

# JioAds SDK for KaiOS

## Table Of Contents

### Document Revision Control:

Version	Description	Date[yyyy-mm-dd]
1.0	Initial draft	2021-04-30
1.5	Adding MultiAd related info	2021-07-16
1.6	Adding Meta data, custom native related info	2021-07-30

# JioAds SDK for KaiOS

This document outlines the method to integrate the JioAds SDK for KaiOS in a privileged javascript app using the official plugin (Beta Version).

Before commencing the integration, ensure that you have created the necessary adspots on the [JioAds Dashboard](#).

## Step 1: Download the SDK

Before proceeding, ensure that you have downloaded the latest version of the JioAds SDK for KaiOS.

Extract all the files from the SDK archive into a separate folder. Plugin is available at below location: *KaiOSSDK*

Do not use any doctype in your html document: (Refer sample below)

```
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-
scale=1.0, maximum-scale=1.0" />
  <title>KaiOS Mobile Web Application</title>
  <link rel="stylesheet" href="css/common.css" />
  <script src="build/KaiOSSDK/jioAds.js"></script>
</head>
<body>
  // Body content goes here
  <div>
    Hello World!
  </div>
</body>
</html>
+ MOAT is Applicable only on Native Interstitial ad-formats.
```

## Step 2: Install the Plugin

Copy the “KaiOSSDK” folder in your app’s root folder and insert the *build/KaiOSSDK/jioAds.js* in your app html as below:

```
<script
type="text/javascript" src="build/KaiOSSDK/jioAds.js"></scr
ipt>
```

## Step 3: Integrate using the Plugin

Copy Paste the below code into your index.html file

```
<ins id="[ENTER-UNIQUE-PLACEMENT-HERE]" data-adspot-
key="[ENTER-ADSPOTKEY-HERE]" data-source="[ENTER-PACKAGE-
NAME/APP-IDENTIFIER]"></ins>
```

Add the below code into your “app.js” file

```

window.addEventListener('DOMContentLoaded', function () {
    var placementId = "unique_ad_5"; //Enter unique placement id
    JioAds.cacheAd(placementId);

    JioAds.onAdPrepared = function(placementId, adUxType)
    { //onAdPrepared is callback event dispatched by sdk, if cacheAd is
      successful
        JioAds.loadAd(placementId);
    } //call this function on 'DOMContentLoaded' is compulsory
}

```

//NOTE: for understanding please go through the app.js given in the SDK  
download app/js folder

## Requesting Ads:

While an user invokes “JioAds.cacheAd” ad requests is being cached;  
wherein “JioAds.loadAd” invocation will render the ad which is obtained from the cache.  
Below function can be used to your desired app event to cache or load an Ad.

```

JioAds.cacheAd( '<unique_placement_id>' );
JioAds.loadAd( '<unique_placement_id>' );

```

## Ad Configuration

Can be used for configuring or overriding any default ad params like log level, ad request timeout, ad rendering timeout and any params specific to some Ad format.

```

JioAds.setConfiguration({
    logLevel: '<log level of SDK>',
    adRequestTimeout: '<timeout in msec>',
    adRenderingTimeout: '<timeout in msec>',
});

```

//NOTE: for details see APPENDIX D

## Native Interstitial

1. Add <ins ...></ins> with appropriate adspotkey and unique placement id (JioAds ad tag)
2. Attach the below function to your desired app event to cache or load an Interstitial Ad

```

JioAds.cacheAd( '<unique_placement_id>' );
JioAds.loadAd( '<unique_placement_id>' );

```

## Pre-roll Video

3. Add <ins ...></ins> with appropriate adspotkey and unique placement id (JioAds ad tag) along with video container id
1. Attach the your media container to JioAds video player for pre-roll experience

```

// Media Container
<div id="video_container_id">
// Video Player Code goes here
</div>
<ins id="[ENTER-UNIQUE-PLACEMENT-HERE]" data-adspot-
key="[ENTER-ADSPOTKEY-HERE]" data-source="[ENTER-PACKAGE-

```

```
NAME/APP-IDENTIFIER]" data-container-id="[VIDEO-CONTAINER-ID]"></ins>
```

2. Attach the below function to your desired app event to cache or load an Ad

```
JioAds.cacheAd( '<unique_placement_id>' );  
JioAds.loadAd( '<unique_placement_id>' );
```

## Rewarded Video

Rewarded Video are opt-in Video Ads that reward users with virtual currency or premium content for viewing video trailers. Ads that reward users with virtual currency or premium content for viewing video trailers. You can use this adspot to load rewarded video ads that can be used to reward users with in-app currency on view or completing actions.

