

# BAISHUN Game Access Documentation

Game access documentation (H5) Version 1.2.8 - 2025.03.24

Version	explain	Date
1.0.3	Added interface for obtaining game information list	2023.05.04
	Added extend field to /v1/api/change_balance interface	2023.05.26
	Added error code 1023, added client_ip to request parameter of 3.2 query app user information interface	2023.06.01
	Added order_id to /v1/api/change_balance interface	2023.06.05
1.0.4	Added NativeBridge access instructions	2023.06.10
	Added billiards and rocket game IDs	2023.06.28
1.0.5	Changed serverSecret to appKey	2023.07.05
1.0.6	Added selection of game official server node	2023.07.12
1.0.7	Improved flow chart	2023.07.27
1.0.8	Changed currency icon size	2023.08.08
	Added 4.3 fuse query interface	2023.08.10
1.0.9	Added msg_type to /v1/api/change_balance interface	2023.08.22
1.1.0	Added android sample code	2023.09.07
	Added fishing game ID	2023.09.13
1.1.1	Added ios sample code	2023.09.14
	Added game id, 1023, 1024, 1025	2023.09.18
1.1.2	Optimized incomplete document display	2023.09.20

1.1.3	1. Deleted 2.6code update interface, added currency type to change_balance interface	2023.11.29
1.1.4	2. Added 3.5 reporting interface	2023.11.30
1.1.5	3. Added 8 casual game interaction flow chart	2023.12.12
	4. Added 3.6.2 Fishing game extended information description	2023.12.18
	5. Add 3.2 balance_list	2023.12.21
1.1.6	Adjust the currency modification interface, modify the spelling error of currency_amount in 3.2 interface	2024.01.03
1.1.7	3.6.1, add ludo msg_type field description	2024.01.11
1.1.8	3.6.1, supplement ludo msg_type= "skin" and "gift", extend detailed description	2024.05.15
1.1.9	4.4 Add ludo game status query interface	2024.05.22
1.2.0	Add user type and conversion conditions in 3.2 interface	2024.05.27
1.2.1	Modify document	2024.07.22
1.2.2	Add Flutter implementation code	2024.09.09
1.2.3	Add ReactNative Android implementation code	2024.09.10
1.2.4	Add ReactNative IOS implementation	2025.01.13
1.2.5	Optimize iOS to pull up the sandbox path file	2025.01.16
1.2.6	Add 4.3 mode query player balance interface	2025.02.07
1.2.7	Added sample code for destroying and releasing webview when the game calls client destroy. (The channel client implements and verifies according to its own variable naming)	2025.02.19
1.2.8	Optimize Android to show and hide Webview	2025.03.24

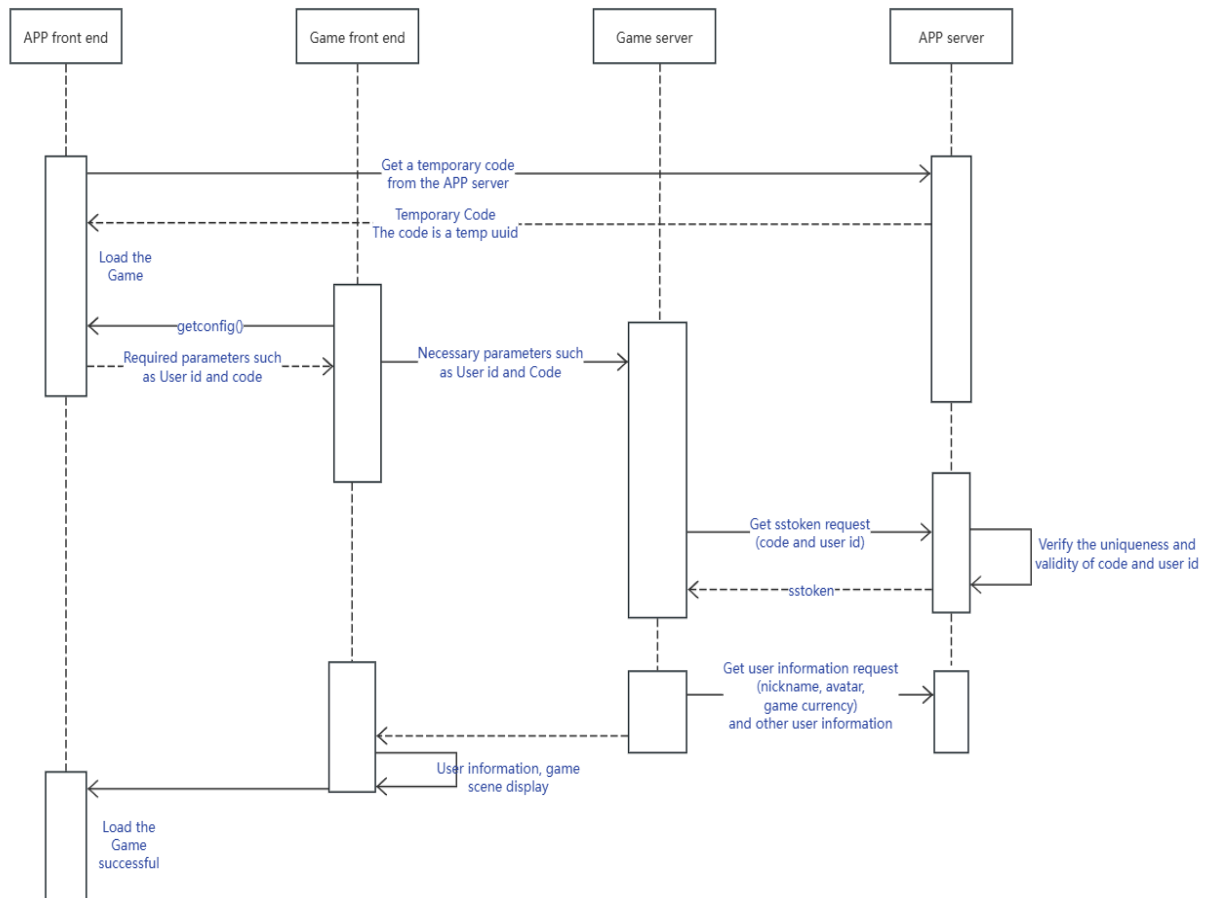
# 1. Overview

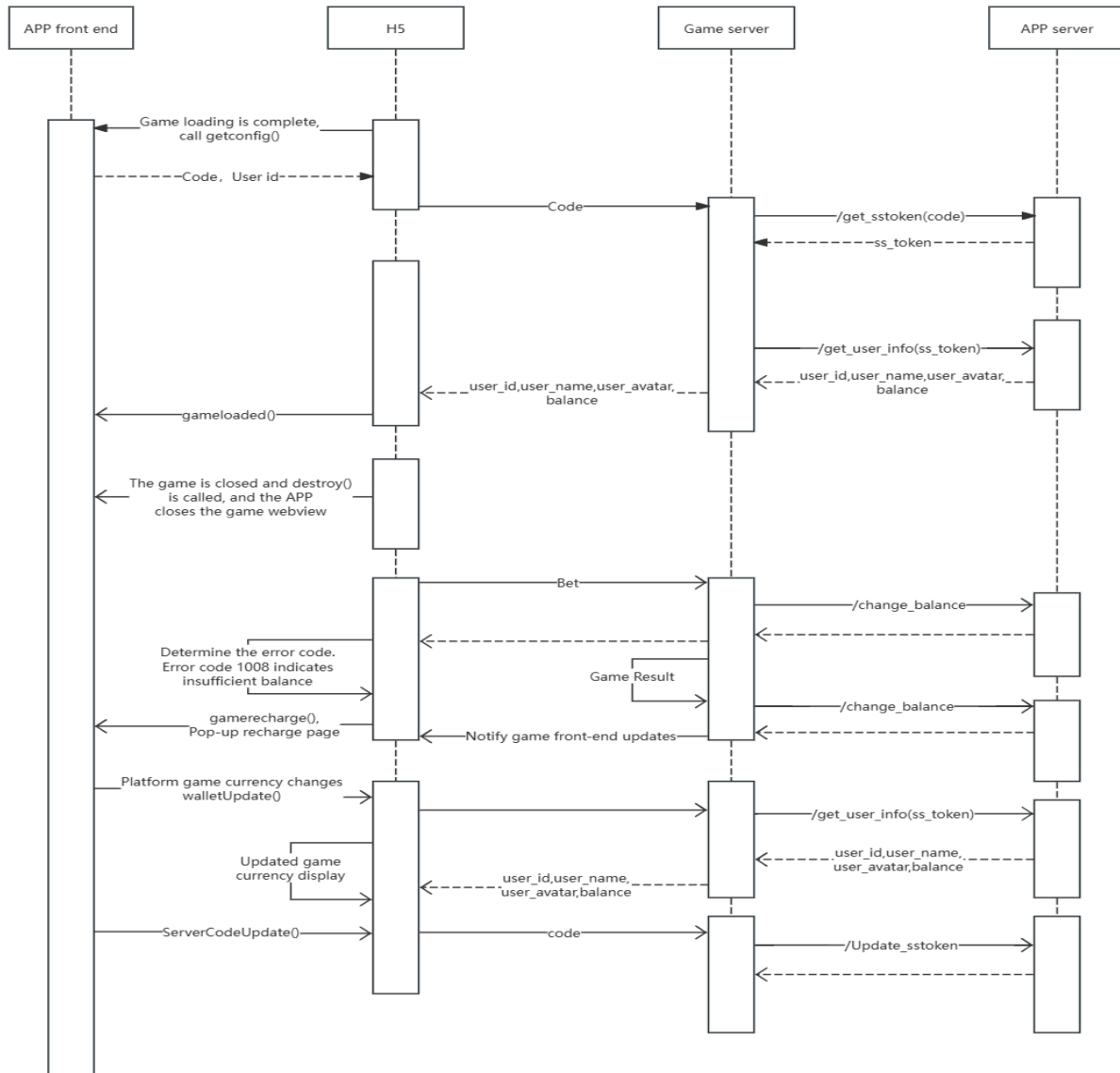
This document contains related information about BAISHUN docking: process description, access format description, and detailed definition of each interface.

