

#SuperG SDK for iOS

Revision history

2.0.32

-Possibility to hide the Thumbr close button (remote configuration) -Improved interstitial ads -Added demo and manual for Ad-only integration

2.0.31

-Personalized Ad serving support -Technical updates

2.0.3

-Ad serving support -SSL support -Technical updates

2.0.22

-Fixed memory leaks, improved memory clean up -Removed refresh button in iPad layout -Catch url's that contain 'openinbrowser' and open them in browser -Send along SDK version number to Thumbr server -Improved orientation behavior -Upgraded to Xcode 4.6.1 requirements

2.0.21

-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

Easy SDK Setup

• Step 1

Clone this repository to you local disk and drag the contents of **** SDK Files **** to the file browser of your XCode project

• Step 2

Open your AppDelegate.h file and add to your imports:

```
#import "Thumbr/Thumbr.h"
```

Then declare (or add to existing delegate):

```
@interface AppDelegate : UIResponder <UIApplicationDelegate, ThumbrSDKDelegate>
```

And register the following methods and variables:

```
FOUNDATION_EXPORT NSString *const sid;
FOUNDATION_EXPORT NSString *const client_id;
FOUNDATION_EXPORT NSString *const country;
FOUNDATION_EXPORT NSString *const locale;
FOUNDATION_EXPORT NSString *const appsflyerNotifyAppID;
FOUNDATION_EXPORT NSString *const registerUrl;
```

```
+(NSDictionary*) getAdSettings;
- (void) initializeSDK;
- (void) animateAdIn:(id)sender;
- (void) animateAdOut:(id)sender;
```

● Step 3

Open your AppDelegate.m and add to your imports:

```
#import "AppsFlyer.h"
```

Add this line to the *** applicationDidBecomeActive *** method:

```
[AppsFlyer notifyAppID: appsflyerNotifyAppID];
```

Below your @synthesize section, add the Thumbr Settings (get them from your Thumbr game manager):

```
//SETTINGS
```

```
NSString *const sid = @"your_sid";
NSString *const client_id = @"your_client_id";
NSString *const appsflyerNotifyAppID = @"your_appsflyer_id";
NSString *const scoreGameID = @"";
```

```
NSString *const registerUrl = @"https://gasp.thumbr.com/auth/authorize?";
NSString *const switchUrl = @"https://gasp.thumbr.com/auth/authorize?";
NSString *const portalUrl = @"https://mobile.thumbr.com/start?";
NSString *const statusBarHidden = @"TRUE";//TRUE or FALSE
```

```
//LOCAL SETTINGS :: LEAVE EMPTY UNLESS SPECIFICALLY REQUIRED
```

```
NSString *const country = @"";
NSString *const locale = @"";
```

```
//AD SERVING SETTINGS
```

```
#define updateTimeInterval @"0"//default number of seconds before Inline Ad refreshes (can be overridden serverside)
#define autocloseInterstitialTime @"600"//number of seconds before interstitial Ad closes
#define showCloseButtonTime @"6"//Number of seconds before the Ad close button appears
#define iPad_Inline_zoneid @"0337178053"
#define iPad_Inline_secret @"F0B4E489B0CFC0BB"
#define iPad_Overlay_zoneid @"8336743053"
#define iPad_Overlay_secret @"BEF5D9D4D3E9B3CC"
#define iPad_Interstitial_zoneid @"0336739057"
#define iPad_Interstitial_secret @"DA018F2094E8189C"
#define iPhone_Inline_zoneid @"5383077054"
#define iPhone_Inline_secret @"C9AC24EF9CB18FFD"
#define iPhone_Overlay_zoneid @"8383057050"
#define iPhone_Overlay_secret @"A2E465BF955D25A5"
#define iPhone_Interstitial_zoneid @"8383057050"
#define iPhone_Interstitial_secret @"A2E465BF955D25A5"
```

```
//init score variables
```

```
NSMutableDictionary *scoreParams;
NSObject *scoreOutput;
```

And add these lines to your *** didFinishLaunchingWithOptions *** method:

```

[self initilizeSDK];

/*
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 behaviour can be influenced server sided)
*/
[Thumbr setAction:@"optional_registration"];

//OPEN THE SDK UPON APP START
[Thumbr startThumbrPortalRegistration];

```

Last, in this file, add (or update) these methods:

```

- (void)applicationWillTerminate:(UIApplication *)application
{
    [Thumbr stop];
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    [Thumbr stop];
}

- (void) initilizeSDK
{
    //remove keyboard
    for (UIView* view in [_viewController.view subviews]) {
        if ([view isKindOfClass: [UITextField class]] ) {
            if ([view isFirstResponder]) {
                [view resignFirstResponder];
            }
        }
    }

    //Implement the SDK
    NSNumber *ThumbrOrientation;
    if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPhone)
    {
        ThumbrOrientation = [NSNumber numberWithInt: UIDeviceOrientationPortrait];
    }
    else
    {
        ThumbrOrientation = [NSNumber numberWithInt: UIDeviceOrientationLandscapeLeft];
    }

    NSDictionary *settings = [NSDictionary dictionaryWithObjectsAndKeys:scoreGameID,scoreGameID,country,ThumbrSett

    [Thumbr initializeSDKWithSettings:settings andDelegate:self];
}

#pragma mark Thumbr SDK delegate
- (void) thumbrSDK:(Thumbr *)sdk didLoginUser:(ThumbrUser *)user
{
    NSLog(@"Game Thumbr user: %@ ", [user description]);
    NSLog(@"uid: %@", user.uid);
    NSLog(@"username: %@", user.username);
    NSLog(@"status: %@", user.status ? @"Registered" : @"Temporary");
}

