GameSDK Installation (Android)

Name	Date	version	remark
Nengjie	2020.3.17	V1.0.0	created
Nengjie	2020.4.8	V1.0.1	Improvements
Nengjie	2020.4.14	V1.0.2	Improvements
Nengjie	2020.4.20	V1.0.4	Improvements
Nengjie	2020.4.23	V1.0.5	Improvements
Nengjie	2020.5.8	V1.0.7	Improvements
Nengjie	2020.5.8	V1.0.8	Add Google login and PayBy payment
Nengjie	2020.5.26	V1.0.10	Improvements
Nengjie	2020.6.8	V1.1.1	Improvements
Nengjie	2020.6.24	V1.1.2	Add facebook login and twitter login
Nengjie	2020.7.6	V.1.1.3	1. Add appsFlyer and tapjoy. 2 Add the ability to block users.
Nengjie	2020.7.19	V1.1.4	Add huawei
Nengjie	2020.8.3	V1.1.5	Add Botim

1. Version update notes:

- In 2.7, add botim configuration, see details.

2 Configuration

2.1 Register and create app

Get the app's appId, appSecret, gameId, provided by AuGame.

2.2 Download resources

Please download the latest resource pack at

2.3 Add SDK

Please copy the SDK/GameSDK-Android(xxx).aar into the lib of the main Module of the project.

2.4 Add the google-services.json file

google-services.json file provided by AmuseGames.

agconnect-services.json file provided by AmuseGames.

1. Add google-services.json file to your module (application level).

2. Add agconnect-services.json file to your module (application level).

Note:

1. If you don't need access to Huawei, you don't need to configure agconnect-services.json.

2.5 Project-level build.gradle configuration

In the build.gradle file at the project level, add the following dependencies

```
buildscript {
    repositories{
        google()
        jcenter()
        maven {url 'https://developer.huawei.com/repo/'}
    }
    dependencies {
        classpath 'com.google.gms:google-services:4.2.0'
        classpath 'com.huawei.agconnect:agcp:1.3.2.301'
    }
}
allprojects{
    repositories{
        google()
        jcenter()
        maven {url 'https://developer.huawei.com/repo/'}
    }
}
```

Note:

1. If you don't need access to Huawei, you don't need to configure "`maven {url 'https://developer.huawei.com/repo/'}`" and "`classpath 'com.huawei.agconnect:agcp:1.3.2.301'`".

2.6 App/build.gradle configuration

In the build.gradle file, add the following dependencies:

```
apply plugin: 'com.google.gms.google-services'
apply plugin: 'com.huawei.agconnect'

android {
    repositories {
        flatDir {
            dirs 'libs'
        }
    }
}
dependencies {
    implementation 'com.android.billingclient:billing:2.0.3'

    implementation 'com.google.android.gms:play-services-auth:18.0.0'

    implementation 'com.facebook.android:facebook-login:7.0.0'

    implementation('com.twitter.sdk.android:twitter:3.3.0@aar') {
        transitive = true
    }

    implementation 'com.google.firebase:firebase-core:17.0.0'
    implementation 'com.google.firebase:firebase-messaging:20.1.0'

    implementation 'com.google.firebase:firebase-dynamic-links:19.0.0'

    implementation 'com.google.firebase:firebase-analytics:17.2.0'

    implementation(name: 'GameSDK-Android(xxx)', ext: 'aar')}

    implementation 'com.appsflyer:af-android-sdk:5.4.0'
    implementation 'com.android.installreferrer:installreferrer:1.1.2'

    implementation 'com.tapjoy:tapjoy-android-sdk:12.4.2@aar'


    implementation 'com.huawei.hms:hwid:4.0.4.300'
    implementation 'com.huawei.hms:iap:4.0.4.301'
    implementation 'com.huawei.hms:game:5.0.0.300'

}
```

Sync Now appears in the upper right corner, click on it.

Note:

1. If you don't need google pay, you don't need to configure the "implementation 'com.android.billingclient:billing:2.0.3'".

2. If you don't need google login, you don't need to configure the "implementation 'com.google.android.gms:play-services-auth:18.0.0'".

3. If you don't need facebook login, you don't need to configure the "implementation

com.facebook.android:facebook-login:7.0.0".

4. If you don't need twitter login, you don't need to configure the "implementation('com.twitter.sdk.android:twitter:3.3.0@aar') { transitive = true}".

5. If you don't need to push, you don't need to configure the "implementation 'com.google.firebase:firebase-messaging:20.1.0'".

6. After inviting friends through sharing, if you configure "implementation 'com.google.firebase-:firebase-dynamic-links:19.0.0'", if your friends do not have the current game installed, you can download the current game through google paly or webpage and install it, after installation, it will generate the corresponding invitation relationship.

7. After configuring the "implementation 'com.google.firebase:firebase-analytics:17.2.0' "After that, our platform performs data analysis on the users of the current game.

