

Unity WebGL小游戏接入文档（24.10.14）

更新简介

[Feature（新增功能/接口）](#)

[Others（其他修改）](#)

如何上线一个 Unity WebGL 小游戏

一、创建 Unity 小游戏应用

二、Unity 游戏内容构建、转换和适配

ALIPAY-SDK

[Unity编辑器设置](#)

[AlipaySDK下载地址：241014](#)

Alipay Build Tool (Unity编辑器工具)

[出包流程](#)

[Loading页接入](#)

[Loading图注意事项](#)

[Loading字段说明](#)

[loadingPageConfig参数](#)

[TextConfig参数](#)

Style参数

[BarConfig参数](#)

Style参数

[MaterialConfig参数](#)

三、游戏预览和上传

3.1、安装支付宝小游戏开发工具

3.2、登录支付宝开发者账号

3.3、预览

3.4、上传

3.5、体验码

四、游戏审核和上线

五、游戏运营

六、Alipay SDK API

一、CanIUse, 判断 API 能否使用

1. API 说明
2. 参数描述
3. 调用示例:

二、获取用户授权

1. API 说明
2. 参数描述
3. 调用示例
4. 返回值示例

三、获取基本信息

1. API 说明
2. 参数描述
3. 调用示例:
4. 返回值示例

四、获取启动参数

1. API 说明
2. 参数描述
3. 调用示例:

五、键盘

1. 显示键盘
2. 使用示例

六、广告

1. 激励广告---RewardAd

- 一、API 说明
- 二、广告错误信息以及解决方案

2. 插屏广告---InterstitialAd

- 一、API 说明
- 二、广告错误信息以及解决方案
3. 调用示例:
4. 注意事项

七、分享

1. API 说明
2. OnShareAppMessage
 - 一、string title
 - 二、JObject param
3. ShowSharePanel
4. 调用示例:

八、游戏内购

1. API 说明
2. 参数描述
3. 注意事项
4. 结果错误码
5. buyQuantity 限制说明
6. 返回数据示例
7. 调用示例

九、跳转到其他小程序（小游戏）

1. API 说明
2. 参数描述
3. 调用示例:

十、退出当前小游戏

1. API 说明
2. 参数描述
3. 调用示例:

十一、文件

本地缓存文件

本地用户文件

清理策略

注意:

(一) AlipayFSManager GetFileSystemManager();

1. 功能描述
2. 返回

(二) Access(异步判断文件/目录是否存在)

1. 功能:

2. 参数:
3. 调用示例:
4. 返回
5. 错误码

(三) ReadFile(读取本地文件内容)

1. 功能:
2. 参数:
3. encoding合法值
4. 调用示例:
5. 返回

(四) WriteFile(写文件)

1. 功能
2. 参数
3. encoding合法值
4. 调用示例
5. 返回

(五) Unlink(删除文件)

1. 功能
2. 参数
3. 调用示例
4. 返回

(六) AppendFile(在文件结尾追加内容)

1. 功能
2. 参数
3. 调用示例

(七) CopyFile(复制文件)

1. 功能
2. 参数
3. 调用示例

(八) GetFileInfo(获取本地临时文件或本地缓存文件信息)

1. 功能
2. 参数

3. 调用示例

(九) GetSavedFileList(获取本地缓存文件列表)

1. 功能

2. 参数

3. 调用示例

(十) Mkdir(创建目录)

1. 功能

2. 参数

3. 调用示例

(十一) Readdir(获取目录内文件列表)

1. 功能

2. 参数

3. 调用示例

(十二) RemoveSavedFile(删除本地缓存文件)

1. 功能

2. 参数

3. 调用示例

(十三) Rename(重命名文件)

1. 功能

2. 参数

3. 调用示例

(十四) Rmdir(删除目录)

1. 功能

2. 参数

3. 调用示例

(十五) SaveFile(保存临时文件到本地)

1. 功能

2. 参数

3. 调用示例

(十六) Unzip(解压文件)

1. 功能

2. 参数

3. 调用示例

十二、音频

(一) AlipayAudioContext CreateInnerAudioContext();

1. 属性

2. 方法

十三、剪切板

1. API 说明

2. 参数描述

3. 调用示例:

十四、震动

1. API 说明

2. 参数描述

3. 调用示例:

十五、屏幕亮度

1. API 说明

2. 参数描述

3. 调用示例:

十六、支付宝Log

1. API 说明

2. 参数描述

3. 调用示例:

十七、网络

1. API 说明

2. 参数描述

十八、支付宝交互

(一) ShowToast

1. API 说明

2. 参数描述

(二) HideToast

1. API 说明

2. 参数描述

(三) Alert

1. API 说明

2. 参数描述

(四) Confirm

1. API 说明

2. 参数描述

(五) Prompt

1. API 说明

2. 参数描述

十九、触摸

1. API 说明

2. 返回对象

3. 调用示例

二十、存储

(一) SetStorage

1. API 说明

2. 参数描述

3. 调用示例:

(二) RemoveStorage

1. API 说明

2. 参数描述

3. 调用示例:

(三) GetStorageInfo

1. API 说明

2. 参数描述

3. 返回StorageInfo类

4. 调用示例:

(四) GetStorage

1. API 说明

2. 参数描述

3. 返回success

4. 调用示例:

(五) ClearStorage

1. API 说明

2. 参数描述

二十一、Socket

(一) AlipaySocket CreateSocket

1. 功能描述

2. AlipaySocket监听事件

(二) AlipaySocket.ConnectSocket

1. 功能:

2. 参数:

3. 错误码

4. 调用示例:

(三) AlipaySocket.CloseSocket

1. 功能:

2. 调用示例:

(四) AlipaySocket.SendSocketMessage

1. 功能:

2. 参数:

二十二、游戏圈

(一) 功能描述

1. API 说明

2. 参数描述

(二) 游戏圈圈子ID

(三) CreateGameClubButtonParam对象

1. 类结构

2. Options 创建参数

3. GameClubButtonStyle按钮样式

(四) GameClubButton对象

1. 方法

2. 使用示例

(五) GetGameClubData

1. API 说明

2. 参数描述

3. 返回

二十三、游戏行为上报行为上报

1. API 说明

2. 调用示例:

二十四、获取当前位置

1. API 说明

2. 参数描述

3. 调用示例:

4. 返回示例

二十五、获取当前电量信息

1. API 说明

2. 参数描述

3. 调用示例:

4. 返回示例

二十六、设备网络状态

1. API 说明

2. 参数描述

3. 调用示例:

4. 返回示例

更新简介

Feature (新增功能/接口)

- 新增导出时unity-namespace.js配置文件 (支持资源预下载功能)

Others (其他修改)

- FS写入较大数据耗时优化
- FS同步读写二进制文件方法优化

如何上线一个 Unity WebGL 小游戏

Unity 小游戏的上线主要分为以下几步：

1. 创建 Unity 小游戏应用
2. Unity WebGL游戏 打包构建、转换和适配支付宝
3. 游戏打包和上传
4. 审核和上线
5. 游戏运营

一、创建 Unity 小游戏应用

目前的流程与普通小游戏类似，可以参考：<https://www.yuque.com/zhifuba Xiaoyouxi-zromu/ts2ry9/cd5abg3aaneeeoopq>

注意事项

- 1、应用类型需要选择为 **Unity 小游戏**
- 2、Unity 游戏的 **appid 应该是 206017 开头**，如果不是，可以检查下选择的的游戏类型是否正确
- 3、需要确保 **最低基础库版本为 1.24.11**

操作路径：应用信息-> 基础设置-> 设置最低基础库版本-> 1.24.11

The screenshot shows the Alipay Mini Game console interface. On the left, the sidebar has '应用信息' (Application Information) selected. The main content area shows the '设置最低基础库版本' (Set Minimum Base Library Version) dialog box. The dialog box contains a table with the following data:

版本	受影响用户UV占比
<input type="radio"/> 1.24.13	0.04%
<input checked="" type="radio"/> 1.24.11	0.0%
<input type="radio"/> 1.24.9	0.0%
<input type="radio"/> 1.24.6	0.0%
<input type="radio"/> 1.24.5	0.0%

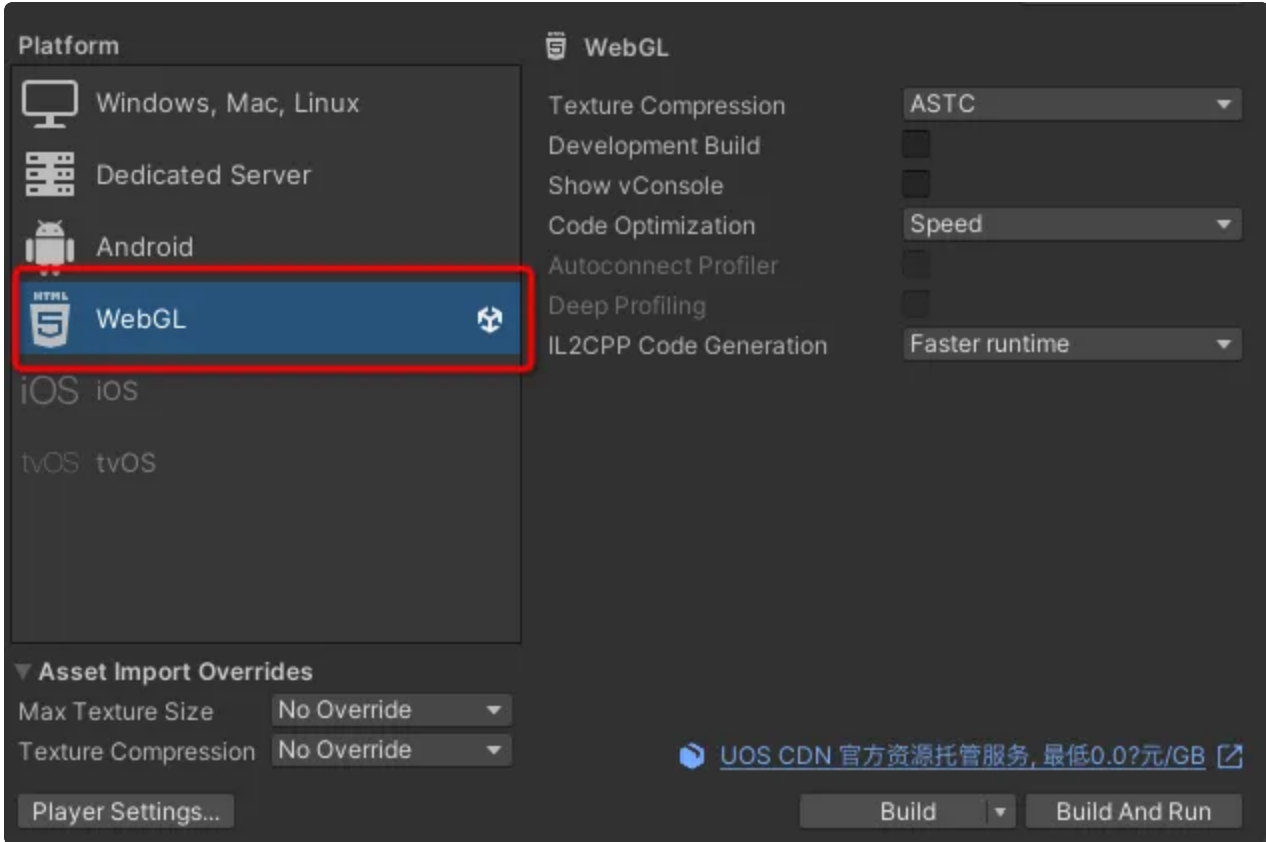
At the bottom of the dialog box, there is a '确定' (Confirm) button. Below the dialog box, a status bar shows '设置最低基础库版本 已设置' (Set Minimum Base Library Version has been set).

二、Unity 游戏内容构建、转换和适配

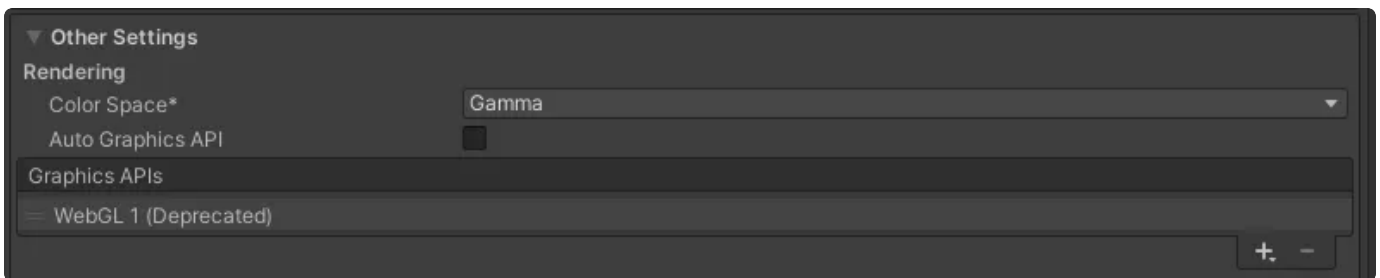
ALIPAY-SDK

Unity编辑器设置

- Unity接入需先将工程转换为WebGL打包。随后将ALIPAYSDK导入Unity



- 支付宝目前只支持WebGL1。请按下图设置选项

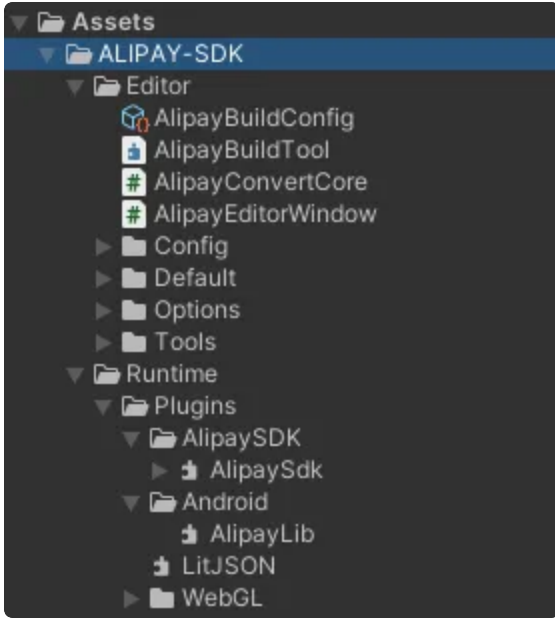


AlipaySDK下载地址：241014

- 下载后解压包体，导入插件到Unity中。确保Unity工程没有报错后，进一步开发接入。

ALIPAY-SDK结构分为Editor与Runtime两部分。其中Editor中代码为Unity编辑器工具，用户将工程导出为在支付宝中运行的结构。Runtime部分则是提供支付宝相关基础功能的API（授权、广告等）。

ALIPAY-SDK在工程中位置，请不要挪动该文件夹，导入后保持默认位置即可。



- 使用打包工具打包出一个正常webgl测试是否能够正常出包。若无问题，开始接入API
- ALIPAYSDK提供的API、具体调用文档请看[第六节-AlipaySDKAPI](#)

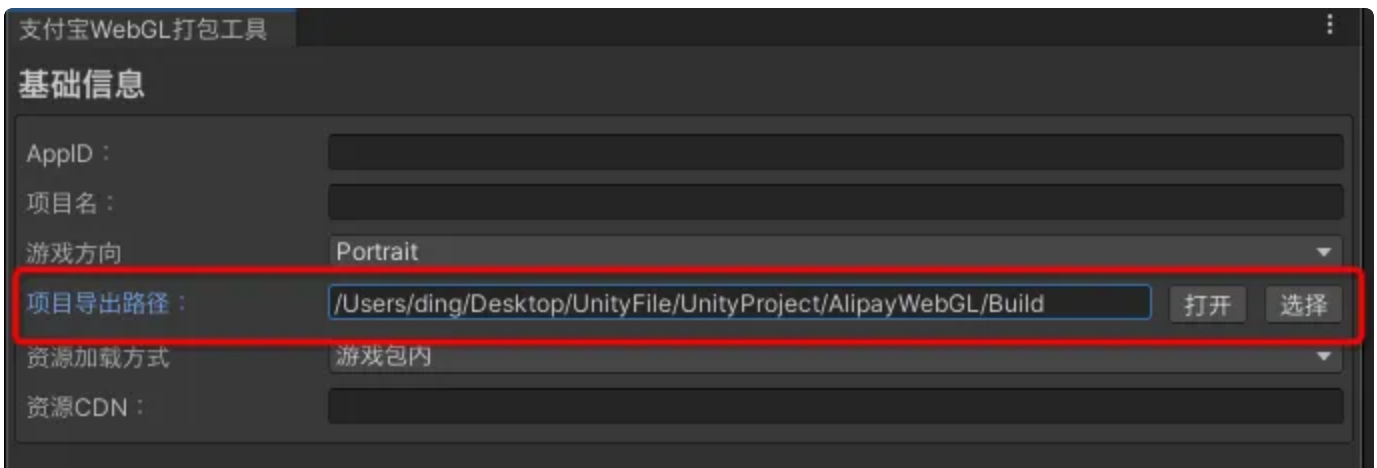
Alipay Build Tool (Unity编辑器工具)

