Integration guide

JioAds Android SDK v1.10.4

Index

Introduction – JioAds Android SDK 6

Integration Steps 6

Download SDK 6

Adding SDK in project 6

Gradle Dependencies 6

Basic API calls 6

Manifest Permission 6

SDK callbacks 7

Error Codes 9

Proguard Configuration 10

Video Instream Format 16

Caching Instream Video Ad 16

Showing Instream Video Ad 17

Handling Instream Video Ad object on Activity life cycle 17

Additional API for Instream Video Ad 17

Requesting Ad Duration API for Instream Video Ad 19

Customizing Video Ads 20

Mute / UnMute 20

Video Container 20

Play / Pause 21

Numerical Video Progress 22

Video Progress Bar 22

Minimize / Maximize 23

Skip Ad Label 23

Click Button 24

Ad Badge 25

Complete XML for reference 25

Native Format 30

Creating Container for Native ads 30

Cache Native ads 31

Showing Native Ad 32
Handling Native Ad object on Activity life cycle 32
Customizing Native Banner Ads 33
Icon layout 33
Title 33
Description 34
CTA Button 35
Complete XML for reference 35
Customizing Native Billboard Ads 38
Media Layout 38
Icon layout 38
Title 39
Description 39
CTA Button 40
Sponsored 40
Complete XML for reference 41
Interstitial Format 44
Caching Interstitial Ad 44
Showing Interstitial Ad 45
Customizing Interstitial Ads 45
Media Layout 45
Video Layout 46
Video Close Button 46
Mute/Unmute Button 46
Play/Pause Button 47
Progress Layout 48
Complete XML for reference 49
Dynamic Display Ad Format 52
Caching Dynamic Display Ad 52
Supported Display Sizes 53
Showing Instream Video Ad 53
Handling Display Ad object on Activity life cycle 53
Customizing Dynamic Display Ads 54
Audio Instream Format 55
Caching Instream Audio Ad 55

Showing Instream Audio Ad 56
Showing Instream Audio Companion Ad 56
Supported Audio Companion Ad Sizes 56
Handling Instream Audio Ad object destroy 57
Additional API for Instream Audio Ad 57
Supporting APIs by SDK 57
Enable SDK Logs 57
Passing App Meta Data 57
Enable Media Caching 58
Clearing Cached Media 59
Request Refresh Rate for Ads 59
Bitrate for Video Ads. 59
Set Request Dampening Limit 59
Request Timeout 60
Media Timeout 60
Set Skip Event Key 60
Custom UserAgent 60
Enabling Custom Show Ad 60
Steps to implement 60
APIs available with JioAd 62
APIs available with NativeAd 62
APIs available with VideoAd 64
APIs available with AdEventTracker 65
Companion Ads 66
Integration steps: 66
Step 1: Add below progurad rules: 66
Step 2: Registering companion Ad Slots: 66
2) Use below API to declare JioAdView as companion slot 67
Requesting VMAP 71
AdMob Mediation 72
IMA Mediation 72
PRISM Ads 73
Caching Prism Ad 73
Shopping Icons Visibility 74
Loading Prism Ad 75

Callbacks 75
Close Prims Ad 75

Introduction – JioAds Android SDK

SDK is common for Smartphone, Tablet and Connected TV platforms.

Integration Steps

Download SDK

You can download latest SDK **here**.

Adding SDK in project

Include *jioadsdk.aar* in lib folder.

Gradle Dependencies

Add library file dependency in app build.gradle

```
implementation files('libs/jioadsdk.aar')
```

Add below dependency in app build.gradle

```
implementation 'androidx.browser:browser:1.3.0
```

If you want to fetch Location targeted ads, then add below file dependency in build.gradle

```
implementation 'com.google.android.gms:play-services-ads:16.0.0'
implementation'com.google.android.gms:play-services-location:16.0.0'
```

Basic API calls

SDK requires to call below basic API once the SDK is integrated in app.

On launch of your app. Preferably in onCreate() of Activity class.

```
JioAds.getInstance().init(context);
```

To be called when app is exiting. Preferably in onDestroy of MainActivity.

```
JioAds.getInstance().release();
```

Manifest Permission

We recommend having below permissions in application's manifest

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

SDK callbacks

Below are the callbacks supported by SDK.

```
@Override
public void onAdFailedToLoad(JioAdView jioAdView, JioAdError jioAdError) {
Log.d("JioAds", "onAdFailedToLoad " + jioAdError.getErrorDescription());
}

@Override
public void onAdClosed(JioAdView jioAdView, oolean isVideoCompleted, oolean
isEligibleForReward) {
    Log.d("JioAds", "onAdClosed() called");
}

@Override
public void onAdPrepared(JioAdView adView) {
    Log.d("JioAds", "onAdPrepared() called");
}

@Override
public void onAdClicked(JioAdView jioAdView) {
    Log.d("JioAds", "onAdClicked called");
}

@Override
public void onAdRender(JioAdView jioAdView) {
    Log.d("JioAds", "onAdRender called");
}

@Override
public void onAdRefresh(JioAdView jioAdView) {
    Log.d("JioAds", "onAdRefresh() called");
}

@Override
public void onAdMediaExpand(JioAdView jioAdView) {
    Log.d("JioAds", "onAdMediaExpand() called");
}
```

```
    @Override
    public void onAdMediaCollapse(JioAdView jioAdView) {
        Log.d("JioAds", "onAdMediaCollapse() called");
    }

    @Override
    public void onAdMediaStart(JioAdView jioAdView) {
        Log.d("JioAds", "onAdMediaStart() called");
    }

    @Override
    public void onAdMediaEnd(JioAdView adView) {
        Log.d("JioAds", "onAdMediaEnd() called");
    }

    @Override
    public void onAdReceived(JioAdView jioAdView) {
        Log.d("JioAds", "onAdReceived() called");
    }

    @Override
    public void onAdSkippable(JioAdView jioAdView) {
        Log.d("JioAds", "onAdSkippable() called");
    }
    @Override
    public void onMediaPlaybackChange(JioAdView ioAdView,JioAdView.MediaPlayBack
mediaPlayBack){
        Log.d("JioAds","onMediaPlaybackChange. MediaPlayBack: "+mediaPlayBack);
    }

   @Override
   public void onAdDataPrepared(JioAd oole,Boolean isLastAd){
         Log.d("JioAds", "onAdDataPrepared() called");
   }

    @Override
   public void onAllAdsExhausted(){
     Log.d("JioAds", "onAllAdsExhausted() called");
    }
    @Override
Public void onAdChange(JioAdView jioAdView,Int adNumber){
  Log.d("JioAds", " onAdChange() called");
}
    @Override
Public void onAdMediaProgress(Long totalDuration,long progress){
  Log.d("JioAds", " onAdMediaProgress () called");
}
```

Error Codes

SDK provides callback *onFailedToLoad(JioAdView jioAdView,JioAdError error)* in any error cases.

