

JSSDK Docking Document

Version: 1.0.0

目录

1 、 Introduction to sdk:	2
1.1、 Ad types supported:	2
1.2、 version:	2
2、 sdk access configuration:	2
2.1、 info.plist configuration:	2
2.2、 build setting configuration:	3
2.3、 iOS14 support (optional):	3
2.4、 Importing related sdk:	3
2.5 Importing system support libraries:	4
AppTrackingTransparency.framework	4
2.6 About JCiOSConfig.plist:	4
3、 SDK Api:	5
3.1、 header file	5
3.2、 Initializing sdk	5
3.3、 splash Ads Api	6
3.4、 banner Ads Api	7
3.5、 Intersitial Ads Api	8
3.6、 RewardView Ads Api	8

3.7、native Ads Api.....	9
3.8、Ad callbacks Api.....	10
3.9、About the GDPR setting in the EU.....	11
4、 Related Error Reporting:	11

1 、 Introduction to sdk:

JCSdk is a set of advertising SDK provided by MS Company, which integrates the advertising SDKs of major advertisers and related statistical SDKs to facilitate the joint operation and data analysis of in-app advertising between platforms.

1.1、 Ad types supported:

Splash Ads、Banner Ads、RewardedVideo Ads、Interstitial Ads、native Ads

1.2、 version:

See: [version](#)

2、 sdk access configuration:

2.1、 info.plist configuration:

Support http network configuration:

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

</dict>

Google configuration:

```
<key>GADApplicationIdentifier</key>
<string>ca-app-pub-9488501426181082/7319780494</string>
<key>GADIsAdManagerApp</key>
<true/>
```

Get location permission configuration

```
<key>NSLocationWhenInUseUsageDescription</key>
<string>The app needs to get your location</string>
```

Get IDFA permissions , iOS14support

```
<key>NSUserTrackingUsageDescription</key>
<string>This identifier will be used to deliver personalized
ads to you.</string>
```

2.2、 build setting configuration:

bitcode Set NO,

“Other linker flags” add “-ObjC”,

2.3、 iOS14 support (optional):

Please refer to the [JCSDK iOS14](#) support documentation for details.

2.4、 Importing related sdk:

MS Advertising Support Library and related documents: check out [JCSDK](#)























JCSDK.framework 、 JCiOSConfig.plist

The following are data platform libraries and related documents: Check out [DataCollection SDK](#)

Third-party advertising support library:

Check out [ADThirdParty SDK](#)

2.5 Importing system support libraries

▼ Frameworks, Libraries, and Embedded Content	
Name	Embed
 Accelerate.framework	Do Not Embed ⌵
 AdSupport.framework	Do Not Embed ⌵
 AVFoundation.framework	Do Not Embed ⌵
 CoreGraphics.framework	Do Not Embed ⌵
 CoreLocation.framework	Do Not Embed ⌵
 CoreMedia.framework	Do Not Embed ⌵
 CoreMotion.framework	Do Not Embed ⌵
 CoreTelephony.framework	Do Not Embed ⌵
 iAd.framework	Do Not Embed ⌵
 libbz2.tbd	
 libc++.tbd	
 libresolv.9.tbd	
 libsqlite3.tbd	
 libxml2.tbd	
 libz.tbd	
 MessageUI.framework	Do Not Embed ⌵
 SafariServices.framework	Do Not Embed ⌵
 Security.framework	Do Not Embed ⌵
 SystemConfiguration.framework	Do Not Embed ⌵
 UIKit.framework	Do Not Embed ⌵
 VideoToolbox.framework	Do Not Embed ⌵
 WebKit.framework	Do Not Embed ⌵
+ —	

AppTrackingTransparency.framework

2.6 About JCIOSConfig.plist:

A local visual cache list made to prevent internal parameter fetching failures.

ReYunAppID、ReYunChannelID cannot be a null value.

V1.0.0 add

Item	Value
appid	Appid required for JCSDK initialization
channelid	ChannelId required for JCSDK initialization
ReYunAppID	Appid required for reyun initialization
ReYunChannelID	channelId required for reyun initialization
UmengAppID	Appid required for UMeng initialization
ShuShuAppID	Appid required for 数数 initialization

Item	Value
TalkingDataAppID	Appid required for TalkingData initialization
V2.0.0 add	

Item	Value
KochavaAppID	Appid required for Kochava initialization
TenJinAppID	Appid required for tenjin initialization
ShowSplashFirst	Whether to display an open–screen ad when opening the app for the first time, bool type: YES/NO
LogLevel	Log level: string type. 1. Close. 2. Open JC log. 3. Open JC+ad log. 4. Open JC+ad+data log

3、SDK Api:

If there is a conflict between the API in the documentation and the

framework, the framework's API should prevail.

