
POKKT Native iOS SDK Developer API Guide

v1.1

Approval Form

THIS DOCUMENT MAY NOT BE REPRODUCED OR COPIED IN WHOLE OR IN PART

Document ID	PokktAndroidSDKDeveloper-PASD01
Revision Number	1.1
Created Date	19 March 2015
File Name	Pokkt SDK API Guide v1.0
Number of Pages	10
Status	Created
Revised Date	15 March 2015
Created By	Pokkt

How to use PokktSDK?

PokktSDK integration steps

Step1 : Add the libPokktSDK.a file to application

Project Header → Targets → Build Phase → Link Binary With Libraries → Plus → Others
→ Choose libPokktSDK.a

Steps 2: Add SDK Public header files to application.
Drag and drop the all .h files to application.

Implementation steps.

- Import the PokktManager.h file into your application . Then create instance for the PokktManager class. While creating instance you can send user security information as dictionary or you can set user security information values into PokktManagerSecurity.plist. (Required)
- By default auto cache is 'Enabled', When we need to disable auto cache , To be pass the user security information with auto cache 'NO'.
- You can set the debug enabled or not. By default it is 'NO'. (Optional)
- Mandatory to set the delegate for callbacks. (Required)
- Mandatory to pass the present controller for playing the videos. (Required)
- When we disabled auto cache. This "cacheVideoCampaign" function need to be called. This will call the service to get the video and cache. (Required)
- After cached To play Incentivised or non Incentivised video's ,call "playVideoCampaign:" function. (Required)
- To show the play video buttons , need to observe "isVideoAvailable" function. (Required)
- To show the skip button on video player , While playing the video. Need to call "setSkipVideo:" as YES or NO. By default it is YES. (Optional)
- You can set the default time to show skip button on video. For that need to call the "setDefaultSkipTime" function. (Optional)
- Need to implement the delegate methods to handle the video presentation and when downloading completed or failed. (Required)

VideoResponse

Purpose

This is a model classes . It will return the response of the gratified videos.

PokktManagerSecurity.plist

Purpose

In this plist file , have to set some security values for getting the videos.

Public class:

PokktManager

Purpose

Getting Instance of Pokkt SDK and executing operations play video campaign.

Constants

```
static NSString * SECURITY_KEY      = @"security_key";
static NSString * APPLICATION_ID    = @"application_id";
static NSString * USER_ID          = @"U_ID";
static NSString * INTEGRATION_TYPE  = @"integration_type";
static NSString * AUTO_CACHE_VIDEO = @"auto_cache_video";
```

Use this constants for when user want to send their security information.

Public methods

+ (id)sharedInstance

Create instance of PokktManager to access pokkt sdk. This will do the auto cache.

Returns

Instance of PokktManager.

Example

```
PokktManager *pokktManager = [PokktManager sharedInstance];
```

+ (id)sharedInstance:(NSDictionary *)userInfoDict

Create instance of PokktManager to access pokkt sdk.

Method Parameters

userInfoDict : This is dictionary, User has to pass the security key,application id,user id and integrity type as a dictionary . Create dictionary with constant string as a keys. Here we can disable the auto cache.

Returns

Instance of PokktManager.

Example

```
NSDictionary *userDict = [NSDictionary dictionaryWithObjectsAndKeys: @"", SECURITY_KEY, @"",
APPLICATION_ID, @"", USER_ID,@"", INTEGRATION_TYPE,[NSNumber numberWithInt:NO],
AUTO_CACHE_VIDEO, nil];
PokktManager *pokktManager = [PokktManager sharedInstance:dict];
```

Otherwise we can make the dictionary with security key and values , This dictionary should pass to

Only for official use

© Private & Confidential

Prepared By: Pokkt Team

pokktmanager instance methods.

In the above steps we should use any one.

PokktResources.bundle

Purpose

In this bundle file , This need to be add into the application. This bundle has image.

-
- (void)setDebugger:(BOOL)isEnabled;
To set debug mode YES or NO . By default, debug mode is YES.

Method Parameters

isEnabled: Whether app should be run in debug mode.

Example

```
[pokktManager setDebugger:YES];
```

- (void)cacheVideoCampaign
It will call video service and then it will cache those video.

Example

```
[pokktManager cacheVideoCampaign];
```

- (void)playVideoCampaign:(BOOL)alsIncentivised withScreenName:(NSString *)screenName
Play cached video campaign.

Method Parameters

alsIncentivised: To set whether user will be gratified or not.

screenName : To be pass the video playable screen or controller name.

Example

```
[pokktManager playVideoCampaign:YES withScreenName:@"ViewController"]; // Gratified Video
```

or

```
[pokktManager playVideoCampaign:NO withScreenName:@"ViewController"]; // Non Gratified Video
```

- (BOOL)isVideoAvailable
Check whether video is cached on device.