8. "implementation 'com.google.firebase:firebase-core:17.0.0'" is the base library for firebase, if one or more of messaging, dynamic-links, analytics are configured, then this must be configured.

9. If you don't need to analyze the installation source, you don't need to configure "implementation 'com.appsflyer:af-android-sdk:5.4.0' " and " implementation com.android.installreferrer:installreferrer:1.1.2".
10. If you don't need to analyze the installation source, you don't need to configure "implementation 'com.tapjoy:tapjoy-android-sdk:12.4.2@aar'".

11. If you don't need access to Huawei, you don't need to configure the "implementation 'com. huawei.hms:hwid:4.0.4.300'" and "implementation 'com.huawei.hms:hwid:4.0.4.300'" and "implementation 'com.huawei.hms:hwid:4.0. 4.300'"

2.7 Configuration project information

Configured in AndroidManifest.xml, example:

```
<application>
```

```xml
<meta-data
    android:name="ttk_game_channel"
    android:value="ttk_your channel" />
<meta-data
    android:name="ttk_game_gameId"
    android:value="ttk_your gameId" />
<meta-data
    android:name="ttk_game_appId"
    android:value="ttk_your appId" />
<meta-data
    android:name="ttk_game_appSecret"
    android:value="ttk_your appSecret" />


<meta-data
    android:name="ttk_game_botim_clientId"
    android:value="ttk_your botim clientId" />



<!--If you don't need a Google login, you don't need to configure this-->
<meta-data
    android:name="ttk_game_google_clientId"
    android:value="ttk_your google clientId" />


<!--If you don't need a facebook login, you don't need to configure this-->
 <meta-data android:name="com.facebook.sdk.ApplicationId"
    android:value="@string/facebook_app_id"/>
 <activity android:name="com.facebook.FacebookActivity"
    android:configChanges="keyboard|keyboardHidden|
    screenLayout|screenSize|orientation"
    android:label="@string/app_name" />
 <activity android:name="com.facebook.CustomTabActivity"
    android:exported="true">
    <intent-filter>
       <action android:name="android.intent.action.VIEW" />
       <category android:name="android.intent.category.DEFAULT" />
       <category android:name="android.intent.category.BROWSABLE" />
```

```xml
                <data android:scheme="@string/fb_login_protocol_scheme" />
            </intent-filter>



        <provider
            android:authorities=
                "com.facebook.app.FacebookContentProvider{
                your facebook_app_id}"
            android:name="com.facebook.FacebookContentProvider"
            android:exported="true"/>
    </activity>


    <!--If you don't need a twitter login, you don't need to configure this-->
    <meta-data android:name="ttk_game_twitter_consumerKey"
        android:value="ttk_your twwiter CONSUMER_KEY"/>
    <meta-data android:name="ttk_game_twitter_consumerSecret"
        android:value="ttk_your twwiter CONSUMER_SECRET"/>
    <meta-data
        android:name="ttk_game_af_key"
        android:value="ttk_your af key" />
    <meta-data
        android:name="ttk_game_tapjoy_key"
        android:value="ttk_your tapjoy key" />
    <meta-data
        android:name="com.huawei.hms.client.appid"
        android:value="huawei appid" />

</application>
```

Configured in String.xml, example:

```xml
<!--If you don't need a facebook login, you don't need to configure this-->
<string name="app_name">your app name</string>
<string name="facebook_app_id" >your facebook_app_id</string>
<string name="fb_login_protocol_scheme">
    fb[your facebook_app_id]</string>
```

```
<!--If you don't need a twitter login, you don't need to configure this-->
<string name="com.twitter.sdk.android.CONSUMER_KEY" >
    your twiter CONSUMER_KEY</string>
<string name="com.twitter.sdk.android.CONSUMER_SECRET" >
    your twiter CONSUMER_SECRET</string>
```

Note:

1. Please replace "your channel" with the current channel, optional values "ttk", "botim", "huawei".

2.Please replace "your gameId" with the gameId assigned by our platform for your game.

3. Please replace "your appId" with the appId assigned by our platform for your game.

4. Please replace "your appSecret" with the appSecret assigned by our platform for your game.

5. Please replace "your google clientId" with the GoogleClientId assigned by our platform for your game. If you don't need a Google login, you don't need to configure this.

6. Please replace "your facebook_app_id" with the facebook appId assigned by our platform for your game. If you don't need a facebook login, you don't need to configure this.

7. Please replace "your twiter CONSUMER_KEY" with the your twiter CONSUMER_KEY assigned by our platform for your game. If you don't need a twitter login, you don't need to configure this.

8. Please replace "your twiter CONSUMER_SECRET" with the your twiter CONSUMER_SECRET assigned by our platform for your game. If you don't need a teitter login, you don't need to configure this.

9. Please replace "your af key" with the appsflyer key assigned by our platform for your game. If you don't need a Google login, you don't need to configure this.

