**Gaminfinity SDK**

**Programming Guide**

V1.2.5.0

*for*

Android SDK v1.1.1.1

iOS SDK v1.1.3.0

**Table of Content**

[1. Introduction 4](#_Toc401758227)

[2. Description 4](#_Toc401758228)

[3. Gaminfinity SDK API 6](#_Toc401758229)

[3.1 getAccountId 6](#_Toc401758230)

[4. Gaminfinity SDK Sequence Diagram 8](#_Toc401758231)

[4.1Play Now 8](#_Toc401758232)

[4.2 Facebook Login 9](#_Toc401758233)

[4.3 Self-Test 9](#_Toc401758234)

[5. Example for Android 10](#_Toc401758235)

[6. Example for iOS 12](#_Toc401758236)

[7. Download Site 14](#_Toc401758237)

[7.1 Gaminfinity SDK for Android 14](#_Toc401758238)

[7.2 Gaminfinity SDK Demo for Android 14](#_Toc401758239)

[7.3 Gaminfinity SDK for iOS 14](#_Toc401758240)

[7.4 Gaminfinity SDK Demo for iOS 14](#_Toc401758241)

[7.5 Facebook SDK for Android 14](#_Toc401758242)

[7.6 Facebook SDK for iOS 14](#_Toc401758243)

[8. Configurations 15](#_Toc401758244)

[8.1 Configuration for Self-Test 15](#_Toc401758245)

[8.2 Configuration for Testing 15](#_Toc401758246)

[8.3 Configuration for Live 15](#_Toc401758247)

[9. Appendix 16](#_Toc401758248)

[9.1 Error Code Definition 16](#_Toc401758249)

[9.2 Example for Facebook Login 17](#_Toc401758250)

文件修訂紀錄

|  |  |  |
| --- | --- | --- |
| 版號 | 日期 | 說明 |
| V1.2.4 | 2014-09-30 | First release in English Version |
| V1.2.5 | 2014-10-22 | Change Facebook Login Sample to WebView Login  Please refer to Chapter 9.2 Example for Facebook Login |

# Introduction

This document describes how to use Gaminfinity SDK to get unique ID to identify a user.

# Description

The number of API in Gaminfinity SDK: 1

Game developer should do the following items:

1. Using Gaminfinity SDK,
2. Using Facebook SDK to implement Facebook Login

How to implement Facebook Login feature, please refer to

https://developers.facebook.com/docs/facebook-login/v2.0

# Gaminfinity SDK API

Platform: Android and iOS。

## 3.1 getAccountId

**Purpose**

acquire an unique ID to identify a user.

**Input Parameters**

String accessToken //Facebook access token

String url //Gaminfinity SDK Server URL

accessToken: null or valid Facebook access token

url: null or Gaminfinity SDK Server URL

**When to use**

There are two options to login: (1) Play Now, (2) Facebook Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Options | accessToken | url | Description |
| 1 | Play Now | NULL | NULL | Retrieve an AccountID = A123456789, only for self-test mode |
| 2 | Valid URL | User ID |
| 3 | Facebook Login | access token | NULL | Retrieve an AccountID = A123456789, only for self-test mode |
| 4 | access token | Valid URL | User ID |

Callback function

void onGetAccountId(int result, String accountId)

**Return Values**

int result

When result = 1，success.

When result = -n, something goes wrong, please refer to Appendix for error code definition

String accountId //use ID

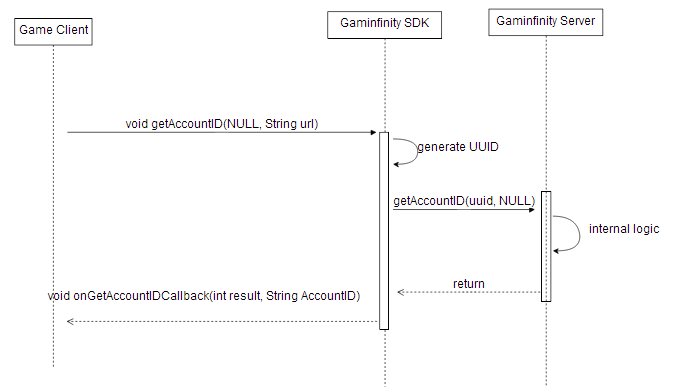
# Gaminfinity SDK Sequence Diagram

There are maximum five roles in the sequence diagram.