For customer docking reference.

BAISHUN gaming platform, hereinafter referred to as "BAISHUN".

Players of customers are hereinafter referred to as "users".





## 1.1. Docking Method

### 1.1.1 Third-party platform configures the operation process for users to play games

- a. Customer calls the interface to obtain game information. Obtain the "game Zip package download address or URL address" and the version number corresponding to the Zip package.
- b. The customer-side client needs to decide whether to update the game package based on the version number, and store it locally in the App.
- c. The user accesses the decompressed local path link and enters the game interface.
- d. For users to bet and settle in the game, the customer needs to provide an API for converting the currency used to participate in the game.

e. The user ends game and returns to the merchant.

### 1.1.2 Docking processr

a. The customer provides test environment server API and client support BAISHUN game H5 protocol method

b. Merchants jointly debug and test the final effect based on the documents of both parties.

c. Provide the production environment server calling interface address.

d. The customer configures the production environment server calling interface address.

e. The customer can finally release it.

### 1.1.3 CDN Optimization

1. The default address of the BAISHUN game offline zip package is the Alibaba Cloud OSS bucket

2. In order to improve user experience, merchants can configure CDN domain name acceleration by themselves.

1. BAISHUN game provides source address

2.The customer configures the CDN Return-to-Origin address and cache policy (more than 3 days)

3. Customer provides accelerated domain names to BAISHUN

4. BAISHUN provides game download addresses for merchants to perform preheating operations.

## 1.2. FAQ

Q:How to set the height of webview?

A: The current running of the game requires the webview to be set to the height of the entire screen.

## 2. BAISHUN Front-end Protocol

- The principle of two-way communication between client and game H5 is achieved by injecting API with the help of JSBridge.

- The APP client needs to add the following protocol method, and the data format returned to the game JS is a JSON string.

- Please refer to the specific code

5. Android Code Accomplishment

6. IOS Code Accomplishment

## 2.1. get Config: To obtain information configuration

The game JS calls the APP client method, and the APP returns information configuration data to JS.

Parameter Name	Type	Note
appChannel	string	Customer channels (provided by BAISHUN and can be viewed in the customer backend)
appId	int64	Customer ID (provided by BAISHUN and can be viewed in the customer backend)
userId	string	UserId
code	string	Authentication token generated by the client (used by the game server to obtain user information)  (code is a one-time parameter and is unique. After the game server uses this parameter to obtain sstoken from the app server, the code is no longer usable. If this code is used to obtain sstoken again, the app server returns an error code 1001)
roomId	string	Room ID (It can be empty when not existed)
game Mode	string	Game Senarios  2:Half Screen(Streamer Show) 3: Full screen (Game Lobby)
language	string	The default language type is English (see Table 3 for details. Multi-language comparison table)

gameConfig	sceneMode:int	Mode Level: Default: 0 0:Mode List 1:Fresh 2:Middle 3:Prime (Only Slots supported Currently: 1 :Maitreya 2 :USA 3 :777)
	currencyIcon:string	Currency icon (external network accessible URL, 60*60 size)
gsp	int	Game official server node 101:Singapore (Alibaba Cloud) 201:Dubai (AWS) 301:Silicon Valley (Alibaba Cloud) 401:Frankfurt (Alibaba Cloud)

## 2.2 destroy destroy the game

Game JS calls APP client methods.

When the game needs to be actively closed, the APP will be notified, and the APP will close the WebView.

## 2.3 gameRecharge Insufficient balance prompt

Game JS calls APP client methods.

When the game coins are insufficient or the gold coin icon is clicked, the APP will be notified, and the APP will open the mall page.

## 2.4 gameLoaded The game is loaded

Game JS calls APP client methods.

When the game is loaded, the APP will be notified when the in-game screen is displayed.

(PS: After launching the game, the APP can display its own loading animation first, and close the loading animation after receiving a notification)

## 2.5 walletUpdate User currency changes

The APP client calls the game JS method.

When the APP completes the recharge or the currency changes and needs to notify the game to refresh the game currency, walletUpdate needs to be called.

Parameter Name	Type	Note
UserId	String	UserId

### 3. The interface that the client server needs to provide (app to access the game, complete the interface in accordance with the following protocol)

After the customer completes the interface, he or she can add the interface configuration in the BAISHUN background, and then debug and run it.

BAISHUN Server domain name:

test server: game-cn-test.jieyou.shop

Authentication and public parameters:

Two-way authentication between app server and BAISHUN server, data communication method is HTTP POST

JSON data format (http Header adds "Content-Type", "application/json")

Signature verifies the identity of the sender of the request by calculating md5 of the random number, AppKey, and time. AppKey must be kept strictly confidential to prevent leakage. The request sent by BAISHUN to the app server will carry the parameters signature\_nonce, timestamp and signature. The app server calculates and compares the signature to verify the BAISHUN identity (the reverse is also true)

Note: Signature is valid within 15 seconds. The signature\_nonce is required not to be repeated within 15 seconds (it is recommended to be globally unique) to prevent replay attacks.

Algorithm: Signature=md5(SignatureNonce+Appkey+Timestamp), The signature is hex encoded (lowercase) and has a length of 32 bits.

signature      Generate reference code:

package main

import (

    "crypto/md5"

    "crypto/rand"

    "encoding/hex"



```

    "fmt"
    "time"
)

func GenerateSignature(signatureNonce string, appKey string, timestamp int64) string {
    data := fmt.Sprintf("%s%s%d", signatureNonce, appKey, timestamp)
    h := md5.New()
    h.Write([]byte(data))
    return hex.EncodeToString(h.Sum(nil))
}

```

```

func main() {
    tempByte := make([]byte, 8)
    rand.Read(tempByte)
    signatureNonce := hex.EncodeToString(tempByte)
    fmt.Println("nonce:", signatureNonce)
    appKey := "8ddcff3a80f4189ca1c9d4d902c3c909"
    timestamp := time.Now().Unix()
    fmt.Println("signature:", GenerateSignature(signatureNonce, appKey, timestamp))
}

```

Public request parameter description:

Name	Type	Must write	Note
signature_nonce	string	Yes	A random string used to prevent replay attacks and is recommended to be globally unique. The mini game will detect an error if there is a request with the same nonce within 15 seconds.
timestamp	int	Yes	Timestamp. If the request timestamp exceeds 15 seconds, an error will be reported.

signature	string	Yes	Request signature, signature algorithm see "Authentication"
provider_name	string	No	Requester identity description

Public return parameter description :

Name	Type	Note
code	int	Error Code
message	string	Error News
unique_id	string	Unique request ID

### 3.1. Obtain SSToken

BAISHUN The server side obtains the long-term token of the calling interface (ss\_token)

post: /v1/api/get\_sstoken

Request parameter description :

Name	Type	Note
app_id	int64	Application ID, customer ID, added and generated by BAISHUN background, passed to h5 front-end from app front-end
user_id	string	Player ID, the player ID passed by the app front-end to the h5 front-end
code	string	The temporary code passed by the app front end to the h5 front end.  (code is a one-time parameter and is unique. After the game server uses this parameter to obtain sstoken from the app server, the code is no longer usable and the app server returns an error code 1001)

Response parameter description :

Name	Type	Note
ss_token	string	Specify user_id to call the long-term token of the app server interface.
expire_date	int64	Expiration time of ss_token, timestamp, milliseconds

Example :

req:

```
{
  "app_id":21397507,
  "user_id":"id1001",
  "code":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ" ,
  "signature":"c62d04ebdb5100e475f45f5ebe8c64ee",
  "signature_nonce":"5f0eb04d7603a9d8",
  "timestamp":1682674598
}
```

rsp:

```
{
  "code": 0,
  "message": "succeed",
  "unique_id": "1682674739807011000",
  "data": {
    "ss_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9",
    "expire_date": 1671096189000
  }
}
```

## 3.2 Query App User Information

The BAISHUN server sends a request to the app server to obtain the user's information

post: /v1/api/get\_user\_info

Request parameter description :

Name	Type	Must write	Note
app_id	int64	Yes	
user_id	string	Yes	
ss_token	string	Yes	ss_token returned by the /v1/api/get_sstoken interface
client_ip	string	Yes	Can be null
game_id	int	Yes	Game ID

Return parametersNote :

Name	Type	Must Write	Note
user_id	string	Yes	
user_name	string	Yes	The user's nickname, which is displayed in the game
user_avatar	string	Yes	The user's avatar loading address is displayed in the game.
balance	float64	Yes	User's game currency
balance_list	balance_info	No	User game currency information array, for multi-currency platforms
user_type	int	No	1: Ordinary users, 2: Whitelist users, 3: Blacklist users
release_cond	int	No	This value is greater than or equal to 0. If user_type is non-0, it will be converted into the number of

			game coins won or lost by ordinary users.
--	--	--	---

- balance\_info Note:

Name	Type	Must Write	Note
name	string	Yes	Currency Name
currency_type	int	Yes	Currency Type definition is consistent with the one passed to the H5 front end
currency_amount	float64	Yes	Balance

Example:

req:

```
{
  "app_id":21397507,
  "user_id":"id1001",
  "ss_token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9",
  "client_ip":"110.86.1.130",
  "game_id":1010,
  "signature":"c62d04ebdb5100e475f45f5ebe8c64ee",
  "signature_nonce":"5f0eb04d7603a9d8",
  "timestamp":1682674598
}
```

rsp:

```
{
  "code": 0,
  "message": "succeed",
  "unique_id": "1603289980002643968",
  "data": {
```

```
    "user_id": "id1001",
    "user_name": "tom",
    "user_avatar": "avatar.com",
    "balance":1000
  }
}
```

### 3.3 ss\_token Update Interface

If ss\_token expires, refresh ss\_token

If ss\_token is not valid for a long time, provide update ss\_token interface

post: /v1/api/update\_sstoken

req:

```
{
  "app_id":21397507,
  "user_id":"id1001",
  "ss_token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9",
  "signature":"c62d04ebdb5100e475f45f5ebe8c64ee",
  "signature_nonce":"5f0eb04d7603a9d8",
  "timestamp":1682674598
}
```

rsp:

```
{
  "code": 0,
  "message": "succeed",
  "unique_id": "1603289541785956352",
  "data": {
    "ss_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9", //updated ss_token
    "expire_date": 1671096189000 //Expiration timestamp, milliseconds
  }
}
```

### 3.4 Currency Modification Interface

BAISHUN Mini games require betting and settlement, and the platform currency of the app needs to be modified.

post: /v1/api/change\_balance

Special note: For any errors, the Error Code (code) should not be 0. Error Code 0 represents success. The Error Code for insufficient balance is: 1008.

