

Mobile SDK for iPhones

INDEX

TOC \t "heading 1, 1,heading 2, 2,heading 3, 3,Überschrift 1 A, 4,Überschrift 23, 5"

Introduction and basics PAGEREF \_Toc \h 3

SDK structure and modules PAGEREF \_Toc1 \h 4

Supported iOS versions PAGEREF \_Toc2 \h 5

Requirements PAGEREF \_Toc3 \h 6

First steps - project setup And IDE configuration PAGEREF \_Toc4 \h 6

General settings PAGEREF \_Toc5 \h 6

Xcode Setup PAGEREF \_Toc6 \h 6

Linked libraries and frameworks PAGEREF \_Toc7 \h 6

Additional settings for testing, compiling and debugging PAGEREF \_Toc8 \h 8

Add linker flags PAGEREF \_Toc9 \h 8

Compiler LLVM PAGEREF \_Toc10 \h 8

Device and simulator targets PAGEREF \_Toc11 \h 8

Integration and implementation of the SDK PAGEREF \_Toc12 \h 10

Ad integration guideline PAGEREF \_Toc13 \h 10

Add GuJ SDK to project PAGEREF \_Toc14 \h 10

GuJAdView Methoden (Version 2.0) PAGEREF \_Toc15 \h 11

GUJAdVIEW WITH GCD-BLOCKS (VERSION 2.0) PAGEREF \_Toc16 \h 11

Instantiate GuJadView withmOcean backfill PAGEREF \_Toc17 \h 12

InstantiateGuJadView without keywords PAGEREF \_Toc18 \h 13

Instantiate GuJadView with keywords PAGEREF \_Toc19 \h 14

Positioning of GuJadViews PAGEREF \_Toc20 \h 14

Positioning of GuJAdViews with CGPoint information PAGEREF \_Toc21 \h 14

Positioning GuJAdView with view replacement PAGEREF \_Toc22 \h 15

Deactivation of GEO-Location Sevices for GuJAdView PAGEREF \_Toc23 \h 17

Compiling and starting PAGEREF \_Toc24 \h 17

Notification usage and handling with GUJAdViewControllerDelegate PAGEREF \_Toc25 \h 17

Interstitialintegration PAGEREF \_Toc26 \h 20

interstitial delegate implementation PAGEREF \_Toc27 \h 20

Interstitial implementation PAGEREF \_Toc28 \h 21

Startstitial Handling PAGEREF \_Toc29 \h 22

Add own requestheader and requestparameter PAGEREF \_Toc30 \h 23

ERROR CODES PAGEREF \_Toc31 \h 24

Known ORMMA Issues PAGEREF \_Toc32 \h 25

# Introduction and basics

The following documentation contains a detailed description for integration and implementation of the G+J EMS iOS Software Development Kits (SDK). The SDK supports the ORMMA Standard. Following advertising formats and features can be realized with the SDK:

* Standard - advertisement
* Interstitials
* Rich Media
* Keyword Targeting
* Geo Targeting
* Custom Targeting
* IP Targeting
* Unique User / Frequency Capping
* In App Browser

## SDK structure and modules

The SDK is structured in several modules:

GuJBaseSDK: contains base functionality for loading and displaying ads. Native device features can also be used for advertisement e.g. location based services, messaging, video and audio, maps etc.

ORMMA Module: extends the GUJ BaseSDK and realize the ORMMA conform handling of ads.Get further information about ORMMA here <http://www.ormma.org>

GuJmOcean Module: extends the GUJ ORMMA SDK and realizes the delivery of ads of the performance network optimobile. Get further information about moceanon <http://code.google.com/p/mocean-sdk-ios/>

GuJ XAXSIS Module: extends the GUJ mOcean SDK and realizes backfill for Interstitial placements with XAXIS inventory.

GuJAdView Context: is a kind of wrapper and contains all mentioned SDKs (Base SDK, ORMMA SDK, mOcean SDK und XAXSIS SDK). Therefore all modules can be easy integrated by opening this module in the project tree.

The special software architecture and -design of the SDK allows the developer the enabling of ads with just a few lines of code.

## Supported iOS versions

The SDK supports in general iOS version 3.0+. It was tested on the following devices and iOS versions.

Hardware

* iPhone 3GS
* iPhone 4
* iPhone 4S
* iPad 1
* iPad 2
* iPad 3
* iPad Air

iOS Version

* iOS 5.x
* iOS 6.x
* iOS 7.x

The SDK-Starter package contains the following components:

* libGUJAdViewContext.aforiOS
* libGUJAdViewContextSimulator.aforthesimulator
* ORMMAResourceBundle.bundle
* VideoAdLib.bundle for the XAXSIS Video SDK
* Required Header-files:
* GUJAdViewContext.h
* Optional Header-files
* GUJAdViewControllerDelegate.h

The libraries libGUJAdViewContext.a and libGUJAdViewContextSimulator are „fat“ Libraries. Content of different architectures (armv7, armv7s, i386) are collected to one big library.  
libGUJAdViewContext (see GuJAdViewContext) contains following libs:

* libGUJBaseSDK
* libGUJORMMASDK
* libGUJmOceanSDK
* libGUJXAXSISSDK
* libAdMobileSDK (externalmOcean SDK)
* libAdSDKLib (external XAXSIS SDK)

If you do not prefer the usage of this fat libraries see the mentioned information in attachment A.

