

SuperG SDK for Android

Revision history

Version	Date	Changes
3.0.0	2013-11-06	-Rebranded to SuperG -Event logging -Push Notifications Support -Separate Thumbr T animation Framework -AppsFlyer made optional -Performance optimizations
2.0.32	2013-07-25	-Possibility to hide the Thumbr close button (remote configuration) -Improved interstitial ads -Added demo and manual for Ad-only integration
2.0.31	2013-05-16	-Personalized Ad serving support -Technical updates
2.0.3	2013-04-29	-Ad serving support -Added SSL support -Some small bug fixes
2.0.22	2013-03-22	-Added support for alternative layout -Catch url's that contain 'openinbrowser' and open them in browser -Send along SDK version number to Thumbr server -Improved orientation behavior
2.0.21	2013-01-24	-Better Thumbr T-button resizing -Added visual SDK version to Thumbr window -Added Customizable Orientation -Added extra parameters to registration flow: default,registration,optional_registration -Added a counter to the 'opens' of the Thumbr SDK window -Let external URL's (within the SDK window) be opened in the default browser -Bug fix: better orientation handling in general -Let Thumbr server know that back end is loaded from within the SDK (via GET param (&sdk=1) + via HEADER (x-thumbr-method)) -Added version header (X-Thumbr-Version)
2.0.2		-Updated version number to match iOS version (for better release planning) -Added Animated Thumbr T-button support -Added SDK version number to Thumbr screen -Bug fixes
1.1		- Added Appsflyer support
1.0.1		- Bug fixes
1.0		- Initial version

SuperG SDK installation

Prerequisites

- Push messaging is enabled by default
- Event logging is enabled by default. You can set custom events by following the guidelines in the part 'Optional 3'

Step 1:

Unzip the SDK package. If you are using the Demo App, be sure to keep the SDK and the Demo App in the same directory.

Step 2:

Add a reference to the SuperG SDK library project. Follow these steps:

1. Import the SuperG SDK project into Eclipse
2. In the Package Explorer, right-click YOUR project and select Properties.
3. In the Properties window, select the "Android" properties group at left and locate the Library properties at right.
4. Click Add to open the Project Selection dialog.
5. From the list of available library project, select the SuperG SDK project and click OK.
6. When the dialog closes, click Apply in the Properties window.
7. Click OK to close the Properties window As soon as the Properties dialog closes, Eclipse rebuilds your project.

Step 3 (optional):

Add the TestApp project to Eclipse to understand the methods better.

Step 4:

Update your AndroidManifest.xml with at least the settings below.

A lot of permissions are asked, this is because the advertising functionality actually requires most of them!!!

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.yourcompany.your_app"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:targetSdkVersion="17" android:minSdkVersion="8" />

    <permission
        android:name="com.yourcompany.your_app.permission.C2D_MESSAGE"
        android:protectionLevel="signature" />
    <uses-permission android:name="com.gkxim.tqhung.thumbt.dev_demo.permission.C2D_MESSAGE" />
    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
    <uses-permission android:name="android.permission.GET_TASKS" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.READ_CALENDAR" />
    <uses-permission android:name="android.permission.WRITE_CALENDAR" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.DISABLE_KEYGUARD"/>
```

```

<application
    android:icon="@drawable/ic_launcher"
    android:debuggable="true"
    android:label="YourLabel" >

    <receiver android:name="com.appsflyer.MultipleInstallBroadcastReceiver" android:exported="true">
        <intent-filter>
            <action android:name="com.android.vending.INSTALL_REFERRER" />
        </intent-filter>
    </receiver>

    <activity
        android:name="com.yourcompany.your_app.yourActivity"
        android:configChanges="orientation|screenSize|keyboardHidden"
        android:screenOrientation="portrait"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name="com.adgoji.mraid.adview.AdActivity"
        android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|uiMode" />

    <receiver
        android:name="com.cliqdigital.supergsdk.utils.GCMReceiver"
        android:permission="com.google.android.c2dm.permission.SEND" >
        <intent-filter>
            <action android:name="com.google.android.c2dm.intent.RECEIVE" />
            <action android:name="com.google.android.c2dm.intent.REGISTRATION" />
            <category android:name="com.yourcompany.your_app" />
        </intent-filter>
    </receiver>
    <service android:name="com.cliqdigital.supergsdk.utils.GCMIntentService" />
</application>