3.1、header file

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

3.2、Initializing sdk

```
/// The platform log switch is off by default. Please set NO when you go
online
/// @param openPlatformLog YES/NO
+ (void) setOpenPlatformLog: (BOOL) openPlatformLog;

/// Initialize the JCSDK API
/// @param appId appId (Provided by the platform)
/// @param channelId channelId (Provided by the platform)
/// @param isOpenInBody Enable or disable Get Ads From Inside
configuration : YES/NO
/// @param block Internal advertising logic request result block
+ (void) jcSDKInitConfigWithAppId: (NSString*) appId
channelId: (NSString*) channelId
isOpenInBody: (BOOL) isOpenInBody block: (void (^)(BOOL
isOk)) block;
```

Note: when 2.0.0, isOpenInBody can not set "NO"

```
- (BOOL) application: (UIApplication *) application
didFinishLaunchingWithOptions: (NSDictionary *) launchOptions {

    [JC_iOSAdApi setOpenPlatformLog: YES];

    [JC_iOSAdApi
    jcSDKInitConfigWithAppId: @"ff48e91e-043e-46fc-8097-eeed0a
    7f3281" channelId: @"IOS" isOpenInBody: YES block: ^(BOOL
    isOk) {
        // It is recommended that the load event for each type
        of ad be called here, as a pre-load process
    }];
}
```

```
return YES;

}
```

3.3、 splash Ads Api

Please be called after the window is loaded.

```
/// load Splash Ads
+(void)loadSplashView;
```

```
[JC_iOSAdApi loadSplashView];
```

3.4、 banner Ads Api

```
/// load banner Ads
+(void)loadBannerConfig;
```

```
/// banner isReady ?
```

```
+(BOOL)bannerIsReady;
```

```
/// show banner
```

```
/// @param bannerCon Load the banner controller.
+(void)showBannerViewWithCon:(UIViewController*)bannerCon
;
```

Description of the banner ad: It is recommended that you call load as soon as possible, and that the controller displaying the banner is set to the current controller.

If you use the delete banner function, then you need to reload and show the banner again.


```

[JC_iOSAdApi loadBannerConfig];

BOOL isReady = [JC_iOSAdApi bannerIsReady];
if (isReady) {

    [JC_iOSAdApi showBannerViewWithCon:con];

}else{
    NSLog(@"banner isReady is fail!");
}

```

3.5、Interstitial Ads Api

```

/// load Interstitial Ads
+ (void)loadInterstitialConfig;

/// isReady Interstitial Ads?
/// return YES/NO
+ (BOOL)interstitialIsReady;

/// show Interstitial Ads
+ (void)showInterstitialView;

```

Interstitial ad description: Sdk has an internal logic to request Interstitial ad circularly, so it only needs to call load once, it is recommended to call load as soon as possible, and then show the ad by judging isReady.

```

[JC_iOSAdApi loadInterstitialConfig];

BOOL isReady = [JC_iOSAdApi interstitialIsReady]

if (isReady){
    [JC_iOSAdApi showInterstitialView];
}

```

3.6、RewardView Ads Api

```

/// load RewaredVideo Ads
+ (void)loadRewardConfig;

```

```
/// RewaredVideo Ads isReady?
```

```
/// return :YES/NO.
```

```
+(BOOL)rewardVIsReady;
```

```
/// show RewaredVideo Ads
```

```
+(void)showRewardView;
```

RewardVideo ads description: Sdk has the logic to loop request rewardVideo ads, so you only need to call load once, it is recommended to call load as soon as possible, and then judge isReady to show the ads.

```
[JC_iOSAdApi loadRewardConfig];

BOOL isReady = [JC_iOSAdApi rewardVIsReady]

if (isReady){
    [JC_iOSAdApi showRewardView];
}
```

3.7、 native Ads Api

```
/// load native Ads
```

```
/// @param size ads size (Please match the size of the ADFrame in the
displayed ad space config, otherwise the display may be incomplete.)
```

```
+(void)loadNativeConfigWithSize:(CGSize)size;
```

```
/// isReady native Ads
```

```
+(BOOL)nativeIsReady;
```

```
/// show native Ads
```

```
/// @param config native Ads config
```

```
+(UIView*)showNativeConfigWithConfig:(JCNativeConfig*)config;
```

Description of native ads: Native ads require users to design the content and layout of the ad view, please upload the ad size for load ads, and please upload the configuration for show ads.