Special note: This interface needs to be protected from concurrent processing. For a single user, this interface may be called multiple times in one second. Modifications of player game coins must have a lock mechanism.

diff\_msg = result The game has a function to make up the order for settlement messages. When making up the order, the order\_id remains unchanged and needs to be handled correctly (the player cannot be settled repeatedly)

Request parameter description :

Name	Type	Must Write	Note
app_id	int	Yes	
user_id	string	Yes	
ss_token	string	Yes	
currency_diff	int64	Yes	Changed game currency, negative value decreases, positive value increases
diff_msg	string	Yes	Reason for change, "bet", "result", "refund", "buyin", "buyout"  "buyin": Fish buy in, "buyout":Fish buy out, If the fishing game is in the take-out mode  This interface is only called when buying and taking out
game_id	int	Yes	Game ID, mini-game ID provided by BAISHUN
game_round_id	string	No	One game id

room_id	string	Yes	Room ID, transparently transmitting the roomid returned by get Config
change_time_at	int64	Yes	changed timestamp
order_id	string	Yes	Order ID, unique
extend	string	No	Expanded numeric field
msg_type	string	No	Define the Type of the message in the extension field
currency_type	int	No	Currency Type, app definition

Return parametersNote :

Name	Type	Must Write	Note
currency_balance	float64	Yes	total remaining value

Example :

req:

```
{
  "app_id":21397507,
  "user_id":"id1001",
  "ss_token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9",
  "currency_diff": -100,
  "game_id":1006,
  "room_id":"room_123",
  "game_round_id": "rlmy01pq-cqkdd39jyrmz",
  "order_id":"2R5PHkx43UQPQydCrml71BVqXwH",
  "change_time_at":1638845715,
  "diff_msg":"bet",
  "signature":"c62d04ebdb5100e475f45f5ebe8c64ee",
  "signature_nonce":"5f0eb04d7603a9d8",
```



```

    "timestamp":1682674598
  }
  rsp:
  {
    "code": 0,
    "message": "succeed",
    "unique_id": "1603289541785956352",
    "data": {
      "currency_balance": 900 //total remaining value
    }
  }
}

```

### 3.6 msg\_type and extend Field specific definition

#### 3.6.1 expanded numeric fieldNote

Games	msg_type	extend	Description
All Games	""	""	Real message
All Games	robot_bet	{"nickname":"Mr. Wang"}	Robot betting
All Games	robot_result	{"nickname":"Mr. Wang"}	Robot settlement
Rocket	user_cash_out	{"odds": 12000, "type": "user_cash_out"}	User cash out (launched )
Rocket	crash_top_5	[{"user_id","", "reward":5, "odds":100},...]	top5 (Launched)
Rocket	crash_robot_result	{"nickname":"Mr. Wang", "odds": 12000}	Robot settlement

Rocket	top_jackpot	{"type":"top_jackpot"}	Real first prize (Launched)
Rocket	robot_top_jackpot	""	Robot first price
Fruit	""	{"fruit":0}	Betting fruit (Launched)
LUDO	bet,result,skin,gift	when msg_type="skin": see 3.6.1.1 when msg_type="gift":see 3.6.1.1	bet: Participation deduction result: settlement reward skin:skin purchase gift:Gift Sent

#### 3.6.1.1 extend Field details Note

Parameters	Type	Note
itemId	int	ID
itemType	string	Type
itemCount	int	Quantity
itemPrice	int	Price
currencyType	int	CurrencyType
roomId	string	RoomID
buyer	string	Buyer
receiver	string	Gift recipient

### 3.6.2 Fish game expanded numeric fieldNote

Example :

parameters	Type	Note
change_type	string	"bet","piggy","cdkey" bet: Includes cannon and fish-hit rewards piggy: Piggy bank to receive rewards cdkey: Redeem code to receive rewards
fire_token	string	Bullet token, pigpy和cdkey, this field is empty
change	string	"bet","win","piggy","cdkey" bet: hit win: Rewards for hitting fish piggy: Piggy bank to receive rewards
fish_type	int	Fish Type
drop_id	int	The boss dropped the item ID. If it is not the boss fish, the value is 0.

## 4.BAISHUN The interface provided by the game backend

### 4.1 Obtain Game Info

Provide a preview icon and loading address of the game

post: /v1/api/one\_game\_info

Request parameter description :

Name	Type	Must	Note
app_channel	string	Yes	Channel Name, configured by BAISHUN backend

app_id	int	yes	Customer ID, configured by BAISHUN backend
game_id	int	yes	Game ID, defined by BAISHUN game server

Return parametersNote :

Name	Type	Must	Note
game_id	int	Yes	Game ID
name	string	Yes	Game Name
preview_url	string	Yes	Game icon Load address
game_version	string	Yes	Game version
download_url	string	Yes	Game package loading address
game_mode	[]int	Yes	Modes supported by the game, 2: Show 3: Game
game_orientation	int	Yes	1 portrait, 2 landscape
safe_height	int	Yes	Game security is high (game effective UI height)
venue_level	[]int	Yes	List of supported venues

req:

```
{
  "app_channel": "yuyue",
  "app_id": 21397507,
  "game_id": 1006,
  "signature": "signature",
  "signature_nonce": "signature_nonce"
```

```

        "timestamp":1675147393,
    }
    rsp:
    {
        "code":0
        "msg":"success"
        "data":{
            "game_id":1006,
            "name":"greedy",
            "preview_url":"https://xxx.xxx.com/xxx/xxx.png",
            "game_version":"x.x.x",
            "download_url":"https://xxx.xxx.com/xxx/index.html",
            "game_mode":[1,2,3],
            "game_orientation":1,
            "safe_height":710,
            "venue_level":[1]
        }
    }
}

```

## 4.2 Get Game Information List

post: /v1/api/gamelist

Request parameter description :

Name	Type	Must	Note
game_list_type	int	Yes	Game list type, 2 games, 3 shows
app_channel	string	Yes	Channel Name, configured by BAISHUN backend
app_id	int	Yes	Customer ID, configured by BAISHUN backend

Return parametersNote :

Name	Type	Must	Note
data	[]obj	Yes	Game list, obj as /v1/api/one_game_info Return parameters

Example :

req:

```
{
  "game_list_type":3
  "app_channel":"yuyue",
  "app_id":21397507,
  "signature":"signature",
  "signature_nonce":"signature_nonce"
  "timestamp":1675147393,
}
```

rsp:

```
{
  "code":0
  "msg":"success"
  "data":[
    {
      "game_id":1004,
      "name":"SLOTS",

      "preview_url":"https://bobilive-com-test.oss-cn-beijing.aliyuncs.com/admin_game/a92aac6e253b4d6d4f73d613526b7fe1.png",
      "game_version":"2.7.0",
      "download_url":"https://bobilive-com-test.jieyou.shop/h5web/GameH5/yomi_dev/slots/1.0.0/index.html?ts=1683201800398350467",
      "game_mode":[2,3],
      "game_orientation":1,
      "safe_height":800,
```

```

        "venue_level": [0]},
    {
        "game_id": 1006,
        "name": "greedy",
        "preview_url": "https://xxx.xxx.com/xxx/xxx.png",
        "game_version": "x.x.x",
        "download_url": "https://xxx.xxx.com/xxx/index.html",
        "game_mode": [1, 2, 3],
        "game_orientation": 1,
        "safe_height": 710,
        "venue_level": [1]
    }
]
}

```

#### 4.3 Get user balance (only valid for games with import category)

post: /v2/api/balance\_info

Request parameters (also required for 3.0 common parameters)

Name	Type	Must	Note
user_id	string	是	玩家ID
app_channel	string	是	渠道名称, 由 BAISHUN 后台配置
app_id	int64	是	商户 ID, 由 BAISHUN 后台配置

Return parameters

Name	Type	Must	Note
data	Json obj	是	
——cur_coin	int	是	玩家余额

Example:

```

req:
{
  "user_id":"102512188",
  "app_id":3074591544,
  "app_channel":"skychat",
  "signature":"27c6741ea9af2a78aef790bc76f0e068",
  "signature_nonce":"123456",
  "timestamp":1738898682
}
rsp:
{
  "code": 0,
  "msg": "success",
  "data": {
    "cur_coin": 7
  }
}

```

## 5. Android implementation code

### 5.1 Instantiate WebView

WebView can be created directly in the editor. And set the size, just set some properties in the code.