1. Add `<ins ...></ins>` with appropriate adspotkey and unique placement id (JioAds ad tag)

```
<ins id="[ENTER-UNIQUE-PLACEMENT-HERE]" data-adspot-  
key="[ENTER-ADSPOTKEY-HERE]" data-source="[ENTER-PACKAGE-  
NAME/APP-IDENTIFIER]"></ins>
```

2. Attach the below function to your desired app event to cache or load an Rewarded video Ad

```
JioAds.cacheAd( '<unique_placement_id>' );  
JioAds.loadAd( '<unique_placement_id>' );
```

3. You shall get the below call back when the video is complete or played till the reward in duration received from delivery in order to gratify the user.

```
JioAds.onAdMediaEnd( '<unique_placement_id>',  
<true_if_video_playback_is_completed_else_false> );
```

```
JioAds.onAdClosed( '<unique_placement_id>',  
<true_if_video_playback_is_completed_else_false> );
```

## Banner and Billboard

Native In-Feed (banner) and Native Content Stream (billboard) ad-format mimic the look and feel of the content stream in which they are placed.

1. Add `<ins ...></ins>` (JioAds ad tag) at appropriate position; where they want AD to be rendered.
2. Call `cacheAd`, `loadAd` method on desired app event ad can be rendered. We recommend it on 'DomContentLoaded' event.

```
document.addEventListener( 'DOMContentLoaded', function(e) {  
    console.info( 'Starting loadAd Event' );  
    JioAds.loadAd( '<unique_placement_id>' );  
});
```

3. App developer should move focus on active elements (i.e. `activeElement`) in order to enable focus.  
If ad is focused and an 'Enter Key' is pressed, call to action will be performed.

## Auto Refresh Ad:

Use the below attribute to enable automatic ad refresh after 'x' seconds. In below case ad will automatically keep on refreshing itself after every 30 seconds.

```
<ins id="[ENTER-UNIQUE-PLACEMENT-HERE]" data-adspot-key="[ENTER-ADSPOTKEY-HERE]" data-refresh-rate="30"></ins>
```

+ Applicable only for Native In-Feed (banner) and Native Content Stream (billboard) ad-formats.

## Click and Close Ad:

The Kai Ads SDK will attach click and close events on 'SoftLeft' and 'SoftRight' keys by default.

+ Not applicable for Native In-Feed (banner) and Native Content Stream (billboard) ad-formats.

## Supported Ad Formats

Currently supported Ad formats are as below:

- Native Ads
  - Interstitial Native Ads
  - Interstitial Native Video Ads with Interstitial Native End Card
  - Native In-Feed (Banner)
  - Native Content Stream (Billboard)
- VAST Ads
  - In-Stream Video Ads
  - Rewarded Video Ads
  - AdPod Video Ads

# APPENDIX A

Below events callbacks can overridden by developers as per their app requirements

## Callbacks Definition

```
JioAds callbacks {  
    onAdFailedToLoad(placementId, options),  
    onAdPrepared(placementId, adUxType),  
    onAdRender(placementId),  
    onAdClosed(placementId, isVideoCompleted, reward),  
    onAdClicked(placementId),  
    onAdMediaStart(placementId),  
    onAdMediaEnd(placementId, reward),  
    onAdRefresh(placementId),  
    onAdSkippable(placementId)  
}
```

## JioAds Ad Event Callback List

Callback	Values
onAdFailedToLoad	Fires when error occurs (Eg: while requesting ad/Caching/Showing)
onAdPrepared	Fires when ad is ready to load (Applicable for all ad formats)
onAdRender	Fires when Ad is ready to rendered
onAdClosed	Fires on closing the ad
onAdClicked	Fires when user interact (Click) with the ad
onAdMediaStart	Fires when media play started. (Applicable only for VAST Videos <i>i.e. Interstitial and Pre-roll</i> )
onAdMediaEnd	Fires when media ends (Applicable only for VAST Videos <i>i.e. Native Interstitial, Rewarded video and Pre-roll Ads</i> )
onAdRefresh	Fires on refreshing the ad (Applicable only for Banner and Billboard)
onAdSkippable	Fired for video ads when skip duration is set and user watched the video till that point and will be able to skip.