```

```

- (void) closedSDKPortalView
{
    NSLog(@"The Game was notified about the closing PortalView");
}
+ (NSDictionary*)getAdSettings{

    NSDictionary *adSettings = [NSDictionary dictionaryWithObjectsAndKeys:showCloseButtonTime,@"showCloseButtonTim
return adSettings;
}

- (void) animateAdIn:(id)sender{
    NSLog(@"animate Ad in");
    _viewController.adView.hidden = NO;
    CGRect adViewFrame = _viewController.adView.frame;
    int frameheight=640;

    if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPad) {
        frameheight = _viewController.view.frame.size.width;
    }
    else{
        frameheight = _viewController.view.frame.size.height;
    }
    adViewFrame.origin.y = frameheight-adViewFrame.size.height;

    CGRect frame = [sender frame];
    if(frame.size.height>10){
        float height = frame.size.height;
        adViewFrame.origin.y = frameheight-height;
    }
    [UIView beginAnimations:nil context:nil];
    [UIView setAnimationDuration:0.5];
    [UIView setAnimationDelay:1.0];
    [UIView setAnimationCurve:UIViewAnimationCurveEaseOut];
    _viewController.adView.frame = adViewFrame;
    [UIView commitAnimations];
}

- (void) animateAdOut:(id)sender{
    NSLog(@"animate Ad out");
    CGRect adViewFrame = _viewController.adView.frame;
    int frameheight=640;
    if ([[UIDevice currentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPad) {
        frameheight = _viewController.view.frame.size.width;
    }
    else{
        frameheight = _viewController.view.frame.size.height;
    }

    adViewFrame.origin.y = frameheight;

    [UIView beginAnimations:nil context:nil];
    [UIView setAnimationDuration:0.5];
    [UIView setAnimationDelay:0];
    [UIView setAnimationCurve:UIViewAnimationCurveEaseOut];
    _viewController.adView.frame = adViewFrame;
    [UIView commitAnimations];
}

```

● Step 4

Open the view controller where the Thumbr button and / or the advertisements will be.

Add this to your imports:

```
#import "Thumbr/Thumbr.h"
#import "Thumbr/AdViewController.h"
```

and add these lines to your ***viewDidLoad*** method:

```
//POSITION OF THE THUMBR BUTTON
//TL = topleft; TR = topright; BL = bottomleft; BR = bottomright
NSString *position=@"TL";
//SIZE OF THE THUMBR BUTTON
//relativeSize=8 will result in a width and height of 1/8 of the portrait screen width (with a max. of 190)
int relativeSize=6;

UIButton *thumbrT = [Thumbr loadThumbrT:relativeSize:position];
[self.view addSubview:thumbrT]; //add the view
[self.view bringSubviewToFront:thumbrT]; //make sure it is on front
```

To add the Thumbr button to a sub view (for custom positions), please consult the Demo application

If you are using inline advertisements (Banners), create the UIViews for this. For iPhone use: 320x50 px For iPad use: 1024x120 pixels

Add a View to your ViewController.xib that will hold the advertisement.

Open the view controller where the Thumbr button and / or the advertisements will be.

Add this to your imports in your viewcontroller.h:

```
#import "Thumbr/Thumbr.h"
#import "Thumbr/AdViewController.h"
#import "AppDelegate.h"
```

```
@property (retain, nonatomic) IBOutlet UIView *adView;
```

Now connect the ‘adView’ IBOutlet to the view that will hold the advertisement. ****** If you are not using ads, simply skip creating the view and the IBOutlet. ******

In your “viewcontroller.m” calling the different advertisements uses these methods:

Overlay advertisement:

```
NSDictionary* adSettings = [AppDelegate getAdSettings];
[[[AdViewController alloc] init] retain] adOverlay:adSettings];
```

Inline advertisement (banner):

```
NSDictionary* adSettings = [AppDelegate getAdSettings];
[[[AdViewController alloc] init] retain] adInline:adSettings adSettings:self.adView];
```

Interstitial advertisement (fullscreen):

```
NSDictionary* adSettings = [AppDelegate getAdSettings];
[[[AdViewController alloc] init] retain] adInterstitial:adSettings];
```

● Step 5 (important!)

Add

```
-all_load -ObjC
```

to the ‘Other linker flags’ in (your project)->Build Settings->Linking

Make sure that the following frameworks are included in your Build Phases->Link Binaries With Libraries:

Frameworks: • CFNetwork • CoreGraphics • Foundation • AdSupport • CoreLocation • CoreTelephony • EventKit • MediaPlayer •
MessageUI • SystemConfiguration • UIKit

Vendor libraries:

- libAppsFlyerLib.a
- LRResty
- MadsSDK
- Thumbr

● REMARKS

If your application has the status bar enabled, the Thumbr close button will not be visible.

To hide the status bar, add

Status bar is initially hidden: YES

to your application Info.plist