## Requirements

The SDK was compiled in Xcode 4.6.2 with ARC (Automatic Reference Counting).

In case of working with another Xcode version you have to make sure the availability of the Apple LLVM 4.2 compiler.

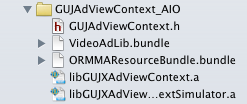
In case of compiling without ARC the -fobj-arc flag has to be set in the compiler settings.

# First steps - project setup And IDE configuration

## General settings

### Xcode Setup

Add the Starter package files to your project folder.

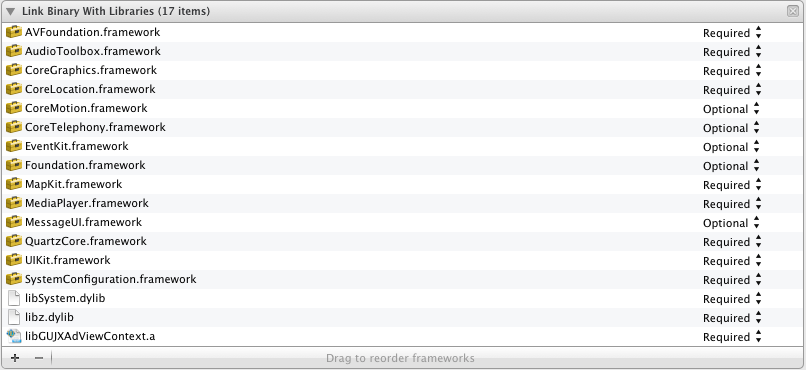


### Linked libraries and frameworks

This step is one of the important once for the integration!  
You have to make sure, that all needed libraries and frameworks are added to your project.

Add the libs and frameworks in the “Build Phase” tab of your project.

* Open projectnavigator
* Open project
* Select „Build Phase“.
* Expand „Link Binary With Libraries“



Add following iOSframerworks to your list.  
  
These frameworks are required!

| Framework | Usedby |
| --- | --- |
| AVFoundation.framework | ORMMA SDK, mOcean SDK, XAXSIS SDK |
| AudioToolbox.framework | XAXSIS SDK |
| CoreGraphics.framework | mOcean SKD |
| CoreLocation.framework | ORMMA SDK, mOcean SDK, XAXSIS SDK |
| CoreMotion.framework | mOcean SKD |
| CoreTelephony.framework | mOcean SKD, XAXSIS SDK |
| EventKit.framework | ORMMA SDK, mOcean SDK |
| MapKit.framework | ORMMA SDK, mOcean SDK |
| MediaPlayer.framework | ORMMA SDK, mOcean SDK, XAXSIS SDK |
| MessageUI.framework | ORMMA SDK, mOcean SDK |
| QuarzCore.framework | mOcean SKD |
| SystemConfiguration.framework | mOcean SDK, XAXSIS SDK |
| libSystem.dylib | mOcean SDK, XAXSIS SDK |
| libz.dylib | XAXSIS SDK |

## Additional settings for testing, compiling and debugging

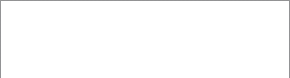
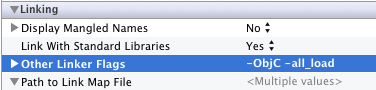
### Add linker flags

Add the following compiler flags to „Other Linker Flags“ in your project.

* -ObjC
* -all\_load

Do the following steps to set them

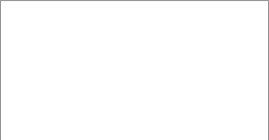
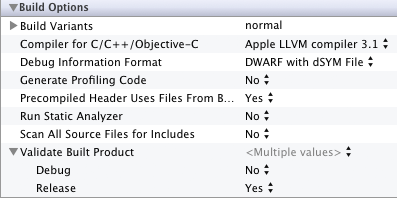
* Open projectnavigator
* Open project
* Select „Build Settings“
* Scroll to „Linking Options“



### Compiler LLVM

Make sure that you are compiling with Apple >= LLVM 4.2 compiler.

* Open projectnavigator
* Open project
* Select „Build Settings
* Scrollen to „Build Options“



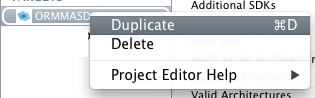
### Device and simulator targets

If you are planning to use the simulator for testing it is needed to add some more libraries to your project. It will be also possible to choose two different targets.

