。如果不为空，赋值到data，然后前端渲染（把musiclist组件拿过来渲染）
2. 确保点下一首的时候播放的是播放历史里的下一首，而不是原来musiclist里的下一首。用set本地存储覆盖一下就可以

查看自己发的博客:

1. find云函数中建个路由，用cloud.getWXContext()拿到wxContext，并把\_openid赋值为wxcontext的openid, 然后从事件的start开始，限制为count，然后逆序排列
2. 去到profile-bloghistory，调用云函数，把开始赋值为当前bloglist的长度，count为10，成功后setdata里设置blog的值
3. 前端渲染

生成小程序码:

1. 建立新的云函数，json中配置permissions的openapi为 “wxacode.getUnlimited”
2. 在index中cloud.openapi.wxacode.getUnlimited中传入对象包含scene属性
3. 到前端绑定事件，事件中调用云函数
4. 云函数中的 cloud.uploadFile把图片的cloudPath从字符串转为png， file content就是buffer，之后return整个file的fileID
5. 到前端调用的方法中，wx.previewImage传入url和current, 且他们的值都为fileId
6. 最后到find的onload里面，输出options.scene就可以看到此值，通过这个值就可以做一些分销功能