出包流程

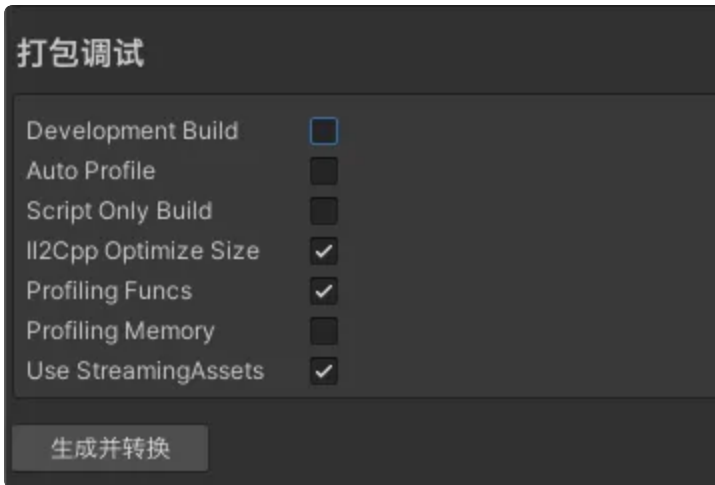
1. 在Unity顶部工具栏总选择支付宝小游戏->WebGL打包工具



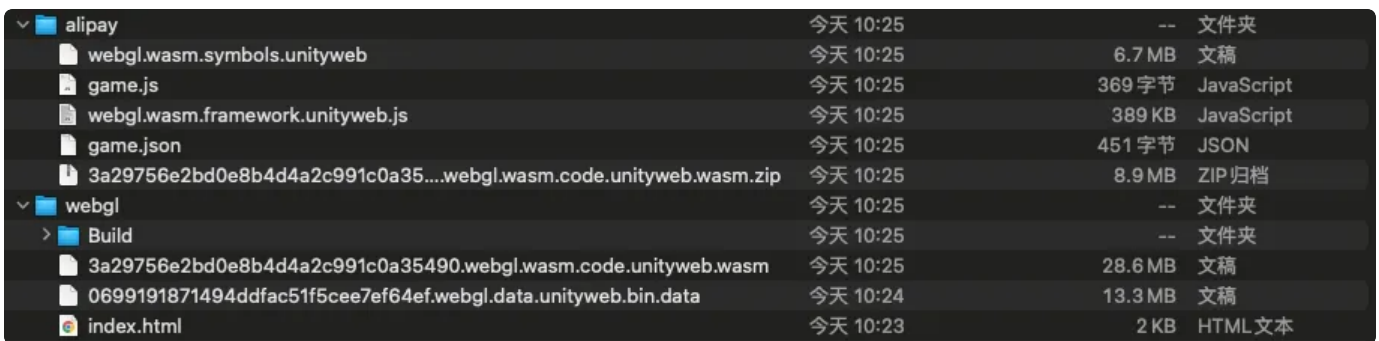
2. 在工具窗口中，点击选择配置好导出的路径。



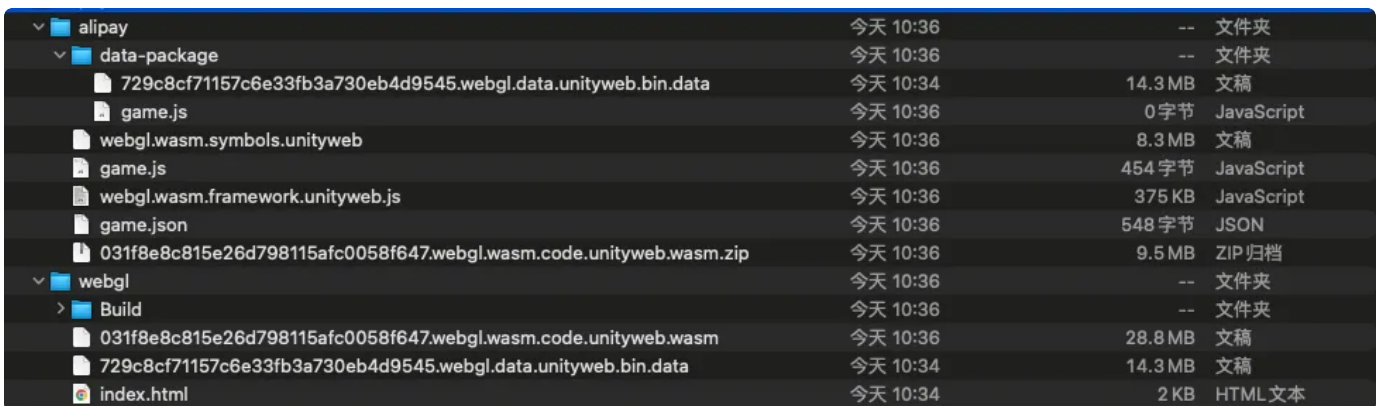
3. 资源加载最好使用CDN，使用cdn加载方式请填写资源CDN一栏。如果有使用到StreamingAssets文件夹资源也需配置CDN。(整包大小不超过30M可以使用游戏包内加载，否则必须使用CDN)
4. 点击**生成并转换**，等待转换完成--->若出现打包问题，可联系支付宝同学共同排查



5. 转换完成后，会在目标路径生成webgl与alipay两个文件夹



资源加载---CDN



资源加载---游戏包内

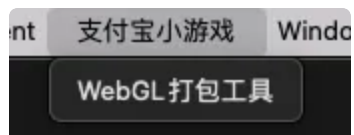
6. 选择CDN模式下，请将webgl文件夹下后缀为 **.bin.data**文件上传至目标cdn的服务中
7. 检查alipay文件夹中game.js中data文件的url路径，以及StreamingAssets文件路径是否与cdn服务器上一致
8. 使用minidev对alipay文件夹进行预览，上传等（Alipay下的symbols文件为分包需要，若出现预览超

时，可删除symbols文件再次尝试)

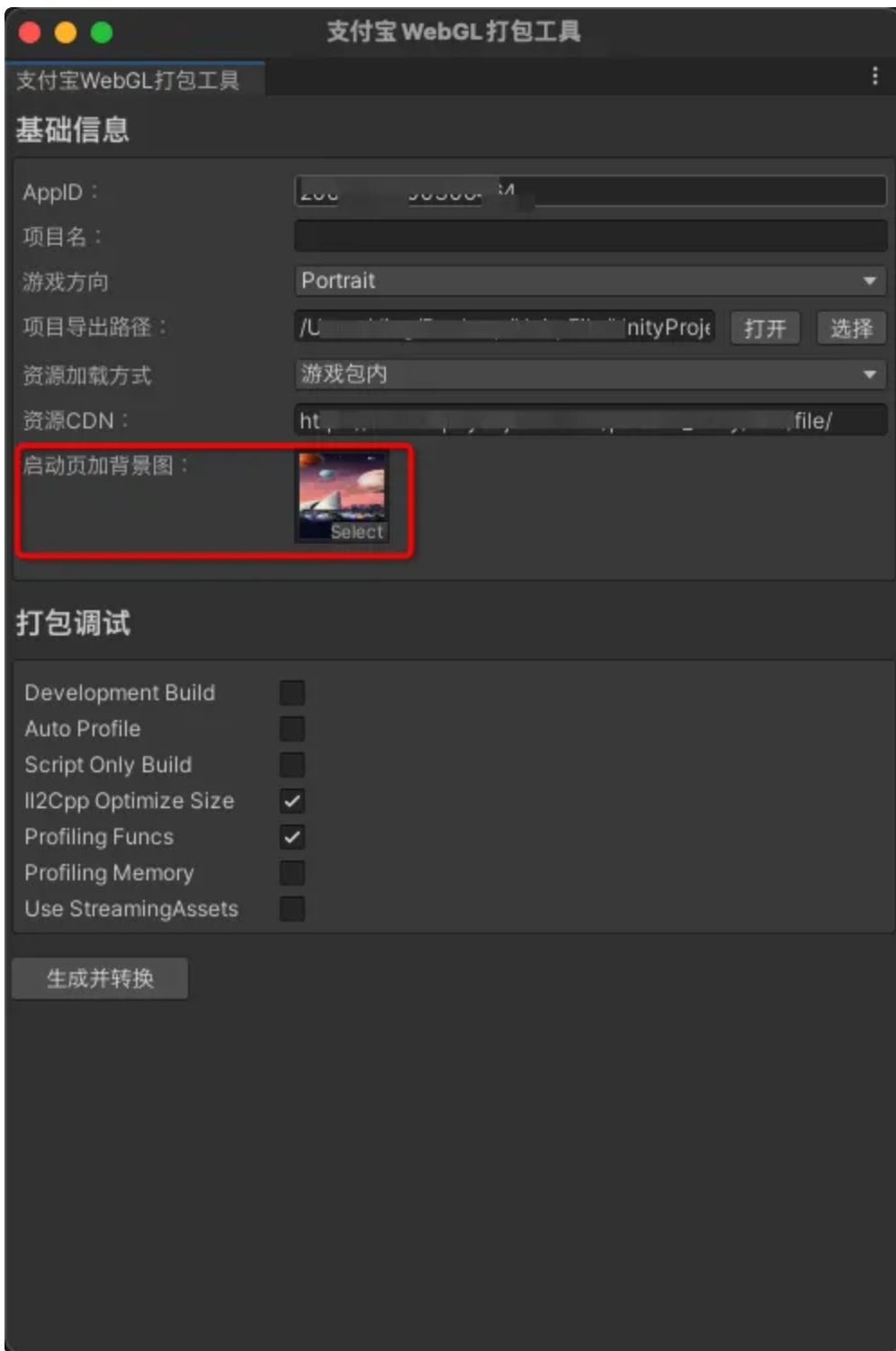
Loading页接入

Loading页面展示功能在支付宝10.6.30版本支持

1. 导入ALIPAYSDK，确保工程正确编译。
2. 顶部工具栏打开支付宝小游戏的WebGL打包工具。



3. 选择loading图。

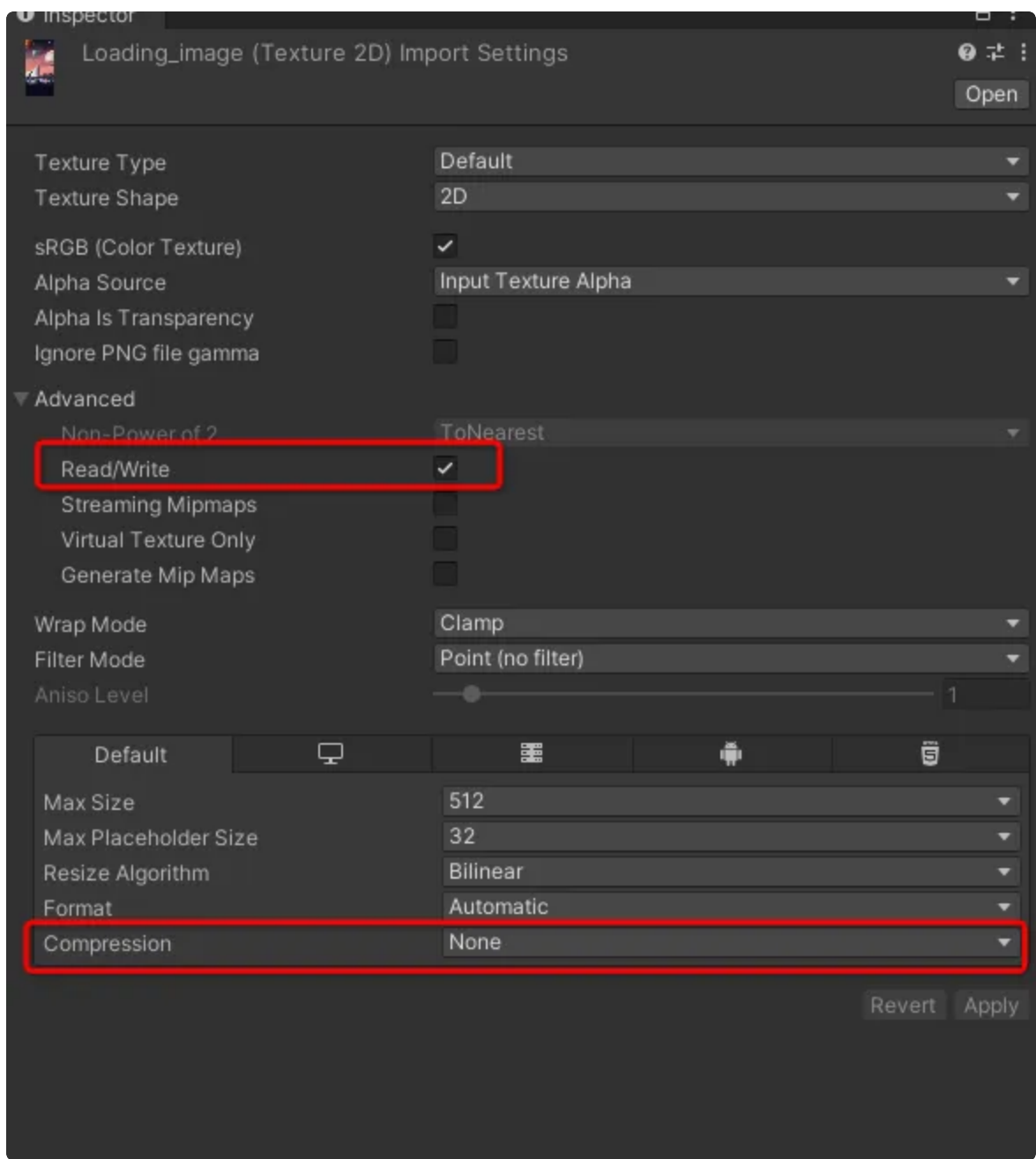


4. 生成产物后，会按照 选定图 + 默认文案 + 默认进度条 的方式呈现loading插件页。如果需要对这些内容进行自定义，则在game.js中修改对应的配置项即可。

Loading图注意事项

1. Loading图不要使用特别大的，影响展示加载时间；
2. Loading图需要勾选Read/Write，否则会编码失败；

3. Compressiony请选择None。



Loading字段说明

game.js文件示例如下：


```

1 ▾ const loaderOptions = {
2     streamingAssetsUrl: '',
3     data: {},
4     loadingPageConfig: {} // loading插件配置入口
5 };
6 ▾ async function main() {
7     const unityInstance = await my.loadUnity(loaderOptions);
8 }
9 ▾ main().catch((err) => {
10     console.error(err);
11 });

```

loadingPageConfig参数

属性	类型	默认值	单位	必填	说明
scaleMode	string	"centerCrop"		否	loading图片的适配模式。 有"centerCrop"、 "fitXY"、"fitCenter" "三种值类型。
designWidth	number	0	px	否	设计宽度，默认全屏
designHeight	number	0	px	否	设计高度，默认全屏
textConfig	TextConfig			否	自定义的文案设置
barConfig	BarConfig			否	自定义的进度条设置
materialConfig	MaterialConfig			是	封面图配置

scaleMode取值说明：

- centerCrop：完全填充。如果图片的宽高比例与屏幕的宽高比例不同，则会从图片的两侧裁剪掉多余的部分，使得图片的宽高比例与图片的宽高比例相同。

- fitXY: 完全填充。会缩放图片以填满屏幕的宽度和高度，当图片的宽高比例与屏幕不一致时，会引起图片的宽高比例变化。
- fitCenter: 非完全填充。会缩放图片以使其宽度和高度都尽可能接近屏幕的宽度和高度，如果图片的宽高比例与与的宽高比例不同，则会将图片缩放到屏幕的宽度和高度中的较小值，然后将图片放在屏幕的中央。

TextConfig参数

属性	类型	默认值	单位	必填	说明
firstStartText	string	"首次加载请耐心等待"		否	首次启动时提示文案
downloadingText	Array<string>	['正在加载资源']		否	加载阶段循环展示的文案，文案切换时间为textDuration
compilingText	string	"编译中"		否	编译阶段文案
completeText	string	"初始化中"		否	初始化阶段文案
textDuration	number	1500	ms	否	downloadingText有多个文案时，每个文案展示时间
style	Style			否	文案样式

Style参数

属性	类型	默认值	单位	必填	说明
width	number	0	px	否	文字宽度
height	number	0	px	否	文字高度
bottom	number	100	px	否	底边距
lineHeight	number	0	px	否	
color	string	"#ffffff"		否	
fontSize	number	14	px	否	字体大小

BarConfig参数

属性	类型	默认值	单位	必填	说明
style	Style			否	进度条样式

Style参数

属性	类型	默认值	单位	必填	说明
width	number	960–竖屏 1800–横屏	px	否	进度条宽度
height	number	40	px	否	进度条高度
padding	number	0	px	否	
bottom	number	40	px	否	进度条底边距
backgroundCo lor	string	"#07C160"		否	进度条颜色（已加载 的进度）
defaultBackgr oundColor	string	"#802b2b2b"		否	进度条颜色（未加载 的进度）

MaterialConfig参数

属性	类型	默认值	单位	必填	说明
backgroundlma ge	string			是	背景图片，使用小游 戏包内图片；（必 传）

三、游戏预览和上传

使用支付宝扫码预览请注意：支付宝版本需要 $\geq 10.6.20$

未来预览与上传都将在支付宝开发者工具IDE支持，并提供分包功能

3.1、安装支付宝小游戏开发工具

```
1 npm install -g minidev@latest
```

3.2、登录支付宝开发者账号

执行 `minidev login` 脚本并扫码登陆开发者账号。



登陆成功后会显示“完成授权”



3.3、预览

```
1 minidev preview -a your_appId -p /path/to/project
```

`your_appId`: 一定是要申请的小游戏的应用 `appId`，不然会预览失败

- 示例

JSON |

```
1 #进入你的项目目录
2 cd /path/to/your/project
3
4 minidev preview -a 206017xxx
```