* Open projectnavigator
* Open project

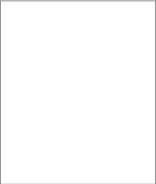
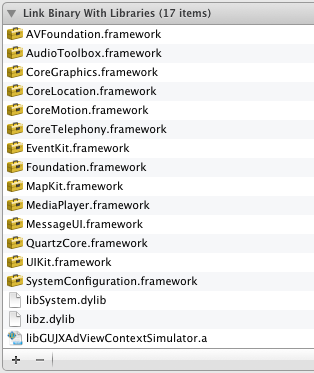
To setup a simple device and test environment do the following steps:

* Create a device target
* Copy this device target to simulator targets



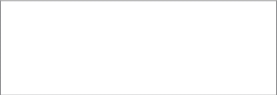
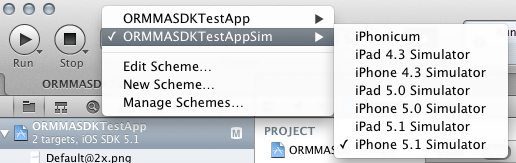
Delete the device libraries from the list ofthe simulator target and add the libraries with simulator suffix.

* Open projectnavigator
* Open project
* Select „Build Phase“.
* Expand „Link Binary With Libraries“



Start the different targets via selection ofthe needed scheme in the dropdown in xCode

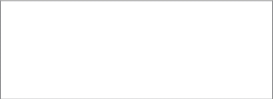
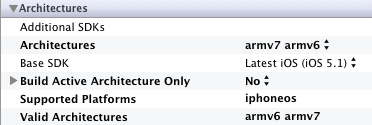
.



Furthermore you can edit there the build settings.

Select Device-Target as followed

* Open projectnavigator
* Open project
* Select „Build Settings
* Select device target
* Scrollto „Architectures“



Select simulator Target as followed

* Open projectnavigator
* Open project
* Select „Build Settings
* Select device target
* Scrollen to „Architectures“



Choose a simulator target instead of a device target.

# Integration and implementation of the SDK

## Ad integration guideline

## Add GuJ SDK to project

Import the GUJAdViewContext.h in your UIViewControlleror UIViewheaderfile.

GuJAdView Methoden (Version 2.0)

With Version 2.0 the developer has the option to control the GUJAdView with the following new methods:

|  |
| --- |
| -(void)show; |

Will show the GUJAdView immediately.

|  |
| --- |
| -(void)showInterstitialView; |

Will show the GUJAdView Interstitial immediately.

|  |
| --- |
| -(void)hide; |

Will hide the GUJAdView (auch Interstitial) immediately.

These methods only takes effect if the developer has added the GUJAdView (not interstitial) to a Super View.

Additionally a reference to the current AdSpaceID is added to the GUJAdView. This will help to identify different AdViews.

|  |
| --- |
| -(NSString\*)adSpaceId; |

## GUJAdVIEW WITH GCD-BLOCKS (VERSION 2.0)

Version 2.0 will now supports Grand-Central-Dispatch (GCD) Blocks with all known GUJAdView constructors.

The completion block that is used by the SDK, is definite as followed in the GUJAdViewContext.h Header-File:

|  |
| --- |
| typedef BOOL (^adViewCompletion)(GUJAdView\* \_adView, NSError\* \_error); |

As developer you have the possibility to examine the returned \_error – if present – or modify and or control the given \_adView before displaying it.  
The Block needs to return a Boolean value that indicates the visibility of the AdView.

If returning YES(true/1) the AdView will be shown imidiatly.  
If returning NO(false/0) the AdView will stay in memory and will not be displayed automaticly.  
You have to call [myAdView show] to display the Ad when returning NO. This can happen anywhere and at anytime in your source “AFTER” assigning the given \_adView variable to a property of you implementing class.

Example:

|  |
| --- |
| [[GUJAdViewContext instanceForAdspaceId:@"12345"] adView:^BOOL(GUJAdView \*\_adView, NSError \*\_error) {  if( \_error != nil ) {  NSLog(@"AdViewError: %@",\_error);  return NO;  } else {  [self.view addSubview:\_adView];  return YES;  }}]; |

Since Version 2.0 GCD-Blocks will available for the following AdView-Methods:

* (void)adView:(adViewCompletion)completion;
* (void)adViewWithOrigin:(CGPoint)origin completion:(adViewCompletion)completion;
* (void)adViewForKeywords:(NSArray\*)keywords completion:(adViewCompletion)completion;
* (void)adViewForKeywords:(NSArray\*)keywords origin:(CGPoint)origin completion:(adViewCompletion)completion;
* (void)interstitialAdViewWithCompletionHandler:(adViewCompletion)completion;

(!)Note:

The Delegate-Method “adViewController:canDisplayAdView“ will NOT executed if using GCD-Block-Calls mentioned before.

