# **微信小游戏排行榜功能快速开发教程**

作者：[Matchvs](http://gad.qq.com/user/index?id=2257559)

链接：<http://gad.qq.com/article/detail/247687>

要做这个好友排行榜.必然要有好友的战绩比分,然后再做排序,最后将数据呈现在UI上 , 可以分为下面几个步骤:

1. 保存每个用户的分数
2. 获取好友列表,并获取好友的分数
3. 渲染排行榜



### **保存每个用户的分数**

保存每个用户的分数,需要调用微信的云存储API,将用户的分数持久化的存起来 .

// 保存用户数据,注意限制单条数据容量不得超过1024字节,

// 单个用户数据总条数不得超过128条

wx.setUserCloudStroage(Object)



//存储最高分

var score = 100;

var kvScore = ｛"key":"score","value":score｝;

wx.setUserCloudStroage(｛"KVDataList":[kvScore]｝,"success":function()｛

//

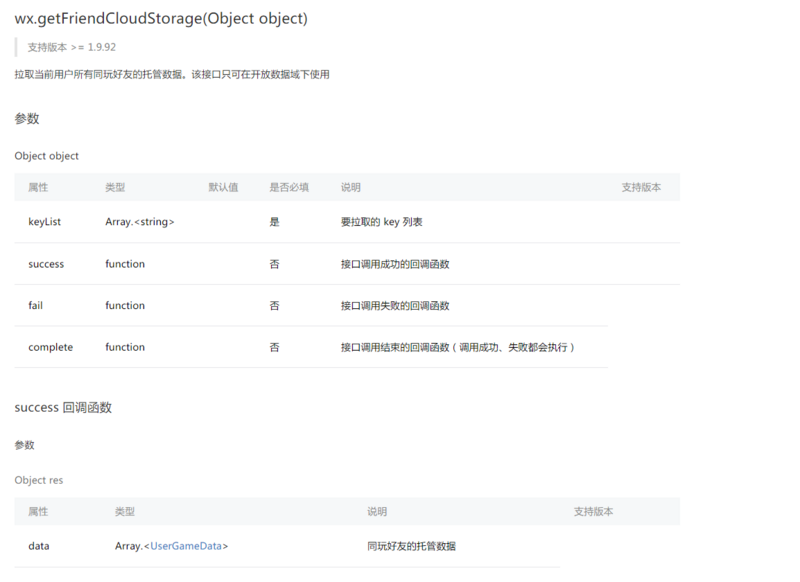
｝);

### **获取好友列表,并获取好友的分数**

​游戏中将玩家的分数保存起来以后,需要调用微信的云存储API wx.getFirendCloudStorage, 获取玩家的微信好友数据,这样就拿到了每个好友的最高分.

值得一提的是,微信的这个接口在内部隐蔽的使用其微信的社交关系链 .

返回列表中的包含的 调用过wx.setUserCloudStroage 这个接口的用户.



### **渲染排行榜**

对分数进行排序 , 得到一个排行榜

let sharedCanvas = wx.getSharedCanvas()

function drawRankList (data) ｛

data.forEach((item, index) => ｛

// ...

｝)

｝

wx.getFriendCloudStorage(｛

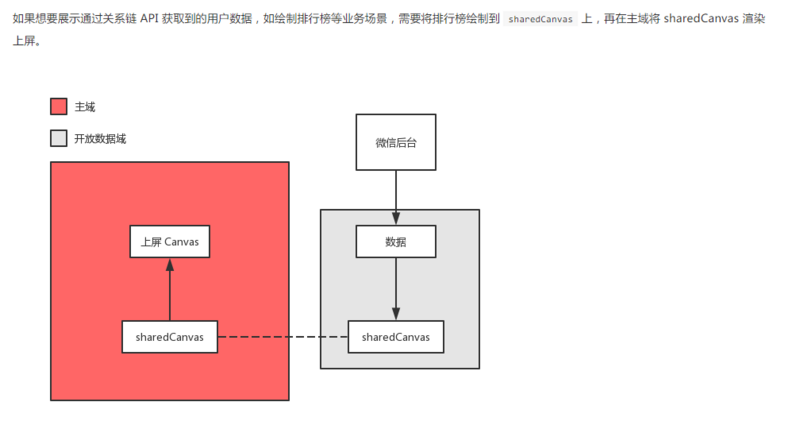
success: res => ｛

let data = res.data

drawRankList(data)

｝

｝)

注意这个 sharedCanvas , 这是独有的画布 ,与小游戏中的画布不是同一个东东.  


使用sharedCanvas来自定义显示玩家的用户排行榜

### **重要说明**

​上述所涉及的微信接口 , 都只能在微信小游戏的 子域 使用 , 微信官网也称之为开放数据域 , 其实应该称之为 封闭数据域 .

为什么这么说呢?  
​

因为子域的js代码执行环境和小游戏本身的代码执行环境是隔离的. 两者之间不能相通 , 子域只能接收外部的消息(如游戏的最高分),不能往外发消息(不能把用户的好友关系链数据发给开发者服务器) , 内存不共享,也就意味着,开发者拿不到微信的社交关系链数据 . 在这样的封闭条件下,开发者能做的只能是在子域的画布上将排行榜数据以个性化的UI元素展示出来 .