✓ 构建成功

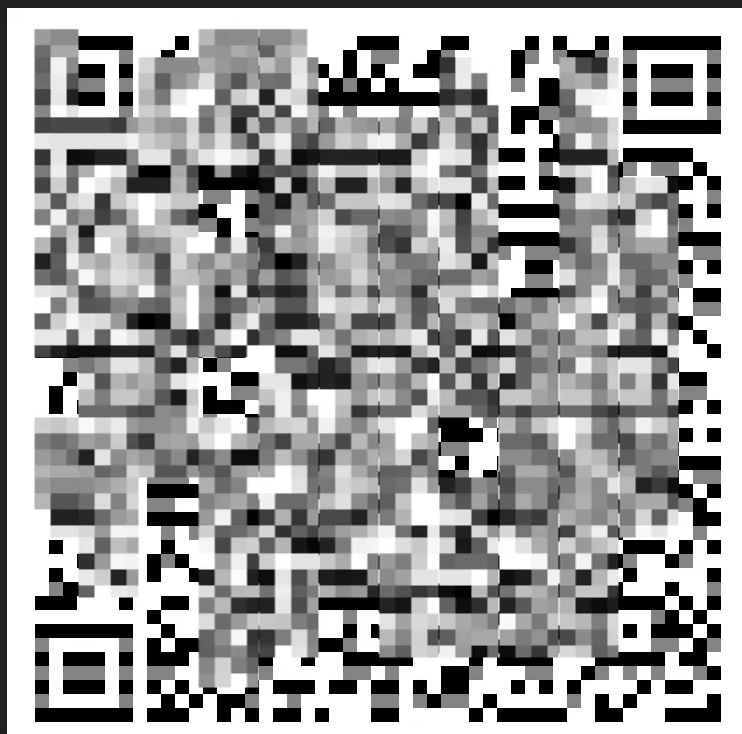
构建产物地址: /var/folders/tv/sxywhwq93f343jfx25s07sd80000gp/T/minidev_dists/build-78c6860ac0

✓ 生成预览码成功

二维码图片地址 <https://mobilecodec.alipay.com/show.htm?code=s5x14617b9bmorjbfmgr1c>

预览包版本: 0.2.2206291704.34

△二维码图片展示失败, 尝试使用 `schema url` 重新生成



3.4、上传

```
1 minidev upload -a your_appId -p /path/to/project --game
```

参数	是否必填	说明
-a	是	指定应用id
--game	是	小游戏应用必填
-p	否	指定工程路径；当前在 <code>game.json</code> 同级目录时，可缺省
-v	否	指定版本

- 示例

```
1 # 假设你已位于项目源码目录：
2 minidev upload -a 206017xxx -v 0.0.1 --game
```

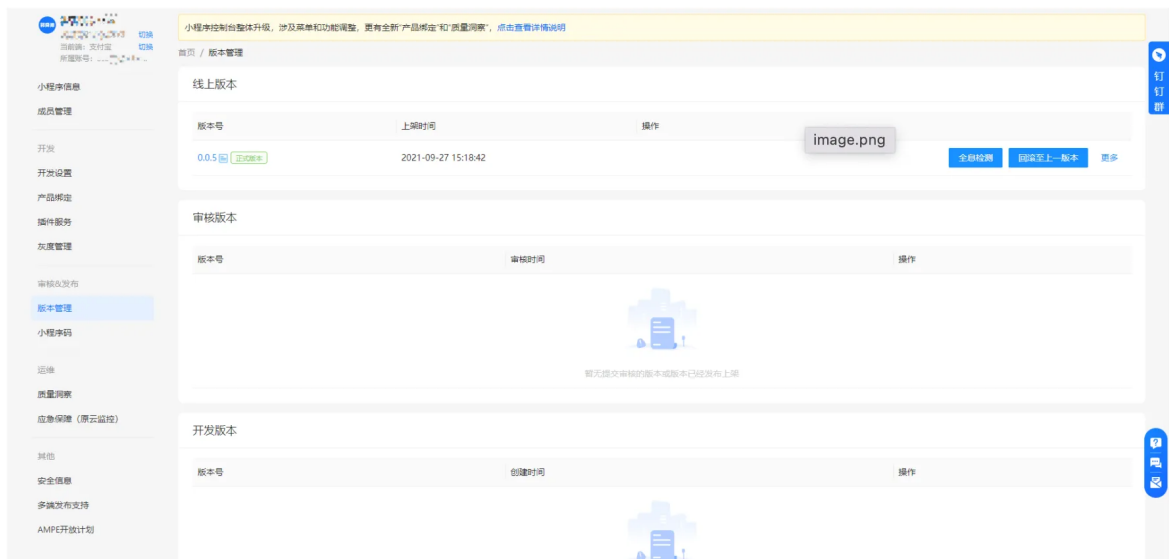
3.5、体验码

上传之后，在开放平台，就可以对上传的版本生成体验码；

设置体验版（可选）

如果在小程序上架之前需要让部分用户先体验小程序，可以设置体验版。

1. 在 **成员管理** 中添加体验成员，一个小程序可以最多添加 50 个体验成员。
2. 在 **开发版本** 中，选择一个开发版本设置为体验版。
3. 体验者可扫描体验版二维码进行体验。



四、游戏审核和上线

审核通过后，在[小程序官网](#)进行发布。



小程序文档 – 支付宝文档中心

支付宝文档中心

支付宝文档中心

五、游戏运营

支付宝中提供了多种游戏运营任务可供接入：

此处为语雀内容卡片，[点击查看](https://www.yuque.com/zhifubaoxiaoyouxi-zromu/ts2ry9/ihmbgogcv7368ukz?view=doc_embed)：https://www.yuque.com/zhifubaoxiaoyouxi-zromu/ts2ry9/ihmbgogcv7368ukz?view=doc_embed

此处为语雀内容卡片，[点击查看](https://www.yuque.com/zhifubaoxiaoyouxi-zromu/ts2ry9/sepa1wm2kb8sf63q?view=doc_embed)：https://www.yuque.com/zhifubaoxiaoyouxi-zromu/ts2ry9/sepa1wm2kb8sf63q?view=doc_embed

此处为语雀内容卡片，点击链接查看：https://www.yuque.com/zhifubaoxiaoyouxi-zromu/ts2ry9/sua2e6lxc8gm24lw?view=doc_embed

六、Alipay SDK API

为了让 Unity 小游戏能够需要使用支付宝平台的一些能力，支付宝开发了一个 Unity SDK。Unity 小游戏业务层只需要集成支付宝这边开发的 Unity SDK，就可以使用 SDK 中的方法能调用支付宝平台的一些能力。

为了降低wasm包大小，运行时内存，新版SDK改用了LitJson替代原来的Newtonsoft。

有改动到的API将JsonObject改为JsonData即可

一、CanIUse，判断 API 能否使用

支付宝支持版本：**10.5.30** 及以上（**10.5.30** 版本之前调用 **CanIUse** 会没有任何反应，需要兼容处理）

检查一个 API 在当前的支付宝版本中是否支持，可以先测试一下当前能否调用CanIUse是否起作用再做API使用。

1. API 说明

Api 名称	Api 声明	Api 参数	Api 返回值
CanIUse	CanIUse(string api, Action<String> result)	1.api 2.result	为Action<String>的返回值，是一个json string

2. 参数描述

参数名	类型	默认值	是否必填	说明
api	string		是	查询的api名称

result	Action<String >	是	接收返回值，返回值为一个json string
--------	--------------------	---	-------------------------

3. 调用示例:

```

1
2  bool CanIUseWork = false;
3  //判断支付宝当前版本是否能调用CanIUse接口是否支持
4  private IEnumerator CanIUseAPICheck()
5  {
6      Debug.Log("判断接口可否使用, 接口: ");
7      AlipaySDK.API.CanIUse("tempAPI", (result) =>
8      {
9          Debug.Log(result);
10         if (!string.IsNullOrEmpty(result))
11         {
12             CanIUseWork = true;
13             Debug.Log("CanIUse 接口可以使用! ");
14         }
15     });
16     yield return new WaitForSeconds(1f);
17     if(!CanIUseWork)
18         Debug.LogError("CanIUse 接口无法使用, 支付宝版本不支持");
19     yield break;
20 }
21
22 private void CanIUse()
23 {
24     if (!CanIUseWork)
25     {
26         StartCoroutine(CanIUseAPICheck());
27         return;
28     }
29     Debug.Log("判断接口可否使用");
30     AlipaySDK.API.CanIUse("接口名", (result) =>
31     {
32         Debug.Log(result);
33     });
34 }

```

二、获取用户授权

支付宝支持版本：**10.5.20** 及以上

GetAuthCode 获取用户信息授权，取得授权码（authCode）。

GetAuthCode 引导用户授权其信息给当前小程序，取得的代表用户授权的授权码（authCode），后续需由小程序服务端使用，向支付宝换取用户信息（如 user_id、头像、昵称、手机号、地区、性别、出生日期等）。

这种方式较通用，但也相对复杂。

服务端对接文档参考：

<https://opendocs.alipay.com/mini/api/openapi-authorize>

<https://opendocs.alipay.com/mini/05dxgc?pathHash=1a3ecb13>

<https://opendocs.alipay.com/mini/02bzs?pathHash=02aefd59>

1. API 说明

Api 名称	Api 声明	Api 参数	Api 返回值
GetAuthCode	GetAuthCode(string[] scopes, Action<String> result)	1.scopes 2.result	为Action<String>的返回值，是一个json string

2. 参数描述

参数名	类型	默认值	是否必填	说明
scopes	string[]		是	1. auth_base：授权获取支付宝会员唯一标识（user_id）。此方式为静默授权，不会弹出授权浮窗。 2. auth_user：授权获取支付宝会员信息。
result	Action<String>		是	接收返回值，返回值为一个json string

3. 调用示例

```
1 private Button userInfoBaseBtn;  
2 private Button userInfoUserBtn;  
3  
4 userInfoBaseBtn.onClick.AddListener(() =>  
5 {  
6     Debug.Log("获取用户信息 base");  
7     string[] scopes = { "auth_base" };  
8     AlipaySDK.API.GetAuthCode(scopes, result =>  
9     {  
10        Debug.Log(result);  
11    });  
12 });  
13 userInfoUserBtn.onClick.AddListener(() =>  
14 {  
15     Debug.Log("获取用户信息 user");  
16  
17     string[] scopes = { "auth_user" };  
18     AlipaySDK.API.GetAuthCode(scopes, result =>  
19     {  
20        Debug.Log(result);  
21    });  
22 });
```

4. 返回值示例

```

1  当请求参数为 "auth_base" 时:
2  {
3      "authCode": "f380cd7xxxxxxxxxxxxcf7RX00",
4      "authErrorScopes": {},
5      "authSuccessScopes": [
6          "auth_base"
7      ],
8      "authcode": "f380cd7xxxxxxxxxxxxcf7RX00"
9  }
10
11  当请求参数为 "auth_user" 时:
12  {
13      "authCode": "4b82ed17xxxxxxxxxxxx9d79EX00",
14      "authErrorScopes": {},
15      "authSuccessScopes": [
16          "auth_user"
17      ],
18      "authcode": "4b82ed17xxxxxxxxxxxx9d79EX00"
19  }

```

三、获取基本信息

支付宝支持版本：**10.5.30** 及以上

1. API 说明

Api 名称	Api 声明	Api 参数	Api 返回值	说明
GetSystemInfo	GetSystemInfo (Action<string> result)	1.result	为 Action<String> 的返回值，是一 个json string	异步 获取基本信 息
GetSystemInfoS ync	string GetSystemInfoSyn c();		返回值，是一个 json string	同步 获取基本信 息
GetAlipayEnv	string GetAlipayEnv();			

2. 参数描述

参数名	类型	默认值	是否必填	说明
result	Action<String >		是	接收返回值，返回值为一个json string

3. 调用示例：

```

1  private void GetSystemInfo()
2  {
3      Debug.Log("获取设备基本信息");
4
5      AlipaySDK.API.GetSystemInfo((result) =>
6      {
7          Debug.Log(result);
8      });
9  }
10
11 private void GetSystemInfoSync()
12 {
13     Debug.Log("同步-----环境检查");
14     var result = AlipaySDK.API.GetSystemInfoSync();
15
16     if (string.IsNullOrEmpty(result))
17     {
18         Debug.Log("获取启动参数失败");
19         return;
20     }
21     Debug.Log("同步-----环境检查-----返回结果: " + result);
22 }
23
24 private void GetAlipayEnv()
25 {
26     var info = AlipaySDK.API.GetAlipayEnv();
27     Debug.Log("获取环境信息: " + info);
28 }
```

4. 返回值示例

```

1  SystemInfo
2  {
3      "system": "17.5.1",
4      "platform": "iOS",
5      "model": "iPhone15,4",
6      "pixelRatio": 3,
7      "windowHeight": 852,
8      "screenWidth": 393,
9      "screenHeight": 852,
10     "version": "10.6.10.00002698",
11     "windowWidth": 393,
12     "language": "",
13     "isIphoneXSeries": false
14 }
15 Env
16 {
17     "USER_DATA_PATH": "https://usr",
18     "renderType": "paladin",
19     "clientName": "ap",
20     "clientVersion": "10.6.50",
21     "language": "zh-Hans",
22     "platform": "android",
23     "pixelRatio": 2.75,
24     "screenHeight": 2400,
25     "screenWidth": 1080,
26     "version": "2.1.22"
27 }

```

- SystemInfo

名称	类型	描述
model	String	手机型号
pixelRatio	Number	设备像素比
windowWidth	Number	窗口宽度
windowHeight	Number	窗口高度
screenWidth	Number	屏幕像素宽度
screenHeight	Number	屏幕像素高度
language	String	支付宝设置的语言

version	String	支付宝版本号
system	String	系统版本
platform	String	Android/iOS

- env

名称	类型	描述
USER_DATA_PATH	String	文件系统中的用户目录路径 (本地路径)。基础库1.13.0开始支持。
renderType	String	渲染框架
clientName	String	客户端名称简写。 支付宝客户端为 'ap'
pixelRatio	Number	设备像素比
screenWidth	Number	屏幕像素宽度
screenHeight	Number	屏幕像素高度
language	String	支付宝设置的语言
clientVersion	String	支付宝版本号
version	String	前端版本
platform	String	Android/iOS

四、获取启动参数

支付宝支持版本：**10.5.50** 及以上

1. API 说明

Api 名称	Api 声明	Api 参数	Api 返回值
--------	--------	--------	---------

GetLaunchOptions	GetLaunchOptions(Action<String> result);	1. result	为Action<String>的返回值，是一个json string
------------------	--	-----------	------------------------------------

2. 参数描述

参数名	类型	是否必填	说明
result	Action<String>	是	接收返回值，返回值为一个json string

3. 调用示例：

```

C#
1 private void GetLaunchOptions()
2 {
3     Debug.Log("获取启动参数");
4     AlipaySDK.API.GetLaunchOptions((result) =>
5     {
6         Debug.Log(result);
7     });
8 }
9
10 //返回格式样例: {"query":{"key1":"value1","key2":"value2"},"referrerInfo":
    {}

```

五、键盘

Api 名称	Api 声明
ShowKeyboard	void ShowKeyboard(string defaultValue, int maxLength, bool multiple, bool confirmHold, string confirmType, Action<string> result);
HideKeyboard	void HideKeyboard(Action<string> result);
OnKeyboardInput	void OnKeyboardInput(Action<String> result);
OnKeyboardConfirm	void OnKeyboardConfirm(Action<String> result);

OnKeyboardComplete	void OnKeyboardComplete(Action<String> result);
OffKeyboardInput	void OffKeyboardInput(Action<String> result);
OffKeyboardConfirm	void OffKeyboardConfirm(Action<String> result);
OffKeyboardComplete	void OffKeyboardComplete(Action<String> result);

1. 显示键盘

参数

属性	类型	默认值	必填	说明
defaultValue	string		是	键盘输入框显示的默认值
maxLength	number		是	键盘中文本的最大长度
multiple	boolean		是	是否为多行输入
confirmHold	boolean		是	当点击确认时键盘是否收起 (false时是收起)
confirmType	string		是	键盘右下角 confirm 按钮的类型, 只影响按钮的文本内容

object.confirmType 的合法值

值	说明	最低版本
done	完成	
next	下一个	
search	搜索	
go	前往	
send	发送	

2. 使用示例

将该脚本挂载至Unity组建InputField上。具体使用可参考以下代码：

```
1 using System;
2 using AlipaySdk;
3 using Newtonsoft.Json.Linq;
4 using UnityEngine;
5 using UnityEngine.EventSystems;
6 using UnityEngine.UI;
7
8 public class AlipayInput : MonoBehaviour, IPointerClickHandler, IPointerExitHandler
9 {
10     private InputField inputField;
11     private bool isShowKeyboard = false;
12
13     public void Awake()
14     {
15         inputField = GetComponent<InputField>();
16     }
17
18     public void OnPointerClick(PointerEventData eventData)
19     {
20         inputField.OnPointerClick(eventData);
21         ShowKeyboard();
22     }
23     public void ShowKeyboard()
24     {
25         if (!isShowKeyboard)
26         {
27             AlipaySDK.API.ShowKeyboard(inputField.text, 100, true, false,
28             "go", (result) =>
29             {
30                 Debug.Log("打开键盘");
31             });
32             AlipaySDK.API.OnKeyboardComplete(OnComplete);
33             AlipaySDK.API.OnKeyboardConfirm(OnConfirm);
34             AlipaySDK.API.OnKeyboardInput(OnInput);
35             isShowKeyboard = true;
36         }
37     }
38     private void OnInput(string result)
39     {
40         Debug.Log("用户输入 OnInput: " + result);
41     }
42     private void OnConfirm(string result)
43     {
44         Debug.Log("用户输入 OnConfirm: " + result);
45     }
46 }
```

```

44         if (!result.Contains("value"))
45     {
46         Debug.LogError("Keyboard result not contains value");
47         return;
48     }
49     var inputVar = JObject.Parse(result);
50     var inputResult = inputVar["value"].ToString();
51     inputField.text = inputResult;
52     HideKeyboard();
53 }
54 private void OnComplete(string result)
55 {
56     Debug.Log("用户输入 OnComplete: " + result);
57     if (!result.Contains("value"))
58     {
59         Debug.LogError("Keyboard result not contains value");
60         return;
61     }
62     var inputVar = JObject.Parse(result);
63     var inputResult = inputVar["value"].ToString();
64     inputField.text = inputResult;
65     HideKeyboard();
66 }
67 public void HideKeyboard()
68 {
69     if (isShowKeyboard)
70     {
71         AlipaySDK.API.OffKeyboardComplete(OnComplete);
72         AlipaySDK.API.OffKeyboardConfirm(OnConfirm);
73         AlipaySDK.API.OffKeyboardInput(OnInput);
74         AlipaySDK.API.HideKeyboard((result) =>
75     {
76         Debug.Log("关闭键盘");
77     });
78         isShowKeyboard = false;
79     }
80 }
81 public void OnPointerExit(PointerEventData eventData)
82 {
83     if (!inputField.isFocused)
84     {
85         HideKeyboard();
86     }
87 }
88 }