// MainActivity onCreate method added

```

this.webView = this.findViewById(R.id.mWebview);
//Prevent web pages from being opened in browsers
this.webView.setWebViewClient(new WebViewClient());
//Enable debugging and remember to comment when publishing
this.webView.setWebContentsDebuggingEnabled(true);
//Set the webview background to be transparent, the default is white
this.webView.setBackgroundColor(0);
//Set the view background to be transparent, the default is white. Optional (needed to be added
when adding a webView component to a separate activity)
binding.getRoot().setBackgroundColor(0);

```

```

WebSettings settings = this.webView.getSettings();
//Setting up JavaScript support
settings.setJavaScriptEnabled(true);
//Set the default text encoding
settings.setDefaultTextEncodingName("UTF-8");
//Set up access to local files

```



```

settings.setAllowFileAccess(true);
//Set to allow Javascript loaded through file url to read all resources (including files, http, https)
settings.setAllowUniversalAccessFromFileURLs(true);
//Set the priority to load cache
settings.setCacheMode(WebSettings.LOAD_CACHE_ELSE_NETWORK);
//Enable HTML5 DOM storage
settings.setDomStorageEnabled(true);
//Set to enable database cache
settings.setDatabaseEnabled(true);
settings.setDatabasePath(getApplicationContext().getFilesDir().getAbsolutePath());
//Set up APP cache
settings.setAppCacheEnabled(true);
settings.setAppCachePath(getApplicationContext().getFilesDir().getAbsolutePath());
//Setting support for zoom
settings.setSupportZoom(true);
//Setting Adaptive
settings.setUseWideViewPort(true);
//Set up autoplay media
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
    settings.setMediaPlaybackRequiresUserGesture(false);
}
//Setting 5.0 and above allows loading mixed http and https pages
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
    settings.setMixedContentMode(WebSettings.MIXED_CONTENT_ALWAYS_ALLOW);
}

// Show Webview
this.webView.setVisibility(View.VISIBLE);

//The class called by the game must be defined as NativeBridge
    this.webView.addJavascriptInterface(new NativeBridge(this), "NativeBridge");

```

## 5.2 Encapsulate calling JS functions

In order to facilitate the unified call of JS functions, you can define such a method

```

/**
 * MainActivity
 * Calling js
 */
public void callJs(final String str){
    WebView webview = this.webView;
    webview.post(new Runnable() {
        @Override
        public void run() {
            webView.loadUrl("javascript:" + str);

```

```

    }
  });
}

```

### 5.3 Instantiate the NativeBridge class

```

class NativeBridge {
    private MainActivity context;

```

```

    NativeBridge(MainActivity context) {
        this.context = context;
    }

```

```

    @JavascriptInterface

```

```

    public void getConfig(String params) {
        Log.d("BSGAME", "Game call getConfig");

```

**The data is only a reference value and needs to be assigned according to your own**

**APP**

```

        try {
            JSONObject obj = new JSONObject(params);
            String jsFunName = obj.optString("jsCallback");

            GetConfigData getConfigData = new GetConfigData();
            getConfigData.appChannel = "mesh";
            getConfigData.appId = 888888888;
            getConfigData.userId = "338426830";
            getConfigData.code =
"SqUdd4KrRa8tPvgoboPEQcJy0V3UQ0ywur612cjHtQhMOsoumtmNUelzAgEa";
            getConfigData.roomId = "20230201";
            getConfigData.gameMode = "3";
            getConfigData.language = "2";
            getConfigData.gameConfig.sceneMode = 0;
            getConfigData.gameConfig.currencyIcon =
"https://bsyx-store.s3.ap-southeast-1.amazonaws.com/game-icons/20230906-163004.png";
            getConfigData.gsp = 101;

```

```

            String str = jsFunName + "(" + new Gson().toJson(getConfigData) + ")";
            this.context.callJs(str);

```

```

        } catch (Exception ex) {
            Log.e("getConfig ERROR", ex.getMessage());
        }
    }
}

```

```

    @JavascriptInterface

```

```

public void destroy(String params) {
    Log.d("BSGAME", "Game call destroy");
    //Close the game TODO client
    if (this.context.webView != null) {
        WebView webview = this.context.webView;
        webview.post(new Runnable() {
            @Override
            public void run() {
                webview.stopLoading(); // Stop the current loading operation
                webview.loadUrl("about:blank"); // Clear current page
                webview.removeAllViews(); // Remove a view from a WebView
                webview.setVisibility(View.GONE); // Hide Webview
            }
        });
    }
}

@JavascriptInterface
public void gameRecharge(String params) {
    Log.d("BSGAME", "Game call gameRecharge");
    //Pull up the TODO client of the recharge mall
}

@JavascriptInterface
public void gameLoaded(String params) {
    Log.d("BSGAME", "Game call gameLoaded");
    //Game loaded TODO client
}

}

class GetConfigData {
    public String appChannel = "";
    public long appld = 0;
    public String userId = "";
    public String code = "";
    public String roomId = "";
    public String gameMode = "";
    public String language = "2";
    public GameConfig gameConfig = new GameConfig();
    public int gsp = 101;
}

class GameConfig {

```

```

    public int sceneMode = 0;
    public String currencyIcon = "";
}

```

## 5.4 Android actively calls walletUpdate

```

//The data is only a reference value and needs to be assigned according to your own APP
Map<String,Object> map = new HashMap<>();
map.put("userId","534206265");
String str = "walletUpdate" + "(" + new Gson().toJson(map) + ")";
this.callJs(str);

```

## 5.5 Android game loading and starting

```

String h5Url = "Https URL address or local path address";
this.webView.loadUrl(h5Url);

```

# 6. IOS implementation code

It is recommended to use WKWebView, instantiate Webview and set parameters to complete the game startup.

## 6.1 Instantiate WebView

```

// viewDidLoad method added
WKUserContentController* wcc = [[WKUserContentController alloc] init];
//Binding method name
[wcc addScriptMessageHandler:self name:@"getConfig"];
[wcc addScriptMessageHandler:self name:@"destroy"];
[wcc addScriptMessageHandler:self name:@"gameRecharge"];
[wcc addScriptMessageHandler:self name:@"gameLoaded"];

WKWebViewConfiguration *config = [WKWebViewConfiguration new];
config.allowsInlineMediaPlayback = YES;
[config setValue:@YES forKey:@"allowUniversalAccessFromFileURLs"];
//The playback of audio and video does not require user gesture triggering, that is, it plays
automatically
config.mediaTypesRequiringUserActionForPlayback = NO;
config.preferences = [[WKPreferences alloc] init];
WKPreferences *preferences = [WKPreferences new];
preferences.javaScriptCanOpenWindowsAutomatically = YES;
config.preferences = preferences;
config.preferences.javaScriptEnabled = YES;
config.userContentController = wcc;

```

```

//Get the width and height of the current screen
int ScreenHeight = [[UIScreen mainScreen] bounds].size.height;
int ScreenWidth = [[UIScreen mainScreen] bounds].size.width;
//Get the height of the status bar
CGRect statusBarRect = [[UIApplication sharedApplication] statusBarFrame];
int statusBarHeight = statusBarRect.size.height;

self.webview = [[WKWebView alloc] initWithFrame:CGRectMake(0,0, ScreenWidth,
ScreenHeight) configuration:config];
[self.webview.scrollView setBackgroundColor:[UIColor clearColor]];
[self.webview setBackgroundColor:[UIColor clearColor]];
[self.webview setUIDelegate:self];
//Set the webpage to be transparent
[self.webview setOpaque:NO];
//Set web page to full screen
if (@available(iOS 11.0, *)) {
    self.webview.scrollView.contentInsetAdjustmentBehavior =
UIScrollViewContentInsetAdjustmentNever;
}
[self.view addSubview:self.webview];

```

## 6.2 Encapsulate calling JS functions

In order to facilitate the unified call of JS functions, you can define such a method

```

// Calling JS
- (void)callJs:(NSString*)method withJavaScriptValue:(nullable id)arguments {
    if (arguments) {
        NSData *data = [NSJSONSerialization dataWithJSONObject:arguments
options:(NSJSONWritingPrettyPrinted) error:nil];
        NSString *jsonStr = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];
        NSString *jsMethods = [NSString stringWithFormat:@"%@"@("%@"), method, jsonStr];
        [self.webview evaluateJavaScript:jsMethods completionHandler:^(id _Nullable resp,
NSError * _Nullable error) {
            NSLog(@"error = %@", , response = %@",error, resp);
        }];
    } else {
        NSString *jsMethods = [NSString stringWithFormat:@"%@"@("{})", method];
        [self.webview evaluateJavaScript:jsMethods completionHandler:^(id _Nullable resp,
NSError * _Nullable error) {
            NSLog(@"error = %@", , response = %@",error, resp);
        }];
    }
}

```

## 6.3 String to Dictionary

```

- (NSDictionary *)dictionaryWithJsonString:(NSString *)jsonString
{
    if (jsonString == nil && jsonString.length == 0) {
        return nil;
    }

    NSData *jsonData = [jsonString dataUsingEncoding:NSUTF8StringEncoding];
    NSError *err;
    NSDictionary *dic = [NSJSONSerialization JSONObjectWithData:jsonData
                                                                options:NSJSONReadingMutableContainers
                                                                error:&err];

    if(err)
    {
        NSLog(@"jsonParsing failed: %@", err);
        return nil;
    }
    return dic;
}

```

## 6.4 Binding protocol method

```

- (void)userContentController:(WKUserContentController *)userContentController
    didReceiveScriptMessage:(WKScriptMessage *)message
{
    NSString* method = [NSString stringWithFormat:@"%@", message.name];
    NSDictionary *dicBody = [self dictionaryWithJsonString:message.body];
    SEL selector = NSSelectorFromString(method);
    if([self respondsToSelector:selector]){
        //Using reverse mapping
        [self performSelector:selector withObject:dicBody];
    }else{
        NSLog(@"Unimplemented methods : %@ --> %@", message.name, message.body);
    }
}

// Get information configuration
- (void) getConfig:(NSDictionary*)args
{
    NSLog(@"BSGAME %s", "The game calls getConfig");
    NSString* method = [args objectForKey:@"jsCallback"];

```

**The data is only a reference value and needs to be assigned according to your own**

### APP

```

NSObject* configData = @{
    @"appChannel":@"mesh",
    @"appId":@"888888888",
    @"userId":@"338426830",