## Instantiate GuJadView withmOcean backfill

The following example describes how to instantiate an adView with a connection for mocean backfill. This means, that the adView will be filled by performance campaigns whether there is no premium campaign delivered by G+J EMS. To activate the backfill for an adView please add a valid site and zone ID by initialization of the GuJAdViewContext.

|  |
| --- |
| [[GUJAdViewContextinstanceForAdspaceId:@"AdSpaceId" site:1234 zone:1234]  Or with delegate for notification:  [[GUJAdViewContextinstanceForAdspaceId:@"AdSpaceId" site:1234 zone:1234 delegate:self] |

The mocean backfill will be automatically activated when delivering a site and zone ID. If you manually like to deactivate the backfill implement the following line of code.

|  |
| --- |
| Aktivate:  [myAdViewContext setMOceanBackFill:YES]  Deaktivate:  [myAdViewContext setMOceanBackFill:NO] |

When you set the backfill active but delivering no valid site and zone ID no exception will be thrown.

## InstantiateGuJadView without keywords

The following example describes how to instantiate an adView without any keywords in your app. Instantiate the GUJAdViewContext and add the AdContainer to the viewDidLoad method of your ViewControllers.

|  |
| --- |
| [self.viewaddSubview:  [[GUJAdViewContex tinstanceForAdspaceId:@"AdSpaceId"] adView]  ];  > OR GCD BLOCK:  [self.viewaddSubview:[[GUJAdViewContext instanceForAdspaceId:@"AdSpaceId"] adView:^BOOL(GUJAdView \*\_adView, NSError \*\_error) {  [self.view addSubView:\_adView];  return YES;  }]; ]; |

## Instantiate GuJadView with keywords

The following example describes how to instantiate an adView including keywords in your app. These keywords are delivered via ad-request and can be used as targeting option by G+J EMS.

|  |
| --- |
| [[GUJAdViewContextinstanceForAdspaceId:@"AdSpaceId"] adViewForKeywords:[NSArray arrayWithObjects:@"key1",@"key2", nil]];  > OR GCD BLOCK:  [[GUJAdViewContext instanceForAdspaceId:@"AdSpaceId"] adViewForKeywords:[NSArray arrayWithObjects:@"key1",@"key2", nil] completion:^BOOL(GUJAdView \*\_adView, NSError \*\_error) {  return YES;  }]; |

## Positioning of GuJadViews

There are two different ways for positioning the adView in general. Both methods are described as followed.

### Positioning of GuJAdViews with CGPoint information

You can deliver a CGPoint via GUJAdViewContext to get access to the position of the ads.

Value x of CGPoints will be ignored because of automatic centration of the container by the SDK. Value y will be delivered to the AdView.

|  |
| --- |
| [[GUJAdViewContextinstanceForAdspaceId:@"AdSpaceId"] adViewWithOrigin:CGPointMake(0.0f, 20.0f)];  > OR GCD BLOCK:  [[GUJAdViewContext instanceForAdspaceId:@"AdSpaceId"] adViewWithOrigin:CGPointMake(0.0f, 20.0f) completion:^BOOL(GUJAdView \*\_adView, NSError \*\_error) {  return YES;  }]; |

The following code shows the implementation of an adView with keywords in a defined position.

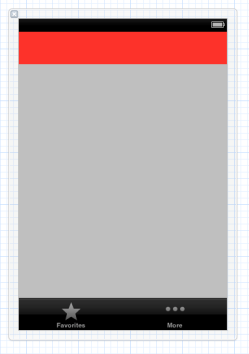
|  |
| --- |
| [[GUJAdViewContextinstanceForAdspaceId:@"AdSpaceId"] adViewForKeywords:[NSArray arrayWithObjects:@"key1",@"key2", nil] origin:CGPointMake(0.0f, 20.0f)];  > OR GCD BLOCK:[[GUJAdViewContext instanceForAdspaceId:@"AdSpaceId"] adViewForKeywords:[NSArray arrayWithObjects:@"key1",@"key2", nil] origin:CGPointMake(0.0f, 20.0f) completion:^BOOL(GUJAdView \*\_adView, NSError \*\_error) {  [self.view addSubview:\_adView];  return YES;  }]; |

### Positioning GuJAdView with view replacement

To integrate a template banner in your interface you can also work with a View-Template in the XCodeInterfaceBuilder

There a UIView will be used as template for GUJAdView. This allows you to have a comfortable overview of the look and feel of your app including the GUJAdView for ad delivery.

Drag and drop a UIView to your interface and fill it with a color for highlighting if you like.