```

六、广告

广告接口与IG方案调用上略有不同，已经接入了ig方案的广告方案需要调整调用。调用方式与普通小游戏类似。

Api 名称	Api 声明	返回	解释
GetAdManager	AlipayAdManager GetAdManager();	AlipayAdManager	获取广告对象
CreateRewardAd	AlipayRewardAdEntity CreateRewardAd(String adUnitId)	AlipayRewardAdEnti ty	创建视频激励 广告
CreateInterstitialAd	AlipayInterstitialAdEntity CreateInterstitialAd(String adUnitId)	AlipayInterstitialAdE ntity	创建插屏广告

1. 激励广告---RewardAd

一、API 说明

API调用	API类型	返回	解释
AlipayRewardAdEntity.Load	方法	void	加载新的广告
AlipayRewardAdEntity.Show	方法	void	展示广告
AlipayRewardAdEntity.OnLoad	事件监听	void	监听激励广告加载事件
AlipayRewardAdEntity.OnError	事件监听	string	监听激励广告错误事件
AlipayRewardAdEntity.OnClose	事件监听	bool	监听用户点击 关闭广告 按钮的事件

二、广告错误信息以及解决方案

错误码是通过onError获取到的错误信息。调试期间，可以通过异常返回来捕获信息。在小游戏发布上线之后，如果遇到异常问题，可以在“运维中心“里面搜寻错误日志，还可以针对异常返回加上适当的监控信息。

错误码	错误信息	解决方案
61000	广告返回空	建议检查广告单元配置是否成功或服务端疲劳度配置
61001	广告请求超时	建议检查网络环境是否正常，服务器是否稳定
61002	广告请求错误	建议检查参数配置，重新请求接口
61003	广告单元重复创建	建议确认是否创建了重复的广告单元，删除重复广告，再进行创建广告视图
61004	广告视图创建失败	建议检查参数配置，重新创建广告视图
2	参数类型错误	建议检查请求时传递的数据是否正常，格式是否正确，可以在请求前打印下入参数据日志
3	未知错误	建议检查参数是否正常，重新请求参数

2. 插屏广告---InterstitialAd

一、API 说明

API调用	API类型	返回	解释
AlipayInterstitialAdEntity.Destroy	方法	void	销毁插屏广告实例
AlipayInterstitialAdEntity.Show	方法	void	展示广告
AlipayInterstitialAdEntity.OnLoad	事件监听	void	监听插屏广告加载事件
AlipayInterstitialAdEntity.OnError	事件监听	string	监听插屏广告错误事件
AlipayInterstitialAdEntity.OnClose	事件监听	bool	监听用户点击 关闭广告 按钮的事件

二、广告错误信息以及解决方案

错误码	错误信息	解决方案
61000	广告返回空	建议检查广告单元配置是否成功或服务端疲劳度配置
61001	广告请求超时	建议检查网络环境是否正常，服务器是否稳定
61002	广告请求错误	建议检查参数配置，重新请求接口

61003	广告单元重复创建	建议确认是否创建了重复的广告单元，删除重复广告，再进行创建广告视图
61004	广告视图创建失败	建议检查参数配置，重新创建广告视图
2	参数类型错误	建议检查请求时传递的数据是否正常，格式是否正确，可以在请求前打印下入参数据日志
3	未知错误	建议检查参数是否正常，重新请求参数

3. 调用示例：

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using AlipaySdk;
4 using AlipaySdk.Ad;
5 using UnityEngine;
6
7 public class AlipayAD
8 {
9
10     private AlipayInterstitialAdEntity adInterstitialEntity;
11     private AlipayRewardAdEntity adRewardEntity;
12     private void CreateRewardAd()
13     {
14         //创建激励广告实体
15         adRewardEntity = AlipaySDK.API.GetAdManager().CreateRewardAd("GAME
16         _TEST_VIDEO_1");
17         adRewardEntity.OnClose += (isEnded) =>
18         {
19             Debug.Log("激励广告关闭, 播放是否成功: " + isEnded);
20             adRewardEntity.Load();
21         };
22         adRewardEntity.OnError += (result) =>
23         {
24             Debug.Log("激励广告异常: " + result);
25         };
26         adRewardEntity.OnLoad += () =>
27         {
28             Debug.Log("激励广告加载完毕");
29         };
30         adRewardEntity.Show();
31     }
32
33     public void LoadRewardAD()
34     {
35         if (adRewardEntity == null)
36         {
37             CreateRewardAd();
38         }
39         else
40         { //预加载激励广告
41             adRewardEntity.Show();
42         }
43     }
44
45     public void LoadInterstitialAD()
```



```

45     {
46         Debug.Log("加载展示插屏广告");
47         //创建插屏广告实体
48         adInterstitialEntity = AlipaySDK.API.GetAdManager().CreateIntersti
49 tialAd("GAME_TEST_TABLE_SCREEN");
50         adInterstitialEntity.OnClose += (isEnded) =>
51     {
52         Debug.Log("插屏广告关闭, 播放是否成功: " + isEnded);
53         adInterstitialEntity.Destroy();
54     };
55     adInterstitialEntity.OnError += (result) =>
56     {
57         Debug.Log("插屏广告异常: " + result);
58     };
59     adInterstitialEntity.OnLoad += () =>
60     {
61         Debug.Log("插屏广告加载完毕 ");
62         adInterstitialEntity.Show();
63     };
64     }
65 }
66

```

4. 注意事项

- 调用广告API时, 填写的的adUnitId应该是spaceCode, 而非APPID



- 开发阶段可以使用测试adUnitId (视频: GAME_TEST_VIDEO_1, 插屏: GAME_TEST_TABLE_SCREEN) 来做测试, 申请到正式的spaceCode后, 再替换掉

七、分享

分享接口与IG方案调用上略有不同，已经接入了ig方案的广告方案需要调整调用。

调用方式与普通小游戏类似。

1. API 说明

Api 名称	Api 声明	Api 参数	Api 返回值
OnShareAppMessage	void OnShareAppMessage(string title, JObject param, Action<String> result);	1. title 2. param 3. result	为Action<String>的返回值，是一个json string
ShowSharePanel	void ShowSharePanel(Action<String> result);	1. result	

2. OnShareAppMessage

设置页面的分享信息，通过OnShareAppMessage注册分享信息，当调用my.showSharePanel 或者点击分享面板的时候会调用。

具体return返回参数，参考：[支小程序文档 - 支付宝文档中心](#)

一、string title

属性	类型	必填	描述
title	String	是	自定义分享标题。

二、JObject param

json对象，非必填项，若无可填null

属性	类型	必填	描述
----	----	----	----

desc	String	否	自定义分享描述：由于分享到微博只支持最大长度 140 个字，因此建议长度不要超过该限制。
query	String	否	查询字符串，必须是 key1=val1&key2=val2 的格式。从这条转发消息进入后，可通过 <code>my.getLaunchOptionsSync()</code> 获取启动参数中的 query。
content	String	否	自定义歧口令文案，最多 28 个字符。
bglmgUrl	String	否	自定义分享预览大图，建议尺寸 750x825，支持：网络图片路径；不支持：base64。
sclmgUrl	String	否	自定义社交图片链接，作为分享到支付宝好友时的主体图片。建议尺寸 376x330。 客户端 10.2.50 开始支持。 可通过 <code>my.canIUse('page.onShareAppMessage.return.sclmgUrl')</code> 进行检测。 支持：网络图片路径；不支持：base64。
imageUrl	String	否	自定义分享小图 icon 元素，支持：网络图片路径；不支持：base64。
searchTip	String	否	生成分享截图的搜索引导，设置该参数后，会在分享图片中增加上支付宝搜“设置关键字”的内容，设置关键字不能超过 5 个字。

3. ShowSharePanel

唤起分享面板；需要先设置分享信息。OnShareAppMessage

4. 调用示例：

```

1  public Button shareBtn;
2  void Start()
3  {
4      shareBtn.onClick.AddListener(() =>
5      {
6          Debug.Log("截图分享");
7          //设置分享信息
8          AlipaySDK.API.OnShareAppMessage("unity WebGL game", null, (result
9          ) =>
10         {
11             Debug.Log("分享信息设置 结果: " + result);
12         });
13         //拉起分享面板
14         AlipaySDK.API.ShowSharePanel(result =>
15         {
16             Debug.Log("显示分享面板: " + result);
17         });
18     });
19 }

```

八、游戏内购

在正式对接游戏内购能力之前，商家需要先签约和准入，详见商家操作手册：

此处为语雀内容卡片，[点击查看](https://www.yuque.com/zhifubaoxiaoyouxi-zromu/ts2ry9/uzf5t7ho6gmnpm5t?view=doc_embed)：https://www.yuque.com/zhifubaoxiaoyouxi-zromu/ts2ry9/uzf5t7ho6gmnpm5t?view=doc_embed

此处为语雀内容卡片，[点击查看](https://www.yuque.com/zhifubaoxiaoyouxi-zromu/ts2ry9/vkqf16z0gkn0zhlg?view=doc_embed)：https://www.yuque.com/zhifubaoxiaoyouxi-zromu/ts2ry9/vkqf16z0gkn0zhlg?view=doc_embed

此处为语雀内容卡片，[点击查看](https://www.yuque.com/zhifubaoxiaoyouxi-zromu/ts2ry9/wuey5abbr0fzcgj1?view=doc_embed)：https://www.yuque.com/zhifubaoxiaoyouxi-zromu/ts2ry9/wuey5abbr0fzcgj1?view=doc_embed

此处为语雀内容卡片，[点击查看](https://www.yuque.com/zhifubaoxiaoyouxi-zromu/ts2ry9/nia4lx40aqq2qvme?view=doc_embed)：https://www.yuque.com/zhifubaoxiaoyouxi-zromu/ts2ry9/nia4lx40aqq2qvme?view=doc_embed

1. API 说明

Api 名称	Api 声明	Api 参数	Api 返回值
--------	--------	--------	---------

RequestGamePayment	RequestGamePayment(String customId, String buyQuantity, Action<String> result)	1.customId 2.buyQuantity 3.result	为Action<String>的返回值，是一个json string
--------------------	--	---	------------------------------------

2. 参数描述

参数名	类型	默认值	是否必填	说明
customId	string		是	游戏开发者自定义的唯一订单号，订单支付成功后通过服务端支付结果回调回传 必须具有唯一性，如果不传或重复传相同值，则会报错
buyQuantity	number		是	金币购买数量，金币数量必须满足：金币数量*金币单价 = 限定价格等级
result	Action<String>		是	接收返回值，返回值为一个json string

3. 注意事项

- 支付时使用扫脸支付，小游戏会被压到后台，需要手动点击返回小游戏。

4. 结果错误码

错误码	描述	解决方案
2	接口入参无效。	检查入参必填字段是否正确。
3	未知异常。	接口调用环境校验异常，建议重试。
4	无权限调用（N22104）。	个人小程序应用没有开放小程序支付能力。
9000	订单处理成功。	不建议根据 my.requestGamePayment 接口同步返回判断是否充值成功，9000 不能判定就是充值成功，建议以服务端回调或者订单查询接口结果为准。

8000	正在处理中。支付结果未知（有可能已经支付成功）。	请调用 alipay.user.gamecenter.payment.querystatus （充值状态查询接口）查询订单的充值状态，以查询接口实际返回的充值状态为准。
4000	订单处理失败。	生成的 tradeNO 无效，请重试
6001	用户中途取消。	建议重新创建一个游戏侧充值订单，让用户重新支付
6002	网络连接出错。	检查网络连接后重试。
6004	处理结果未知（有可能已经成功）。	请调用 alipay.user.gamecenter.payment.querystatus （充值状态查询接口）查询订单的充值状态，以查询接口实际返回的充值状态为准。
60001	生成订单接口请求异常。	服务请求异常，建议重试。
60002	生成订单接口请求失败。	服务请求失败，建议重试。
60004	生成订单接口请求异常。	服务请求异常，建议重试。
15002	请求参数非法	请检查请求参数是否错误
15204	充值功能受限	请咨询游戏中心平台业务
15003	未开通游戏币充值业务	请向游戏中心平台开通充值业务
15203	下单速度过快，请勿频繁操作	请防止用户重复快速触发充值支付
15202	该订单已关闭，请重新发起新的交易	请重新创建一个游戏侧充值订单
15205	该订单处理中，请勿重新请求	建议接入服务端回调，或者调用 alipay.user.gamecenter.payment.querystatus （充值状态查询接口）查询订单的充值状态，以查询接口实际返回的充值状态为准。
15201	充值数量不符合规则	请检查充值金币数量是否符合限定价格等级规则
15206	订单参数异常，请重新发起新的交易	请检查请求参数是否错误

15004	系统繁忙，请稍后重试	订单处理中，建议接入服务端回调，或者调用 alipay.user.gamecenter.payment.querystatus (充值状态查询接口) 查询订单的充值状态，以查 询接口实际返回的充值状态为准。
-------	------------	--

5. buyQuantity 限制说明

购买游戏币的数量，换算成 RMB 必须满足以下价格档位，即 $\text{buyQuantity} * \text{游戏币单价} = \text{限定价格等级}$ 。
如：游戏币单价为 0.1 元，一次购买数量至少是 10 个。

限定价格等级 (单位: 元)
1
3
6
8
12
18
25
30
40
45
50
60
68
73
78
88
98

108
118
128
148
168
188
198
328
648
1288
1998
2998

6. 返回数据示例

发起内购创建订单成功返回的json数据示例：

```
Plain Text |  
1  {  
2    "extendInfo":{  
3      "doNotExit":true,  
4      "isDisplayResult":true,  
5      "tradeNo":"xxxxxxxxxxxxx"  
6    },  
7    "memo":"","  
8    "result":"app_name=\"alipay\"&trade_no=\"xxxxxxx\"&  
9    biz_type=\"trade\"&success=\"true\"",  
10   "resultCode":"9000"  
11  }
```

发起内购创建订单失败返回的json数据示例（不符合限定价格等级）：


```

1  {
2      "success": false,
3      "errorCode": "15201",
4      "errorMsg": "游戏币充值数量不符合限定价格等级",
5      "retryable": false,
6      "serializedSize": 61,
7      "unknownFieldsSerializedSize": 0
8  }

```

7. 调用示例

```

1  public Button topupBtn;
2  void Start()
3  {
4      topupBtn.onClick.AddListener(() =>
5      {
6          Debug.Log("内购");
7          String customId = GetTime() + "";
8          AlipaySDK.API.RequestGamePayment(customId, 100, (result) =>
9          {
10             Debug.Log(result);
11         });
12     });
13 }

```

九、跳转到其他小程序（小游戏）

1. API 说明

Api 名称	Api 声明	Api 参数	Api 返回值
NavigateToMiniProgram	void NavigateToMiniProgram(string appId, string paramKey, JObject param, Action<String> result);	1. result	为Action<String>的返回值，是一个json string
OpenURL	void OpenURL(string url, Action<string> result);	1. url	

2. 参数描述

- NavigateToMiniProgram

参数名	类型	是否必填	说明
appid	string	是	跳转的目标appid
paramKey	string	是	参数key ("extraData")
param	JSONObject	是	传递的参数值
result	Action<String>	是	回调

- OpenURL

参数名	类型	是否必填	说明
url	string	是	跳转的链接
result	Action<String>	是	回调

3. 调用示例:

```
1 private void NavigateToMiniProgram()
2 {
3     Debug.Log("跳转其他小程序");
4     JObject param = new JObject();
5     param.Add("data1", "test");
6     AlipaySDK.API.NavigateToMiniProgram("APPID", "extraData", param, (result) =>
7     {
8         Debug.Log($" NavigateToMiniProgram : {result}");
9     });
10 }
11
12 private void OpenURL()
13 {
14     Debug.Log("复访任务");
15     string openURL = string.Empty;
16     string systemInfo = AlipaySDK.API.GetSystemInfoSync();
17
18     Debug.Log("当前 SystemInfo : " + systemInfo);
19     if (systemInfo.Contains("iOS"))
20     {
21         openURL = "alipays://platformapi/startapp?appId=2060090000285522&url=https%3A%2F%2Frender.alipay.com%2Fp%2Fyuyan%2F180020010001210691%2Findex.html%3FcaprMode%3Dsync&sourceAppId=2021003125685383&sourceUrl=alipays%3A%2F%2Fplatformapi%2Fstartapp%3FappId%3D2021003125685383%26url%3Dhttps%253A%252F%252Frender.alipay.com%252Fp%252Fyuyan%252F180020010001206617%252Findex.html%253FcaprMode%253Dsync%26chInfo%3Dreturnvisit%26sms%3DYES%26appClearTop%3Dfalse%26startMultApp%3DYES";
22     }
23     else
24     {
25         openURL = "alipays://platformapi/startapp?appId=2060090000285522&url=https%3A%2F%2Frender.alipay.com%2Fp%2Fyuyan%2F180020010001210691%2Findex.html%3FcaprMode%3Dsync&sourceAppId=2021003125685383&sourceUrl=alipays%3A%2F%2Fplatformapi%2Fstartapp%3FappId%3D2021003125685383%26url%3Dhttps%253A%252F%252Frender.alipay.com%252Fp%252Fyuyan%252F180020010001206617%252Findex.html%253FcaprMode%253Dsync%26chInfo%3Dreturnvisit%26sms%3DYES%26appClearTop%3Dfalse";
26     }
27
28     AlipaySDK.API.OpenURL(openURL, (result) =>
29     {
30         Debug.Log("打开URL返回信息: " + result);
31     });
32 }
```

十、退出当前小游戏

1. API 说明

Api 名称	Api 声明	Api 参数	Api 返回值
ExitProgram	void ExitProgram(Action<String> result);	1. result	为Action<String>的返回值，是一个json string

2. 参数描述

参数名	类型	是否必填	说明
result	Action<String>	是	回调

3. 调用示例：