</manifest>

```

Step 5:

If you are using Advertisements add these ad_views to your layout (s)

```

<RelativeLayout
    android:id="@+id/ad_view"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerInParent="true"
    android:layout_gravity="bottom"
    android:background="@color/transparent" >
</RelativeLayout>

<RelativeLayout
    android:id="@+id/ad_view_interstitial"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"

```

```
        android:background="@color/transparent" >
</RelativeLayout>
```

If you are using Thumbr registration add this button (inside LinearLayout)

```
<LinearLayout
    android:id="@+id/ThumbrButtonWrapper"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView3"
    android:layout_centerHorizontal="true"
    android:minHeight="50dp"
    android:minWidth="50dp"
    android:orientation="horizontal" >

    <ImageButton
        android:id="@+id/bt_re"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/transparent"
        android:tag="ThumbrLogo"
        android:text="Register" />

</LinearLayout>
```

Step 6 ACTIVITY FILE:

Update the *activity file*, where SuperG will be called (usually your main activity) Look at the demo application to see a [complete implementation](#)

```
package com.yourcompany.yourapp;
```

Import at least these libraries

```
import java.util.Locale;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Context;
import android.content.DialogInterface;
import android.content.DialogInterface.OnDismissListener;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.ActivityInfo;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.FrameLayout;
import android.widget.ImageButton;
import android.widget.RelativeLayout;
import android.widget.Toast;
import com.cliqdigital.supergsdk.SuperG;
import com.cliqdigital.supergsdk.SuperG.OnInterstitialCloseListener;
import com.cliqdigital.supergsdk.SuperG.OnPushMessageListener;
import com.cliqdigital.supergsdk.components.AppsFlyerHelper;
import com.cliqdigital.supergsdk.utils.EVA;
import com.cliqdigital.supergsdk.utils.ProfileObject;
```

Open your class and make sure to implement OnClickListener, OnDismissListener

```
public class yourActivity extends Activity implements OnClickListener, OnDismissListener{

    private static final String TAG = "DEBUGTAG";

    /*
     * SETTINGS
     */
}
```

Enter your app specific settings (provided by your SuperG Game Manager)

```
//APP SPECIFIC SETTINGS
private String sid = "com.yourcompany.your_app";
private String client_id = "84758475-476574";

//DEFAULT GAME ORIENTATION (USED IN onDismiss() BELOW, TO SWITCH BACK AFTER SDK CLOSE)
private int gameOrientation = ActivityInfo.SCREEN_ORIENTATION_PORTRAIT;
//TELL THE SUPERG SDK IN WHICH ORIENTATION TO OPEN
private int superGOrientation = ActivityInfo.SCREEN_ORIENTATION_PORTRAIT;
//HIDE THE CLOSE BUTTON (ONLY USE IN-SDK PLAY BUTTON)
private boolean showButtonClose = true;
//THUMBR BUTTON WIDTH/HEIGHT, RELATIVE TO SCREEN WIDTH (MAX. 120PX)
private double buttonWidth = 0.2;
```

Add more generic settings.

THE AD SERVING SETTINGS NEED TO BE REPLACED. THE SETTINGS BELOW ARE DEMO SETTINGS ONLY

```
/*
 * OTHER, MORE GENERIC SETTINGS
 */
private String action = "registration";
private String appsFlyerKey = "9ngR4oQch5qz7qxcFb7ftd";
private Boolean debug = false; //OPTIONALLY SWITCH TO 'true' DURING DEVELOPMENT
private String registerUrl = "http://gasp.superg.mobi/auth/authorize?";
private String switchUrl = "http://gasp.superg.mobi/auth/authorize?";
private String portalUrl = "http://m.thumbr.com?";
private String SDKLayout = "thumbr";

//AD SERVING SETTINGS
private int updateTimeInterval = 15; //number of seconds before Ad refresh
private int autocloseInterstitialTime = 600; //number of seconds before interstitial Ad closes
private int showCloseButtonTime = 6; //Number of seconds before the Ad close button appears
private String tablet_Inline_zoneid = "1356888057";
private String tablet_Inline_secret = "20E1A8C6655F7D3E";
private String tablet_Overlay_zoneid = "3356907052";
private String tablet_Overlay_secret = "ADAA22CB6D2AFDD3";
private String tablet_Interstitial_zoneid = "7356917050";
private String tablet_Interstitial_secret = "CB45B76FE96C8896";
private String phone_Inline_zoneid = "0345893057";
private String phone_Inline_secret = "04F006733229C984";
private String phone_Overlay_zoneid = "7345907052";
private String phone_Overlay_secret = "AEAAA69F395BA8FA";
private String phone_Interstitial_zoneid = "9345913059";
private String phone_Interstitial_secret = "04B882960D362099";