  
The red space defines the future position for ad delivery

The following code describes how this UIView will be overwritten by the GUJAdView without loosing the position in your interface. Die variable „containerView“ is a reference on the UIView template (red space) in the InterfaceBuilder.

|  |
| --- |
| - (void)\_loadAdview {  adViewCTX\_ = [GUJAdViewContextinstanceForAdspaceId:@"12345" delegate:self];  CGRect frame = containerView.frame;  [containerViewremoveFromSuperview];  containerView = [adViewCTX\_ adViewWithOrigin:CGPointMake(0.0, frame.origin.y)];  [self.view addSubView:containerView];  > OR GCD BLOCK:  [[GUJAdViewContext instanceForAdspaceId:@"AdSpaceId"] adViewWithOrigin:CGPointMake(0.0f, frame.origin.y) completion:^BOOL(GUJAdView \*\_adView, NSError \*\_error) {  [self.viewaddSubview:\_adView]; return YES;  }];} |

## Deactivation of GEO-Location Sevices for GuJAdView

If you do not like to deliver Geo information of the current location of the user in each ad-request: use the following lines of code to disable the GEO location services.

(!) Note: In Version 2.0 the static methods enable- and disableLocationService are now non-static.

|  |
| --- |
| Aktiviate:  [myAdViewContext enableLocationService]  Deaktivate:  [myAdViewContext disableLocationService] |

## Compiling and starting

After compiling and launching the app you will see advertisement at the position of the GUJAdView in the defined position.

Note! If you decide to implement the GUJAdView without using the delegate methods of the GUJAdViewControllerDelegate it won’t be possible for you to analyze the current status of the ad. You won’t be able to define whether an ad is loaded or not or if any kinds of exceptions were thrown while loading.

Please read detailed information about the GUJAdViewController delegate to enable status handling for the ads.

## Notification usage and handling with GUJAdViewControllerDelegate

The GUJAdViewControllerDelegate is able to communicate with the GUJAdView and deliver several controller messages to your app. Furthermore it allows you to control the ad delivery. To enable the usage of these delegate methods add the optional headerfile „GUJAdViewControllerDelegate.h“ to your project.

You have to implement the GUJAdViewControllerDelegateprotocol as followed.

|  |
| --- |
| [GUJAdViewContextinstanceForAdspaceId:@"AdSpaceId" delegate:self] |

In the implementing Class or ViewController add:

|  |
| --- |
| @interface ViewController : UIViewController<GUJAdViewControllerDelegate> |

Furthermore you have to add following delegate methods to your UIViewController class.

|  |
| --- |
| - (void)adViewController:(GUJAdViewController\*)adViewController didConfigurationFailure:(NSError\*)error;  - (void)bannerView:(GUJAdView\*)bannerView didFailLoadingAdWithError:(NSError\*)error;  - (void)interstitialView:(GUJAdView\*)interstitialView didFailLoadingAdWithError:(NSError\*)error; |

Additionally you can implement the following delegate methods to collect additional information of the GUJAdView.

|  |
| --- |
| - (BOOL)adViewController:(GUJAdViewController\*)adViewController canDisplayAdView:(GUJAdView\*)adView;  - (void)bannerViewInitialized:(GUJAdView\*)bannerView;  - (void)bannerViewDidShow:(GUJAdView\*)bannerView;  - (void)bannerViewDidHide:(GUJAdView\*)bannerView;  - (void)bannerViewWillLoadAdData:(GUJAdView\*)bannerView;  - (void)bannerViewDidLoadAdData:(GUJAdView\*)bannerView;  - (void)bannerView:(GUJAdView\*)bannerView receivedEvent:(GUJAdViewEvent\*)event; |

If the configuration of the adViewController is incorrect: the delegate method adViewController:didConfigurationFailure: will be called.

You will get detailed information about several errors by analyzing the NSError. NSError is a param of this method. In many cases the error caused by a missing AdSpaceId.

If an error occurred while loading an ad: the delegate method bannerView:didFialLoadingAdWithError: will be called.

Optional you are able to identify errors by

* Reload an ad
* Delete the AdViewController and reimplement it
* Do nothing

When the GUJAdView is loaded, and no GCD-Blocks where used you can control the visibility of the upcoming adView via the delegate method: (BOOL)adViewController:canDisplayAdView   
Here, you have to return a boolean value for assumed visibility state.  
When returning NO(false/0) the AdView will stay in memory but not displayed on the implementing View.  
Returning YES(true/1) the AdView will be shown immediately.

If the GUJAdView is loaded and added to your app as SubView: the delegate method bannerViewInitialized: will be called.

This allows you to collect detailed information about the GUJAdView. This information can be read by parameter bannerView of this method.

Since the Adview started to load the ads: the delegate method bannerViewWillLoadAdData: will be called.

The app is in process to connect to the adserver of G+J EMS.

If the GUJAdView successfully finished loading the ad: the delegate method bannerViewDidLoadAdData will be called.