```
▼ C# |  
1 private void NavigateToMiniProgram()  
2 {  
3     Debug.Log("点击退出小游戏");  
4     AlipaySDK.API.ExitProgram((result) =>  
5     {  
6         Debug.Log($"退出小游戏结果: {result}");  
7     });  
8 }
```

十一、文件

本地缓存文件

本地缓存文件只能通过调用特定接口产生，不能直接写入内容。

本地缓存文件通过文件虚拟协议 <https://resource> 操作。

- 本地缓存文件产生后，重启之后仍可用。
- 本地缓存文件只能通过 SaveFile（不指定入参 filePath）接口将本地临时文件保存获得。
- 除非用户主动删除小程序，否则本地缓存文件不会被删除。
- 本地缓存文件和本地用户文件共计最多可存储 50 MB 文件，若超出该大小，可通过 RemoveSavedFile 删除本地缓存文件。
- 单个本地缓存文件最大可存储 10 MB。

注意：本地缓存文件是最初的设计，从 [基础库 1.13.0](#) 版本开始，提供了可完全覆盖 **本地缓存文件** 使用场景的 **本地用户文件** 以及对应的管理接口 GetFileSystemManager()，建议使用本地用户文件。

本地用户文件

本地用户文件通过文件虚拟协议 `https://usr` 操作。

- 本地用户文件与本地缓存文件共计最多可存储 200 MB 文件，若超出该大小，需要删除本地缓存文件获取更多存储空间。
- 单个本地用户文件最大可存储 10 MB。

清理策略

- 本地临时文件仅支持在小程序当前生命周期内进行保存，一旦小程序被关闭就可能被清理，即下次冷启动不保证可用。
- 本地缓存文件和本地用户文件只有在代码包被清理的时才会被清理。

注意：

- 以下接口均有同步实现接口，如：AccessSync。

(一) AlipayFSManager GetFileSystemManager();

1. 功能描述

获取全局唯一的文件管理器。

2. 返回

AlipayFSManager 文件管理对象

(二) Access(异步判断文件/目录是否存在)

1. 功能:

异步判断文件/目录是否存在。

2. 参数:

名称	类型	必填	描述
path	String	是	文件夹路径或者文件路径

3. 调用示例:

```
C# |
1 private AlipayFSManager alipayFSManager;
2 private string filePath = "loading.text";
3
4 void Access()
5 {
6     alipayFSManager.Access(filePath, (result) =>
7     {
8         Debug.Log("Access result: " + result);
9     });
10 }
```

4. 返回

存在文件返回:

```
C# |
1 {
2     "success": true
3 }
```

不存在文件返回:

```

1 {
2     "error": 10022,
3     "errorMessage": "指定的 filePath 文件不存在 https://usr/loading.text",
4     "message": "指定的 filePath 文件不存在 https://usr/loading.text"
5 }

```

5. 错误码

值	说明
10022	文件/目录不存在
10024	传入的路径没有读的权限

(三) ReadFile(读取本地文件内容)

1. 功能:

读取本地文件内容。

2. 参数:

名称	类型	必填	描述
filePath	String	是	文件路径
encoding	String	否	指定读取文件的字符编码，如果不传 encoding，则以 ArrayBuffer 格式读取文件的二进制内容

3. encoding合法值

值
ascii
base64
hex

binary

ucs2/ucs-2/utf16le/utf-16le

utf-8/utf8

latin1

4. 调用示例:

```
▼ C# |
1 private AlipayFSManager alipayFSManager;
2 private string filePath = "loading.text";
3 void ReadFile()
4 {
5     alipayFSManager.ReadFile(filePath, (result) =>
6     {
7         UnityEngine.Debug.Log("ReadFile result: " + result);
8     });
9 }
```

5. 返回

正确读取:

```
▼ C# |
1 {
2     "data": "this is a test txt",
3     "dataType": "string",
4     "success": true
5 }
```

文件不存在:

```
▼ C# |
1 {
2     "error": 10022,
3     "errorMessage": "指定的 filePath 文件不存在 https://usr/loading.text",
4     "message": "指定的 filePath 文件不存在 https://usr/loading.text"
5 }
```


(四) WriteFile(写文件)

1. 功能

写文件

2. 参数

名称	类型	必填	默认值	描述
filePath	String	是		要追加内容的文件路径
data	String/ArrayBuffer	是		要追加的文本或二进制数据
encoding	String	否	utf8	

3. encoding合法值

值
ascii
base64
hex
binary
ucs2/ucs-2/utf16le/utf-16le
utf-8/utf8
latin1

4. 调用示例

```

1 private AlipayFSManager alipayFSManager;
2 private string filePath = "loading.text";
3 void WriteFile()
4 {
5     string info = "this is a test txt";
6     alipayFSManager.WriteFile(filePath, info, (result) =>
7     {
8         Debug.Log("WriteFile result: " + result);
9     });
10 }

```

5. 返回

```

1 {
2     "success": true
3 }

```

(五) Unlink(删除文件)

1. 功能

删除文件

2. 参数

名称	类型	必填	描述
filePath	String	是	文件路径

3. 调用示例

```
1 private AlipayFSManager alipayFSManager;  
2 private string filePath = "loading.text";  
3 void Unlink()  
4 {  
5     alipayFSManager.Unlink(filePath, (result) =>  
6     {  
7         Debug.Log("Unlink result: " + result);  
8     });  
9 }
```

4. 返回

删除成功:

```
1 {  
2     "ariver_ext": "com.alibaba.ariver.commonability.file.jsapi.FSManageExt  
    nsion",  
3     "success": true  
4 }
```

要删除的文件不存在:

```
1 {  
2     "error": 10022,  
3     "errorMessage": "文件不存在 https://usr/loading.text",  
4     "message": "文件不存在 https://usr/loading.text"  
5 }
```

(六) AppendFile(在文件结尾追加内容)

1. 功能

2. 参数

3. 调用示例

(七) CopyFile(复制文件)

1. 功能

2. 参数

3. 调用示例

(八) GetFileInfo(获取本地临时文件或本地缓存文件信息)

1. 功能

2. 参数

3. 调用示例

(九) GetSavedFileList(获取本地缓存文件列表)

1. 功能

2. 参数

3. 调用示例

(十) Mkdir(创建目录)

1. 功能

2. 参数

3. 调用示例

(十一) Readdir(获取目录内文件列表)

1. 功能

2. 参数

3. 调用示例

(十二) RemoveSavedFile(删除本地缓存文件)

1. 功能

2. 参数

3. 调用示例

(十三) Rename(重命名文件)

1. 功能

2. 参数

3. 调用示例

(十四) Rmdir(删除目录)

1. 功能

2. 参数

3. 调用示例

(十五) SaveFile(保存临时文件到本地)

1. 功能

2. 参数

3. 调用示例

(十六) Unzip(解压文件)

1. 功能

2. 参数

3. 调用示例

十二、音频

(一) AlipayAudioContext CreateInnerAudioContext();

1. 属性

属性	类型	只读	说明
Src	string	否	音频的数据链接，用于直接播放。
StartTime	double	否	开始播放的位置（单位：s），默认 0
Autoplay	bool	否	是否自动开始播放，默认 false
Loop	bool	否	是否循环播放，默认 false
ObeyMuteSwitch	bool	否	是否遵循系统静音开关，当此参数为 false 时，即使用户打开了静音开关，也能继续发出声音，默认值 true (仅 iOS 支持)
Volume	double	否	音量。范围 0~1。

IsRecordAudioPlayState	bool	否	标记是否记录当前url播放进度到本地存储,记录的可以通过接口查询,以便恢复播放现场; (Since 10.1.62)
duration	double	是(通过GetDuration方法获取)	当前音频的长度(单位: s), 只有在当前有合法的 src 时返回
currentTime	double	是(通过GetDuration方法获取)	当前音频的播放位置(单位: s), 只有在当前有合法的 src 时返回, 时间不取整
paused	bool	是(通过GetDuration方法获取)	当前是是否暂停或停止状态, true 表示暂停或停止, false 表示正在播放
buffered	double	是(通过GetDuration方法获取)	音频缓冲的时间点, 仅保证当前播放时间点到此时间点内容已缓冲。

属性赋值:

```

C#
1 private AlipayAudioContext alipayAudioContext;
2 void AudioTest()
3 {
4     alipayAudioContext = AlipaySDK.API.CreateInnerAudioContext();
5     alipayAudioContext.Src = "https://mdn.alipayobjects.com/paladin_unity/af
6     fts/file/TTesQq1mcBoAAAAAAAAAAAAADkqsAQBr";
7     alipayAudioContext.Autoplay = false;
8     alipayAudioContext.Loop = true;
9     alipayAudioContext.Volume = 0.5f;
9 }

```

获取属性值:

```

1  private AlipayAudioContext alipayAudioContext;
2  private Toggle LoopToggle;
3  private Toggle AutoplayToggle;
4  private Slider VolumeSlider;
5
6  void AudioTest()
7  {
8      alipayAudioContext = AlipaySDK.API.CreateInnerAudioContext();
9      alipayAudioContext.GetAutoplay((result) =>
10     {
11         Debug.Log("alipayAudioContext GetAutoplay:" + result);
12
13         AutoplayToggle.isOn = result == "true" ? true : false;
14     });
15     alipayAudioContext.GetLoop((result) =>
16     {
17         Debug.Log("alipayAudioContext GetLoop:" + result);
18
19         LoopToggle.isOn = result == "true" ? true : false;
20     });
21     alipayAudioContext.GetVolume((result) =>
22     {
23         Debug.Log("alipayAudioContext GetVolume:" + result);
24
25         VolumeSlider.value = (float)Convert.ToDouble(result);
26     });
27 }

```

2. 方法

方法	参数	说明
Play	无	播放
Pause	无	暂停
Stop	无	停止
Seek	position	跳转到指定位置，单位 s。精确到小数点后 3 位，即支持 ms 级别精确度
Destroy	无	销毁当前实例

OnCanPlay	Action	音频进入可以播放状态，但不保证后面可以流畅播放
OnPlay	Action	音频播放事件
OnPause	Action	音频暂停事件
OnStop	Action	音频停止事件
OnEnded	Action	音频自然播放结束事件
OnTimeUpdate	Action<string>	音频播放进度更新事件
OnError	Action<string>	音频播放错误事件
OnWaiting	Action	音频加载中事件，当音频因为数据不足，需要停下来加载时会触发
OnSeeking	Action	音频进行 seek 操作事件
OnSeeked	Action	音频完成 seek 操作事件

十三、剪切板

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
SetClipboard	void SetClipboard(string text, Action<string> result);	1. text 2. result	设置剪贴板数据
GetClipboard	void GetClipboard(Action<string> result);	1. result	获取剪贴板数据

2. 参数描述

- SetClipboard

参数名	类型	是否必填	说明
text	string	是	设置剪切板的内容

result	Action<String>	是	回调
--------	----------------	---	----

- GetClipboard

参数名	类型	是否必填	说明
result	Action<String>	是	回调

3. 调用示例:

```

C#
1  public Button SetClipboard;
2  public Button GetClipboard;
3
4  void Start()
5  {
6      SetClipboard.onClick.AddListener(() =>
7      {
8          Debug.Log("设置剪贴板数据: 8888");
9          AlipaySDK.API.SetClipboard("8888", (result) =>
10         {
11             Debug.Log($" 设置剪贴板 结果: {result}");
12         });
13     });
14
15     GetClipboard.onClick.AddListener(() =>
16     {
17         Debug.Log("获取剪贴板数据");
18         AlipaySDK.API.GetClipboard((result) =>
19         {
20             Debug.Log($" 获取剪贴板数据 结果: {result}");
21         });
22     });
23 }

```

十四、震动

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
--------	--------	--------	--------

VibrateShort	void VibrateShort(Action<string> result);	1. result	较短时间的振动 (40ms)
VibrateLong	void VibrateLong(Action<string> result);	1. result	较长时间的振动 (400ms)

2. 参数描述

参数名	类型	是否必填	说明
result	Action<String>	是	回调

3. 调用示例:

```

C#
1  public Button VibrateShort;
2  public Button VibrateLong;
3
4  void Start()
5  {
6      VibrateShort.onClick.AddListener(() =>
7      {
8          Debug.Log("较短时间的振动 (40ms)");
9          AlipaySDK.API.VibrateShort((result) =>
10         {
11             Debug.Log($" 较短时间的振动 (40ms) 结果: {result}");
12         });
13     });
14
15     VibrateLong.onClick.AddListener(() =>
16     {
17         Debug.Log("较长时间的振动 (400ms) ");
18         AlipaySDK.API.VibrateLong((result) =>
19         {
20             Debug.Log($" 较长时间的振动 (400ms) 结果: {result}");
21         });
22     });
23 }

```

十五、屏幕亮度

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
SetKeepScreenOn	void SetKeepScreenOn(bool keepScreenOn, Action<string> result);	1. keepScreenOn 2. result	设置是否保持屏幕长亮状态。仅在当前小游戏生效，离开小游戏后失效。
SetScreenBrightness	void SetScreenBrightness(float brightness, Action<string> result);	1. brightness 2. result	设置屏幕亮度
GetScreenBrightness	void GetScreenBrightness(Action<string> result);	1. result	获取屏幕亮度

2. 参数描述

- SetKeepScreenOn

参数名	类型	是否必填	说明
result	Action<String>	是	回调
keepScreenOn	bool	是	设置是否长亮

- SetScreenBrightness

参数名	类型	是否必填	说明
result	Action<String>	是	回调
brightness	float	是	亮度值：0-1

- GetScreenBrightness

参数名	类型	是否必填	说明
result	Action<String>	是	回调

3. 调用示例:

```
1 public Button SetKeepScreenOn;
2 public Button SetScreenBrightness;
3 public Button GetScreenBrightness;
4
5 void Start()
6 {
7     SetKeepScreenOn.onClick.AddListener(() =>
8     {
9         Debug.Log("设置是否保持屏幕长亮状态");
10        bool keepOn = true;
11        AlipaySDK.API.SetKeepScreenOn(keepOn, (result) =>
12        {
13            Debug.Log($" 设置是否保持屏幕长亮状态 结果: {result}");
14        });
15    });
16
17    SetScreenBrightness.onClick.AddListener(() =>
18    {
19        Debug.Log("设置屏幕亮度");
20        float brightness = UnityEngine.Random.Range(0.1f, 1.0f);
21        AlipaySDK.API.SetScreenBrightness(brightness, (result) =>
22        {
23            Debug.Log($" 设置屏幕亮度 结果: {result}");
24        });
25    });
26
27    GetScreenBrightness.onClick.AddListener(() =>
28    {
29        Debug.Log("获取屏幕亮度 ");
30        AlipaySDK.API.GetScreenBrightness((result) =>
31        {
32            Debug.Log($" 获取屏幕亮度 结果: {result}");
33        });
34    });
35 }
```

十六、支付宝Log

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
LocalLog	void LocalLog(LogType logType, string logMessage);	1. logType 2. logMessage	打印日志输出到支付宝，方便日志排查

2. 参数描述

参数名	类型	是否必填	说明
logType	LogType	是	日志类型，Unity原生LogType
logMessage	string	是	日志信息

3. 调用示例：