///LEAVE THESE VALUES EMPTY, UNLESS YOU KNOW WHAT YOU'RE DOING
```

```
private String country = ""; //eg: DE or NL
private String locale = ""; //eg: nl_NL or de_DE
private String appsFlyerId="";
/*
 * END SETTINGS
 */
```

Define the 'superG' object and add / edit onPause and onResume functions

```
SuperG superG;

@Override
protected void onPause() {
    superG.pause(this);
    super.onPause();
}
@Override
protected void onResume(){
    superG.resume();
    super.onResume();
}
```

Define the view bt

```
private View bt;
```

Modify your onCreate function

```
@Override
protected void onCreate(Bundle savedInstanceState) {

    Locale l = Locale.getDefault();
    if(country == ""){country = l.getCountry();}
    if(locale == ""){locale = l.getLanguage()+"_"+country;}

    AppsFlyerHelper appsflyerhelper = new AppsFlyerHelper();
    appsflyerhelper.sendTracking(this,appsFlyerKey);
    appsFlyerId = appsflyerhelper.getAppsFlyerId(this);

    //SET AD SERVER SETTINGS
    SharedPreferences settings = this.getSharedPreferences("SuperGSettings", Context.MODE_PRIVATE);
    settings.edit().putInt("updateTimeInterval", updateTimeInterval).commit();
    settings.edit().putInt("autocloseInterstitialTime", autocloseInterstitialTime).commit();
    settings.edit().putInt("showCloseButtonTime", showCloseButtonTime).commit();
    settings.edit().putString("tablet_Inline_zoneid", tablet_Inline_zoneid).commit();
    settings.edit().putString("tablet_Inline_secret", tablet_Inline_secret).commit();
    settings.edit().putString("tablet_Overlay_zoneid", tablet_Overlay_zoneid).commit();
    settings.edit().putString("tablet_Overlay_secret", tablet_Overlay_secret).commit();
    settings.edit().putString("tablet_Interstitial_zoneid", tablet_Interstitial_zoneid).commit();
    settings.edit().putString("tablet_Interstitial_secret", tablet_Interstitial_secret).commit();
    settings.edit().putString("phone_Inline_zoneid", phone_Inline_zoneid).commit();
    settings.edit().putString("phone_Inline_secret", phone_Inline_secret).commit();
    settings.edit().putString("phone_Overlay_zoneid", phone_Overlay_zoneid).commit();
    settings.edit().putString("phone_Overlay_secret", phone_Overlay_secret).commit();
    settings.edit().putString("phone_Interstitial_zoneid", phone_Interstitial_zoneid).commit();
    settings.edit().putString("phone_Interstitial_secret", phone_Interstitial_secret).commit();
    settings.edit().putString("sid", sid).commit();
```

```

settings.edit().putString("client_id", client_id).commit();

// SET VIEW FOR YOUR APPLICATION
super.onCreate(savedInstanceState);
setContentView(com.yourcompany.your_app.homescreen);

//CREATE BUTTON LISTENERS
try{
    bt = (Button) findViewById(R.id.bt_re);
    //Normal Button
}
catch (ClassCastException cce){
    bt = (ImageButton) findViewById(R.id.bt_re);
    //Animated Thumbr Logo
}

//CALL THE SUPERG LIBRARY AND REGISTER SETTINGS
superG=new SuperG(this, 2);
superG.setEnableButtonClose(showButtonClose);
superG.setToastDebug(debug);
superG.setOrientation(superGOrientation);
superG.setLayout(SDKLayout);
superG.setAction(action);

/*
    AT ANY POINT IN THE GAME, YOU CAN CHANGE THE ACTION BEHIND THE THUMBR BUTTON
    ALL POSSIBLE VALUES:

    -default (the default behaviour)
    -registration (forced registration form)
    -optional_registration (registration is optional.
    A counter will determine when the registration will be shown again)

    */
superG.setAction(action);