If bannerViewDidLoadAdData failed you can detect errors during the loading process. In case of an error the Method won’t be called.

If the ad will be displayed in the ViewController: the delegate method bannerViewDidShow: will be called.

Here you are able to do an action caused by the visibility of an ad.

If the ad won’t be displayed or was displayed and is now set to visible=false: the delegate method bannerViewDidHide: will be called.

In this case you can cause an action because of the invisibility of the ad. This is useful to fill the free space with new content or new UIElements.

If an ad is receiving a system message: the message will be forwarded to delegate method bannerView:receivedEvent:.

System messages can be message from 3rd party components like mocean or XAXIS.

## Interstitialintegration

### interstitial delegate implementation

An interstitial will be displayed during the process of switching from one View to another. To be able to control the entry and exit points for the interstitial display you have to implement the following delegate methods. Otherwise you won’t be able to go to your target View after showing an interstitial.

Add the following *GUJAdViewControllerDelegate* method to your UIViewController:

|  |
| --- |
| - (void)interstitialViewInitialized:(GUJAdView\*)interstitialView; - (void)interstitialViewWillLoadAdData:(GUJAdView\*)interstitialView; - (void)interstitialViewDidLoadAdData:(GUJAdView\*)interstitialView;- (void)interstitialViewWillAppear;- (void)interstitialViewDidAppear;- (void)interstitialViewWillDisappear;- (void)interstitialViewDidDisappear;  - (void)interstitialViewReceivedEvent:(GUJAdViewEvent\*)event; |

interstitialViewDidFailLoadingWithError is deprecated in Version 2.0 and will be deleted in further Versions. Please use interstitialView:didFailLoadingAdWithError: instead.

If an error occured while loading: the delegate method interstitialViewDidFailLoadingWithError: will be called.

You will get detailed information about several errors by analyzing the NSError. NSError is a parameter of this method. In case of an error the interstitialView won’t be displayed.

If the InterstitialView is loaded: the delegate method interstitialViewWillAppear: will called immediately. Start here to disable the user interaction for the current view while displaying the interstitial.

Since the Adview started to load the ad, the delegate method interstitialViewWillLoadAdData: is called.

The app is in process to connect to the ad-server of G+J EMS.

If the GUJAdView successfully finished loading the ad, the delegate method interstitialViewDidLoadAdData is called.

If bannerViewDidLoadAdData failed you can detect errors during the loading process. In case of an error the Method won’t be called.

If the Interstitialview successfully finished loading the interstitial: the delegate method interstitialViewDidAppear is called.

Once, the Interstitialview will close itself (after 7seconds in general): the delegate method interstitialViewWillDisappear is called.

At this time the app should be prepared for loading all contents oft he target view and be able to display this target view.

If the Interstitialview is closing: the delegate method interstitialViewDidDisappear is called. This call gives you the signal to display the target view.

If an ad is receiving a system message: the message will be forwarded to delegate method interstitialView:receivedEvent:

System messages can be message from 3rd party components like mocean or XAXIS.

### Interstitial implementation

The easiest way to implement the Interstitialview, is using the GUJAdViewContext as followed:

|  |
| --- |
| [[GUJAdViewContextinstanceForAdspaceId:@"AdSpaceId"] interstitialAdView]  > OR GCD BLOCK:  [[GUJAdViewContext instanceForAdspaceId:@"AdSpaceId"] interstitialAdViewWithCompletionHandler:^BOOL(GUJAdView \*\_adView, NSError \*\_error) {  return YES;  }]; |

Optional you can deliver keywords with the Interstitialview. This allows G+J EMS to target Interstitial campaigns in dependency of these keywords as target.

|  |
| --- |
| [[GUJAdViewContextinstanceForAdspaceId:@"AdSpaceId"] interstitialAdViewForKeywords:  [NSArray arrayWithObjects:@"key1",@"key2", nil]  ];  > OR GCD BLOCK:  [[GUJAdViewContext instanceForAdspaceId:@"AdSpaceId"] interstitialAdViewForKeywords:[NSArray arrayWithObjects:@"key1",@"key2", nil] completion:^BOOL(GUJAdView \*\_adView, NSError \*\_error) {  return YES;  }]; |

In case of a keyword mismatch (keyword parameter ⬄ keywords in adserver) no InterstitialView will be displayed and the delegate method interstitialViewDidFailLoadingWithError: will called.

|  |
| --- |
| [[GUJAdViewContextinstanceForAdspaceId:@"AdSpaceId"] interstitialAdViewForKeywords:  [NSArrayarrayWithObjects:@"key1",@"key2", nil]  ]; |