JioAdError has below 3 APIs.

```
Error.getErrorTitle();
error.getErrorDescription();
error.getErrorCode();
```

Below are the list of error codes which SDK provides to app.

m101 – Error while fetching advid

m102 – Error while fetching UID

m103 – Error fetching vast meta data

m104 – No Fill

m105 – Internet is not available

m106 – Timeout while fetching ad

m107 – Wrong UX type

m108 – Network Error

m109 – Mandatory parameters missing

m110 – Ad rendition error

m111 – Parsing error

m112 – Unknown error

m113 – Error while downloading ad

m114 – Ad spot does not exist

m115 – AD request not allowed

m116 – Adspot is not linked to app

m117 – Request from Invalid/restricted origin

m118 – UA is Invalid/doesn’t exist

m119 – OS is Invalid/doesn’t exist

m120 – Error while fetching advid or uid

m221 – AdPod Timeout error

m122 – Restricted api, please contact your account manager for more details

m123 – Player preparation failed

m124 – User daily impression limit reached
m125 – user lifetime impression limit reached
m126 – user click daily limit reached
m127 – user click lifetime limit reached
m128 – user completed view daily limit reached
m129 – user completed view lifetime limit reached
m130 – user minute wise impression limit reached
m131 – user hourly impression limit reached
m132 – user click minute wise limit reached
m133 – user click hourly limit reached
m134 – user completed view minute wise limit reached
m135 – user completed view hourly limit reached
m136 – Invalid Ad Request Parameters

Proguard Configuration

```
-keepattributes
*Annotation*,JavascriptInterface,Exceptions,InnerClasses,Signature,*Annotation*,EnclosingMeth
od,*Annotation*,Signature

-dontwarn com.google.**
-dontwarn com.google.firebase.**
-dontwarn com.google.android.gms.**

-keep public class com.jio.unity.plugin.android.** {
  public *;
}

-keep public interface com.jio.unity.plugin.android.** {*;}

-keep public class com.jio.jioads.adinterfaces.JioAdPartner {
    public <fields>;
    public <methods>;
}

-keep public class com.jio.jioads.mediation.partners.videoutils.JioMediationVideoController {
```

```
    public <fields>;
    public <methods>;
}

-keep public class com.jio.jioads.mediation.partners.JioMediationListener {
    public <fields>;
    public <methods>;
}

-keep public class com.jio.jioads.mediation.partners.JioMediationAd {
    public <fields>;
    public <methods>;
}


-keep public class com.jio.jioads.adinterfaces.JioAdView {
    public <fields>;
    public <methods>;
}

-keep public class com.jio.jioads.adinterfaces.JioAdError {
    public <fields>;
    public <methods>;
}

-keep public class com.jio.jioads.adinterfaces.AdMetaData {
    public <fields>;
    public <methods>;
}

-keep public class com.jio.jioads.instreamads.vastparser.model.CtaUrl {
    public <fields>;
    public <methods>;
}

-keep public class com.jio.jioads.adinterfaces.JioAds {
        public <fields>;
        public <methods>;
  }

-keep public enum  com.jio.jioads.adinterfaces.JioAds$Companion {
    public <fields>;
    public static <methods>;
}
```

```
-keep public class com.jio.jioads.adinterfaces.JioAdListener {
   public <fields>;
   public <methods>;
}

-keep public class com.jio.jioads.adinterfaces.AdEventTracker {
   public <fields>;
   public <methods>;
}


-keep public class com.jio.jioads.util.Constants {
   public <fields>;
   public <methods>;
}

-keep public class com.jio.jioads.util.Constants$** {
   public <fields>;
   public <methods>;
}

-keep public class com.jio.jioads.network.NetworkTaskListener {
   public <fields>;
   public <methods>;
}
-keep public enum com.jio.jioads.adinterfaces.JioAdView$AD_TYPE {
     <fields>;
      public static **[] values();
      public static ** valueOf(java.lang.String);
  }

-keep public enum com.jio.jioads.adinterfaces.JioAdView$AdState {
  <fields>;
  public static **[] values();
  public static ** valueOf(java.lang.String);
}

-keep public enum com.jio.jioads.adinterfaces.JioAdView$MediaPlayBack{
   <fields>;
   public static **[] values();
   public static ** valueOf(java.lang.String);
}
```

```
-keep public enum com.jio.jioads.adinterfaces.JioAdView$ORIENTATION_TYPE{
   <fields>;
   public static **[] values();
   public static ** valueOf(java.lang.String);
}

-keep public enum  com.jio.jioads.adinterfaces.JioAds$LogLevel {
   <fields>;
    public static **[] values();
    public static ** valueOf(java.lang.String);
}

-keep public enum com.jio.jioads.adinterfaces.JioAds$MediaType {
     <fields>;
     public static **[] values();
     public static ** valueOf(java.lang.String);
}

-keep public enum com.jio.jioads.adinterfaces.JioAds$Environment{
  <fields>;
}

-keep public class com.jio.jioads.adinterfaces.JioAd{
   public <fields>;
   public <methods>;
}

-keep public class com.jio.jioads.adinterfaces.JioAd$NativeAd{
   public <fields>;
   public <methods>;
}

-keep public class com.jio.jioads.adinterfaces.JioAd$VideoAd{
   public <fields>;
   public <methods>;
}

-keep public class com.jio.jioads.adinterfaces.JioAd$VideoAd$Media{
   public <fields>;
   public <methods>;
}

-keep public enum com.jio.jioads.adinterfaces.JioAdError$JioAdErrorType{
  <fields>;
```

```
}

-keep public class com.google.android.gms.location.FusedLocationProviderClient {
   public <fields>;
   public <methods>;
}

#companion story
-keep public class com.jio.jioads.adinterfaces.JioCompanionListener {
   public <fields>;
   public <methods>;
}
-keep public class com.jio.jioads.adinterfaces.JioCompanionListener {
   public <fields>;
   public <methods>;
}
#vmap story
-keep public class com.jio.jioads.adinterfaces.JioAdsLoaderListener {
   public <fields>;
   public <methods>;
}

#Chrome Custom Tab
-keep public class androidx.browser.customtabs.CustomTabsIntent {
   public <fields>;
   public <methods>;
}

#GMA c2s
-keep public class com.jio.jioads.mediation.partners.GooglePlayServicesBanner{
   public <fields>;
   public <methods>;
}
-keep class com.google.** { *; }
-keep interface com.google.** { *; }
-keep class com.google.ads.interactivemedia.v3.api.** { *;}
-keep interface com.google.ads.interactivemedia.** { *; }
-dontwarn com.google.ads.interactivemedia.v3.api.**

-keepclassmembers,allowshrinking,allowobfuscation class com.android.volley.NetworkDispatcher
{
   void processRequest();
}
-keepclassmembers,allowshrinking,allowobfuscation class com.android.volley.CacheDispatcher {
```

```
    void processRequest();
}
# Keep names - Native method names. Keep all native class/method names.
-keepclasseswithmembers class * {
    native <methods>;
}
-keep public class com.google.android.gms.tasks.OnSuccessListener {
    <fields>;
    <methods>;
}

-keep public class com.google.android.gms.** {
    <fields>;
   <methods>;
}
-keep public class com.google.ads.** {
    public <fields>;
    public <methods>;
}
-keep public class com.google.android.gms.common.internal.safeparcel.SafeParcelable {
    public static final *** NULL;
}
-keep,allowshrinking @com.google.android.gms.common.annotation.KeepName class *

-keepclassmembers,allowshrinking,allowobfuscation class com.android.volley.NetworkDispatcher
{
    void processRequest();
}
-keepclassmembers,allowshrinking,allowobfuscation class com.android.volley.CacheDispatcher {
    void processRequest();
}
```

Video Instream Format

Caching Instream Video Ad

This section covers integration steps required to fetch Instream Video ad.