```

```

@"code":@"SqUdd4KrRa8tPvgoboPEQcJy0V3UQ0ywur612cjHtQhMOsoumtmNUelzAgEa",
  @"roomId":@"20230901",
  @"gameMode":@"3",
  @"language":@"2",
  @"gameConfig":@{
    @"sceneMode":@0,

    @"currencyIcon":@"https://bsyx-store.s3.ap-southeast-1.amazonaws.com/game-icons/2
0230906-163004.png",
  },
  @"gsp":@101,
};
[self callJs:method withJavaScriptValue:configData];
}

```

// Destruction Game

- (void) destroy:(NSDictionary\*)args

```

{
  NSLog(@"BSGAME %s", "Game call destroy");
  //Close the game TODO client
  [self.webview stopLoading];
  [self.webview.configuration.userContentController removeAllUserScripts];
  self.webview.navigationDelegate = nil;
  self.webview.UIDelegate = nil;
  [self.webview removeFromSuperview];
  self.webview = nil;
}

```

// Insufficient balance

- (void) gameRecharge:(NSDictionary\*)args

```

{
  NSLog(@"BSGAME %s", "Game call gameRecharge");
  //Pull up the TODO client of the recharge mall
}

```

// Game loading completed

- (void) gameLoaded:(NSDictionary\*)args

```

{
  NSLog(@"BSGAME %s", "Game call gameLoaded");
  // Game loaded TODO client
}

```

## 6.5 IOS actively calls walletUpdate

//The data is only a reference value and needs to be assigned according to your own APP

```
NSObject* map = @{
    @"userId":@"534206265"
};
[self callJs:@"walletUpdate" withJavaScriptValue:map];
```

## 6.6 IOS game loading and starting

### URL method

```
NSString *h5Url =@" https URL address";
NSURL *url = [NSURL URLWithString:h5Url];
NSURLRequest *request = [[NSURLRequest alloc] initWithURL:url];

[self.webview loadRequest:request];
```

### Zip package Local sandbox path method

```
NSURL *documentPath = [[[NSFileManager defaultManager]
URLsForDirectory:NSDocumentDirectory inDomains:NSUserDomainMask] firstObject];
NSString *basePath = [NSString stringWithFormat: @"%@/%@", documentPath,
@" /web-mobile"];

NSURL *baseUrl = [NSURL fileURLWithPath: basePath isDirectory: YES];
NSString *filePath = [NSString stringWithFormat:
@"%@/index.html#/?xxx=%@&sss=%@", basePath, @"xxx", @"sss"];
NSURL *fileUrl = [NSURL URLWithString: filePath];
[self.webview loadFileURL: fileUrl allowingReadAccessToURL: baseUrl];
```

# 7. Flutter implementation code

It is recommended to use the official plugin `webview_flutter`, instantiate `Webivew` and set parameters to complete the game startup.

## 7.1 Import plugin

Add dependencies to the main project's `pubspec.yaml` file and import the corresponding `webview_flutter` version based on the project's own Dart version.

dependencies:

```
webview_flutter: ^4.4.2
```

Execute Pub get to complete the import.

## 7.2 Add a reference

Add the reference to the main.dart header.

```
import 'dart:convert';
import 'dart:io';
```



```

import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:webview_flutter/webview_flutter.dart';
// #docregion platform_imports
// Import for Android features.
import 'package:webview_flutter_android/webview_flutter_android.dart';

// Import for iOS features.
import 'package:webview_flutter_wkwebview/webview_flutter_wkwebview.dart';

```

### 7.3 Instantiate WebView

```

class _WebViewExampleState extends State<WebViewExample> {
  late final WebViewController controller;
  @override
  void initState() {
    super.initState();

    // #docregion webview_controller
    controller = WebViewController()
      ..setJavaScriptMode(JavascriptMode.unrestricted)
      ..setBackgroundColor(const Color(0x00000000))

    ..loadRequest(Uri.parse('https://game-center-test.jieyou.shop/game-packages/common-web/rocket/1.3.3/web-mobile/index.html'));
  }

  // #docregion webview_widget
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Flutter Jsbridge')),
      body: WebViewWidget(controller: controller),
    );
  }
}

```

### 7.3 Flutter Android

It is not possible to directly use Flutter's built-in JavaScriptChannel to communicate with the game. The following modifications are required.

### 7.3.1 Adding Network Permissions

path:android/app/src/main/AndroidManifest.xml, Add the following code.

```
<uses-permission android:name="android.permission.INTERNET"/>
```

### 7.3.2 Allow network requests using plaintext traffic

path:android/app/src/main/AndroidManifest.xml, Add the following code.

```
<application
...
    android:usesCleartextTraffic="true">
...
</application>
```

### 7.3.3 Native JAVA modification

path:webview\_flutter\_android-3.12.1/android/src/main/java/io/flutter/plugins/webviewflutter/WebViewHostApiImpl.java

After modify [WebViewHostApiImpl.java file](#)

#### 7.3.3.1 Import references

```
import android.webkit.JavascriptInterface;
import android.os.Handler;

import android.os.Looper;
```

#### 7.3.3.2 Binding NativeBridge

```
@Override
public void loadUrl(
    @NonNull Long instanceId, @NonNull String url, @NonNull Map<String, String> headers) {
    final WebView webView = Objects.requireNonNull(instanceManager.getInstance(instanceId));
    webView.loadUrl(url, headers);
    //baishun
    setNativeBridge(instanceId);
}

@SuppressLint("JavascriptInterface")
public void setNativeBridge(Long instanceId){
    final WebView webView = (WebView) instanceManager.getInstance(instanceId);
    webView.addJavascriptInterface(new NativeBridge(context), "NativeBridge");
}
```

#### 7.3.3.3 Add NativeBridge Class

```
class NativeBridge {
```

```

private Context ctx;
NativeBridge(Context ctx) {
    this.ctx = ctx;
}
// Add JS calling interface
@JavascriptInterface
public void getConfig(String msg) {
    Handler mainHandler = new Handler(Looper.getMainLooper());
    mainHandler.post(new Runnable() {
        @Override
        public void run() {
            WebViewFlutterPlugin.bsEventChannel.sendEvent(msg);
        }
    });
}

```

```

@JavascriptInterface
public void destroy(String msg) {
    Handler mainHandler = new Handler(Looper.getMainLooper());
    mainHandler.post(new Runnable() {
        @Override
        public void run() {
            WebViewFlutterPlugin.bsEventChannel.sendEvent(msg);
        }
    });
}

```

```

@JavascriptInterface
public void gameRecharge(String msg) {
    Handler mainHandler = new Handler(Looper.getMainLooper());
    mainHandler.post(new Runnable() {
        @Override
        public void run() {
            WebViewFlutterPlugin.bsEventChannel.sendEvent(msg);
        }
    });
}

```

```

@JavascriptInterface
public void gameLoaded(String msg) {
    Handler mainHandler = new Handler(Looper.getMainLooper());
    mainHandler.post(new Runnable() {
        @Override
        public void run() {
            WebViewFlutterPlugin.bsEventChannel.sendEvent(msg);
        }
    });
}

```

```
}
```

#### 7.3.3.4 Add BSEventChannel class

path:webview\_flutter\_android-3.12.1/android/src/main/java/io/flutter/plugins/webviewflutter/BSEventChannel.java

```
package io.flutter.plugins.webviewflutter;
```

```
import android.util.Log;
```

```
import io.flutter.embedding.engine.plugins.FlutterPlugin;
```

```
import io.flutter.plugin.common.EventChannel;
```

```
import io.flutter.plugin.common.PluginRegistry;
```

```
public class BSEventChannel implements EventChannel.StreamHandler {
```

```
    public static final String CHANNEL = "baishunChannel";
```

```
    public static EventChannel.EventSink bsSink;
```

```
    // private Activity activity;
```

```
    public static EventChannel channel;
```

```
    public static BSEventChannel registerWith(FlutterPlugin.FlutterPluginBinding binding) {
```

```
        channel = new EventChannel(binding.getBinaryMessenger(), CHANNEL);
```

```
//        channel = new EventChannel(registrar.messenger(), CHANNEL);
```

```
        BSEventChannel instance = new BSEventChannel();
```

```
        channel.setStreamHandler(instance);
```

```
        return instance;
```

```
    }
```

```
    @Override
```

```
    public void onListen(Object arguments, EventChannel.EventSink events) {
```

```
        // The Flutter app has opened the Event Channel
```

```
        bsSink = events;
```

```
    }
```

```
    @Override
```

```
    public void onCancel(Object arguments) {
```

```
        // The Flutter app has closed the Event Channel
```

```
        bsSink = null;
```

```
    }
```

```
    public void sendEvent(Object o) {
```

```
        if (bsSink != null) {
```

```
            bsSink.success(o);
```

```
        }
```

```
    }
```

```
}
```

#### 7.3.3.5 Register BSEventChannel

path:webview\_flutter\_android-3.12.1/android/src/main/java/io/flutter/plugins/webviewflutter/WebViewFlutterPlugin.java

```
public class WebViewFlutterPlugin implements FlutterPlugin, ActivityAware {  
    ...  
    public static BSEventChannel bsEventChannel;  
    @Override  
    public void onAttachedToEngine(@NonNull FlutterPluginBinding binding) {  
        pluginBinding = binding;  
        ...  
        bsEventChannel = BSEventChannel.registerWith(binding);  
    }  
    @Override  
    public void onDetachedFromEngine(@NonNull FlutterPluginBinding binding) {  
        if (instanceManager != null) {  
            instanceManager.stopFinalizationListener();  
            instanceManager = null;  
            BSEventChannel.channel.setStreamHandler(null);  
        }  
    }  
    ...  
}
```

