通用API接口文档整合

注：本文档中所有的JS交互post过来的数据统一用json串格式，如无数据则json里面可以为空.

注:以下所有的API建议封装成一个工具类或者一个webView基类的形式，方便其他APP套用，所有的response返回参数格式统一如下：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| responseCode | 0或者1 |  |
| responseMessage | 提示信息例如成功失败 | 提示信息例如成功失败 |
| responseData | {  “key1”:”value1”,  “key2”:”value2”  ….  } | 返回数据一般是json键值对的形式 |

1.位置上传 methodName :uploadLocation

json：{}

接收到uploadLocation指令即POST设备所在经纬度和位置信息字符串给后台接收

上传request参数:

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| longitude | 123 | 经度 |
| Latitude | 123 | 纬度 |

2.录音 methodName：recordAudio

json:{}

接收到recordAudio指令即跳转到录音界面，进行录音，录音完毕之后上传给接口

上传request参数:

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| File | file | 文件上传根目录 |

数据以数据流的形式进行上传

3.拍摄（照片和视频）：

methodName:takePicuture

methodName：takeVideo

接收到methodName指令，进行相关原生页面的跳转，操作完之后，POST给后台接口

上传request参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| File | file | 文件上传根目录 |

数据以数据流的形式进行上传

4.支付

methodName：pay

json格式:

后台返回的参数列表名称：

微信支付 ：

{

“type”:1,（微信支付）

“sign”: 支付签名串，

“appid”:”jii4n3njfd”,

“nonceStr”:”dwret”,

“partenrid”：”rereretrf9isif”,

“prepareID”:”E23JIFJNDF”,

“timeStamp”:1289473843,

}

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| type | 1/2 | 0/1 代表支付宝和微信 |
| sign | Xxxx | 支付签名参数 |
| appid | Xxx | 支付appid |
| prepareID | Xxx | 预支付id |
| timeStamp | 1308434 | 时间戳 |

支付宝支付：

{

“type”:2,（AliPay 支付宝支付）

“orderSign”:”32rfddvdv(支付参数)”,

}

支付宝支付后台签名参数:

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| type | 2 | 支付宝支付 |
| orderSign | xxxxxx | 订单支付签名参数 |

在H5页面中点支付的时候， 后台负责签名支付参数（加密），并以json形式发送给移动端，移动端进行原生的调用以完成支付。

5.分享

methodName:share

json数据格式：

{

title；”分享标题”，

“url”:”分享链接”,

“thumbImage”:”缩略图”,

“content”：”分享内容”,

“callBack”:”分享失败的JS回调代码，可由移动端自己加载”,

}

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| title | xxxx | 分享标题 |
| thumbImage | http://www.xxx.png | 分享图片缩略图地址 |
| url | http://www.qianft.com | 分享url地址 |
| content | Xxx | 分享内容标注 |
| callBack | Xxxx | 分享之后的回调 用于刷新js页面比如弹框分享成功等等 |

移动端配置好三方分享的代码，key值初步固定，后续可以通过接口的形式重新初始化三方key，分享平台分别为微信分享（朋友圈、收藏、发送给朋友）、QQ分享、QQ空间分享，点击哪个即分享到哪个平台（相关的配置和代码由移动端负责配置）

6.手势解锁:

json :{}

methodName:settingGestureLock

上传参数：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| gestureLockCode | 2346434 | 手势锁密码 |

移动端跳转到设置手势解锁的页面，设置完成之后，将手势密码POST给后台

7.清除手势密码 ：

methodName:clearGestureLock

json {}

接收到clearGestureLock指令，清空本地保存的手势密码 ，并移除手势解锁页面

8.指纹解锁:

methodName:settingTouchID

json {}

移动端负责保存一个开关，接收到settingTouchID，打开开关，并在每次进入App的时候，弹出touchID认证页面，

9.清除指纹解锁:

methodName：clearTouchID

json：{}

移动端关闭开关，移除touchID认证页面

10.登录

methodName：loginVerify

token是后台发送过来的，移动端负责保存，一些页面需要进行登录验证，由于登录是原生做的，后台负责发送loginVerify，移动端接收到这个指令就弹出原生的登录注册页面，

如果是原生的登录注册页面的上传参数如下:

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| username | Chan | 用户名 |
| Password | 123456 | 密码 |
| VerifyCode | 435465 | 验证码 |
| tel | 18720977712 | 手机号码 |

11.本地数据（增删查改）

安卓用xml，ios用数据库或者plist保存数据在本地，后台可以通过指令对设备保存的本地数据进行增删查改，

methodName；addData (增加)

json:{

“key1”:”value”.