10. Please replace "your tapjoy key" with the tapjoy key assigned by our platform for your game. If you don't need a Google login, you don't need to configure this.

11. Please replace the "huawei appid" with the Huawei appid assigned by our platform for your game. If you don't need access to Huawei, you don't need to configure this.

12. Please replace "your botim clientId" with the botim clientId assigned by our platform for your game. If you don't need a Botim, you don't need to configure this.

2.8 Configuring the first start of an activity

Please configure the blue section in the first activity you start, example：

```
<application>
  <activity android:name=".MainActivity">
    <intent-filter>
       <action android:name="android.intent.action.MAIN" />
       <category android:name="android.intent.category.LAUNCHER" /
    </intent-filter>

  <intent-filter android:autoVerify="true">
      <action android:name="android.intent.action.VIEW" />
      <category android:name="android.intent.category.DEFAULT" />
      <category android:name="android.intent.category.BROWSABLE" />
      <data
         android:scheme="your applicationId"
         android:host="web.game"
         android:pathPrefix="/your gameId/"
       />
      <data
         android:scheme="https"
         android:host="domain"
         android:pathPrefix="/web/game/your gameId/"
```

```
        />
      </intent-filter>
  </activity>
</application>
```

Note:

1. After configuring this, you can launch the current game directly in Game Center TTK APP

2.Please replace "your applicationId" with the applicationId assigned to you by our platform.

3.Please replace "your gameId" with the gameId assigned by our platform for your.

4.In the test environment: replace "domain" with "t-www.amusgame.net".

5.In a production environment: replace "domain" with "www.falcongames.net".

3 API Usage

3.1 Initialization and configuration

Step 1: In the API document, turn log output on or off via 4.1 and 4.2 (off by default), set the server environment via 4.3 or 4.4 (default production environment), and set the payment environment via 4.5 or 4.6 (default production environment).

Step 2: Please initialize via API document 3.2.

Step 3: The initialization callback function's initComplete determines whether the initialization is complete and receives the push message in onMessageReceived.

Step 4:　The SDK will execute the onLoginExpired function after the access token is invalid, please add the relevant operation of re-login in this function.

Step 5: When abnormal behavior is detected, it will disable the current user and trigger onForbiddenCallback, please perform logout and close the game after logging out in onForbiddenCallback.

3.2 Add life cycle functions

Please call the required function in the lifecycle function of the homepage。Please call the required function in the lifecycle function of the homepage. For specific

functions, please refer to 5.1, 5.2, 5.3, 5.4 of API Document.

3.3 Game version updated

Step 1: Upload new version information on the Game Center platform, such as whether updates are mandatory, download address, etc.

Step 2: Get the version information of the game through the method provided by the SDK, see 4.7 of the API Document for version information.

Step 3: After obtaining the latest version information, determine if a mandatory update is required, and if so, prompt the user to update directly, otherwise affecting the use of other methods of SDK.

3.4 Account-related APIs

Step 1: Before logging in, the Privacy Policy and User Agreement page can be opened via 6.2 in the API documentation.

Step 2: In the API documentation, sex login options are provided, namely 6.3, 6.4,6.5 6.6, 6.7 and 6.8, with 6.4 login via TTK app, 6.5 via google account, 6.6 via facebook login, 6.7 via twitter login，6.8 via huawei login.

Step 3: In API Document, after a successful login, the login user information is obtained by 6.10 and the login type is obtained by 6.11.

Step 4: Add a forced downline callback to the account, which is triggered when the current account is logged in elsewhere，Please see 6.9 in the API document.

Step 5: In the API Document, exit the current account via 6.12.

3.5 Payment-related APIs

Step 1: You can check the product details via 7.6 before making a Google payment.

Step 2: Add the product to the Google Play Console current app if Google Payments

is required , and maintain the product information on the appropriate server of the application, then get the product id to be paid, and enter the payment via 7.2.

Step 3: If you need to make Huawei payment, add the product to the current application in Huawei's administration backend, and maintain the product in the application. In the product information of the corresponding server, then get the product id of the product you want to pay for, and make the payment via 7.3.

Step 4: In the API documentation, through 7.4, go to the multiple payment methods page, select the corresponding payment method and pay. Need to configure the commodity id of the corresponding platform.

Step 5: Through 7.5, enter the payment order query.

3.6 Share relevant APIs

Step 1: In API Ducument, get a list of all friends of TTK according to 8.2.1, or a list of friends who are currently playing this game according to 8.2.2.

Step 2: Choose the right model based on the type of sharing. For example, if you invite a friend to play a game, choose the InviteObject model; for example, if you want to give a gift and tell the other person, choose the NoticeObject model; for example, if you want to share a picture, choose the ImageObject model. For specific model fields, refer to API document 8.1.

Step 3: Select different sharing platforms based on different models. shareToServer supports NoticeObject and InviteObject and shareToApp supports ImageObject and NoticeObject. Please refer to 8.2.3 and 8.2.4 of the API documentation for details.