### Startstitial Handling

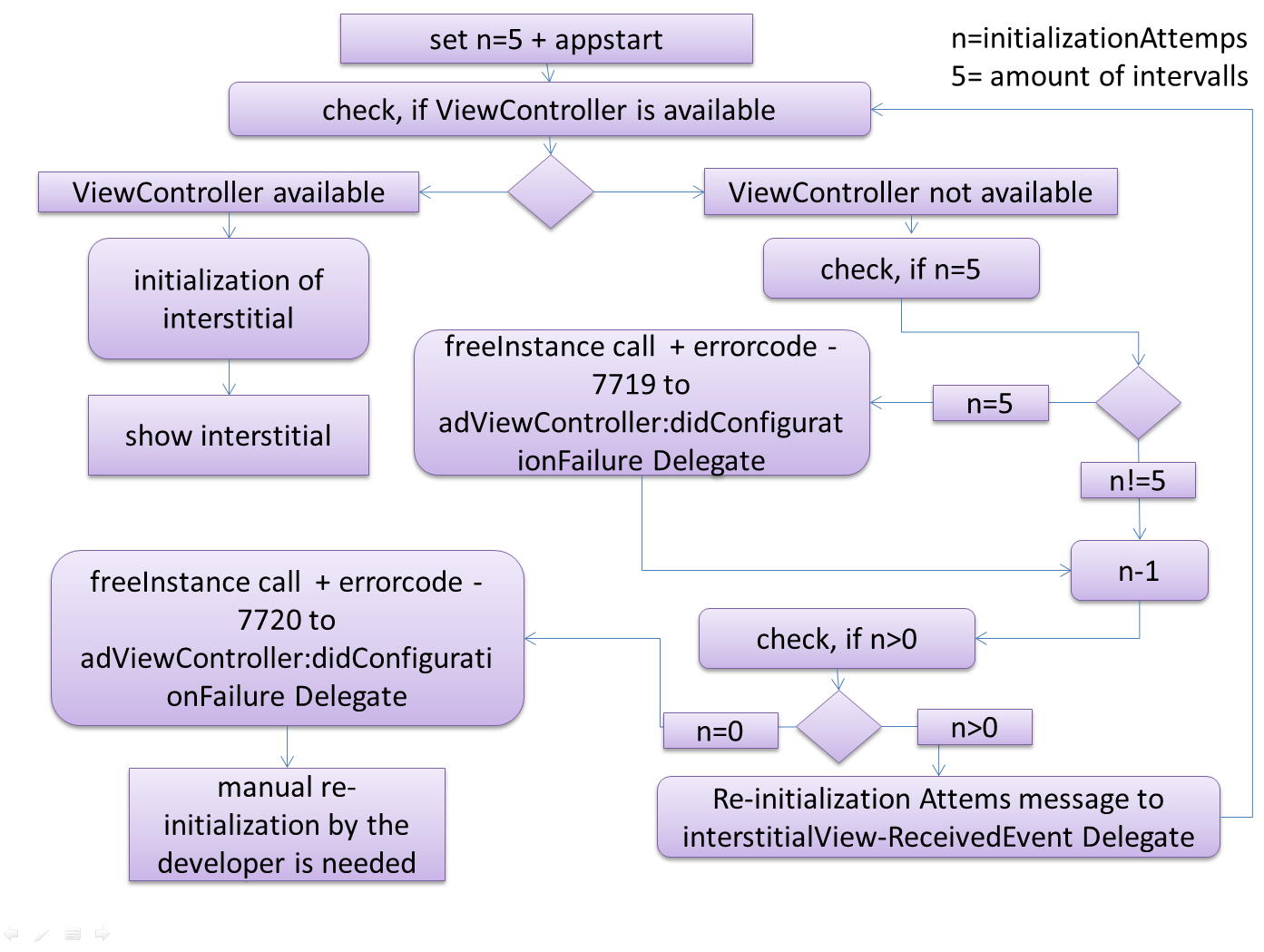
When presenting a Start-interstitial without using GCD-Block-Calls, the AdView appears first when a ViewController (RootViewController) becomes responsible. It takes different time after start of an app until the ViewController is available. To avoid loading of interstitials before this ViewController is accessible the SDK provides an initialization-waiting queue. This is needed for each interstitialView going to be handled in the app delegate class.

The initialization-waiting-queue will check whether the ViewController becomes available or not. The interval is one (1) second. The developer is able to set the amount of intervals in an initializationAttemps property as followed.

|  |
| --- |
| [gujAdViewContext\_ initalizationAttempts:5]; |

For example: the number „5“ will cause five (5) checks for a valid ViewController.

In case of an unsuccessful search for a ViewController the SDK is going to quit. This process is realized as calling the *freeInstance* method. After this, the developer should do a re-initialization programmatically.

In the initialization-waiting queue the SDK fires several errorcodes and events to different delegates. In the following graphic the control flow for this queue is described.