1. Game Client：a game app runs on user device
2. Facebook SDK：the SDK provided by Facebook
3. Gaminfinity SDK：the SDK provided by Gaminfinity
4. Gaminfinity Server: the server which provides service to Gaminfinity SDK
5. Facebook： Servers owned by Facebook

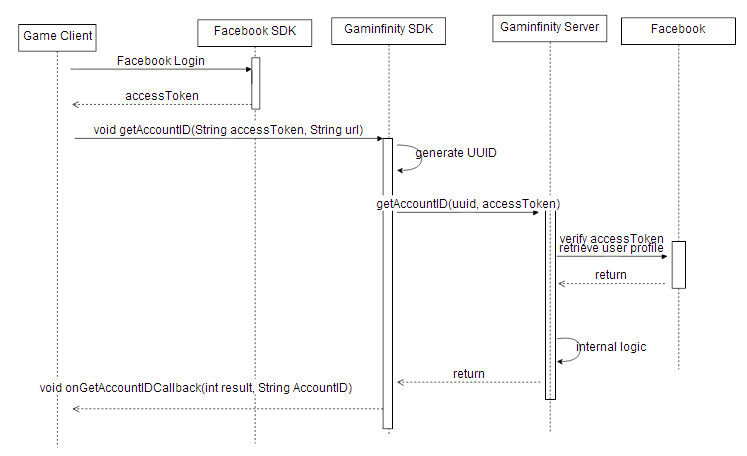
## 4.1Play Now

When a user touches【Play Now】button



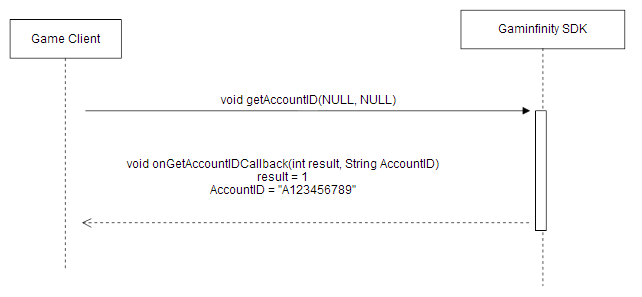
## 4.2 Facebook Login

When a user touches【Facebook Login】button



## 4.3 Self-Test

The purpose of Self-Test is to make sure the Gaminfinity SDK is loaded to your game project and can be called properly.



# Example for Android

* Development tools：Android SDK ADT: adt-bundle-windows-x86-20140624
* SDK Version
  + Facebook SDK for Android: v3.15.0
  + Gaminfinity SDK for Android

Import Gaminfinity SDK to game app project

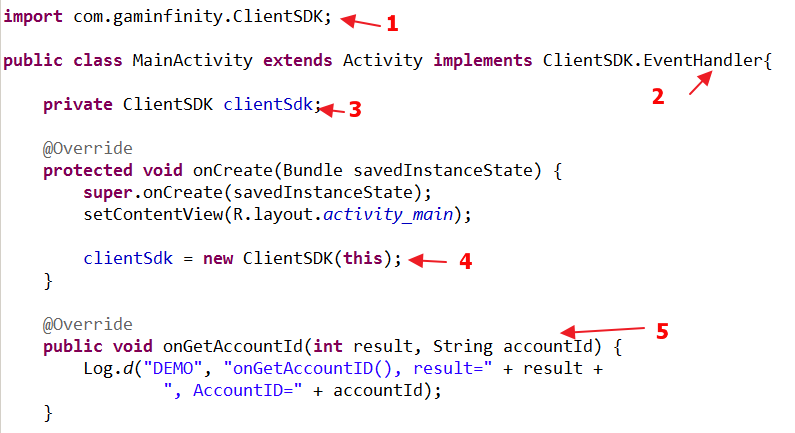
Gaminfinity SDK file name：GaminfinitySDK.jar

Import Facebook SDK to game app project

Please refer to Appendix: Example for Android Facebook Login

Before calling Gaminfinity SDK,

1. import Gaminfinity SDK package
2. extends Gaminfinity SDK interface
3. declare a variable of ClientSDK
4. create a ClientSDK instance
5. implement Gaminfinity SDK interface



Calling Gaminfinity SDK API

(1) Play Now

clientSdk.getAccountId(null, "http://GaminfinityServerURL");

Note：

accessToken should be null

url: please refer to Configurations for valid URL

(2) Facebook Login

clientSdk.getAccountId("Facebook accessToken", "http:// GaminfinityServerURL");