//(OPTIONAL) AUTOMATICALLY START THE SUPERG SDK WHEN OPENING THE APPLICATION.
//ONLY THE FIRST TIME, IN THIS EXAMPLE
if(superG.getCount() < 2){
    superG.setLinkRegister(registerUrl+"response_type=token&country="+country+"&local
    +locale +"&sid="+sid+"&client_id="+client_id+"&handset_id="+appsFlyerId);

    superG.adInit();//initialize Ads if you are showing Ads

    superG.setInterstitialCloseListener(new OnInterstitialCloseListener(){
        public void onEvent(){
            //Log.i(TAG,"The interstitial advertisement was closed. You can resume your game.
            Toast.makeText(getApplicationContext(), "The game has been notified about closing
            //resumeYourGame();
        }
    });

//THIS IS A LISTENER FOR PUSH MESSAGES. YOU CAN SET CERTAIN TASKS BASED ON THE 'action' F
superG.setPushMessageListener(new OnPushMessageListener(){
    @Override
    public void onPushEvent(Intent intent) {
        // TODO Auto-generated method stub
        String action = intent.getExtras().getString("action");
        String message = intent.getExtras().getString("message");

        if(action.equals("customEvent")){

```

```

        Log.i(TAG,"Action '"+action+"' was received. Message is:"+message);
    }
}
});
//UNCOMMENT FOLLOWING LINE NOT TO OPEN THE WINDOW ON APP STARTUP
//superG.buttonREGISTER();

}

}

```

Catch the button clicks - make sure to use your local paths

```

//CATCH BUTTON CLICKS
@Override
public void onClick(View v) {
    if(v.getId()==R.id.bt_re){
        //CHANGE THIS PATH TO THE THUMBR BUTTON IN YOUR LAYOUT!!
        superG.setLinkRegister(registerUrl+"response_type=token&country="+country+"&local
+locale+"&sid="+sid+"&client_id="+client_id+"&handset_id="+appsFlyerId);
        superG.buttonREGISTER();
    }
}

```

Add onDismiss function to handle the SuperG SDK window close

```

/*
 * THIS FUNCTION IS CALLED WHEN THE SUPERG SDK WINDOW CLOSES.
 */
@Override
public void onDismiss(DialogInterface dialog) {
    Log.i(TAG,"dismiss listener called this function");
    //SET THE ORIENTATION BACK TO GAME PREFERENCE
    setRequestedOrientation(gameOrientation);
    //CALL THE EXAMPLE USER DATA FUNCTION
    getUserData();

    //AFTER FIRST OPEN, LET THE SDK KNOW REGISTRATION IS OPTIONAL FROM NOW ON.
    //REGISTRATION FLOW WILL STILL BE SHOWN EVERY #N TIMES (#N = DETERMINED SERVER SIDE)
    superG.setAction("optional_registration");
}

```

This is an example function that shows which data is sent back in the user object

```

//EXAMPLE FUNCTION OF RETURN VALUES
public void getUserData(){
    ProfileObject obj=superG.didLoginUser();
    if(obj!=null){

        //DO SOMETHING WITH THE USER OBJECT
        if(debug == true){Toast.makeText(this,"User ID: "+obj.getmID(), 4000).show();}
        Log.i("ID",obj.getmID());
        Log.i("UserName",obj.getmUserName());
        Log.i("Status",obj.getmStatus());
        Log.i("Email",obj.getmEmail());
    }
}

```



```

        Log.i("Surname",objb.getmSurname());
        Log.i("Gender",objb.getmGender());
        Log.i("DOB",objb.getmDOB());
        Log.i("Locale",objb.getmLocale());
        Log.i("City",objb.getmCity());
        Log.i("Address",objb.getmAddress());
        Log.i("ZipCode",objb.getmZipCode());
        Log.i("NewsLetter",objb.getmNewsLetter());
        Log.i("FirstName",objb.getmFirstName());
        Log.i("Sisdh",objb.getmSisdh());
        Log.i("Housenr",objb.getmHousenr());
        Log.i("Message",objb.getmMessage());
        Log.i("Code",objb.getmCode());
        Log.i("Description",objb.getmDesscription());
    }else{
        if(debug == true){Toast.makeText(this,"Not logged in yet...", 4000).show();}
    }
}

```

Closing the class

```

}

```

Advertisements

To show Advertisements use these functions at appropriate points in your application