#### 7.3.4 Binding protocol method

```
EventChannel _eventChannelPlugin = EventChannel('baishunChannel');  
if(Platform.isAndroid){  
    _eventChannelPlugin.receiveBroadcastStream().listen((event) {  
        {  
            final obj = json.decode(event);  
            String jsFunName = obj['jsCallback'];  
            if (jsFunName.contains('getConfig')) {  
                print("BSGAME Game call getConfig main.dart");  
                GetConfigData _configData = GetConfigData(  
                    appChannel: "mesh",  
                    appld: 88888888,  
                    userId: "338426830",  
                    gameMode: "3",  
                    language: "0",  
                    gsp: 101,  
                    roomId: '20240101',  
                    code:
```

```

        "SqUdd4KrRa8tPvgoboPEQcJy0V3UQ0ywur612cjHtQhMOsoumtmNUelzAgEa",
        gameConfig: GameConfig(
            sceneMode: 0,
            currencyIcon:
                "https://bsyx-store.s3.ap-southeast-1.amazonaws.com/game-icons/20230906-163004.png"));
        String jsUrl = jsFunName + "({$toJsonEncode(_configData.toJson())})";
        controller!.runJavaScript(jsUrl);
    } else if (jsFunName.contains('destroy')) {
        print("BSGAME Game call destroy main.dart");
        //Close the game TODO client
        controller.loadRequest(Uri.parse('about:blank'));
        super.dispose();
    } else if (jsFunName.contains('gameRecharge')) {
        print("BSGAME Game call gameRecharge main.dart");
        //Pull up the payment mall TODO client
    } else if (jsFunName.contains('gameLoaded')) {
        print("BSGAME Game call gameLoaded main.dart");
        //Game loaded TODO client
    }
}
});
}

```

### 7.3.5 If you encounter problems when accessing, you can turn on browser debugging to troubleshoot

```

if (controller.platform is AndroidWebViewController) {
    // TODO android turns on debugging, needs to be changed to false when publishing
    AndroidWebViewController.enableDebugging(false);
    (controller.platform as AndroidWebViewController)
        .setMediaPlaybackRequiresUserGesture(false);
}

```

## 7.4 Flutter IOS

You can directly use Flutter's built-in JavaScriptChannel to communicate with the game.

### 7.4.1 Add support for HTTP web pages

Since HTTPS is enabled by default in iOS version 9.0, you need to add the following code to the ios/Runner/Info.plist file.

```
<key>io.flutter.embedded_views_preview</key><string>YES</string>
```

### 7.4.2 Allow opening external links

Add the following code to the ios/Runner/Info.plist file.

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
</dict>
```

### 7.4.3 Binding protocol method

```
controller.addJavaScriptChannel("getConfig", onMessageReceived: (JavaScriptMessage
jsonMessage){
  print("BSGAME Game call getConfig main.dart ios");
  final obj = json.decode(jsonMessage.message);
  String jsFunName = obj['jsCallback'];
  GetConfigData _configData = GetConfigData(
    appChannel: "mesh",
    appld: 88888888,
    userId: "338426830",
    gameMode: "3",
    language: "0",
    gsp: 101,
    roomId: '20240101',
    code:
      "SqUdd4KrRa8tPvgoboPEQcJy0V3UQ0ywur612cjHtQhMOsoutmNUelzAgEa",
    gameConfig: GameConfig(
      sceneMode: 0,
      currencyIcon:

"https://bsyx-store.s3.ap-southeast-1.amazonaws.com/game-icons/20230906-163004.png");
    String jsUrl = jsFunName + "(${jsonEncode(_configData.toJson())})";
    controller!.runJavaScript(jsUrl);
  });
controller.addJavaScriptChannel("destroy", onMessageReceived: (JavaScriptMessage
message){
  print("BSGAME Game call destroy main.dart ios");
  controller.loadRequest(Uri.parse('about:blank'));
  super.dispose();
});
controller.addJavaScriptChannel("gameRecharge", onMessageReceived: (JavaScriptMessage
message){
  print("BSGAME Game call gameRecharge main.dart ios");
});
controller.addJavaScriptChannel("gameLoaded", onMessageReceived: (JavaScriptMessage
message){
  print("BSGAME Game call gameLoaded main.dart ios");
```

```
});
```

#### 7.4.4 If you encounter problems when accessing, you can turn on browser debugging to troubleshoot

```
path:ios/.symlinks/plugins/webview_flutter_wkwebview/lib/src/webkit_webview_controller.dart
@override
Future<void> setJavaScriptMode(JavascriptMode javascriptMode) {
  //TODO ios turns on debugging, needs to be changed to false when publishing
  this.setInspectable(false);
  switch (javascriptMode) {
    case JavascriptMode.disabled:
      return _webView.configuration.preferences.setJavaScriptEnabled(false);
    case JavascriptMode.unrestricted:
      return _webView.configuration.preferences.setJavaScriptEnabled(true);
  }
}
```

#### 7.4.5 Touch penetration issue on iOS

When using the webview\_flutter plugin or similar WebView solutions in Flutter, when the game overlays other business UIs, WebView on iOS usually prioritizes touch events. You may encounter touch penetration issues on iOS, where even if a GestureDetector is placed on top of WebView, it cannot completely prevent WebView from receiving touch events.

##### 7.4.5.1 Setting up the Platform Channel in Flutter

First, define a Platform Channel in the Flutter project to communicate with the native iOS code. This Channel will be used to call methods in the native code to control the touch behavior of the WebView.

```
import 'package:flutter/services.dart';
```

```
class WebViewTouchControl {
  static const MethodChannel _channel =
    MethodChannel('webview_touch_control');

  static Future<void> disableWebViewTouch() async {
    try {
      await _channel.invokeMethod('disableWebViewTouch');
    } on PlatformException catch (e) {
      print("Failed to disable WebView touch: '${e.message}'.");
    }
  }

  static Future<void> enableWebViewTouch() async {
    try {
```



```

        await _channel.invokeMethod('ableWebViewTouch');
    } on PlatformException catch (e) {
        print("Failed to disable WebView touch: '${e.message}'.");
    }
}

}

```

#### 7.4.5.2 Implementing plugins in iOS native code

On iOS, you need to create a plugin class to handle Flutter method calls and implement the corresponding methods to find and disable touch events for WKWebView.

##### **WebViewTouchControlPlugin.h:**

```
#import <Flutter/Flutter.h>
```

```
@interface WebViewTouchControlPlugin : NSObject<FlutterPlugin>
```

```
@end
```

##### **WebViewTouchControlPlugin.m:**

```
#import "WebViewTouchControlPlugin.h"
```

```
#import <WebKit/WebKit.h>
```

##### **@implementation WebViewTouchControlPlugin**

```

+ (void)registerWithRegistrar:(NSObject<FlutterPluginRegistrar>*)registrar {
    FlutterMethodChannel* channel = [FlutterMethodChannel
        methodChannelWithName:@"webview_touch_control"
        binaryMessenger:[registrar messenger]];
    WebViewTouchControlPlugin* instance = [[WebViewTouchControlPlugin alloc] init];
    [registrar addMethodCallDelegate:instance channel:channel];
}

```

```

- (void)handleMethodCall:(FlutterMethodCall*)call result:(FlutterResult)result {
    if ([@"disableWebViewTouch" isEqualToString:call.method]) {
        [self disableWebViewTouch];
        result(nil);
    } else if ([@"ableWebViewTouch" isEqualToString:call.method]){
        [self ableWebViewTouch];
        result(nil);
    } else {
        result(FlutterMethodNotImplemented);
    }
}

```

```

- (void)disableWebViewTouch{
    NSLog(@"Disable webview touch. . . . .");
}

```

```

    UIViewController *rootViewController =
    UIApplication.sharedApplication.keyWindow.rootViewController;
    WKWebView *webView = [self findWebViewInView:rootViewController.view];
    if (webView) {
        webView.userInteractionEnabled = NO;
        // You can further disable other gestures or events as needed
    }
}

```

```

- (void)enableWebViewTouch{
    NSLog(@"Enable webview touch. . . . .");
    UIViewController *rootViewController =
    UIApplication.sharedApplication.keyWindow.rootViewController;
    WKWebView *webView = [self findWebViewInView:rootViewController.view];
    if (webView) {
        webView.userInteractionEnabled = YES;
        // Other gestures or events can be further enabled as needed
    }
}

```

```

- (WKWebView *)findWebViewInView:(UIView *)view {
    for (UIView *subview in view.subviews) {
        if ([subview isKindOfClass:[WKWebView class]]) {
            return (WKWebView *)subview;
        } else {
            WKWebView *foundWebView = [self findWebViewInView:subview];
            if (foundWebView) {
                return foundWebView;
            }
        }
    }
    return nil;
}

```

**@end**

#### 7.4.5.3 Registering a plugin

Register the plugin class in the [AppDelegate.m](#) file:

```

#import "AppDelegate.h"
#import "GeneratedPluginRegistrant.h"
#import "WebViewTouchControlPlugin.h"
@implementation AppDelegate

```

```

- (BOOL)application:(UIApplication *)application

```

```

didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    [GeneratedPluginRegistrant registerWithRegistry:self];
    // Registering a plugin
    [WebViewTouchControlPlugin registerWithRegistrar:[self
registrarForPlugin:@"WebViewTouchControlPlugin"]];
    // Override point for customization after application launch.
    return [super application:application didFinishLaunchingWithOptions:launchOptions];
}

@end

```

#### 7.4.5.4 Calling native methods in Flutter

Calling `WebViewTouchControl.disableWebViewTouch()` method to disable touch events of WebView;

Calling `WebViewTouchControl.ableWebViewTouch()` method to enable touch events for WebView.

```
import 'package:flutterdemo_webview/webview_touch_control.dart';
```

```
//Declaring variables
```

```
bool disableWebViewTouch = false;
```

```
//Handling WebViewTouch
```

```

void _handleWebViewTouch() async {
  if (Platform.isIOS) {
    disableWebViewTouch = !disableWebViewTouch;
    if(disableWebViewTouch){
      WebViewTouchControl.disableWebViewTouch();
    }else{
      WebViewTouchControl.ableWebViewTouch();
    }
  }
}

```

## 8. ReactNative implementation code

After RN was upgraded to 0.61.5, webview was deleted from the react native official API, and the official recommendation is to use react-native-webview.

The current sample code uses "react-native": "0.70.1", "react-native-webview": "^13.10.2"

### 8.1 Adding Dependencies

Enter the project root directory and execute

```
yarn add react-native-webview
```

or

```
npm install --save react-native-webview
```

## 8.2 Linking native dependencies

As of react-native 0.60, autolink will take care of the linking step, but don't forget to run pod install.