Making a Cache Request for Ad.

```
JioAdView jioAdViewInstreamVideo = new JioAdView (this,"<ADSPOT_KEY_GOES_HERE>",
JioAdView.AD_TYPE.INSTREAM_VIDEO); // context pass here should be Activity context.

jioAdViewInstreamVideo.setAdListener(new JioAdListener() {

        @Override
        public void onAdFailedToLoad(JioAdError jioAdError,JioAdView jioAdView) {
        // Error Callback
        }

        @Override
        public void onAdPrepared(JioAdView jioAdView) {
        // Success callback
        }

        @Override
        public void onAdClosed(JioAdView jioAdView,oolean isVideoCompleted, oolean
        isEligibleForReward) {
        // When Ad is closed
        }

        @Override
        public void onAdRender(JioAdView jioAdView){
        // When ad is rendered on screen
        }

        @Override
        public void onAdMediaEnd(JioAdView jioAdView)          {
        // When media ended
        }

        });

jioAdViewInstreamVideo.cacheAd();
```

Showing Instream Video Ad

When loadAd() API is called, SDK will try to render the ad inside the container shared.

```
Framelayout instreamAdContainer =(FrameLayout)findViewById(R.id.instreamAdContainer);

if(jioAdViewInstreamVideo.getAdState() == JioAdView.AdState.PREPARED) {

        // Providing a container to video ads to show.
        instreamAdContainer.addView(jioAdViewInstreamVideo);

        jioAdViewInstreamVideo.loadAd();

}
```

Handling Instream Video Ad object on Activity life cycle

Call below snippet to destroy the instream video ad object. Usually this is called in onDestroy() of activity.

```
@Override
protected void onDestroy() {
    if (jioAdViewInstreamVideo!= null) {
        jioAdViewInstreamVideo.onDestroy();
    }
super.onDestroy();
}
```

Additional API for Instream Video Ad

Use the below APIs to pause or resume Instream Ad

```
//to hide all the video controls
jioAdViewInstreamVideo pauseAd();

//to show all the video controls
jioAdViewInstreamVideo.resumeAd();
```

Use the below APIs to hide or show all the overlay controls

```
//to hide all the video controls
jioAdViewInstreamVideo.hideControls();

//to show all the video controls
jioAdViewInstreamVideo.showControls();
```

Use below APIs to Expand or Collapse the In-Stream video ads

```
//to expand the video ad
jioAdViewInstreamVideo.expandAd();

//to collapse the video ad
jioAdViewInstreamVideo.collapseAd();
```

During Orientation change

```
@Override
public void onConfigurationChanged(Configuration newConfig) {

        if(jioAdViewInstreamVideo!=null && jioAdViewInstreamVideo.isMediaInProgress()) {

            if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {

                  jioAdViewInstreamVideo.expandAd();

             } else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {

                  jioAdViewInstreamVideo.collapseAd();

            }

        }

super.onConfigurationChanged(newConfig);

}
```

Requesting Ad Duration API for Instream Video Ad

Use the below API to request the video ads of specific duration in seconds. This API is **Only available for in-stream video.**

```
jioAdViewInstreamVideo.  setRequestedAdDuration();
```

Note- Any input less than 2 seconds will be ignored.

Customizing Video Ads

This section covers customizing any video ads. Video could be part of Instream, Native, Interstitial and so on.

Below API need to be called to enable customization.

```
jioadView.setCustomInstreamAdContainer(R.layout.custom_instream_layout)
```

Add required elements with their tags to support individual functionality.

Mute / UnMute

```
<ImageView
        android:id="@+id/iv_sound_unmute_button"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="10dp"
        android:tag="@string/jio_video_volume_icon"
        android:src="@drawable/jio_unmute"
        android:background="@drawable/jio_mute" />
```

android:tag="***@string/jio_video_volume_icon***" detects the ImageView is used for displaying and controlling the Mute / unmute state.

In case app wants to update the drawable icons for both UnMute and Mute, they can pass their drawable within src and background attribute.

Video Container

```
<RelativeLayout
        android:id="@+id/fl_video_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
```

```
        android:layout_centerInParent="true"
        android:tag="@string/jio_video_player_container"
        android:layout_gravity="center_vertical">
```

android:tag="***=”@string/jio_video_player_container***" detects player position.

RelativeLayout can be positioned as per app need.

Play / Pause

```
<ImageView
        android:id="@+id/adPlayback"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_centerVertical="true"
        android:layout_margin="5dp"
        android:contentDescription="@string/playpause"
        android:src="@drawable/jio_vast_pause"
        android:background="@drawable/jio_vast_play"
        android:tag="@string/jio_video_playback_icon"
        android:visibility="gone" />
```

android:tag="***@string/jio_video_playback_icon***" detects the ImageView is used for displaying and controlling the Play / Pause state.

In case app wants to update the drawable icons for both Pause and Play, they can pass their drawable within src and background attribute.

Numerical Video Progress

```
<TextView
        android:id="@+id/progressCount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_gravity="center"
        android:layout_toRightOf="@+id/adText"
        android:padding="5dp"
        android:tag="@string/jio_video_progresscount"
        android:text="00:00"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:textColor="@android:color/white"
        android:textSize="12sp"
        android:visibility="gone" />
```

android:tag=" ***@string/jio_video_progresscount***" indicates this view will be used for showing progress count of video ad.

Video Progress Bar

```
<ProgressBar
        android:padding="5dp"
        android:layout_centerVertical="true"
        android:tag="@string/jio_video_progressbar"
        style="@style/Widget.AppCompat.ProgressBar.Horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_toStartOf="@+id/adSizeToggle"
        android:layout_toEndOf="@+id/progressCount" />
```

android:tag=" ***@string/ jio_video_progressbar***" indicates this view will be used for showing progress of video ad.

Minimize / Maximize

```
<ImageView
        android:id="@+id/adSizeToggle"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true"
        android:layout_gravity="center"
        android:layout_marginRight="5dp"
        android:padding="5dp"
        android:visibility="gone"
        android:src="@drawable/jio_fullscreen_icon"
        android:tag="@string/jio_video_resize_icon"
        android:background="@drawable/jio_minimize_icon"
        />
```

android:tag="***@string/jio_minimize_icon***" detects the ImageView is used for displaying and controlling the Maximize / Minimize state.

In case app wants to update the drawable icons for both maximize and minimize, they can pass their drawable within src and background attribute.

Skip Ad Label

```
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/adDetailsLayout"
```

```
        android:layout_alignParentRight="true"
        android:layout_marginRight="5dp"
        android:layout_marginBottom="3dp"
        android:background="#55000000"
        android:contentDescription="Skip Ad"
        android:drawableRight="@drawable/jio_pre_roll_skip"
        android:drawablePadding="3dp"
        android:gravity="center_vertical"
        android:padding="8dp"
        android:tag="@string/jio_video_skip_element"
        android:text="You can skip ad in SKIP_COUNTER"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:textColor="@android:color/white"
        android:visibility="gone" />
```

android:tag="***@string/jio_video_skip_element***" detects the TextView is used to manage the Skip labels.

Android:contentDescription attribute contains the Skip label which User should see once the ad is skippable.

Android:arawableRight attribute contains the skip drawable used along with Skip Ad text. Publisher can pass their own drawable if needed.