Returns

Returns YES if video is cached on device else returns NO.

Example

```
BOOL blsVideoAvailable = [pokktManager isVideoAvailable];
```

- (void)setDefaultSkipTime:(int)timeInSec;
Set the default skip time in seconds.

Method Parameters

timeInSec: Time for skip the video.

Example

```
[pokktManager setDefaultSkipTime:10];
```

-
- (void)setSkipVideo:(BOOL)boolVal;
To set skip video functionality while playing. By default, skip button is visible.

Method Parameters

shouldSkip: YES for disable skip button.

Example

```
[pokktManager setSkipVideo:YES];
```

- (void)setPokktDelegate:(id<PokktDelegate>)mDelegate;
To set Pokkt Delegate functionality. for call back the delegates. while playing video.

Method Parameters

mDelegate : Current class object for call back delegates.

Example

```
[pokktManager setPokktDelegate:self];
```

- (void)setPresentView:(UIViewController *)controller;
To set the presentview for player view.

Method Parameters

controller : Present controller object

Example

```
[pokktManager setPresentView:self];
```

Properties of PokktManager

@property (nonatomic, strong, readonly) NSString * applicationId;
This property is used to read the application id.

@property (nonatomic, strong, readonly) NSString * UID;
This property is used to read the user id.

@property (nonatomic, strong, readonly) NSString * SDKVersion;
This property is used to read the SDKVersion.

@property (nonatomic, strong, readonly) NSString * videoVC;
This property is used to read the VideoVC.

@property (nonatomic, strong, readonly) NSString * securityKey;

`@property (nonatomic, readonly) BOOL isEnabled;`
This property is used to read the isEnabled.

`@property (nonatomic, strong) NSString * name;`
Optional to set the name property.

`@property (nonatomic, strong) NSString * age;`
Optional to set the age property.

`@property (nonatomic, strong) NSString * sex;`
Optional to set the sex property.

`@property (nonatomic, strong) NSString * location;`
Optional to set the location property.

`@property (nonatomic, strong) NSString * martial_Status;`
Optional to set the martial_Status property.

`@property (nonatomic, strong) NSString * birthday;`
Optional to set the birthday property.

`@property (nonatomic, strong) NSString * facebook_ID;`
Optional to set the facebook_ID property.

`@property (nonatomic, strong) NSString * twitter_Handle;`
Optional to set the twitter_Handle property.

`@property (nonatomic, strong) NSString * education_Information;`
Optional to set the education_Information property.

`@property (nonatomic, strong) NSString * nationality;`
Optional to set the nationality property.

`@property (nonatomic, strong) NSString * employment_Status;`
Optional to set the employment_Status property.

`@property (nonatomic, strong) NSString * content_type;`
Optional to set the content_type property.

`@property (nonatomic, strong) NSString * opt_userId;`
Optional to set the opt_userId property.

PokketController:

Delegate methods

DownloadCompleteListener (PokktDelegate)

Purpose

Set delegate for downloading.

Public Methods

- (void)didFinishedVideoDownload:(BOOL)isFinished;
Called when video downloading is completed.

Method Parameters

isFinished: Bool value of the video downloaded .

- (void)didFailedVideoDownload:(NSString *)errorMsg;
Called when video downloading has failed.

Method Parameters

message: error message for failure.

Example

```
- (void)didFinishedVideoDownload:(BOOL)isFinished
{
    // Do operation when video downloading finished
}

- (void)didFailedVideoDownload:(NSError *)error
{
    // Do operation when video downloading failed
}
```

VideoDisplayedDelegate (PokktDelegate)

Purpose

Set delegate on video playback display.

Public methods

- (void)onVideoDisplayed
Called when video is started.

Example

```
- (void)onVideoDisplayed
{
    // Do Operation on when video displayed
}
```

VideoCompletedDelegate (PokktDelegate)

Purpose

Set delegate on video playback completion after full view.

Public methods

- (void)onVideoCompleted:(BOOL)isCompleted

Called when video playing is completed after full view.

Example

```
- (void)onVideoCompleted:(BOOL)isCompleted
{
    // Do operation when video compleated
}
```

VideoSkippedDelegate (PokktDelegate)

Purpose

Set delegate on video playback is skipped.

Public methods

- (void)onVideoSkipped

Called when video is skipped using skip button.

Example

```
- (void)onVideoSkipped
{
    // Do operation when video skipped
}
```

VideoGratifiedDelegate (PokktDelegate)

Purpose

Set delegate to know whether to gratify user.

Public methods

- (void)onVideoGratified:(VideoResponse *)videoResponse

Called when video playing is completed and user is to be gratified for the view.

Example

```
- (void)onVideoGratified:(VideoResponse *)videoResponse
{
    // Received the videoresponse model classes
}
```