```

C#
1  public Button logTestBtn;
2
3  void Start()
4  {
5      logTestBtn.onClick.AddListener(() =>
6      {
7          AlipaySDK.API.LocalLog(LogType.Log, "这是一个测试支付宝本地测试Log");
8          AlipaySDK.API.LocalLog(LogType.Warning, "Warning! 这是一个测试支付宝本地测试Log");
9          AlipaySDK.API.LocalLog(LogType.Error, "Error! 这是一个测试支付宝本地测试Log");
10     });
11 }

```

十七、网络

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
--------	--------	--------	--------

Request	void Request(string url, Action<string> result, JsonData headers = null, string method = "GET", object data = null, int timeout = 30000, string dataType = "json");	<ol style="list-style-type: none"> 1. url 2. result 3. headers 4. method 5. data 6. timeout 7. dataType 	小游戏网络请求
Request	void Request(string url, Action<string> result, JsonData param = null);		
DownloadFile	void DownloadFile(string url, JsonData header, Action<string> result);	<ol style="list-style-type: none"> 1. url 2. header 3. result 	下载文件资源到本地
UploadFile	void UploadFile(string url, string filePath, string fileName, string fileType, Action<string> result, JsonData header = null, JsonData formData = null);	<ol style="list-style-type: none"> 1. url 2. filePath 3. fileName 4. fileType 5. header 6. formData 7. result 	上传本地资源到开发者服务器

2. 参数描述

- Request

参数名	类型	是否必填	说明
url	string	是	目标服务器url
result	Action<String>	是	回调

headers	JsonData	否	设置请求的 HTTP 头, 默认 {'content-type': 'application/json'}
method	string	否	默认GET, 目前支持GET/POST
data	object	否	请求参数
timeout	int	否	超时时间, 单位ms, 默认30000
dataType	string	否	期望返回的数据格式, 默认json, 支持 json, text, base64, arraybuffer

- DownloadFile

参数名	类型	是否必填	说明
url	string	是	目标服务器url
result	Action<String>	是	回调
header	JsonData	是	HTTP 请求 Header

- UploadFile

参数名	类型	是否必填	说明
url	string	是	开发者服务器地址
result	Action<String>	是	回调
filePath	string	是	要上传文件资源的本地定位符
fileName	string	是	文件名, 即对应的 key, 开发者在服务器端通过这个 key 可以获取到文件二进制内容
fileType	string	是	文件类型, image / video / audio
formData	JsonData	否	HTTP 请求中其他额外的 form 数据
header	JsonData	否	HTTP 请求 Header

十八、支付宝交互

(一) ShowToast

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
ShowToast	<code>void ShowToast(string content, string type, int duration, Action<string> result);</code>	<ol style="list-style-type: none">1. content2. type3. duration4. result	显示一个弱提示，可选择多少秒之后消失

2. 参数描述

参数名	类型	是否必填	说明
content	string	是	文字内容
type	string	是	toast 类型，展示相应图标，默认 none，支持 success / fail / exception / none'。其中 exception 类型必须传文字信息
duration	int	是	显示时长，单位为 ms，默认 2000
result	Action<string>	是	回调

(二) HideToast

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
HideToast	<code>void HideToast(Action<string> result);</code>	<ol style="list-style-type: none">1. result	隐藏弱提示

2. 参数描述

参数名	类型	是否必填	说明
-----	----	------	----

result	Action<string>	是	回调
--------	----------------	---	----

(三) Alert

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
Alert	void Alert(string title, string content, string buttonText, Action<string> result);	1. result 2. content 3. buttonText 4. title	alert警告框

2. 参数描述

参数名	类型1·	是否必填	说明
result	Action<string>	是	回调
content	string	是	alert框的内容
buttonText	string	是	按钮文字，默认确定
title	string	是	alert框的标题

(四) Confirm

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
--------	--------	--------	--------

Confirm	<pre>void Confirm(string title, string content, string okButtonText, string cancelButtonText, Action<string> result);</pre>	<ol style="list-style-type: none"> 1. title 2. content 3. okButtonText 4. cancelButtonText 5. result 	confirm 确认框
---------	---	---	-------------

2. 参数描述

参数名	类型	是否必填	说明
result	Action<string>	是	回调
title	string	是	confirm框的标题
content	string	是	confirm框的内容
okButtonText	string	是	确认按钮文字，默认‘确定’
cancelButtonText	string	是	确认按钮文字，默认‘取消’

(五) Prompt

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
--------	--------	--------	--------

Prompt	<pre>void Prompt(string title, string message, string placeholder, string align, string okButtonText, string cancelButtonText, Action<string> result);</pre>	<ol style="list-style-type: none"> 1. title 2. message 3. placeholder 4. align 5. okButtonText 6. cancelButtonText 7. result 	弹出一个对话框，让用户在对话框内输入文本
--------	--	---	----------------------

2. 参数描述

- Prompt

参数名	类型	是否必填	说明
result	Action<string>	是	回调
title	string	是	prompt框标题
message	string	是	prompt框文本，默认‘请输入内容’
placeholder	string	是	输入框内的提示文案
align	string	是	message对齐方式，可用枚举 left/center/right, iOS ‘center’, android ‘left’
okButtonText	string	是	确认按钮文字，默认‘确定’
cancelButtonText	string	是	确认按钮文字，默认‘取消’

十九、触摸

1. API 说明

支付宝点击事件

Api 名称	Api 声明	Api 参数	API 说明
OnTouchStart	<code>void OnTouchStart(Action<OnTouchListenerResult> onTouchStartResult);</code>	Action<OnTouchListenerResult> onTouchStartResult	监听开始触摸事件
OffTouchStart	<code>void OffTouchStart(Action<OnTouchListenerResult> onTouchStartResult = null);</code>	非必填	取消监听开始触摸事件
OnTouchMove	<code>void OnTouchMove(Action<OnTouchListenerResult> onTouchMoveResult);</code>	Action<OnTouchListenerResult> onTouchStartResult	监听触点移动事件
OffTouchMove	<code>void OffTouchMove(Action<OnTouchListenerResult> onTouchMoveResult = null);</code>	非必填	取消监听触点移动事件
OnTouchCancel	<code>void OnTouchCancel(Action<OnTouchListenerResult> onTouchCancelResult);</code>	Action<OnTouchListenerResult> onTouchStartResult	监听触点失效事件
OffTouchCancel	<code>void OffTouchCancel(Action<OnTouchListenerResult> onTouchCancelResult = null);</code>	非必填	取消监听触点失效事件
OnTouchEnd	<code>void OnTouchEnd(Action<OnTouchListenerResult> onTouchEndResult);</code>	Action<OnTouchListenerResult> onTouchStartResult	监听触摸结束事件
OffTouchEnd	<code>void OffTouchEnd(Action<OnTouchListenerResult> onTouchEndResult = null);</code>	非必填	取消监听触摸结束事件

2. 返回对象

OnTouchListenerResult

类型	属性	说明
Touch[]	touches	当前所有触摸点的列表
Touch[]	changedTouches	触发此次事件的触摸点列表（暂无返回，请忽略）
long	timeStamp	事件触发时的时间戳（暂无返回，请忽略）

Touch

类型	属性	说明
float	clientX	触点相对于可见视区左边沿的 X 坐标。
float	clientY	触点相对于可见视区上边沿的 Y 坐标。
int	identifier	Touch 对象的唯一标识符，只读属性。一次触摸动作（我们值的是手指的触摸）在平面上移动的整个过程中，该标识符不变。可以根据它来判断跟踪的是否是同一次触摸过程。
float	pageX	触点相对于页面左边沿的 X 坐标。
float	pageY	触点相对于页面上边沿的 Y 坐标。

3. 调用示例

```
1 private void StartTouchListen()  
2 {  
3     AlipaySDK.API.OnTouchStart(OnTouchStart);  
4     AlipaySDK.API.OnTouchMove(OnTouchMove);  
5     AlipaySDK.API.OnTouchEnd(OnTouchEnd);  
6     inListenTouch = true;  
7 }  
8  
9 private void StopTouchListen()  
10 {  
11     AlipaySDK.API.OffTouchStart(OnTouchStart);  
12     AlipaySDK.API.OffTouchMove(OnTouchMove);  
13     AlipaySDK.API.OffTouchEnd(OnTouchEnd);  
14     inListenTouch = false;  
15 }  
16  
17 private void OnTouchStart(OnTouchListenerResult onTouchResult)  
18 {  
19     Debug.Log("OnTouchStart: " + JsonUtility.ToJson(onTouchResult));  
20 }  
21  
22 private void OnTouchMove(OnTouchListenerResult onTouchResult)  
23 {  
24     Debug.Log("OnTouchMove: " + JsonUtility.ToJson(onTouchResult));  
25 }  
26  
27 private void OnTouchEnd(OnTouchListenerResult onTouchResult)  
28 {  
29     Debug.Log("OnTouchEnd: " + JsonUtility.ToJson(onTouchResult));  
30 }
```

二十、存储

- 缓存数据本地加密存储，通过 API 读取时会自动解密返回。
- 覆盖安装支付宝(不是先删除再安装)，不会导致小程序缓存失效；
- 支付宝设置中心清除缓存不会导致小程序缓存失效；
- 小程序缓存默认具有支付宝账号和小程序id两级隔离；
- iOS 客户端支持 iTunes 备份。Ā

(一) SetStorage

1. API 说明

将数据存储在本机缓存中指定的 key 中的异步接口，会覆盖掉原来该 key 对应的数据。单条数据转换成字符串后，字符串长度最大为 200*1024。对于同一个支付宝用户的同一个小程序缓存，存总上限为 10MB。

Api 名称	Api 声明	Api 参数	说明
SetStorage	<code>void SetStorage(string key, string data, Action<string> result);</code>	1. key 2. data 3. result	异步设置本地缓存
SetStorageSync	<code>bool SetStorageSync(string key, string data)</code>		设置本地缓存的同步方法

2. 参数描述

参数名	类型	是否必填	说明
result	Action<string>	是	回调
key	string	是	缓存数据的key
data	string	是	要缓存的数据

3. 调用示例:


```

1 private void SetStorage()
2 {
3     AlipaySDK.API.SetStorage("test", "test", (result) =>
4     {
5         JsonData messageJson = JsonMapper.ToObject(result);
6         if (messageJson["success"].IsBoolean)
7         {
8             var isSuccess = messageJson["success"].IsBoolean;
9             Debug.Log("设置存储信息: " + isSuccess);
10        }
11    });
12 }
13
14 private void SetStorageSync()
15 {
16     var isSuccess = AlipaySDK.API.SetStorageSync("test", "test");
17     var isSuccess2 = AlipaySDK.API.SetStorageSync("test2", "test2");
18     UnityEngine.Debug.Log("设置存储信息: " + isSuccess + "---" + isSuccess2
19 );
20 }

```

(二) RemoveStorage

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
RemoveStorage	void RemoveStorage(string key, Action<string> result);	1. key 2. result	删除缓存数据的接口
RemoveStorageSync	bool RemoveStorageSync(string key);		同步 删除缓存数据

2. 参数描述

参数名	类型	是否必填	说明
result	Action<string>	是	回调

key	string	是	缓存数据的key
-----	--------	---	----------

3. 调用示例:

```

1  private void RemoveStorage()
2  {
3      AlipaySDK.API.RemoveStorage("test1", (result) =>
4      {
5          JsonData messageJson = JsonMapper.ToObject(result);
6          if (messageJson["success"].IsBoolean)
7          {
8              var isSuccess = messageJson["success"].IsBoolean;
9              Debug.Log("删除缓存数据, 结果: " + isSuccess);
10         }
11     });
12 }
13
14 private void RemoveStorageSync()
15 {
16     var isSuccess = AlipaySDK.API.RemoveStorageSync("test");
17     UnityEngine.Debug.Log("删除存储信息: " + isSuccess + "----" + isSuccess2
18 );
19 }

```

(三) GetStorageInfo

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
GetStorageInfo	void GetStorageInfo(Action<StorageInfo> result);	1. result	获取当前storage的相关信息的接口。
GetStorageInfoSync	StorageInfo GetStorageInfoSync();		同步 获取StorageInfo

2. 参数描述

参数名	类型	是否必填	说明
result	Action<StorageInfo>	是	回调

3. 返回StorageInfo类

名称	类型	说明
keys	String Array	当前storage中所有的key
currentSize	Number	当前占用的空间大小, 单位KB
limitSize	Number	限制的空间大小, 单位KB

4. 调用示例:

```

C# |
1  private void GetStorageInfo()
2  {
3      AlipaySDK.API.GetStorageInfo((result) =>
4      {
5          UnityEngine.Debug.Log("异步获取存储信息: " + JsonSerializer.ToJson(result));
6      });
7  }
8
9  private void GetStorageInfoSync()
10 {
11     var info = AlipaySDK.API.GetStorageInfoSync();
12     UnityEngine.Debug.Log("获取存储信息: " + JsonSerializer.ToJson(info));
13 }

```

(四) GetStorage

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
GetStorage	void GetStorage(string key, Action<string> result);	1. result 2. key	获取缓存数据的接口

GetStorageSync	string GetStorageSync(string key);	同步获取缓存数据
----------------	------------------------------------	----------

2. 参数描述

参数名	类型	是否必填	说明
result	Action<string>	是	回调
key	string	是	缓存数据的key

3. 返回success

名称	类型	说明	
success	boolean	boolean	true: 获取成功 false: 获取失败
data(如果success)	Object/String	key对应的内容	

4. 调用示例:

```

C# |
1  private void GetStorage()
2  {
3      AlipaySDK.API.GetStorage("test", (result) =>
4      {
5          JsonData messageJson = JsonMapper.ToObject(result);
6          if (messageJson["success"].IsBoolean)
7          {
8              var data = messageJson["data"];
9              Debug.Log("获取存储信息: " + data);
10         }
11     });
12 }
13
14 private void GetStorageSync()
15 {
16     var value = AlipaySDK.API.GetStorageSync("test");
17     UnityEngine.Debug.Log("获取存储信息: " + value + "----" + value2);
18 }

```

(五) ClearStorage

1. API 说明

Api 名称	Api 声明	Api 参数	API 说明
ClearStorage	void ClearStorage(Action<string> result);	1. result	清除本地数据缓存
ClearStorageSync	bool ClearStorageSync();		同步清除本地数据

2. 参数描述

参数名	类型	是否必填	说明
result	Action<string>	是	回调

二十一、Socket

主体：企业支付宝小游戏、个人支付宝小游戏

相关文档：[网络 API 使用须知](#)

(一) AlipaySocket CreateSocket

1. 功能描述

获取AlipaySocket对象，该对象提供Alipay的Socket相关功能

2. AlipaySocket监听事件

API	Api 声明	监听事件	说明
OnSocketOpen	void OnSocketOpen(Action<string> result)	Action<string>	监听 WebSocket 连接打开事件。

OffSocketOpen	void OffSocketOpen(Action<string> result)	Action<string>	取消监听 WebSocket 连接打开事件。
OnSocketError	void OnSocketError(Action<string> result)	Action<string>	监听 WebSocket 错误。
OffSocketError	void OffSocketError(Action<string> result)	Action<string>	取消监听 WebSocket 错误。
OnSocketMessage	void OnSocketMessage(Action<string> result)	Action<string>	监听 WebSocket 接受到服务器的消息事件。
OffSocketMessage	void OffSocketMessage(Action<string> result)	Action<string>	取消监听 WebSocket 接受到服务器的消息事件。
OnSocketClose	void OnSocketClose(Action<string> result)	Action<string>	监听 WebSocket 关闭。
OffSocketClose	void OffSocketClose(Action<string> result)	Action<string>	取消监听 WebSocket 关闭。

(二) AlipaySocket.ConnectSocket

1. 功能：

创建一个 [WebSocket](#) 的连接。一个支付宝小游戏同时只能保留一个 WebSocket 连接，如果当前已存在 WebSocket 连接，会自动关闭该连接，并重新创建一个新的 WebSocket 连接。

Api 名称	Api 声明	Api 参数	Api 返回值
--------	--------	--------	---------

ConnectSocket	void ConnectSocket(string url, Action<string> result, object data = null, object header = null)	<ol style="list-style-type: none"> 1. url 2. result 3. data 4. header 	为Action<String>的返回值, 是一个json string
---------------	---	---	-------------------------------------

2. 参数:

名称	类型	必填	描述
url	string	是	目标服务器url
result	Action<string>	是	回调: Action<String>的返回值, 是一个json string
data	object	否	请求的参数
header	object	否	设置请求的头部

3. 错误码

error	描述
1	未知错误
2	网络连接已经存在
3	URL参数为空
4	无法识别的URL格式
5	URL必须以ws或者wss开头
6	连接服务器超时
7	服务器返回的https证书无效
8	服务端返回协议头无效
9	WebSocket请求没有指定Sec-WebSocket-Protocol请求头
10	网络连接没有打开, 无法发送消息

11	消息发送失败
12	无法申请更多内存来读取网络数据

4. 调用示例：

```

1
2  void ConnectSocket()
3  {
4      Debug.Log("打开socket");
5      var socket = AlipaySDK.API.CreateSocket();
6      socket.ConnectSocket("*****", (re) =>
7      {
8          Debug.Log("连接socket: " + re);
9      });
10 }

```

(三) AlipaySocket.CloseSocket

1. 功能：

关闭socket连接。

Api 名称	Api 声明	Api 参数	Api 返回值
CloseSocket	CloseSocket(Action<string> result)	1. result	为Action<String>的返回值, 是一个json string

2. 调用示例：