Note：

accessToken should be valid Facebook access token

url: please refer to Configurations for valid URL

# Example for iOS

* Development tools：XCode 5.1.1
* SDK Version
  + Facebook SDK for iOS: v3.17.0
  + Gaminfinity SDK for iOS

Import Gaminfinity SDK to game app project

Gaminfinity SDK file name：libGaminfinitySDK.a and GaminfinitySDK.h

Before calling Gaminfinity SDK,

1. Using Gaminfinity SDK include file

#import "GaminfinitySDK/GaminfinitySDK.h"

1. Extends Gaminfinity SDK Interface

@interface ViewController : UIViewController **<EventHandler>**

1. declare a variable of ClientSDK

GaminfinitySDK\* gaminfinitySdk;

1. create a ClientSDK instance

gaminfinitySdk = [[GaminfinitySDK alloc]init];

gaminfinitySdk.delegate = self;

1. implement Gaminfinity SDK interface

- (void) onGetAccountId:(int)result AccountId:(NSString \*)accountId{

…

}

Calling Gaminfinity SDK API

(1) Play Now

[gaminfinitySdk getAccountId:nil ServerUrl: @"http://GaminfinityServerURL"];

Note：

accessToken should be null

url: please refer to Configurations for valid URL

(2) Facebook Login

[gaminfinitySdk getAccountId:fbAccessToken ServerUrl: @"http:// GaminfinityServerURL"];

Note：

accessToken should be valid Facebook access token

url: please refer to Configurations for valid URL

# Download Site

## 7.1 Gaminfinity SDK for Android

<https://github.com/stevechen0923/GaminfinitySDK/tree/master/Android/>

## 7.2 Gaminfinity SDK Demo for Android

https://github.com/stevechen0923/GaminfinitySdkDemo/tree/master/Android

## 7.3 Gaminfinity SDK for iOS

<https://github.com/stevechen0923/GaminfinitySDK/tree/master/iOS>

## 7.4 Gaminfinity SDK Demo for iOS

https://github.com/stevechen0923/GaminfinitySdkDemo/tree/master/iOS

## 7.5 Facebook SDK for Android

<https://developers.facebook.com/docs/android>

## 7.6 Facebook SDK for iOS

<https://developers.facebook.com/docs/ios>

# Configurations

## Configuration for Self-Test

The purpose of Self-Test is to make sure the Gaminfinity SDK is loaded to your game project and can be called properly.

In this mode, there is no need internet access because no connection to Gaminfinity Server，

Please note that this mode is only for development phase.

Gaminfinity Server URL = NULL

## 8.2 Configuration for Testing

Gaminfinity SDK connects to Testing Gaminfinity Server for developing and testing purpose.

Gaminfinity Server URL = <http://60.199.161.41/sns/new_bind_uuid.php>

## 8.3 Configuration for Live

Gaminfinity SDK connects to Live Gaminfinity Server

Please note app on app store should use this configuration.

Gaminfinity Server URL = <https://ests-sdk.wartown.com.tw/sns/new_bind_uuid.php>

# Appendix

## 9.1 Error Code Definition

1 Success

**Error code from Gaminfinity Server**

-1 parameter error

-2 invalid Facebook Access Token

-3 Facebook binding error

-99 Server internal Error

**Error code from Gaminfinity SDK**

-100 general error(including network error)

-101 server connection timeout(timeout=10 seconds)

-102 network disconnected

-103 network transport error

-104 data encryption/decryption error

-110 Internal Error

## 9.2 Example for Facebook Login

Implement Items：

1. Custom Facebook Login Button
2. Retrieve Facebook access token
3. Custom Facebook Logout Button

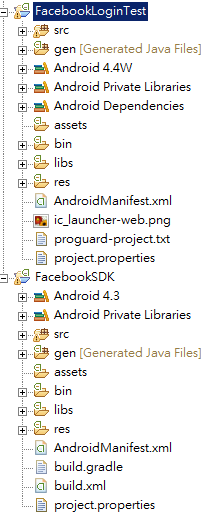
Description：

This example is only available for Android platform，which uses Facebook SDK for Android(v3.15.0)。Please refer for detailed information on Facebook Developer Web Site

Please apply for Facebook App ID，and generates Key hashes。

Project in Eclipse

FacebookLoginTest is the example project name。FacebookSDK folder is Facebook SDK for Android.



This example only focus on those files in FacebookLoginTest folder，and these is no need any modification in FacebookSDK.

Do not forget to add FacebookSDK as an library to FacebookLoginTest project.