Andoird:text attribute contains the text label used for showing text before the Ad is skippable. i.e. You can skip ad in X secs. **SKIP_COUNTER** is a macro which SDK will replace with skip duration and gradually decrease as media plays.

Click Button

```
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_margin="10dp"
        android:padding="3dp"
```

```
android:tag="@string/jio_video_cta"
android:text="Visit Advertiser"
android:textColor="@color/jio_white"
android:textStyle="bold" />
```

android:tag="***@string/jio_video_cta***" is used to place the CTA label for Ad.

Ad Badge

```
<TextView
android:id="@+id/adText"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerVertical="true"
android:layout_gravity="center"
android:layout_toRightOf="@id/adPlayback"
android:maxLength="15" android:padding="5dp"
android:text="Ad :"
android:textAppearance="?android:attr/textAppearanceSmall"
android:textColor="@android:color/white"
android:textSize="12sp"
android:visibility="gone" />
```

Complete XML for reference

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@android:color/black">
```

```
<RelativeLayout
android:id="@+id/adLayout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_centerInParent="true"
android:layout_gravity="center" />

<RelativeLayout
android:id="@+id/adDetailsLayout"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignBottom="@id/adLayout"
android:background="#55000000"
android:orientation="horizontal">

    <ImageView
    android:id="@+id/adPlayback"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_centerVertical="true"
    android:layout_margin="5dp"
    android:background="@drawable/jio_vast_play"
    android:contentDescription="@string/playpause"
    android:src="@drawable/jio_vast_pause"
    android:tag="@string/jio_video_playback_icon"
    android:visibility="gone" />

    <TextView
    android:id="@+id/adText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:layout_gravity="center"
```

```
android:layout_toRightOf="@id/adPlayback"
android:maxLength="15"
android:padding="5dp"
android:tag="VastAdProgressLabel"
android:text="Ad :"
android:textAppearance="?android:attr/textAppearanceSmall"
android:textColor="@android:color/white"
android:textSize="12sp"
android:visibility="gone" />

<TextView
android:id="@+id/progressCount"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerVertical="true"
android:layout_gravity="center"
android:layout_toRightOf="@+id/adText"
android:padding="5dp"
android:tag="@string/jio_video_progresscount"
android:text="00:00"
android:textAppearance="?android:attr/textAppearanceSmall"
android:textColor="@android:color/white"
android:textSize="12sp"
android:visibility="gone" />

<ProgressBar
android:padding="5dp"
android:layout_centerVertical="true"
android:tag="@string/jio_video_progressbar"
style="@style/Widget.AppCompat.ProgressBar.Horizontal"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_toStartOf="@+id/adSizeToggle"
```

```
            android:layout_toEndOf="@+id/progressCount" />

            <ImageView
            android:id="@+id/adSizeToggle"
            android:layout_width="30dp"
            android:layout_height="30dp"
            android:layout_alignParentEnd="true"
            android:layout_alignParentRight="true"
            android:layout_centerVertical="true"
            android:layout_gravity="center"
            android:layout_marginRight="5dp"
            android:background="@drawable/jio_minimize_icon"
            android:padding="5dp"
            android:src="@drawable/jio_fullscreen_icon"
            android:tag="@string/jio_video_resize_icon"
            android:visibility="gone" />
        </RelativeLayout>
        <ImageView
        android:id="@+id/iv_sound_unmute_button"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="10dp"
        android:tag="@string/jio_video_volume_icon"
        android:layout_alignParentLeft="true"
        android:src="@drawable/jio_unmute"
        android:background="@drawable/jio_mute"/>

        <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/adDetailsLayout"
        android:layout_alignParentRight="true"
```

```
android:layout_marginRight="5dp"
android:layout_marginBottom="3dp"
android:background="#55000000"
android:contentDescription="Skip Ad"
android:drawableRight="@drawable/jio_pre_roll_skip"
android:drawablePadding="3dp"
android:gravity="center_vertical"
android:padding="8dp"
android:tag="@string/jio_video_skip_element"
android:text="You can skip ad in SKIP_COUNTER"
android:textAppearance="?android:attr/textAppearanceSmall"
android:textColor="@android:color/white"
android:visibility="gone" />

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_above="@+id/adDetailsLayout"
android:layout_alignParentLeft="true"
android:layout_marginRight="5dp"
android:layout_marginBottom="3dp"
android:background="#55000000"
android:contentDescription="Ad Counter"
android:drawablePadding="3dp"
android:gravity="center_vertical"
android:padding="8dp"
android:textSize="14dp"
android:tag="@string/jio_video_ad_counter"
android:textAppearance="?android:attr/textAppearanceSmall"
android:textColor="@android:color/white"/>

</RelativeLayout>
```

Native Format

This section covers integration steps required to fetch and render Native ads.

Native ads are of below types

1. Native Banner
    a. This are usually Native In feeds and of size 320x50.
    b. Customization can be applied to be style the ad basis the app layout.
2. Native Billboard
    a. This are usually Native content stream ads and of size 300x250
    b. Customization can be applied to be style the ad basis the app layout.
    c. Can be used to display rendering Mastheads or carousal ads.

In case customization on these native ads is not applied, then SDK will render the ad in its pre-defined layout.

Native Ad is divided into 3 sections.

1. Showing Native Banner and Native Billboard ads using pre-defined layout.
2. Customization for Native Banner
3. Customization for Native Billboard

Creating Container for Native ads

This is common for all sections. Add below Relative Layout in your app layout where you would like the Native ad to render.

```
<RelativeLayout
        android:id="@+id/adview"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:background="@android:color/transparent"
        android:layout_gravity="center"
        android:visibility="gone" />
```

Cache Native ads

Covers integration steps required to fetch Native ad. Its common for all 3 sections.

Making a Cache Request for Ad.

```
JioAdView jioAdViewNative = new JioAdView(this, <ADSPOT_KEY_GOES_HERE>,
<UX_TYPE_GOES_HERE>); // context passed here will be a activity contexr

jioAdViewNative.setAdListener(new JioAdListener() {

        @Override
        public void onAdFailedToLoad(JioAdError jioAdError,JioAdView jioAdView) {
        // Error callback
        }

        @Override
        public void onAdPrepared(JioAdView jioAdView) {
        // Success callback
        }

        @Override
        public void onAdRender(JioAdView jioAdView){
        // When ad is rendered
        }

        @Override
        public void onAdMediaEnd(JioAdView jioAdView,long reward)    {
        // When native video ad is ended
        }

        @Override
        public void onAdRefresh(JioAdView jioAdView)   {
        // When ad is refreshed
        }


});

// Creating object of the ad container.
Final RelativeLayout nativeContainer=(RelativeLayout)findViewById(R.id.adview);

LayoutInflater layoutInflater = (LayoutInflater)
getSystemService(Context.LAYOUT_INFLATER_SERVICE);

nativeContainer.removeAllViews();
nativeContainer.addView(jioAdViewNative);
```

```
jioAdViewNative.cacheAd();
```

Showing Native Ad

Common to all 3 sections. When loadAd() API is called, SDK will try to render the ad inside the container shared. This can be called when an app event occurs or inside onAdPrepared() callback.

```
If(jioAdViewNative.getAdState() == JioAdView.AdState.PREPARED) {

jioAdViewNative.loadAd();

}
```

Handling Native Ad object on Activity life cycle

Call below snippet to destroy the instream video ad object. Usually this is called in onDestroy() of activity. Common to all 3 sections.