```
react-nativelink react-native-webview
```

*Note: If you need to uninstall React Native WebView, run react-native unlink react-native-webview to unlink it.*

iOS: If using CocoaPods, run pod install in the ios directory.

## 8.3 Import references

```
import WebView from 'react-native-webview';
```

## 8.4 Instantiate WebView

```
const webViewRef = useRef(null);
const [showWebView, setShowWebView] = useState(true); // Control whether WebView is displayed
return (
  <SafeAreaView style={{ flex: 1 }}>
    {showWebView && (<WebView
      ref={(r) => (this.webViewRef = r)}
      source={{ uri:
'https://game-center-test.jieyou.shop/game-packages/common-web/rocket/1.3.3/web-mobile/index.html' }}
      onMessage={handlemessage}
      javaScriptEnabled={true}
      //TODO Enable debugging, change to false when publishing
      webViewDebuggingEnabled = {false}
    />)}
  </SafeAreaView>);
```

## 8.5 Binding protocol method

```
//Receive H5 messages
const handlemessage = (event) => {
  const message = JSON.parse(event.nativeEvent.data);
  const jsFunName = message["jsCallback"];
  console.log(jsFunName)
  if (jsFunName.includes('getConfig')) {
    console.log("BSGAME Game call getConfig")
    const configData = {
      appChannel: "mesh",
      appId: 88888888,
```

```

    userId: "338426830",
    code: "SqUdd4KrRa8tPvgoboPEQcJy0V3UQ0ywur612cjHtQhMOsoumtmNUelzAgEa",
    roomId: "202307",
    gameMode: "3",
    language: "2",
    gameConfig: {
      sceneMode: 0,
      currencyIcon:
"https://bobilive-com-test.jieyou.shop/h5web/GameH5/showDemo/game_bean.png"
    },
    gsp: 101
  }
  const configDataStr = JSON.stringify(configData);
  this.webViewRef.injectJavaScript(`window.${jsFunName}(${configDataStr})`);
}
else if (jsFunName.includes('destroy')) {
  console.log("BSGAME Game call destroy");
  //Close the game TODO client
  setShowWebView(false);
}
else if (jsFunName.includes('gameRecharge')) {
  console.log("BSGAME Game call gameRecharge")
  //Pull up the payment mall TODO client
}
else if (jsFunName.includes('gameLoaded')) {
  console.log("BSGAME Game call gameLoaded")
  //Game loaded TODO client
}
};

```

## 8.6 ReactNative Android

### 8.6.1 Solution to link native dependency failure

#### 1.Manually edit the settings.gradle file

path:/android/settings.gradle,Add the following two lines  
 rootProject.name = 'reactNativeDemo'

```

...
include ':react-native-webview'
project(':react-native-webview').projectDir = new File(rootProject.projectDir,
'../node_modules/react-native-webview/android')
...

```

```
include ':app'
```

## 2. Manually edit the build.gradle file

path:/android/app/build.gradle, Add the following line  
dependencies {

```
...
    implementation project(':react-native-webview')
...
}
```

## 3. Manually edit the MainApplication.java file

path:/android/app/src/main/java/com/reactnatedemo/MainApplication.java, Add Import  
import com.reactnativecommunity.webview.RNCWebViewPackage;

### 8.6.2 Binding NativeBridge

path:/node\_modules/react-native-webview/android/src/main/java/com/reactnativecommunity/webview/RNCWebView.java

```
public class RNCWebView extends WebView implements LifecycleEventListener {
```

```
    NativeBridge bridge_bs;
```

```
    protected RNCWebViewBridge createRNCWebViewBridge(RNCWebView webView) {
        if (bridge == null) {
            bridge = new RNCWebViewBridge(webView);
            addJavascriptInterface(bridge, JAVASCRIPT_INTERFACE);
            //baishun
            bridge_bs = new NativeBridge(webView);
            addJavascriptInterface(bridge_bs, "NativeBridge");
        }
        return bridge;
    }
}
```

```
}
```

### 8.6.3. Add NativeBridge Class

path:/node\_modules/react-native-webview/android/src/main/java/com/reactnativecommunity/webview/RNCWebView.java

```
protected class NativeBridge {
    RNCWebView mWebView;
```

```

NativeBridge(RNCWebView c) {
    mWebView = c;
}
// Add JS calling interface
// Add JS calling interface
@JavascriptInterface
public void getConfig(String msg) {
    if (mWebView.getMessagingEnabled()) {
        mWebView.onMessage(msg);
    }
}
@JavascriptInterface
public void destroy(String msg) {
    if (mWebView.getMessagingEnabled()) {
        mWebView.onMessage(msg);
    }
}
@JavascriptInterface
public void gameRecharge(String msg) {
    if (mWebView.getMessagingEnabled()) {
        mWebView.onMessage(msg);
    }
}
@JavascriptInterface
public void gameLoaded(String msg) {
    if (mWebView.getMessagingEnabled()) {
        mWebView.onMessage(msg);
    }
}
}

```

## 8.7 ReactNative IOS

### 8.7.1 Binding protocol method

After modify [RNCWebViewImpl.m file](#)

path:/node\_modules/react-native-webview/apple/RNCWebViewImpl.m

Add a statement

```

static NSString *const getConfig_bs = @"getConfig";
static NSString *const destroy_bs = @"destroy";
static NSString *const gameRecharge_bs = @"gameRecharge";
static NSString *const gameLoaded_bs = @"gameLoaded";

```

//Add Remove

```

- (void)removeFromSuperview
#endif
{
    if (_webView) {

        [_webView.configuration.userContentController
removeScriptMessageHandlerForName:MessageHandlerName];
        [_webView.configuration.userContentController
removeScriptMessageHandlerForName:getConfig_bs];
        [_webView.configuration.userContentController
removeScriptMessageHandlerForName:destroy_bs];
        [_webView.configuration.userContentController
removeScriptMessageHandlerForName:gameRecharge_bs];
    }

    //Handling message dispatch
- (void)userContentController:(WKUserContentController *)userContentController
    didReceiveScriptMessage:(WKScriptMessage *)message
{
    ...

    else if ([message.name isEqualToString:getConfig_bs] ||
        [message.name isEqualToString:destroy_bs] ||
        [message.name isEqualToString:gameRecharge_bs] ||
        [message.name isEqualToString:gameLoaded_bs]) {
        if (_onMessage) {
            NSMutableDictionary<NSString *, id> *event = [self baseEvent];
            [event addEntriesFromDictionary: @[@"data": message.body]];
            _onMessage(event);
        }
    }
    ...
}

//Add Listener
- (void)resetupScripts:(WKWebViewConfiguration *)wkWebViewConfig {
    ...

    [wkWebViewConfig.userContentController
removeScriptMessageHandlerForName:getConfig_bs];
    [wkWebViewConfig.userContentController
removeScriptMessageHandlerForName:destroy_bs];

```



```

[webViewConfig.userContentController
removeScriptMessageHandlerForName:gameRecharge_bs];
[webViewConfig.userContentController
removeScriptMessageHandlerForName:gameLoaded_bs];
[webViewConfig.userContentController
addScriptMessageHandler:[RNCWeakScriptMessageDelegate alloc] initWithDelegate:self]
name:getConfig_bs];
[webViewConfig.userContentController
addScriptMessageHandler:[RNCWeakScriptMessageDelegate alloc] initWithDelegate:self]
name:destroy_bs];
[webViewConfig.userContentController
addScriptMessageHandler:[RNCWeakScriptMessageDelegate alloc] initWithDelegate:self]
name:gameRecharge_bs];
[webViewConfig.userContentController
addScriptMessageHandler:[RNCWeakScriptMessageDelegate alloc] initWithDelegate:self]
name:gameLoaded_bs];
...
}

```

## 9. Unity implementation code

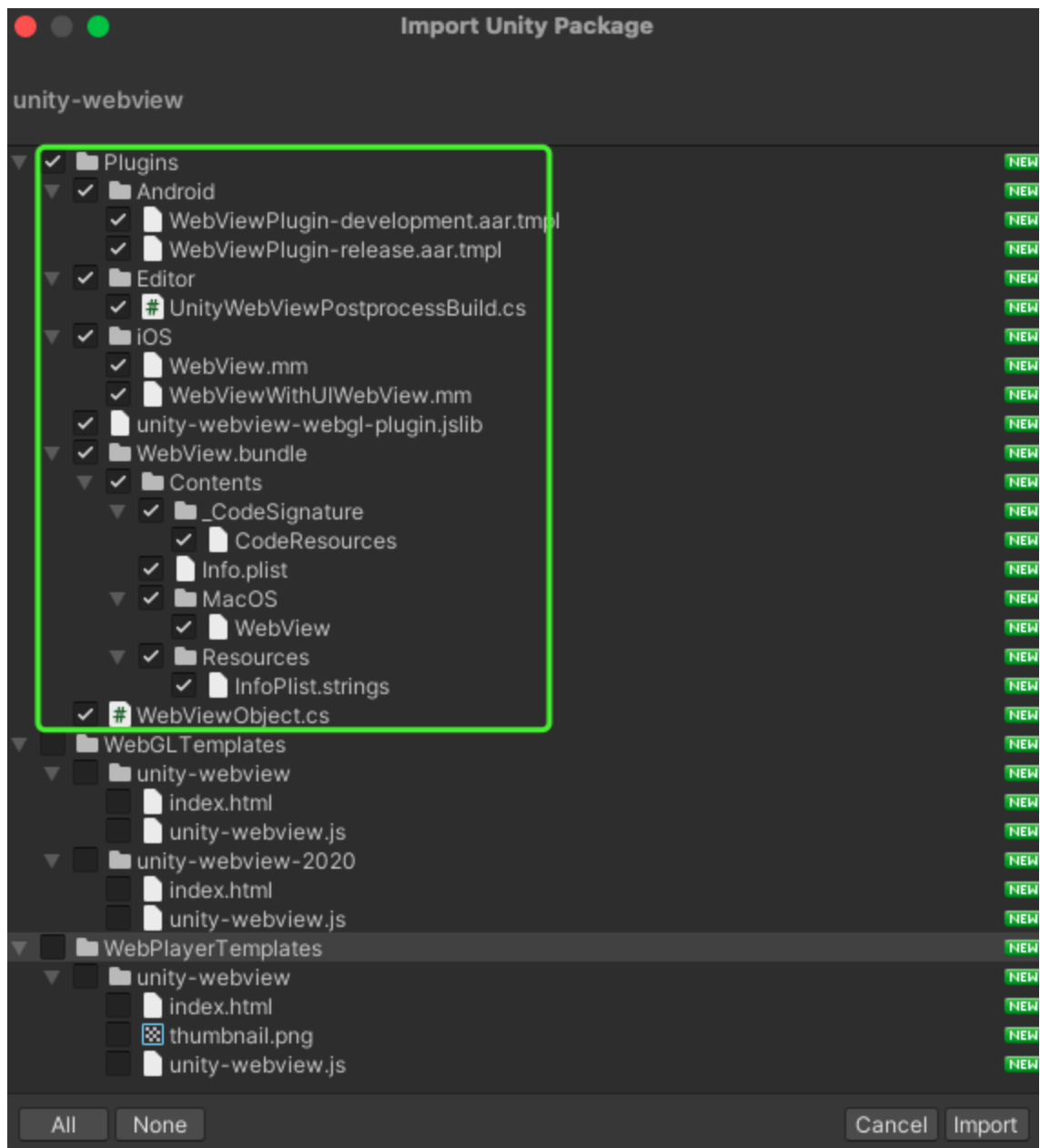
### 9.1 Import plugin

#### 9.1.1 Download the plugin to your local computer.

Download: [下载地址](#)

#### 9.1.2 Importing to Unity

Click the Assets menu, select Import Package and choose Custom Package, and select the previously downloaded package to import.



## 9.2 Initialize WebView and register message listener