```

1 void CloseSocket()
2 {
3     var socket = AlipaySDK.API.CreateSocket();
4     socket.CloseSocket((result) =>
5     {
6         Debug.Log("关闭socket: " + result);
7     });
8 }

```

(四) AlipaySocket.SendSocketMessage

1. 功能：

通过 WebSocket 连接发送数据，需要先使用 [my.connectSocket](#) 发起建连，并在 [my.onSocketOpen](#) 回调之后再发送数据

Api 名称	Api 声明	Api 参数	Api 返回值
SendSocketMessage	void SendSocketMessage(string data, Action<string> result, bool isBuffer = false)	1. data 2. result 3. isBuffer	为Action<String>的返回值，是一个json string

2. 参数：

名称	类型	必填	描述
data	string	是	需要发送的内容：普通的文本内容 String 或者经 base64 编码后的 String
result	Action<string>	是	回调：Action<String>的返回值，是一个json string
isBuffer	bool	否	如果需要发送二进制数据，需要将入参数据经 base64 编码成 String 后赋值 data，同时将此字段设置为true，否则如果是普通的文本内容 String，不需要设置此字段

二十二、游戏圈

(一) 功能描述

创建游戏圈按钮。游戏圈按钮被点击后会跳转到小游戏的游戏圈。此接口需要挂载小游戏基础包，支付宝10.5.90及以上版本开始支持。

1. API 说明

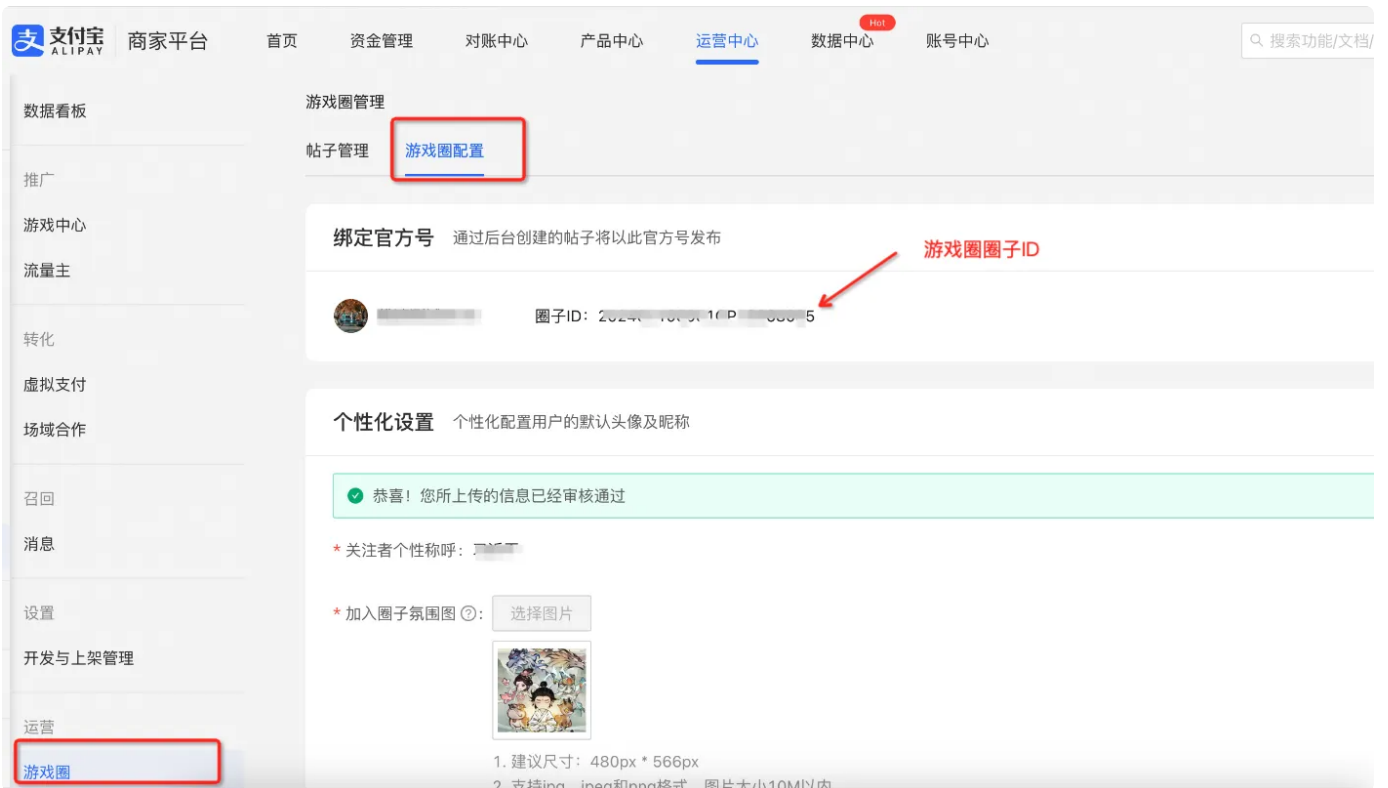
Api 名称	Api 声明	Api 参数	Api 返回值
CreateGameClubButton	<pre>void CreateGameClubButton(CreateGameClubButtonP aram param, Action<GameClubButton > result);</pre>	1.param 2.result	回调中返回 GameClubButton对象
CreateGameClubButton	<pre>GameClubButton CreateGameClubButton(CreateGameClubButtonP aram param);</pre>	1. param	直接返回 GameClubButton对象

2. 参数描述

参数名	类型	是否必填	说明
param	CreateGameClubButtonParam	是	
result	Action<GameClubButton>	是	

(二) 游戏圈圈子ID

链接：<https://b-pre.alipay.com/page/game-management/game-club>



(三) CreateGameClubButtonParam对象

1. 类结构

```
1 [Serializable]
2     public class CreateGameClubButtonParam
3     {
4         public Options options;
5     }
6 [Serializable]
7     public class Options
8     {
9         /// <summary>
10        /// 必填, 按钮类型
11        /// </summary>
12        public GameClubButtonType type;
13        /// <summary>
14        /// 按钮上的文本, 仅当 type 为 text 时有效
15        /// </summary>
16        public string text;
17        /// <summary>
18        /// 按钮的背景图片, 仅当 type 为 image 时有效
19        /// </summary>
20        public string image;
21        /// <summary>
22        /// 必填, 游戏圈按钮的图标, 仅当 object.type 参数为 image 时有效。
23        /// </summary>
24        public GameClubButtonIcon icon;
25        /// <summary>
26        /// 必填, 按钮的样式
27        /// </summary>
28        public GameClubButtonStyle style;
29        /// <summary>
30        /// 跳到对应的活动页面
31        /// </summary>
32        public string openlink;
33    }
34
35    public enum GameClubButtonType
36    {
37        /// <summary>
38        /// 可以设置背景色和文本的按钮
39        /// </summary>
40        text,
41        /// <summary>
42        /// 只能设置背景贴图的按钮, 背景贴图会直接拉伸到按钮的宽高
43        /// </summary>
44        image
45    }
```

```

46 [Serializable]
47 public struct GameClubButtonStyle
48 {
49     public int left;
50
51     public int top;
52
53     public int width;
54
55     public int height;
56
57     public string backgroundColor;
58
59     public string borderColor;
60
61     public int borderWidth;
62
63     public int borderRadius;
64
65     public string color;
66
67     public GameClubButtonTextAlign textAlign;
68
69     public int fontSize;
70 }
71
72 public enum GameClubButtonTextAlign
73 {
74     /// <summary>
75     ///  居左
76     /// </summary>
77     left,
78     /// <summary>
79     ///  居中
80     /// </summary>
81     center,
82     /// <summary>
83     ///  居右
84     /// </summary>
85     right
86 }
87
88
89 public enum GameClubButtonIcon
90 {
91     blue,
92     white,
93     dark,

```

```
    light
}
```

2. Options 创建参数

字段	是否必填	含义	有效值
type	是	按钮类型	“text”（文本样式）、“image”（图片样式）
text	type=“text”时必填	按钮文案	type=“text”有效，string
image	否	自定义按钮背景图，配置以后icon字段将失效	type=“image”有效，仅支持网络图片路径
icon	否	按钮样式图标	type=“image”且“image”字段未配置时有效，“blue”（蓝色的图标）、“white”（白色的图标）、“dark”（有黑色圆角背景的白色图标）、“light”（有白色圆角背景的蓝色图标）
openlink	否	游戏圈传参数，进入到指定圈子下	string，传游戏圈的圈子ID
style	是	按钮的样式	GameClubButtonStyle

3. GameClubButtonStyle按钮样式

字段	是否必填	含义	有效值
left	否	按钮离屏幕左边距	number
top	否	按钮离屏幕上边距	number
width	是	按钮宽度	number
height	是	按钮高度	number
backgroundColor	否	按钮背景颜色	格式为 6 位 16 进制数，例：“#ff0000”

borderColor	否	按钮边框颜色	格式为 6 位 16 进制数，例: "#ff0000"
borderWidth	否	按钮边框宽度	number
borderRadius	否	按钮圆角幅度	number
color	否	文案颜色	type="text"有效，格式为 6 位 16 进制数，例: "#ff0000"
textAlign	否	文案对齐方式	type="text"有效，“left”（左对齐）、“center”（居中对齐）、“right”（右对齐）
fontSize	否	文案字体大小	type="text"有效，number

(四) GameClubButton对象

1. 方法

方法	参数	说明
Show	无	显示游戏圈按钮
Hide	无	隐藏游戏圈按钮
Destroy	无	销毁游戏圈按钮
OnTap	Action onTapAction	游戏圈按钮点击的事件
OffTap	Action offTapAction = null	非必填
SetProperty	string key, object value	刷新按钮属性
SetStyle	string key, object value	刷新Style属性

2. 使用示例

- 该测试 demo，使用了自定义实现GetCtrl，用以获取组件。请使用自己项目中获取组件的方式。
- 同时可以存在多个游戏圈按钮。不过推荐一个界面只展示一个游戏圈按钮。

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using AlipaySdk;
4 using UnityEngine;
5 using UnityEngine.UI;
6
7 public class PlaycirclePanel : TestBase
8 {
9     private CanvasGroup canvasGroup;
10    private List<GameClubButton> gameClubButtons;
11
12    private Button ShowGameClubBtn => GetCtrl<Button>("ShowGameClubBtn");
13    private Button HideGameClubBtn => GetCtrl<Button>("HideGameClubBtn");
14    private Button ExitPlaycirclePanelBtn => GetCtrl<Button>("ExitPlaycirclePanelBtn");
15
16    private InputField textInput => GetCtrl<InputField>("textInput");
17    private Button clubTextBtn => GetCtrl<Button>("clubTextBtn");
18
19    private InputField urlInput => GetCtrl<InputField>("urlInput");
20    private Button urlTextBtn => GetCtrl<Button>("urlTextBtn");
21
22    private int sceneWidth;
23    private int sceneHeight;
24
25    public string newImageUrl;
26    private Button UpdateImageBtn => GetCtrl<Button>("UpdateImageBtn");
27
28    private Slider leftSlider => GetCtrl<Slider>("leftSlider");
29    private Slider topSlider => GetCtrl<Slider>("topSlider");
30    private Slider widthSlider => GetCtrl<Slider>("widthSlider");
31    private Slider heightSlider => GetCtrl<Slider>("heightSlider");
32    private Slider borderWidthSlider => GetCtrl<Slider>("borderWidthSlider");
33    private Slider borderRadiusSlider => GetCtrl<Slider>("borderRadiusSlider");
34    private Slider fontSizeSlider => GetCtrl<Slider>("fontSizeSlider");
35
36    private Toggle bgBlueToggle => GetCtrl<Toggle>("bgBlueToggle");
37    private Toggle bgRedToggle => GetCtrl<Toggle>("bgRedToggle");
38    private Toggle bgGreenToggle => GetCtrl<Toggle>("bgGreenToggle");
39
40    private Toggle bBlueToggle => GetCtrl<Toggle>("bBlueToggle");
41    private Toggle bRedToggle => GetCtrl<Toggle>("bRedToggle");
42    private Toggle bGreenToggle => GetCtrl<Toggle>("bGreenToggle");
```



```

43
44
45     private Toggle BlueToggle => GetCtrl<Toggle>("BlueToggle");
46     private Toggle RedToggle => GetCtrl<Toggle>("RedToggle");
47     private Toggle GreenToggle => GetCtrl<Toggle>("GreenToggle");
48
49     private Toggle leftToggle => GetCtrl<Toggle>("leftToggle");
50     private Toggle centerToggle => GetCtrl<Toggle>("centerToggle");
51     private Toggle rightToggle => GetCtrl<Toggle>("rightToggle");
52
53     private Toggle iconBlueToggle => GetCtrl<Toggle>("iconBlueToggle");
54     private Toggle iconWhiteToggle => GetCtrl<Toggle>("iconWhiteToggle");
55     private Toggle iconDarkToggle => GetCtrl<Toggle>("iconDarkToggle");
56     private Toggle iconLightToggle => GetCtrl<Toggle>("iconLightToggle");
57
58     public void Awake()
59     {
60         canvasGroup = this.GetComponent<CanvasGroup>();
61         canvasGroup.alpha = 0;
62         canvasGroup.interactable = false;
63         canvasGroup.blocksRaycasts = false;
64         gameClubButtons = new List<GameClubButton>();
65
66         sceneWidth = Screen.width;
67         sceneHeight = Screen.height;
68     }
69
70     public void Start()
71     {
72         leftSlider.value = 0;
73         topSlider.value = 0;
74         widthSlider.value = 0;
75         heightSlider.value = 0;
76         borderWidthSlider.value = 0;
77         borderRadiusSlider.value = 0;
78         fontSizeSlider.value = 0;
79
80         bgBlueToggle.isOn = true;
81         bBlueToggle.isOn = true;
82         BlueToggle.isOn = true;
83         leftToggle.isOn = true;
84         iconBlueToggle.isOn = true;
85
86         leftSlider.minValue = 0;
87         leftSlider.maxValue = sceneWidth;
88         leftSlider.onValueChanged.AddListener((f) =>
89         {
90             for (int i = 0; i < gameClubButtons.Count; i++)

```

```

91     {
92     }
93     gameClubButtons[i].SetStyle("left", (int)f);
94     }
95 });
96
97 topSlider.minValue = 0;
98 topSlider.maxValue = sceneHeight;
99 topSlider.onValueChanged.AddListener((f) =>
100 {
101     for (int i = 0; i < gameClubButtons.Count; i++)
102     {
103     }
104     gameClubButtons[i].SetStyle("top", (int)f);
105     }
106 });
107
108 widthSlider.minValue = 50;
109 widthSlider.maxValue = sceneWidth;
110 widthSlider.onValueChanged.AddListener((f) =>
111 {
112     for (int i = 0; i < gameClubButtons.Count; i++)
113     {
114     }
115     gameClubButtons[i].SetStyle("width", (int)f);
116     }
117 });
118
119 heightSlider.minValue = 50;
120 heightSlider.maxValue = sceneHeight;
121 heightSlider.onValueChanged.AddListener((f) =>
122 {
123     for (int i = 0; i < gameClubButtons.Count; i++)
124     {
125     }
126     gameClubButtons[i].SetStyle("height", (int)f);
127     }
128 });
129
130 borderWidthSlider.minValue = 1;
131 borderWidthSlider.maxValue = 10;
132 borderWidthSlider.onValueChanged.AddListener((f) =>
133 {
134     for (int i = 0; i < gameClubButtons.Count; i++)
135     {
136     }
137     gameClubButtons[i].SetStyle("borderWidth", (int)f);
138     }
139 });
140
141 borderRadiusSlider.minValue = 1;

```

```

139     borderRadiusSlider.maxValue = 10;
140     borderRadiusSlider.onValueChanged.AddListener((f) =>
141     {
142         for (int i = 0; i < gameClubButtons.Count; i++)
143         {
144             gameClubButtons[i].SetStyle("borderRadius", (int)f);
145         }
146     });
147
148     fontSizeSlider.minValue = 10;
149     fontSizeSlider.maxValue = 50;
150     fontSizeSlider.onValueChanged.AddListener((f) =>
151     {
152         for (int i = 0; i < gameClubButtons.Count; i++)
153         {
154             gameClubButtons[i].SetStyle("fontSize", (int)f);
155         }
156     });
157
158     #region Toggle参数
159
160     bgBlueToggle.onValueChanged.AddListener((b) =>
161     {
162         for (int i = 0; i < gameClubButtons.Count; i++)
163         {
164             gameClubButtons[i].SetStyle("backgroundColor", "#0000ff"
165     );
166         }
167     });
168     bgRedToggle.onValueChanged.AddListener((b) =>
169     {
170         for (int i = 0; i < gameClubButtons.Count; i++)
171         {
172             gameClubButtons[i].SetStyle("backgroundColor", "#ff0000"
173     );
174         }
175     });
176     bgGreenToggle.onValueChanged.AddListener((b) =>
177     {
178         for (int i = 0; i < gameClubButtons.Count; i++)
179         {
180             gameClubButtons[i].SetStyle("backgroundColor", "#00ff00"
181     );
182         }
183     });
184     bBlueToggle.onValueChanged.AddListener((b) =>

```

```

184         for (int i = 0; i < gameClubButtons.Count; i++)
185     {
186         gameClubButtons[i].SetStyle("borderColor", "#0000ff");
187     }
188 });
189 bRedToggle.onValueChanged.AddListener((b) =>
190 {
191     for (int i = 0; i < gameClubButtons.Count; i++)
192     {
193         gameClubButtons[i].SetStyle("borderColor", "#ff0000");
194     }
195 });
196 bGreenToggle.onValueChanged.AddListener((b) =>
197 {
198     for (int i = 0; i < gameClubButtons.Count; i++)
199     {
200         gameClubButtons[i].SetStyle("borderColor", "#00ff00");
201     }
202 });
203
204 BlueToggle.onValueChanged.AddListener((b) =>
205 {
206     for (int i = 0; i < gameClubButtons.Count; i++)
207     {
208         gameClubButtons[i].SetStyle("color", "#0000ff");
209     }
210 });
211 RedToggle.onValueChanged.AddListener((b) =>
212 {
213     for (int i = 0; i < gameClubButtons.Count; i++)
214     {
215         gameClubButtons[i].SetStyle("color", "#ff0000");
216     }
217 });
218 GreenToggle.onValueChanged.AddListener((b) =>
219 {
220     for (int i = 0; i < gameClubButtons.Count; i++)
221     {
222         gameClubButtons[i].SetStyle("color", "#00ff00");
223     }
224 });
225
226
227 leftToggle.onValueChanged.AddListener((b) =>
228 {
229     for (int i = 0; i < gameClubButtons.Count; i++)
230     {
231         gameClubButtons[i].SetStyle("textAlign", "left");