For more information, see: JCNativeConfig.h

```

[JC_iOSAdApi
loadNativeConfigWithSize:CGSizeMake(CGRectGetWidth(self.view.bounds), 350)];

BOOL isReady = [JC_iOSAdApi nativeIsReady]

JCNativeConfig *config = [[JCNativeConfig alloc] init];
    config.ADFrame = CGRectMake(.0f, 200.0f,
CGRectGetWidth(self.view.bounds), 350.0f);

        config.mediaViewFrame = CGRectMake(0, 120.0f,
CGRectGetWidth(self.view.bounds), 350.0f - 120.0f);

            config.renderingViewClass = [[[CustomView
alloc] init] class];

                config.rootViewController = self;

UIView *adview = [JC_iOSAdApi
showNativeConfigWithConfig:config];

If (isReady){

[self.view addSubview:self.nativeADView];

}

```

3.8、 Ad callbacks Api

The sdk uses notifications to tell the user that each ad is acting accordingly.

For more information, please refer to: JCAadCallBackHeader.h

Here's an example of a callback api using splash ads, Please follow the format for other ad callbacks

```

[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(msAdLoadCallBack:) name:MSSplashADKey
object:nil];

```

```

-(void)msAdLoadCallBack:(NSNotification*)noti{
    NSLog(@"%@",noti.userInfo);
    NSInteger code = [noti.userInfo[@"status"]
integerValue];
    switch (code) {
        case MSAd_splashDidShow:
        {
            NSLog(@"MSAd_splashDidShow");
        }
        break;

        default:
            break;
    }
}

```

3.9、 About the GDPR setting in the EU

```

/// Determine if it is EU territory API
/// @param block callback isEU? YES / NO
+ (void)getLocationIsEU: (void (^)(BOOL isEU))block;

/// the GDPR interface API
/// @param dismissblock close Interface callback
/// @param failBlock show Fail callback
+ (void)jcSDKShowGDPRWithDismissblock: (void (^)(void))dismi
ssblock loadFailblock: (void (^)(NSError *error))failBlock;

```

```

[JC_iOSAdApi getLocationIsEU:^(BOOL isEU) {
    if (isEU) {
        [JC_iOSAdApi jcSDKShowGDPRWithDismissblock:^(
            } loadFailblock:^(NSError * _Nonnull error) {

        }]];
    }else{
    }
}
}];

```

4、 Related Error Reporting:

If you are connected to KSAAdSDK and you get an error when uploading the AppStore package, Apple does not support emulator binaries, then you need to add a script to the xcode project to remove the emulator binaries from the sdk.as below:

```
APP_PATH="${TARGET_BUILD_DIR}/${WRAPPER_NAME}"

# This script loops through the frameworks embedded in the
application and
# removes unused architectures.
find "$APP_PATH" -name '*.framework' -type d | while read -r
FRAMEWORK
do
    FRAMEWORK_EXECUTABLE_NAME=$(defaults read
"$FRAMEWORK/Info.plist" CFBundleExecutable)
    FRAMEWORK_EXECUTABLE_PATH="$FRAMEWORK/$FRAMEWORK_EXEC
UTABLE_NAME"
    echo "Executable is $FRAMEWORK_EXECUTABLE_PATH"

    EXTRACTED_ARCHS=()

    for ARCH in $ARCHS
    do
        echo "Extracting $ARCH from
$FRAMEWORK_EXECUTABLE_NAME"
        lipo -extract "$ARCH" "$FRAMEWORK_EXECUTABLE_PATH"
-o "$FRAMEWORK_EXECUTABLE_PATH-$ARCH"
        EXTRACTED_ARCHS+=("$FRAMEWORK_EXECUTABLE_PATH-$ARC
H")
    done

    echo "Merging extracted architectures: ${ARCHS}"
    lipo -o "$FRAMEWORK_EXECUTABLE_PATH-merged" -create
"${EXTRACTED_ARCHS[@]}"
    rm "${EXTRACTED_ARCHS[@]}"

    echo "Replacing original executable with thinned version"
```

```
rm "$FRAMEWORK_EXECUTABLE_PATH"  
mv "$FRAMEWORK_EXECUTABLE_PATH-merged"  
"$FRAMEWORK_EXECUTABLE_PATH"
```

done

After adding the script block, check the Run script only when installing and repackaging it for submission.