‘key2’:”value2”,

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| insertKey1 | 要增加的值1 |  |
| insertKey2 | 要增加的值2 |  |
| insertKey3 | 要增加的值3 |  |

“key3”:”value3”,

…

}

methodName：deleteData(删除)

json:

{

“deleteKey”:value1

}

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| deleteKey | xxx | 要删除的key |

methodName:queryData(查询)

json:

{

queyKey:”key1”,

}

methodName:udpateData (更新)

json:

{

key:”key”,

value:newValue

…

}

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| key | xxx | 要更新的key |
| Value | xxx | 要等新的value |

移动端根据methodName来分别处理，根据json里面的key，去移动端存贮的本地数据中先查找到对应的值，再分别对其进行增伤查改

12.拨打电话：

methodName： dialog

json:

{

“tel”:18720977712

}

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| tel | 18720977712 | 要拨打的电话号码 |

13.扫描二维码/条形码

methodName:scanQRCode

json:{}

移动端跳转到扫描二维码页面，二维码好条形码和后台约定好json数据格式，扫描到二维码里的数据进行跳转处理等等

14.上传设备UDID：

methodName：uploadUDID

request参数:

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| UDID | JFIHG-3FDFDD-SGJSGNS | 手机唯一标识符 |

移动端负责POST设备唯一标识符给后台。

15.三方登录

methodName；thirdLogin(QQ/微信)

json:

{

type:1,2(代表QQ/微信)

}

在H5页面中吊起原生的三方登录，登录完成获取到三方的用户信息之后，将账户信息POST给后台，进行登录认证，

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| nickname | Chan | 昵称 |
| Token | 394jfgjsfb | 三方登录返回的唯一标示 |
| Sex | Male、female | 性别 |
| Age | 20 | 年龄 |
| avatarPath | http://www.xxx.png | 头像地址 |

16.头部显示隐藏、按钮的样式

methodName:hideTooBar

json:

{

hideToolBar:0/1

leftButton:标题1，标题2…..

rightButton:标题1.标题2….

}

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| HideToolBar | 0/1 | 是否隐藏头部导航栏 |
| leftButtonItem | Action1 | 左边按钮点击事件（后台与安卓ios约定好） |
| rightButtonItem | Action2 | 右边按钮点击事件（后台与安卓ios约定好） |
| url | http://www.qianft.com | 指定跳转页面url |

移动端负责接收hideToolBar的bool值，是0就用H5的导航栏，是1就用原生的导航栏，左右标题的设置根据json里的leftButton和rightbutton而定，这里的标题要跟后台协商好点击之后的事件

返回按钮点击事件：

methodName：backButtonAction

json:{

[url:’xxx://xxx.com](url:'xxx://xxx.com)’

}

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| url | http:www.xxx.com | 要跳转的url |

注： 这里的返回按钮点击 默认是返回上一级页面，当接收到url的时候，优先跳转到指定的url页面

17.白名单过滤

进入APP去后台拉白名单，每次加载H5页面去拦截url，只要url不在白名单里面的，不允许加载，这里的白名单过滤的作用就是过滤一些三方网站的加载，比如：屏蔽广告

18.长按复制

methodName：avoidCopy

json:

{

}

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| forbideCopy | 0/1 | 是否禁止拷贝 |

接收到voidCopy，移动端负责屏蔽加载页面的长按复制功能，不允许用户长按复制等操作

19.屏幕旋转

移动端负责适配

20.启动图、引导图

移动端负责拉接口来获取启动图和引导图

21.推送

移动端负责

推送消息的参数列表：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| badgeNumber | 100应用图标提醒数 | 应用图标提醒数 |
| apnsContent | http://www.xxx.xx.com | 点击推送后的页面跳转URL |

22.加载失败和加载动画

移动端负责

23.版本更新

后台存最新上架的版本号（分安卓和ios），移动端每次启动去后台数据库取这个最新版本号和自己的去做对比，若对比比这个版本号小，就弹更新按钮，

response参数列表：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| appType | Ios/android | Ios或者android类型 |
| versionNumber | 2.0 | 最新版本号 |

24.扫描用户设备安装的所有APP名单，并获取APP是否正在运行

移动端负责获取到设备的APP名单，并POST给后台

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| appListBundel | [com.app1,com.app2…] | 用户安装的所有app名单 |

25弹出到外部浏览器访问：

methodName:

outterBrowserJump

{

url:http://www.baidu.com

}

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| url | http://www.baidu.com | 跳转到外部浏览器访问该url |

26.点击图片保存图片/下载图片

methodName:downloadPhoto

下拉弹出下载图片的选项（安卓/ios实现）

{

imgUrl:http://www.baidu.com

}

|  |  |  |
| --- | --- | --- |
| 参数名称 | 参数值 | 参数描述 |
| imgurl | http://www.xxx.png | 下载图片url |

移动端实现