```
@Override
protected void onDestroy() {
      if (jioAdViewNative != null) {
            jioAdViewNative.onDestroy();
       }
super.onDestroy();
}
```

Customizing Native Banner Ads

This section covers customizing native banner ads.

Below API need to be called to enable customization.

```
jioadView. setCustomNativeAdContainer(container: Int)
```

Add required elements with their tags to support individual functionality.

Icon layout

```
<FrameLayout
    android:padding="2dp"
    android:id="@+id/native_icon_layout"
    android:layout_width="48dp"
    android:layout_height="48dp"
    android:layout_centerVertical="true"
    android:layout_gravity="center_vertical"
   android:tag="@string/jio_native_icon_layout" />
```

android:tag="***@string/jio_native_icon_layout***" is used to place the CTA label for Ad.

Title

```
<TextView
        android:id="@+id/jio_ads_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="2dp"
        android:ellipsize="end"
        android:gravity="center"
        android:maxLines="1"
```

```
        android:paddingStart="2dp"
        android:paddingEnd="1dp"
        android:tag="@string/jio_native_title"
        android:textColor="@color/jio_black"
        android:textSize="10sp"
        android:textStyle="bold"
        tools:ignore="SmallSp" />
```

android:tag="***@string/jio_native_tilte***" is used to place the CTA label for Ad.

Description

```
<TextView
        android:id="@+id/jio_ads_desc"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/jio_ads_title"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="3dp"
        android:ellipsize="end"
        android:gravity="center"
        android:paddingStart="2dp"
        android:paddingEnd="1dp"
        android:tag="@string/jio_native_description"
        android:textColor="@color/jio_text_color"
        android:textSize="7sp" />
```

android:tag="***@string/jio_native_description***" is used to place the CTA label for Ad.

CTA Button

```
<Button
    android:id="@+id/jio_cta"
    style="@style/style_jio_button_infeed"
    android:layout_width="wrap_content"
    android:layout_height="25dp"
    android:layout_alignParentEnd="true"
    android:layout_centerVertical="true"
    android:layout_gravity="center"
    android:layout_marginEnd="10dp"
    android:focusable="true"
    android:gravity="center"
    android:tag="NativeCTA"
    android:text="Install Now"
    android:textAlignment="center" />
```

android:tag="***@string/NativeCTA***" is used to place the CTA label for Ad.

Complete XML for reference

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="320dp"
  android:layout_height="50dp"
  android:background="@drawable/jio_nativeborder">

  <FrameLayout
    android:padding="2dp"
    android:id="@+id/native_icon_layout"
    android:layout_width="48dp"
    android:layout_height="48dp"
```

```
    android:layout_centerVertical="true"
    android:layout_gravity="center_vertical"
    android:tag="@string/jio_native_icon_layout" />

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="2dp"
    android:layout_toStartOf="@id/jio_cta"
    android:layout_toEndOf="@id/native_icon_layout"
    android:gravity="center"
    android:orientation="vertical">

    <TextView
        android:id="@+id/jio_ads_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="2dp"
        android:ellipsize="end"
        android:gravity="center"
        android:maxLines="1"
        android:paddingStart="2dp"
        android:paddingEnd="1dp"
        android:tag="@string/jio_native_title"
        android:textColor="@color/jio_black"
        android:textSize="10sp"
        android:textStyle="bold"
        tools:ignore="SmallSp" />

    <TextView
        android:id="@+id/jio_ads_desc"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_below="@+id/jio_ads_title"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="3dp"
        android:ellipsize="end"
        android:gravity="center"
        android:paddingStart="2dp"
        android:paddingEnd="1dp"
        android:tag="@string/jio_native_description"
        android:textColor="@color/jio_text_color"
        android:textSize="7sp" />

    </RelativeLayout>

    <Button
      android:id="@+id/jio_cta"
      style="@style/style_jio_button_infeed"
      android:layout_width="wrap_content"
      android:layout_height="25dp"
      android:layout_alignParentEnd="true"
      android:layout_centerVertical="true"
      android:layout_gravity="center"
      android:layout_marginEnd="10dp"
      android:focusable="true"
      android:gravity="center"
      android:tag="NativeCTA"
      android:text="Install Now"
      android:textAlignment="center" />
</RelativeLayout>
```

Customizing Native Billboard Ads

This section covers customizing native billboard ads. Ads can be image or video.

Below API need to be called to enable customization.

```
jioadView. setCustomNativeAdContainer(container: Int)
```

Add required elements with their tags to support individual functionality.

Media Layout

```
<RelativeLayout
    android:id="@+id/fb_media_view"
    android:layout_width="300dp"
    android:layout_height="157dp"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="1.5dp"
    android:layout_marginRight="1.5dp"
    android:descendantFocusability="afterDescendants"
    android:focusable="true"
    android:gravity="center"
    android:tag="@string/jio_native_media_layout"
    android:visibility="visible" />
```

Icon layout

```
<FrameLayout
        android:id="@+id/native_icon_layout"
        android:layout_width="48dp"
        android:layout_height="48dp"
        android:tag="@string/jio_native_icon_layout" />
```

android:tag="***@string/jio_native_icon_layout***" is used to place the CTA label for Ad.

Title

```
<TextView
        android:id="@+id/jio_ads_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="2dp"
        android:ellipsize="end"
        android:gravity="center"
        android:maxLines="1"
        android:paddingStart="2dp"
        android:paddingEnd="1dp"
        android:tag="@string/jio_native_title"
        android:textColor="@color/jio_black"
        android:textSize="10sp"
        android:textStyle="bold"
        tools:ignore="SmallSp" />
```

android:tag="***@string/jio_native_tilte***" is used to place the CTA label for Ad.

Description

```
<TextView
            android:id="@+id/jio_tv_desc_cs"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="2dp"
            android:ellipsize="end"
            android:maxLines="2"
            android:paddingTop="1dp"
            android:paddingRight="4dp"
```

```
android:tag="@string/jio_native_description"
android:textColor="@color/jio_black"
android:textSize="12sp"
android:visibility="visible" />
```

android:tag="***@string/jio_native_description***" is used to place the CTA label for Ad.

CTA Button

```
<Button
android:id="@+id/jio_cta_cs"
style="@style/style_jio_button"
android:layout_width="wrap_content"
android:layout_height="35dp"
android:layout_gravity="center_horizontal"
android:background="@drawable/jio_button"
android:focusable="true"
android:gravity="center"
android:tag="@string/jio_native_cta" />
```

android:tag="***@string/*** *jio_native_cta*" is used to place the CTA label for Ad.