## Attributes that can be set under <ins> tag

Attributes for <ins>	Values
id	Any unique string, without space, for the page
data-adspot-key	Adspot key. You can find either in step 3 or in adspot listing page under ID:
data-source	App identifier in which you are integrating JioAds SDK
data-refresh-rate	Refresh the ad after every x seconds.
data-container-id	Video Ad container div id
data-customization	"set-ad-data" / "get-ad-data" Refer Appendix G for setting other custom attributes
data-ad-break-duration	Midroll / ad break duration
data-adMetaData	Key value meta data Refer Appendix F for more details

# APPENDIX B

## Rendition precedence for Native Interstitial Ad Markup

Rendition precedence for *Native Interstitial* ad-format will be followed in below order:

1. Video element will take priority among all; after video play native interstitial experience will be shown (*similar to end-card*) to an end user.
2. Best fit image will be delivered; if any of additional creative element is present (i.e. 240x260)
3. Ad template with any of below available elements:
  - Icon, Title, Image, Description, CTA from the native json response
  - a. Ad template image 1200x627; if specified image is present
  - b. Ad template image 300x250; if specified image is present

## Rendition for Native In-Feed

Native elements used for rendition are as below:

- Icon
- Title
- Description
- CTA Button

## Rendition for Native Content Stream

Native elements used for rendition are as below:

- Icon
- Title
- Description
- CTA Button
- Image

# APPENDIX C

## Error Object for onAdFailedToLoad

You can update your *onAdFailedToLoad* interface to provide error details by adding *Error Object Parameter*.

```
/**
 * Method will be invoked if, Error occurred.
 * @param {Object} options - Error object returned by Ad SDK.
 */

JioAds.onAdFailedToLoad(placementId, options) {
    console.log(options);
}
```

Sample: *ErrorObject*

```
{
  placementId: "unique_ad_5",
  ErrorCode: "1008",
  ErrorDescription: "Network error"
}
```

Error Code	Error Description
1001	No Fill
1002	Ad request not allowed
1003	Manifest entry missing
1004	Request Timeout
1005	Internal server error
1006	SDK initialization error
1008	Network error
1009	Mandatory params missing
1010	Rendition error
1011	Parsing error / VAST not parsed
1012	Unknown error
1013	Invalid request arguments
1014	Invalid Ad request/Arguments
1015	Invalid Media File
1021	Invalid Adspot
1022	Invalid package name
1023	Capping reached

## APPENDIX D

### List of Ad configuration params with setConfiguration

```
JioAds.setConfiguration({
    ifa: 'bc6b78a1-4df0-407e-9a20-88bc64907232',
    reqType: "stg", //stg, prod
    ai: "com.jio.parentpkg",
    logLevel: "debug",
    adRequestTimeout: 6000,
    adRenderingTimeout: 5000,
    globalMetaData: { .... }
});
```

### JioAds Ad configuration params List

Params	Values
ifa	App configuring ifa to be used in Ad request
logLevel	debug, log, info, trace, warn, error enable the level of SDK logs
adRequestTimeout	Ad request timeout for XHR calls
adRenderingTimeout	Ad rendering timeout
reqType	Server endpoint for verification stg / prod
ai	Parent package name if different from child package name
la	Latitude
lo	Longitude
acc	Location accuracy
gts	Location timestamp
globalMetaData	Global meta data key value configuration Refer Appendix F for supported key meta data



# APPENDIX E

## Adbreak / Midroll Support - Instream Ads

You can set the Ad break within the INS tag **data-ad-break-duration= [value in secs] attribute**, in order to create an midroll. Midroll are instream ads that are played back to back till the duration completes.

This feature is used to monetize during live streaming of events where the duration of the ad breaks are unknown to the publisher.

To initiate the Ad break you will need to use the below code into your application:

```
<ins id="[ENTER-UNIQUE-PLACEMENTID-HERE]" data-adspot-key="[ENTER-ADSPOTKEY-HERE]" data-container-id=[ENTER-CONATINER-ID-WHERE-YOU-WANT-TO-SHOW-AN-AD] data-ad-break-duration=[ENTER-AD-BREAK-DURATION]></ins>
```

data-ad-break-duration: ad break duration for a midroll

data-container-id : this is an placement where you want to show your Ad

Kindly ensure the minimum value for data-ad-break-duration is 2 or greater than 2 secs else the will not create a video Ad break while throwing appropriate error.