```

```

232     }
233 });
234 centerToggle.onValueChanged.AddListener((b) =>
235 {
236     for (int i = 0; i < gameClubButtons.Count; i++)
237     {
238         gameClubButtons[i].SetStyle("textAlign", "center");
239     }
240 });
241 rightToggle.onValueChanged.AddListener((b) =>
242 {
243     for (int i = 0; i < gameClubButtons.Count; i++)
244     {
245         gameClubButtons[i].SetStyle("textAlign", "right");
246     }
247 });
248
249 iconBlueToggle.onValueChanged.AddListener((b) =>
250 {
251     for (int i = 0; i < gameClubButtons.Count; i++)
252     {
253         gameClubButtons[i].SetProperty("icon", "blue");
254     }
255 });
256 iconWhiteToggle.onValueChanged.AddListener((b) =>
257 {
258     for (int i = 0; i < gameClubButtons.Count; i++)
259     {
260         gameClubButtons[i].SetProperty("icon", "white");
261     }
262 });
263 iconDarkToggle.onValueChanged.AddListener((b) =>
264 {
265     for (int i = 0; i < gameClubButtons.Count; i++)
266     {
267         gameClubButtons[i].SetProperty("icon", "dark");
268     }
269 });
270 iconLightToggle.onValueChanged.AddListener((b) =>
271 {
272     for (int i = 0; i < gameClubButtons.Count; i++)
273     {
274         gameClubButtons[i].SetProperty("icon", "light");
275     }
276 });
277
278 #endregion
279

```

```

280 HideGameClubBtn.onClick.AddListener(() =>
281 {
282     HideGameClubClick();
283 });
284
285 ShowGameClubBtn.onClick.AddListener(() =>
286 {
287     ShowGameClubClick();
288 });
289
290 ExitPlaycirclePanelBtn.onClick.AddListener(() =>
291 {
292     HidePanel();
293 });
294
295 UpdateImageBtn.onClick.AddListener(() =>
296 {
297     if (string.IsNullOrEmpty(newImageUrl))
298     {
299         return;
300     }
301
302     for (int i = 0; i < gameClubButtons.Count; i++)
303     {
304         gameClubButtons[i].SetProperty("image", newImageUrl);
305     }
306 });
307
308 clubTextBtn.onClick.AddListener(() =>
309 {
310     if (!string.IsNullOrEmpty(textInput.text))
311     {
312         for (int i = 0; i < gameClubButtons.Count; i++)
313         {
314             gameClubButtons[i].SetProperty("text", textInput.text
315 );
316         }
317     });
318
319
320 urlTextBtn.onClick.AddListener(() =>
321 {
322     for (int i = 0; i < gameClubButtons.Count; i++)
323     {
324         gameClubButtons[i].SetProperty("openlink", urlInput.text
325 );
326     }

```

```

326     });
327 }
328
329
330 public void ShowPanel()
331 {
332     canvasGroup.alpha = 1;
333     canvasGroup.interactable = true;
334     canvasGroup.blocksRaycasts = true;
335 }
336
337 public void HidePanel()
338 {
339     canvasGroup.alpha = 0;
340     canvasGroup.interactable = false;
341     canvasGroup.blocksRaycasts = false;
342
343     for (int i = 0; i < gameClubButtons.Count; i++)
344     {
345         gameClubButtons[i].Destroy();
346     }
347     gameClubButtons.Clear();
348 }
349
350
351 public void ShowPanelOneButton()
352 {
353     ShowPanel();
354     SetOneImageBtn();
355 }
356
357 public void ShowPanelOneIconButton()
358 {
359     ShowPanel();
360     SetOneIconBtn();
361 }
362
363 public void ShowPanelRandomButton()
364 {
365     ShowPanel();
366     SetTwoGameClubBtn();
367 }
368
369 public void ShowPanelButton()
370 {
371     ShowPanel();
372     SetGameClubBtn();
373 }

```

#region 设置

```

374
375     public void SetOneImageBtn()
376     {
377         CreateGameClubButtonParam param = new CreateGameClubButtonParam
378         {
379             options = new Options
380             {
381                 type = GameClubButtonType.image,
382                 image = "https://mdn.alipayobjects.com/paladin_unity/aft
383 s/img/1e7STKNyP1cAAAAAAAAAAAAADkqsAQBr/original",
384                 icon = GameClubButtonIcon.blue,
385                 style = new GameClubButtonStyle
386                 {
387                     left = 150,
388                     top = 400,
389                     width = 100,
390                     height = 100,
391                     backgroundColor = "#000000",
392                     borderColor = "#ffffff",
393                     borderWidth = 2,
394                     borderRadius = 5,
395                     color = "#ffffff",
396                     textAlign = GameClubButtonTextAlign.center,
397                     fontSize = 18
398                 },
399                 openlink = "Lv-X010gAuqztP4pRyKfZnY2aJKe9aE1",
400             },
401         };
402
403         AlipaySDK.API.CreateGameClubButton(param, (clubBtn) =>
404         {
405             if (clubBtn == null)
406             {
407                 Debug.LogError("游戏圈按钮创建失败!!");
408             }
409             else
410             {
411                 clubBtn.OnTap(() =>
412                 {
413                     Debug.Log("触发游戏圈按钮点击事件: " + clubBtn.buttonId);
414                 });
415                 clubBtn.Show();
416                 gameClubButtons.Add(clubBtn);
417             }
418         });
419     }
420

```



```

421 public void SetOneIconBtn()
422 {
423     CreateGameClubButtonParam param = new CreateGameClubButtonParam
424     {
425         options = new Options
426         {
427             type = GameClubButtonType.image,
428             icon = GameClubButtonIcon.blue,
429             style = new GameClubButtonStyle
430             {
431                 left = 150,
432                 top = 400,
433                 width = 100,
434                 height = 100,
435                 backgroundColor = "#000000",
436                 borderColor = "#ffffff",
437                 borderWidth = 2,
438                 borderRadius = 5,
439                 color = "#ffffff",
440                 textAlign = GameClubButtonTextAlign.center,
441                 fontSize = 18
442             },
443             openlink = "Lv-X010gAuqztP4pRyKfZnY2aJKe9aE1",
444         },
445     };
446
447     AlipaySDK.API.CreateGameClubButton(param, (clubBtn) =>
448     {
449         if (clubBtn == null)
450         {
451             Debug.LogError("游戏圈按钮创建失败!!");
452         }
453         else
454         {
455             clubBtn.OnTap(() =>
456             {
457                 Debug.Log("触发游戏圈按钮点击事件: " + clubBtn.buttonId);
458             });
459             clubBtn.Show();
460             gameClubButtons.Add(clubBtn);
461         }
462     });
463 }
464
465
466
467 public void SetGameClubBtn()
468 {

```

```

469 CreateGameClubButtonParam param = new CreateGameClubButtonParam
470 {
471     options = new Options
472     {
473         type = GameClubButtonType.text,
474         text = "Test",
475         icon = GameClubButtonIcon.blue,
476         style = new GameClubButtonStyle
477         {
478             left = 150,
479             top = 400,
480             width = 100,
481             height = 50,
482             backgroundColor = "#ff0000",
483             borderColor = "#ffffff",
484             borderWidth = 2,
485             borderRadius = 5,
486             color = "#000000",
487             textAlign = GameClubButtonTextAlign.center,
488             fontSize = 18
489         },
490         openlink = "202404150001GP64239517",
491     },
492 };
493
494
495 AlipaySDK.API.CreateGameClubButton(param, (clubBtn) =>
496 {
497     if (clubBtn == null)
498     {
499         Debug.LogError("游戏圈按钮创建失败!!");
500     }
501     else
502     {
503         clubBtn.OnTap(() =>
504         {
505             Debug.Log("触发游戏圈按钮点击事件: " + clubBtn.buttonId);
506         });
507         clubBtn.Show();
508         gameClubButtons.Add(clubBtn);
509     }
510 });
511 }
512
513 public void SetTwoGameClubBtn()
514 {
515     CreateGameClubButtonParam param = new CreateGameClubButtonParam
516     {

```

```

517     options = new Options
518     {
519         type = GameClubButtonType.text,
520         text = "Test",
521         icon = GameClubButtonIcon.blue,
522         style = new GameClubButtonStyle
523         {
524             left = 150,
525             top = 400,
526             width = 100,
527             height = 50,
528             backgroundColor = "#ff0000",
529             borderColor = "#ffffff",
530             borderWidth = 2,
531             borderRadius = 5,
532             color = "#000000",
533             textAlign = GameClubButtonTextAlign.center,
534             fontSize = 18
535         },
536         openlink = "Lv-X010gAuqztP4pRyKfZnY2aJKe9aE1",
537     },
538 };
539
540
541 AlipaySDK.API.CreateGameClubButton(param, (clubBtn) =>
542 {
543     if (clubBtn == null)
544     {
545         Debug.LogError("游戏圈按钮创建失败!!");
546     }
547     else
548     {
549         clubBtn.OnTap(() =>
550         {
551             Debug.Log("触发游戏圈按钮点击事件: " + clubBtn.buttonId);
552         });
553         clubBtn.Show();
554         gameClubButtons.Add(clubBtn);
555     }
556 }
557
558 CreateGameClubButtonParam param2 = new CreateGameClubButtonPa
559 ram
560 {
561     options = new Options
562     {
563         type = GameClubButtonType.text,
564         text = "Test2",

```

```

564         icon = GameClubButtonIcon.dark,
565         style = new GameClubButtonStyle
566     {
567         left = 150,
568         top = 550,
569         width = 100,
570         height = 50,
571         backgroundColor = "#00ff00",
572         borderColor = "#ffffff",
573         borderWidth = 2,
574         borderRadius = 5,
575         color = "#000000",
576         textAlign = GameClubButtonTextAlign.center,
577         fontSize = 18
578     },
579     openlink = "Lv-X010gAuqztP4pRyKfZnY2aJKe9aE1",
580 },
581 },
582 };
583
584 AlipaySDK.API.CreateGameClubButton(param2, (clubBtn) =>
585 {
586     if (clubBtn == null)
587     {
588         Debug.LogError("游戏圈按钮创建失败!!");
589     }
590     else
591     {
592         clubBtn.OnTap(() =>
593         {
594             Debug.Log("触发游戏圈按钮点击事件: " + clubBtn.button
595 Id);
596             });
597         clubBtn.Show();
598         gameClubButtons.Add(clubBtn);
599     }
600 });
601 });
602 }

```

(五) GetGameClubData

获取游戏圈相关数据

1. API 说明

Api 名称	Api 声明	Api 参数	Api 返回值
GetGameClubData	void GetGameClubData(int[] dataTypeList, Action<string> result)	int[] dataTypeList Action<string> result	result

2. 参数描述

参数名	类型	是否必填	说明
dataTypeList	int[]	是	需要获取的数据指标的对象数组
result	Action<string>	是	回调

- dataTypeList说明

结构属性	type
类型	number
必填	是
说明	见type表格说明

- dataTypeList type说明

type取值	说明	GameClubDataByType.value
1	用户最新加入游戏圈的时间	秒级Unix时间戳
3	用户禁言状态	0: 正常 1: 禁言

4	当天(自然日)点赞贴子数	
5	当天(自然日)评论贴子数	
6	当天(自然日)发表贴子数	
7	当天(自然日)发表视频贴子数	
8	当天(自然日)赞官方贴子数	
9	当天(自然日)评论官方贴子数	

3. 返回

- success

属性	类型	说明
dataType	number	数据类型, 见type说明
value	number	数据值

- fail

属性	类型	说明
error	number	错误码
errorMessage	string	响应错误信息

- 错误码

error	描述
1	请求参数非法
2	系统繁忙，请稍后重试
3	查询用户交互数据限流

二十三、游戏行为上报行为上报

该方法可以在Unity游戏中对支付宝上报用户行为节点进程，有助于游戏运营数据盘点

1. API 说明

Api 名称	Api 声明	Api 参数	含义
ReportLoadingCompleted	ReportLoadingCompleted();		报告游戏加载完成
ReportAuthorized	ReportAuthorized();		初次启动游戏时，默认授权或主动授权完成时进行上报
ReportGameCharacterCreated	ReportGameCharacterCreated(bool initial,);	1.initial 是否是第一次创建角色进入(新用户)	报告游戏角色创建完成
ReportGamePlay	ReportGamePlay();		报告成功进入游戏游玩界面
ReportCustomEvent	ReportCustomEvent(string eventId, JsonData extData)	1. eventId (事件名称) 2. extData (参数)	

2. 调用示例:

```

1 public void GameActionReport()
2 {
3     AlipaySDK.API.ReportLoadingCompleted();
4
5     AlipaySDK.API.ReportAuthorized();
6
7     AlipaySDK.API.ReportGameCharacterCreated(true);
8
9     AlipaySDK.API.ReportGamePlay();
10
11    AlipaySDK.API.ReportCustomEvent("revisit_icon_expo");
12 }

```

二十四、获取当前位置

1. API 说明

Api 名称	Api 声明	Api 参数	Api 返回值
GetLocation	void GetLocation(Action<string> result, int type = 0, int cacheTimeout = 30)	1. result 2. type 3. cacheTimeout	为Action<String>的返回值，是一个json string

2. 参数描述

参数名	类型	是否必填	说明
result	Action<String>	是	回调
type	int	否	0: 默认，获取经纬度 1: 获取经纬度和详细到区县级别的逆地理编码数据 2: 获取经纬度和详细到街道级别的逆地理编码数据，不推荐使用 3: 获取经纬度和详细到POI级别的逆地理编码数据，不推荐使用

cacheTimeout	int	否	支付宝客户端经纬度定位缓存过期时间，单位秒。默认 30s。使用缓存会加快定位速度，缓存过期会重新定位
--------------	-----	---	--

3. 调用示例：

```

C#
1 private void GetLocation()
2 {
3     Debug.Log("获取位置信息 ");
4     AlipaySDK.API.GetLocation((result) =>
5     {
6         Debug.Log("当前位置 返回信息: " + result);
7     }, 1);
8 }

```

4. 返回示例

- success 返回值

名称	类型	描述
longitude	String	经度
latitude	String	纬度
accuracy	String	精确度，单位m
horizontalAccuracy	String	水平精确度，单位m
country	String	国家(type>0生效)
countryCode	String	国家编号 (type>0生效)
province	String	省份(type>0生效)
city	String	城市(type>0生效)
cityAdcode	String	城市级别的地区代码(type>0生效)

- 错误码描述

error	描述	解决方案
11	请确认定位相关权限已开启	提示用户打开定位权限
12	网络异常，请稍后再试	提示用户检查当前网络
13	定位失败，请稍后再试	
14	业务定位超时	提示用户再次尝试

```

C#
1 {
2   "accuracy": "30",
3   "altitude": 0,
4   "bearing": 0,
5   "ccode": "0731",
6   "city": "长沙市",
7   "cityAdcode": "430100",
8   "country": "中国",
9   "countryCode": "156",
10  "district": "岳麓区",
11  "districtAdcode": "430104",
12  "latitude": "28.111732",
13  "locationTime": 1724999152885,
14  "locationType": "wifi",
15  "longitude": "112.937668",
16  "province": "湖南省",
17  "provinceAdcode": "430000",
18  "horizontalAccuracy": "30"
19 }

```

二十五、获取当前电量信息

1. API 说明

Api 名称	Api 声明	Api 参数	Api 返回值
GetBatteryInfo	void GetBatteryInfo(Action<string> result);	1. result	为Action<String>的 返回值，是一个json string

GetBatteryInfoSync	string GetBatteryInfoSync();		返回的是一个json string
--------------------	------------------------------	--	-------------------

2. 参数描述

参数名	类型	是否必填	说明
result	Action<String>	是	回调

3. 调用示例:

```

C#
1  private void GetBatteryInfo()
2  {
3      Debug.Log("获取电量信息 ");
4      AlipaySDK.API.GetBatteryInfo((result) =>
5      {
6          Debug.Log("当前电量 返回信息: " + result);
7      });
8  }
9
10 private void GetBatteryInfoSync()
11 {
12     Debug.Log("获取电量信息 ");
13     var result = AlipaySDK.API.GetBatteryInfoSync();
14     Debug.Log("当前电量 返回信息: " + result);
15 }

```

4. 返回示例

参数名称	参数类型	参数描述
level	int	当前设备电量
isCharging	bool	当前设备是否充电中

```

1 {
2     "isCharging": true,
3     "level": 31
4 }

```

二十六、设备网络状态

1. API 说明

Api 名称	Api 声明	Api 参数	Api 返回值
GetNetworkType	void GetNetworkType(Action<string> result);	1. result	为Action<String>的返回值，是一个json string
OnNetworkStatus Change	void OnNetworkStatusChange(Action<s tring> onAction);	1. onActio n	监听网络变化的事件
OffNetworkStatu sChange	void OffNetworkStatusChange(Action< string> offAction = null);	1. offActio n	取消监听网络变化的事件（可不填）

2. 参数描述

- GetNetworkType

参数名	类型	是否必填	说明
result	Action<String>	是	回调

- OnNetworkStatusChange

参数名	类型	是否必填	说明
onAction	Action<String>	是	监听事件，网络变化后触发的回调事件

- OffNetworkStatusChange

参数名	类型	是否必填	说明
offAction	Action<String>	否	

3. 调用示例:

```

C#
1 private void GetNetworkType()
2 {
3     Debug.Log("获取网络类型 ");
4     AlipaySDK.API.GetNetworkType((result) =>
5     {
6         Debug.Log("当前网络类型 返回信息: " + result);
7     });
8 }
9
10 private void OnNetworkStatusChange()
11 {
12     Debug.Log("监听网络状态变化");
13     AlipaySDK.API.OnNetworkStatusChange((result) =>
14     {
15         Debug.Log("当前网络状态变化 返回信息: " + result);
16     });
17 }
18
19 private void OffNetworkStatusChange()
20 {
21     Debug.Log("取消监听网络状态变化");
22     AlipaySDK.API.OffNetworkStatusChange();
23 }
24
25

```

4. 返回示例

名称	类型	描述
networkAvailable	Boolean	网络是否可用
networkType	String	网络类型值 unknown / notreachable / wifi / 3g / 2g / 4g / wwan

```
1 {  
2   "err_msg": "network_type:wifi",  
3   "hasSystemProxy": false,  
4   "networkAvailable": true,  
5   "networkInfo": "WIFI",  
6   "networkType": "WIFI",  
7   "signalStrength": -62  
8 }
```