Sponsored

```
<TextView
android:id="@+id/jio_ads_sponspored"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />
```

Complete XML for reference

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="300dp"
   android:layout_height="250dp"
   android:background="@drawable/jio_nativeborder"
   android:orientation="vertical">

   <RelativeLayout
      android:id="@+id/fb_media_view"
      android:layout_width="300dp"
      android:layout_height="157dp"
      android:layout_alignParentTop="true"
      android:layout_marginLeft="1.5dp"
      android:layout_marginRight="1.5dp"
      android:descendantFocusability="afterDescendants"
      android:focusable="true"
      android:gravity="center"
      android:tag="@string/jio_native_media_layout"
      android:visibility="visible" />


      <TextView
         android:id="@+id/jio_ads_title_cs"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:ellipsize="end"
         android:maxLines="1"
         android:tag="@string/jio_native_title"
         android:textColor="@color/jio_black"
         android:textSize="12sp"
```

```
    android:textStyle="bold" />

<LinearLayout
    android:id="@+id/jio_cta_container"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_alignParentEnd="true"
    android:gravity="center_vertical"
    android:orientation="vertical">


    <Button
        android:id="@+id/jio_cta_cs"
        style="@style/style_jio_button_infeed"
        android:layout_width="wrap_content"
        android:layout_height="35dp"
        android:layout_gravity="center_horizontal"
        android:background="@drawable/jio_button"
        android:focusable="true"
        android:gravity="center"
        android:tag="@string/jio_native_cta" />
</LinearLayout>


<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/jio_tv_title_cs"
    android:layout_marginTop="2dp"
    android:layout_toLeftOf="@+id/jio_cta_container"
    android:orientation="horizontal">

    <FrameLayout
```

```
        android:id="@+id/native_icon_layout"
        android:layout_width="48dp"
        android:layout_height="48dp"
        android:tag="@string/jio_native_icon_layout" />

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_marginStart="8dp"
        android:layout_marginEnd="8dp"
        android:layout_weight="1"
        android:orientation="vertical">

        <TextView
            android:id="@+id/jio_ads_desc_cs"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="2dp"
            android:ellipsize="end"
            android:maxLines="2"
            android:paddingTop="1dp"
            android:paddingRight="4dp"
            android:tag="@string/jio_native_description"
            android:textColor="@color/jio_black"
            android:textSize="12sp"
            android:visibility="visible" />

    </LinearLayout>

  </LinearLayout>


</RelativeLayout>
```

```
<TextView
    android:id="@+id/jio_ads_sponspored"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

</RelativeLayout>
```

Interstitial Format

Caching Interstitial Ad

This section covers integration steps required to fetch Interstitial ad.

Making a Cache Request for Ad.

```
JioAdView jioAdViewInterstitial = new JioAdView (this,"<ADSPOT_KEY_GOES_HERE>",
JioAdView.AD_TYPE.INTERSTITIAL); // context pass here should be Activity context.

jioAdViewInterstitial.setAdListener(new JioAdListener() {

        @Override
        public void onAdFailedToLoad(JioAdError jioAdError,JioAdView jioAdView) {
        // Error Callback
        }

        @Override
        public void onAdPrepared(JioAdView jioAdView) {
        // Success callback
        }

        @Override
        public void onAdClosed(JioAdView jioAdView,oolean isVideoCompleted, oolean
        isEligibleForReward) {
        // When Ad is closed
        }

        @Override
        public void onAdRender(JioAdView jioAdView){
        // When ad is rendered on screen
        }

        @Override
        public void onAdMediaEnd(JioAdView jioAdView)          {
```

```
        // When media ended
        }

        });

jioAdViewInterstitial.cacheAd();
```

Showing Interstitial Ad

When loadAd() API is called, SDK will try to render the ad inside the container shared.

```
jioAdViewInterstitial.loadAd();
```

Customizing Interstitial Ads

This section covers customizing interstitial ads. Ads can be image or video.

Below API need to be called to enable customization.

```
jioadView. setCustomInterstitialAdContainer(portraitLayoutId: Int, landscapeLayoutId: Int, adCategory: Int)
```

Add required elements with their tags to support individual functionality.

Media Layout

```
    <RelativeLayout
        android:id="@+id/media_view"
        android:layout_width="match_parent"
        android:layout_height="240dp"
        android:clickable="true"
        android:gravity="center"
        android:tag="@string/jio_native_media_layout" />
```

Video Layout

```
<RelativeLayout
    android:id="@+id/fl_video_container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_centerInParent="true"
    android:layout_gravity="center"
    android:tag="@string/jio_video_player_container" />
```

Video Close Button

```
<TextView
    android:id="@+id/iv_close_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_margin="10dp"
    android:drawableRight="@drawable/jio_close_advertisement_video"
    android:gravity="center_vertical"
    android:tag="@string/jio_video_skip_element"
    android:text="@string/you_can_skip_ad_in_skip_counter"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:textColor="@android:color/white"
    android:visibility="gone" />
```

Mute/Unmute Button

```
<ImageView
    android:id="@+id/iv_sound_unmute_button"
    android:layout_width="30dp"
    android:layout_height="30dp"
```

```
    android:layout_marginStart="10dp"
    android:layout_marginTop="10dp"
    android:background="@drawable/jio_mute"
    android:contentDescription="@string/jio_video_volume_icon"
    android:src="@drawable/jio_unmute"
    android:tag="@string/jio_video_volume_icon"
    android:visibility="gone" />
```

Play/Pause Button

```
<ImageView
    android:id="@+id/adPlayback"
    android:layout_width="30dp"
    android:layout_height="30dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentBottom="true"
    android:layout_centerVertical="true"
    android:layout_marginStart="10dp"
    android:layout_marginBottom="10dp"
    android:background="@drawable/jio_vast_play"
    android:contentDescription="@string/jio_video_playback_icon"
    android:src="@drawable/jio_vast_pause"
    android:tag="@string/jio_video_playback_icon"
    android:visibility="gone" />
```

Progress Layout

```
<LinearLayout
    android:id="@+id/progressLayout"
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="5dp"
    android:layout_marginBottom="10dp"
    android:layout_toEndOf="@id/adPlayback">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="@string/ad_text_with_colon"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:textColor="@android:color/white"
        android:textSize="12sp" />

    <TextView
        android:id="@+id/progressCount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:contentDescription="@string/jio_video_progresscount_down"
        android:padding="5dp"
        android:tag="@string/jio_video_progresscount"
        android:text="@string/_00_00"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:textColor="@android:color/white"
        android:textSize="12sp" />
  </LinearLayout>
```

Complete XML for reference

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/rootLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/black">

    <RelativeLayout
        android:id="@+id/fl_video_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_centerInParent="true"
        android:layout_gravity="center"
        android:tag="@string/jio_video_player_container" />

    <TextView
        android:id="@+id/iv_close_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_margin="10dp"
        android:drawableRight="@drawable/jio_close_advertisement_video"
        android:gravity="center_vertical"
        android:tag="@string/jio_video_skip_element"
        android:text="@string/you_can_skip_ad_in_skip_counter"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:textColor="@android:color/white"
        android:visibility="gone" />
```

```
<ImageView
   android:id="@+id/iv_sound_unmute_button"
   android:layout_width="30dp"
   android:layout_height="30dp"
   android:layout_marginStart="10dp"
   android:layout_marginTop="10dp"
   android:background="@drawable/jio_mute"
   android:contentDescription="@string/jio_video_volume_icon"
   android:src="@drawable/jio_unmute"
   android:tag="@string/jio_video_volume_icon"
   android:visibility="gone" />

<ImageView
   android:id="@+id/adPlayback"
   android:layout_width="30dp"
   android:layout_height="30dp"
   android:layout_alignParentStart="true"
   android:layout_alignParentBottom="true"
   android:layout_centerVertical="true"
   android:layout_marginStart="10dp"
   android:layout_marginBottom="10dp"
   android:background="@drawable/jio_vast_play"
   android:contentDescription="@string/jio_video_playback_icon"
   android:src="@drawable/jio_vast_pause"
   android:tag="@string/jio_video_playback_icon"
   android:visibility="gone" />

<LinearLayout
android:id="@+id/progressLayout"
android:layout_width="wrap_content"
android:layout_height="30dp"
android:layout_alignParentBottom="true"
android:layout_marginStart="5dp"
```

```
    android:layout_marginBottom="10dp"
    android:layout_toEndOf="@id/adPlayback">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="@string/ad_text_with_colon"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:textColor="@android:color/white"
        android:textSize="12sp" />

    <TextView
        android:id="@+id/progressCount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:contentDescription="@string/jio_video_progresscount_down"
        android:padding="5dp"
        android:tag="@string/jio_video_progresscount"
        android:text="@string/_00_00"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:textColor="@android:color/white"
        android:textSize="12sp" />

</LinearLayout>

</RelativeLayout>
```

Dynamic Display Ad Format

Caching Dynamic Display Ad

This section covers integration steps required to fetch Interstitial ad.