* Fehlercode -7719 (Missing (Root)ViewController) is sent to the „*adViewController:didConfigurationFailure:“*delegate, when the ViewController availability is first time checked. In case of developers wish for manual initialization the developer can react on this error-code.
* During theinitialization-waiting queue the SDK firesthe information „*Re-Initalization Attempt*“ to the „***interstitialViewReceivedEvent***“ for each intervalic check.
* In case of last time checking the availability of the ViewController and the unsuccess of initialization the errorcode-7720is sent to the „*adViewController:didConfigurationFailure*:“ delegate. After receiving this code the developer should manually re-initialize the InterstitialView.

## Add own requestheader and requestparameter

It is possible to define own request header and request parameter for the ad-request. This is needed e.g. for the delivery of the current temperature for temperature targeting. With the following lines of code you are able to define and implement them.

TheRequestHeaderFieldis a parameter delivered as http-Header in the ad-request.

TheRequestParameter is a parameter delivered as QueryString as a part of the request URL.

|  |
| --- |
| [[GUJAdViewContextinstance] addAdServerRequestHeaderField:@"name" value:@"key1"]  [[GUJAdViewContextinstance] addAdServerRequestParameter:@"name" value:@"key1"] |

The parameter should set before sending the ad-request to the ad-server. This means the parameter should declared before calling adView:, adViewWithOrigin:, adViewForKeywords:, adViewForKeywords:origin, interstitialAdView orinterstitialAdViewForKeywords.

# ERROR CODES

| Error Code | SDK - Internal Name | Description |
| --- | --- | --- |
| -1000 | GUJ\_ERROR\_CODE\_GENERAL\_UNDEFINED | An undefined / unknown Error occur. This may a invalid ad view response, iOS related or programmatically related error. |
| -1004 | GUJ\_ERROR\_CODE\_UNABLE\_TO\_COMPLETE | A function or request was unable to completed by the SDK. Can occur when the AdView creation process or SDK initialization process has interrupted. Also occurs when the ad server request is interrupted by a connection time out. |
| -1005 | GUJ\_ERROR\_CODE\_COMMAND\_FAILED\_OR\_UNKNOWN | If an ORMMA-Command fails. |
| -1006 | GUJ\_ERROR\_CODE\_UNAVAILABLE | If an ORMMA function or command and / or a resource is not available. |
| -1007 | GUJ\_ERROR\_CODE\_FAILED\_TO\_ASSIGN\_OBJ | If the SDK can not assign an ad view object to a third party SDK and / or function. |
| -1109 | GUJ\_ERROR\_CODE\_ORMMA\_CALL\_UNHANDLED | If an ORMMA - Call / Function is available but can not be handled by the SDK |
| 1 | GUJ\_ERROR\_CODE\_ADSPACE\_ID | If the Ad-Space-ID is not well configured. |
| 1005 | GUJ\_ERROR\_CODE\_SERVER\_ERROR | If the Ad-Server responses in an unknown format. |
| 1006 | GUJ\_ERROR\_CODE\_INCORRECT\_AD\_FORMAT | If the Ad-Server responses with an unknown Ad-Format. |
| 1008 | GUJ\_ERROR\_CODE\_INVALID\_AD\_FORMAT\_HEADER | If the Ad-Server response an empty body, error or unknown Flight-Header-Code. |
| 1009 | GUJ\_ERROR\_CODE\_MISSING\_ADCONFIGURATION | If the Ad-Configuration is erroneous. |
| 1010 | GUJ\_ERROR\_CODE\_INVALID\_AD\_SERVER\_RESPONSE | If the Ad-Server response is valid by its flights-Header but does not conform to an valid ad body format. |
| 22 | GUJ\_ERROR\_CODE\_MOCEAN\_AD\_FAILD\_LOADING | If the SDK can NOT load the mOcean SDK binaries. |
| 400 | GUJ\_ERROR\_CODE\_CORE\_LOCATION | If the core-location framework is erroneous or can not be loaded and / or initialized. |
| 2003 | GUJ\_ERROR\_CALENDAR\_UNAVAILABLE | If the calendar framework is erroneous or can not be loaded and / or initialized. |

# Known ORMMA Issues

(ISSUE 9000.001)

The ORMMA Javascript (V1.1) delivers no response according to several native functioncallbacks like video or audio, maps orevents.

In this case the AdDesigner has to implement additional javascript-handler to be able to cause actions after relevant notifications of the SDK.

(ISSUE 9000.002)

When implementing the click-to-calendar (ormma.createEvent()) functionality the AdDesigner hasto implement a notification to give the app-user a signal when a new entry was created in their calendar.

(ISSUE 9000.003)

The delivered ads should contain information about the size ofthe ad e.g. width:300px and height: 50px. Otherwise the SDK won’t be able to calculate the size ofthe Ormmaview. The default Ormmaview size and height will be 1px to 1px.

Therefore no ad can be displayed and an error with error-code: 9002 would be thrown.