```

//INLINE ADVERTISEMENT:
    final RelativeLayout ad_view = (RelativeLayout) findViewById(R.id.ad_view);
    superG.adInline(ad_view);

//OVERLAY ADVERTISEMENT:
    final RelativeLayout ad_view = (RelativeLayout) findViewById(R.id.ad_view);
    superG.adOverlay(ad_view);

//INTERSTITIAL ADVERTISEMENT:
    final RelativeLayout ad_view = (RelativeLayout) findViewById(R.id.ad_view_interstitial);
    superG.adInterstitial(ad_view);

```

Optional: Animated Thumbr T

To change the button "Registration Thumbr" to te Animated Thumbr T, like in previous versions follow these steps:

Step 1:

Add a reference to the ThumbrT animation library project. Follow these steps:

1. Import the ThumbrT project into Eclipse
2. In the Package Explorer, right-click YOUR project and select Properties.
3. In the Properties window, select the "Android" properties group at left and locate the Library properties at right.
4. Click Add to open the Project Selection dialog.
5. From the list of available library project, select the ThumbrT project and click OK.

6. When the dialog closes, click Apply in the Properties window.
7. Click OK to close the Properties window As soon as the Properties dialog closes, Eclipse rebuilds your project.

Step 2:

Open the layout of your activity and create the Thumbr registration button.

```
<!-- Change tag if you want your own ImageButton,
otherwise it will become the animated thumbr logo -->
<ImageButton
    android:id="@+id/bt_re"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:tag = "ThumbrLogo"
    android:background = "@drawable/ic_launcher"
    android:text="Register Thumbr" />
```

Make sure the button is in a LinearLayout, so create one if it isn't. Even if the button is it's only child

Step 3:

Add this to the Activity:

Load the animated Thumbr T-button and attach it to 'bt_re'

```
// LOAD AND ANIMATE THE THUMBR LOGO :: RESIZE IT TO A PERCENTAGE OF THE
// SCREEN WIDTH
@Override
public void onFocusChanged(boolean hasFocus) {

    // THUMBR BUTTON WIDTH/HEIGHT, RELATIVE TO SCREEN WIDTH (MAX. 120PX)
    double buttonWidth = 0.2;

    try{
        if(bt.getTag().equals("ThumbrLogo")){
            logoanimation.animation anim = new logoanimation.animation();
            anim.animateLogo((ImageButton)bt,this,buttonWidth);
        }
    }
    catch (ClassCastException cce){
        Log.i("ThumbrLogo","Animation not used");
    }
}
```

Optional 2: AppsFlyer

If you want to use appsflyer, Fill out the 'appsFlyerKey' variable and add the appsflyer library to your build path. That's it.

To do so, take **Vendor/AF-Android-SDK-v1.3.9.jar** and add it to the lib-folder of your Android project. Don't forget to add it to the Java Build Path AND check it in 'Order and Export' tab in the Java Build Path

Optional 3: Event logging

Basis events are logged and stored by default. You can use the following extra methods to store other events:

Achievement event

```
EVA eva = new EVA();  
//usage: eva.achievementEarned(context, achievement_name)  
eva.achievementEarned(this, "FoundGold");
```

Click event

```
EVA eva = new EVA();  
//usage: eva.click(context, clicked_item)  
eva.click(this, "SomeButton");
```

Purchase event

```
EVA eva = new EVA();  
//usage: eva.purchase(context, currency, payment_method, price, purchased_item)  
eva.purchase(this, "EUR", "in-app-purchase", "0.99", "ProStatus");
```

Start Level event

```
EVA eva = new EVA();  
//usage: eva.startLevel(context, app_mode, level, score_type, score_value)  
eva.startLevel(this, "basic", "1", "points", "0");
```

Finish Level event

```
EVA eva = new EVA();  
//usage: eva.finishLevel(context, app_mode, level, score_type, score_value)  
eva.finishLevel(this, "basic", "1", "points", "120");
```

Upsell event

```
EVA eva = new EVA();  
//usage: eva.upSell(context, currency, payment_method)  
eva.upSell(this, "EUR", "in-app-purchase");
```

Optional 4: Proguard

Add these lines to your Proguard configuration:

```
-dontwarn com.unity3d.**  
-dontwarn com.appsflyer.**
```

If Proguard still gives you errors, please look at the proguard.cfg in the Demo project.