Making a Cache Request for Ad.

```
JioAdView jioAdViewDynamicDisplay = new JioAdView (this,"<ADSPOT_KEY_GOES_HERE>",
JioAdView.AD_TYPE.DYNAMIC_DISPLAY); // context pass here should be Activity context.

jioAdViewDynamicDisplay.setAdListener(new JioAdListener() {

        @Override
        public void onAdFailedToLoad(JioAdError jioAdError,JioAdView jioAdView) {
        // Error Callback
        }

        @Override
        public void onAdPrepared(JioAdView jioAdView) {
        // Success callback
        }

        @Override
        public void onAdClosed(JioAdView jioAdView,oolean isVideoCompleted, oolean
        isEligibleForReward) {
        // When Ad is closed
        }

        @Override
        public void onAdRender(JioAdView jioAdView){
        // When ad is rendered on screen
        }

        @Override
        public void onAdMediaEnd(JioAdView jioAdView)          {
        // When media ended
        }

        });
jioAdViewDynamicDisplay.setDisplayAdSize(Constants.DynamicDisplaySize.<ACTUAL_SIZE>)
jioAdViewDynamicDisplay.cacheAd();
```

Supported Display Sizes

Use below API to set dynamic display ad size

```
jioAdViewDynamicDisplay.setDisplayAdSize(Constants.DynamicDisplaySize)

//Possible values for DynamicDisplaySize

// Constants.DynamicDisplaySize.SIZE_320x50
// Constants.DynamicDisplaySize.SIZE_300x50
// Constants.DynamicDisplaySize.SIZE_300x250
// Constants.DynamicDisplaySize.SIZE_728x90
// Constants.DynamicDisplaySize.SIZE_320x100
// Constants.DynamicDisplaySize.SIZE_300x600
// Constants.DynamicDisplaySize.SIZE_970x90
// Constants.DynamicDisplaySize.SIZE_970x250
//  Constants.DynamicDisplaySize.SIZE_160x00
```

Showing Instream Video Ad

When loadAd() API is called, SDK will try to render the ad inside the container shared.

```
Framelayout displayAdContainer =(FrameLayout)findViewById(R.id.displayAdContainer);

if(jioAdViewDynamicDisplay.getAdState() == JioAdView.AdState.PREPARED) {

        // Providing a container to display ads to show.
        displayAdContainer.addView(jioAdViewDynamicDisplay);

        jioAdViewDynamicDisplay.loadAd();

}
```

Handling Display Ad object on Activity life cycle

Call below snippet to destroy the display ad object. Usually this is called in onDestroy() of activity.

```
@Override
protected void onDestroy() {
    if (jioAdViewDynamicDisplay!= null) {
        jioAdViewDynamicDisplay.onDestroy();
    }
super.onDestroy();
}
```

Customizing Dynamic Display Ads

This section covers customizing dynamic display ads. Ads can be image or video.

Below API need to be called to enable customization.

jioAdViewDynamicDisplay.setCustomDisplayAdContainer(int nativeContainer, int videoContainer)

Example-
jioAdViewDynamic.setCustomDisplayAdContainer(R.layout.custom_dynamic_display_layout, R.layout.custom_dynamic_video_layout);

Add required elements with their tags to support individual functionality.

Note

1. In above API pass custom layout to 1st argument to customize image ads and pass 2nd argument to customize video ads.

2. For complete details (TAG and example) to customize image-based ads please refer Customizing Native Banner Ads section (page no-30-34) or Native Billboard ads section (page no-35-40)

3. For complete details (TAG and example) to customize video-based ads please refer Customizing Video Ads section (page no-17-26)

Audio Instream Format

Caching Instream Audio Ad

This section covers integration steps required to fetch Instream audio ad.

Making a Cache Request for Ad.

```
JioAdView jioAdViewInstreamAudio = new JioAdView (this,"<ADSPOT_KEY_GOES_HERE>",
JioAdView.AD_TYPE.INSTREAM_AUDIO); // context pass here should be Activity context.

jioAdViewInstreamAudio.setAdListener(new JioAdListener() {

        @Override
        public void onAdFailedToLoad(JioAdError jioAdError,JioAdView jioAdView) {
        // Error Callback
        }

        @Override
        public void onAdPrepared(JioAdView jioAdView) {
        // Success callback
        }

        @Override
        public void onAdClosed(JioAdView jioAdView,oolean isVideoCompleted, oolean
        isEligibleForReward) {
        // When Ad is closed
        }

        @Override
        public void onAdRender(JioAdView jioAdView){
        // When ad is rendered on screen
        }

        @Override
        public void onAdMediaEnd(JioAdView jioAdView)          {
        // When media ended
        }

        });

jioAdViewInstreamAudio.cacheAd();
```

Showing Instream Audio Ad

When loadAd() API is called, SDK will try to play audio ad.

```
If(jioAdViewInstreamAudio.getAdState() == JioAdView.AdState.PREPARED) {
        jioAdViewInstreamVideo.loadAd();
}
```

Showing Instream Audio Companion Ad

```
Framelayout audioCcompanionAdContainer
=(FrameLayout)findViewById(R.id.companionAdContainer);

  // Providing a container to companion ads to show.
   jioAdViewInstreamAudio.setAudioCcompanionContainer(audioCompanionAdContainer,
<COMPANION_AD_SIZE>,DEFAULT_PORTRAIT_IMAGE,DEFAULT_LANDSCAPE_IMAGE)
```

Supported Audio Companion Ad Sizes

Use below API to set dynamic display ad size

```
jioAdViewDynamicDisplay.setDisplayAdSize(Constants.DynamicDisplaySize)

//Possible values for DynamicDisplaySize

// Constants.CompanionAdSize.SIZE_320x480
// Constants.CompanionAdSize.SIZE_480x320
// Constants.CompanionAdSize.SIZE_300x250
// Constants.CompanionAdSize.SIZE_240x260
// Constants.CompanionAdSize.SIZE_320x184
// Constants.CompanionAdSize.SIZE_1024x768
// Constants.CompanionAdSize.SIZE_768x1024
// Constants.CompanionAdSize.SIZE_1920x1080
//  Constants.CompanionAdSize.SIZE_1080x1920
```

Handling Instream Audio Ad object destroy

Call below snippet to destroy the instream Audio ad object.

```
If (jioAdViewInstreamVideo!= null) {
        jioAdViewInstreamAudio.onDestroy();
 }
```

Additional API for Instream Audio Ad

Use the below APIs to pause or resume Instream Audio Ad

```
//to pause Instream Audio Ad
jioAdViewInstreamAudio pauseAd();

//to show all the video controls
jioAdViewInstreamAudio.resumeAd();
```

Supporting APIs by SDK

Enable SDK Logs

Below snippet helps to enabled / disable SDK logging.

```
JioAds.getInstance().setLogLevel(LogLevel logLevel);

//Possible values for logLevel
// JioAds.LogLevel.NONE  - No logs will be printed

// JioAds.LogLevel.DEBUG -To enable debug logs
```

Passing App Meta Data

Below snippet will help passing app meta-data in request.