1. **in the \res\values\strings.xml，add your Facebook app id**

<string name="app\_id">716592228435883</string>

1. **In the AndroidManifest.xml, add Internet access permission，as well as Activity and meta-data for Facebook use.**

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

<meta-data android:name="com.facebook.sdk.ApplicationId" android:value="@string/app\_id" />

<activity android:name="com.facebook.LoginActivity"/>

1. **In the \res\layout\activity\_main.xml, add two Buttons**

<Button

android:id="@+id/buttonLogin"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:onClick="onLoginButtonClick"

android:text="login" />

<Button

android:id="@+id/buttonLogout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_below="@+id/buttonLogin"

android:onClick="onLogoutButtonClick"

android:text="logout" />

**(4)In the MainActivity.java, add the following codes,**

**package** com.example.facebooklogintest;

**import** android.app.Activity;

**import** android.content.Intent;

**import** android.os.Bundle;

**import** android.view.Menu;

**import** android.view.MenuItem;

**import** android.view.View;

**import** android.widget.Toast;

**import** com.facebook.\*;

**public** **class** MainActivity **extends** Activity {

**private** Session.StatusCallback statusCallback = **new** SessionStatusCallback();

**private** **class** SessionStatusCallback **implements** Session.StatusCallback {

@Override

**public** **void** call(Session session, SessionState state, Exception exception) {

**if** (session.isOpened()) {

Toast.*makeText*(MainActivity.**this**, "AccessToken=" + session.getAccessToken(), Toast.*LENGTH\_SHORT*).show();

}

}

}

@Override

**public** **void** onStart() {

**super**.onStart();

Session.*getActiveSession*().addCallback(statusCallback);

}

@Override

**public** **void** onStop() {

**super**.onStop();

Session.*getActiveSession*().removeCallback(statusCallback);

}

@Override

**public** **void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {

**super**.onActivityResult(requestCode, resultCode, data);

Session.*getActiveSession*().onActivityResult(**this**, requestCode, resultCode, data);

}

@Override

**protected** **void** onSaveInstanceState(Bundle outState) {

**super**.onSaveInstanceState(outState);

Session session = Session.*getActiveSession*();

Session.*saveSession*(session, outState);

}

**public** **void** onLoginButtonClick(View v){

**Session session = new Session(this);**

**Session.setActiveSession(session);**

**Session.OpenRequest openRequest = new Session.OpenRequest(this);**

**openRequest.setPermissions(Arrays.asList("email", "public\_profile", "user\_friends"));**

**openRequest.setLoginBehavior(SessionLoginBehavior.SUPPRESS\_SSO);**

**session.openForRead(openRequest.setCallback(statusCallback));**

}

**public** **void** onLogoutButtonClick(View v){

Session session = Session.*getActiveSession*();

**if** (!session.isClosed()) {

session.closeAndClearTokenInformation();

}

}

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

Settings.*addLoggingBehavior*(LoggingBehavior.*INCLUDE\_ACCESS\_TOKENS*);

Session session = Session.*getActiveSession*();

**if** (session == **null**) {

**if** (savedInstanceState != **null**) {

session = Session.*restoreSession*(**this**, **null**, statusCallback, savedInstanceState);

}

**if** (session == **null**) {

session = **new** Session(**this**);

}

Session.*setActiveSession*(session);

**if** (session.getState().equals(SessionState.*CREATED\_TOKEN\_LOADED*)) {

session.openForRead(**new** Session.OpenRequest(**this**).setCallback(statusCallback));

}

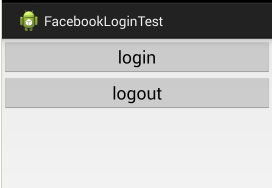
}

}

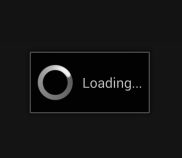
}

**(5)Compile and Run**

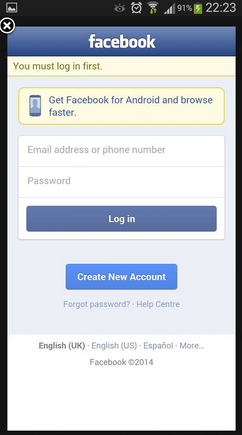
FacebookLoginTest Screenshot is shown below.



While pressing login button, a Loading sprite is shown，which is owned by Facebook SDK



No matter Facebook App is installed or not in the device，Facebook will show Web Login



After login Facebook，a valid access token will be shown on the bottom for 1 -2 seconds.