# APPENDIX F

## Global Ad Meta Key value Settings

Push keyword data to *JioAds SDK through ins tag or set them through setConfiguration API*. Put your app and user relevant info, data will be injected and will be applicable for subsequent *cacheAd* call.

```
<ins id="[ENTER-UNIQUE-PLACEMENTID-HERE]" data-adspot-key="[ENTER-ADSPOTKEY-HERE]" data-container-id=[ENTER-CONATINER-ID-WHERE-YOU-WANT-TO-SHOW-AN-AD] data-adMetaData=[ENTER-AD-KEY-VALUE-META-JSON-OBJECT]></ins>
```

```
JioAds.setConfiguration({
    ifa: 'bc6b78a1-4df0-407e-9a20-88bc64907232',
    reqType: "stg", //stg, prod
    globalMetaData: {
        ag: 28,
```

```

        ci: 'mumbai',
        gn: 'M',
        pc: 400063,
        chnm: "zee5",
        screenname: "football",
        jiomoney: "games",
        jiotv: "display",
        jioplatform: "jiopages"
    }
};

```

## Key Value Meta object definition:

Labels	Keys
Channel ID	chid
Channel Name	chnm
Show Name	shnm
Page Category	pcat
Section Category	scat
Language of Article	loa
Language	lang
Content ID	ctid
Content Type	ctype
Vendor	vnm
Actor	act
Objects	obj
Is Kids Protected	iskp
App Version	avr
Genre	gnr
State	st
City	ci
Age	ag
Gender	gn
Country	co
Pincode	pc
Device Brand	dvb
Device Model	dvm
Serial ID	srid
isQwerty	isq
Hour	hr
Minute	min
Keywords	kwrds
Network Type	nt
Custom Meta Data	

# APPENDIX G

## Customizing Native Ads

Jio Ads SDK will render native ad in its default layout. Developer can further extend the rendition and create a customized experience.

Developer can access native elements via `data-native-element`.

Eg: `<span data-native-element="title"></span>`

### Example layout

```
<ins id="[ENTER-UNIQUE-PLACEMENTID-HERE]" data-adspot-key="[ENTER-ADSPOTKEY-HERE]" data-customization="[set-ad-data/get-ad-data]"
click-element="[customClickButton]" sec-click-
element="[customSecClickButton]" close-
element="[customCloseButton]" video-element="[customVideoTag]">

  <video class="customVideoTag" controls data-native-
element="video" style="width:100%; display: block;"></video>

  <img class="customCloseButton" style="position:absolute;
display: block;" />

  <img data-native-element="mainimage"/>

  <button class="customClickButton" data-native-element="ctatext"
></button>

  <button class="customSecClickButton" data-native-
element="secctatext" ></button>

  <div data-native-element="desc"><strong>Description:
</strong></div>

</ins>
```

### Note:

- If the Ad has no data-customization detected inside `<ins>` tag; it will render the default layout for either “Native In-feed” or “Native Content Stream” whatever is applicable. For more details.

- It is absolutely mandatory to wrap all elements inside html tag and apply css to hide them. i.e. style="display:none;"
- It is mandatory to use <div> tag to render text elements and <img> tag to render image elements i.e. imageIcon, customImage
- For displaying video ads it's mandatory to fill video-element with the class name in ins tag.
- For ad close control need to fill close-element with the class name
- For primary cta need to fill click-element
- For secondary cta need to fill sec-click-element
- To ensure that the custom layout is properly filled, it is expected that the developer would set all required ad elements as mandatory at adspot level.

#### List of Native Elements

data-native-element	Values
title	Text for Title. Max 30 chars.
video	Video url
ctaText	Text for Call To Action.
secctatext	Secondary text for call to action
desc	Description. Max 90 chars.
desc2	Tagline for ad
rating	Rating for app.
downloads	Download count for app.
price	Price in the default format.
iconimage	Url for image icon. Dimension: 80x80.
mainimage	Url for image. Dimension: IAB.
sponsored	Sponsored text
likes	like
salePrice	Price in USD after discount.
phone	Phone number.
address	Address.
customImage	Custom image