```
using System.Collections;  
using UnityEngine;  
using UnityEngine.Networking;  
using UnityEngine.UI;  
using System.Collections.Generic;
```

```
public class WebViewController : MonoBehaviour  
{
```

```

private WebViewObject webViewObject;

[System.Serializable]
public class GameMessage
{
    public string jsCallback;
}

[System.Serializable]
public class GetConfigData
{
    public string appChannel = "";
    public int appld = 0;
    public string userId = "";
    public string code = "";
    public string roomId = "";
    public string gameMode = "";
    public string language = "2";
    public GameConfig gameConfig = new GameConfig();
    public int gsp = 101;
}

[System.Serializable]
public class GameConfig
{
    public int sceneMode = 0;
    public string currencyIcon = "";
}

void Start()
{
    Debug.Log("WebViewController initialization");
}

void showWebview()
{
    //Prevent multiple webviews from existing at the same time due to consecutive clicks
    hideWebview();
    webViewObject = (new
GameObject("WebViewObject")).AddComponent<WebViewObject>();
    webViewObject.Init(
        cb: msg =>
        {
            GameMessage messsage = JsonUtility.FromJson<GameMessage>(msg);

```

```
string jsFunName = message.jsCallback;
if (jsFunName.Contains("getConfig"))
{
```

```
    Debug.Log("BSGAME Game call getConfig");
```

The data is only a reference value and needs to be assigned according to your own

APP

```
    GetConfigData getConfigData = new GetConfigData();
```

```
    getConfigData.appChannel = "mesh";
```

```
    getConfigData.appId = 913522301;
```

```
    getConfigData.userId = "338426830";
```

```
    getConfigData.code =
```

```
"SqUdd4KrRa8tPvgoboPEQcJy0V3UQ0ywur612cjHtQhMOsoutmNUelzAgEa";
```

```
    getConfigData.roomId = "10023";
```

```
    getConfigData.gameMode = "3";
```

```
    getConfigData.language = "2";
```

```
    getConfigData.gameConfig.sceneMode = 0;
```

```
    getConfigData.gameConfig.currencyIcon =
```

```
"https://bsyx-store.s3.ap-southeast-1.amazonaws.com/game-icons/20230906-163004.png";
```

```
    getConfigData.gsp = 101;
```

```
    //Return data to game JS
```

```
    string str = jsFunName + "(" + JsonUtility.ToJson(getConfigData, true) + ")";
```

```
    webViewObject.EvaluateJS(str);
```

```
}
```

```
else if (jsFunName.Contains("destroy"))
```

```
{
```

```
    Debug.Log("BSGAME Game Call destroy");
```

```
    hideWebview();
```

```
}
```

```
else if (jsFunName.Contains("gameRecharge"))
```

```
{
```

```
    Debug.Log("BSGAME Game Call gameRecharge");
```

```
    //Pull up the TODO client of the recharge mall
```

```
}
```

```
else if (jsFunName.Contains("gameLoaded"))
```

```
{
```

```
    Debug.Log("BSGAME Game Call gameLoaded");
```

```
    //Game loaded TODO client
```

```
}
```

```
},
```

```
transparent: true, //Background transparency can be turned on (default is white)
```

```
//ios
```

```
enableWKWebView: true
```

```
);
```

```

// Set the webview size
// left, top, right, bottom Unit: px
webViewObject.SetMargins(0, 0, 0, 0);
webViewObject.SetTextZoom(100);
webViewObject.SetMixedContentMode(2);
webViewObject.SetVisibility(true);
string gameUrl =
"https://game-center-test.jieyou.shop/game-packages/common-web/fishing/2.6.2/web-mobile/in
dex.html?isUnity=1";
if (gameUrl.StartsWith("http"))
{
    webViewObject.LoadURL(gameUrl.Replace(" ", "%20"));
}
}

void hideWebview()
{
    var webViewObject = GameObject.Find("WebViewObject");
    if (webViewObject)
    {
        Destroy(webViewObject);
    }
}
}

```

**Special note: When opening the game link gameUrl, you need to add ?isUnity=1 after index.html.**

### 9.3 Actively call walletUpdate

```

//walletUpdate User currency changes
Dictionary<string, object> dictionary = new Dictionary<string, object>();
dictionary["userId"] = "xxxx";
string str = "walletUpdate" + "(" + JsonUtility.ToJson(dictionary, true) + ")";
webViewObject.EvaluateJS(str);

```

### 9.4 Set loading to allow loading local file URLs

- Android is enabled by default, no additional configuration is required
- iOS is disabled by default. WKWebView does not allow access to file URLs via XMLHttpRequest. You need to define the preprocessor macro UNITYWEBVIEW\_IOS\_ALLOW\_FILE\_URLS.

- From the Unity menu bar, choose **Edit> Project Settings**.
- In the **Project Settings** window, choose **Player** setting, choose iOS platform, Find **Other Settings** part, Expand **Scripting Define Symbols**, Add to **UNITYWEBVIEW\_IOS\_ALLOW\_FILE\_URLS**.
- Rebuild the project.

## 9.5 If you encounter problems when accessing, you can turn on browser debugging to troubleshoot

Click the File menu, select Build Settings, and check Development Build.

Android local debugging reference: [https://blog.csdn.net/qq\\_38238956/article/details/128325141](https://blog.csdn.net/qq_38238956/article/details/128325141)

IOS local debugging reference: <https://www.ngui.cc/el/792246.html?action=onClick>

## 10. Common Docking Issues

### 10.1 System/browser Version Requirements

#### Recommended to Use

- a. IOS11 above
- b. Android 5 above

### 10.2 The game cannot be opened normally

- a. White Screen
- b. Stay on the loading page

Android Local Debugging Reference:

[https://blog.csdn.net/qq\\_38238956/article/details/128325141](https://blog.csdn.net/qq_38238956/article/details/128325141)

IOS Local debugging reference: <https://www.ngui.cc/el/792246.html?action=onClick>

After debugging, please provide specific error message screenshots or log feedback to BAISHUN.

## 11. List

### 11.1 Gamelits

Number(game_id)	Game
1001	Texas Hold'em

1004	Slots
1005	Joy Bounce
1006	Fruit Slot
1007	Angel and Devil
1008	Whack-a-Mole
1009	Gesture Guess
1010	Lottery
1012	Digital War
1013	Horse Racing
1014	Fast 3
1015	Billiards
1016	Rocket
1017	Fruit 2
1018	Horse Racing 2
1019	Slots 2
1020	Fruit Ranch
1021	Lucky Gift
1022	Fishing
1023	Lava Slots

1024	Camel Racing
1025	Dog Racing
1026	Roulette Chest
1027	Witch Slots
1028	LUDO
1029	Fruit Roulette

## 11.2 Error List

code	Description
0	Sucessful
1001	Data Error
1002	Reading Data Error
1003	Signature Error
1004	Failed to send
1005	Parsing failed
1006	Failed to serialize data
1007	app not added
1008	Insufficient assets
1012	Channel game not found
1019	The game is under maintenance



1020	User is banned
1021	ErrorAPIServer, API Server internal error
1022	Player account function restrictions
1023	Restricted IP

### 11.3 Multilingual Comparison Table

Language Type	Type	Value
Chinese	string	0
Traditional Chinese	string	1
English	string	2
Indonesian	string	3
Malay	string	4
Thai	string	5
Vietnamese	string	6
Arabic	string	7
Filipino	string	8
Portuguese	string	9
Turkish	string	10
Urdu	string	11
Japanese	string	12
Russian	string	13

Spanish	string	14
Hindi	string	38