Passing data globally for all adspot within the app.

```
HashMap<String,String> oolean =new HashMap<>();

customData.put("key1","value1");

customData.put("key2","value2");

JioAds.getInstance().setMetaData(oolean);
```

Passing data for specific AdView within the app.

```
HashMap<String,String> oolean=new HashMap<>();
oolean.put("key1","value1");
oolean.put("key2","value2");
jioAdView. setMetaData(oolean);
```

Enable Media Caching

Section covers to enable or disable media caching of creatives. SDK will try to download image \ video both creatives on usage of right API.

onAdPrepared() callback will be triggered once the media is cached and onAdReceived() callback is triggered once ad is received but media creatives aren't cached.

API need to be called before calling cacheAd().

```
jioAdView.enableMediaCaching(JioAds.MediaType cacheMode);
//Default is cache disabled
```

JioAds.MediaType.IMAGE	1. SDK to download Image creatives during cacheAd() call. 2. Applicable for Native ads.
JioAds.MediaType.VIDEO	1. SDK to download video creatives during cacheAd() call. 2. Applicable for Native and Instream ads.
JioAds.MediaType.ALL	1. SDK to download both image and video creatives during cacheAd() call. 2. Applicable for Native and Instream ads.

Clearing Cached Media

SDK tries to auto clean up the cached media within a day. But still if app needs to clear out cached media the below API can be called.

```
JioAds.getInstance().clearCachedMedia(context, JioAds.MediaType);
```

JioAds.MediaType.IMAGE	Clear cached Images
JioAds.MediaType.VIDEO	Clear cached video creatives
JioAds.MediaType.ALL	Clear all creatives

Request Refresh Rate for Ads

App can set the custom refresh rate for each adspot. Default is 30 secs.

```
jioAdView.setRefreshRate(int timeout_value_in_secs);
```

Bitrate for Video Ads.

SDK tries to play video of bitrate lesser or equal to the value specified in this API. In case there are no video with <= bitrate specified then closest highest will be selected.

```
jioAdView.setVideoBitRate(int videoBitRate);
```

Set Request Dampening Limit

SDK tries to limit request in case when server responds No-Fill multiple times. Internally SDK will try to gradually auto-increment the refresh rate to higher value. API provides a limit to the max value. Default is 1440 mins.

```
jioAdView.setDampeningLimit(long limit_in_mins);
```

Request Timeout

App can set the custom timeout for request call. Default is 20 secs.

```
jioAdView.setRequestTimeout(int timeout_value_in_secs);
```

Media Timeout

App can set the custom timeout for media fetching. Default is 20 secs.

```
jioAdView.setMediaTimeout(int timeout_value_in_secs);
```

Set Skip Event Key

Applicable for STB platform. Flexibility to change the keycode basis which the Skip button can be managed.

```
jioAdView.setSkipEventKey(int KeyCode);
```

Custom UserAgent

App can set user agent using below API

```
JioAds.getInstance().setCustomUserAgent (String customUserAgent);
```

Enabling Custom Show Ad

SDK allows publisher to show ads on their own by using custom implementation. This feature is available for Video and Native Ads. This can be used with following variations:

- Regular Native Ads
- Regular Instream Video Ads
- Instream Video Ads with Finite duration
- Instream Video Ads with Infinite duration

Steps to implement

1. Initialize JioAdView Object as per requirement:

i. For Regular Native Ads:

```
JioAdView jioAdView = new JioAdView(this,"<Ad spot ID Goes Here>",
        JioAdView.AD_TYPE.<Any Native AD_TYPE>);
```

ii. For Instream Video Ads:

```
JioAdView jioAdView = new JioAdView(this,"<Ad spot ID Goes Here>",
        JioAdView.AD_TYPE.INSTREAM_VIDEO);
```

2. Add AdListener on jioAdView. Sample code as below:

```
jioAdView.setAdListener(new JioAdListener() {
    @Override
    public void onAdFailedToLoad (JioAdView jioAdView, JioAdError
                    jioAdError) {

            //This callback is called when any error occurs while fetching ads

    }

    @Override
    public void onAdPrepared (JioAdView jioAdView) {

    //This callback is called when next ad is ready to be consumed
    //When this callback is received following API can be called to fetch next ad
            jioAdView. fetchNextAdData ();

    }

    @Override
    public void onAdDataPrepared (JioAd oole, oolean isLastAd) {

      //This API is called:
        1. When loadCustomAd() is called and data is ready.
        2.  When fetchNextAdData () is called.
     //If requested duration is exhausted isLastAd will come as true
     //otherwise it will come as false. For Native ad it will be true
     //Always as native ad will be delivered once.  Please refer
    //point 3 to check how to use JioAd object

    }
```

```
        @Override
        public void onAllAdsExhausted () {
                // In case server does not have any more ads to serve when
        jioAdView. fetchNextAdData() is called, this callback will be given
        }
 });
```

3. Once the implementation is done and JioAd object is obtained to show ad, while showing ads appropriate event tracking URLs should be fired by using get APIs available by yourself or should be instructed to SDK to fire event tracking URLs using AdEventTracker.

APIs available with JioAd

JioAd objects have following methods which can be used to access data to show custom ads

1. getAdCategory() :
   It gives type of JioAd NATIVE or VIDEO.
2. getAdId():
   It gives Ad Id.
3. getMetadata():
   It gives metadata/ad data in format of JSONObject.
4. getVideoAd():
   When ad category is VIDEO, this method can be called to get VideoAd object, which gives complete information of video ad to be played.
5. getNativeAd():
   When ad category is NATIVE, this method can be called to get NativeAd object, which gives complete information of native ad to be played.
6. getAdEventTracker():
   If you want SDK to handle network calls for event tracking urls, you can request to get AdEventTracker object and use APIs available to instruct SDK to fire particular events.

APIs available with NativeAd

1. getCampaignId():
   Returns String for campaign Id
2. getTitle():
   Returns String for title
3. getDescription():
   Returns String for Description 1

4. getDescription2():
   Returns String for Description 2
5. getCtaText():
   Returns String for CTA text
6. getCtaUrl():
   Returns String for CTA/Click Url. This can be Deeplink url or http/https url.
7. getFallbackLink():
   Returns String for fallback click http/https url
8. getIconImage():
   Returns String for icon image url
9. getMainImage():
   Returns String for main image url
10. getCustomImage():
    Returns String for required custom image url as per size if publisher has defined custom size using setCustomImageSize() api on JioAdView else returns null
11. getCustomImages():
    Returns HashMap<String,String> for all the custom image urls available
12. getObjective():
    Returns String for objective text
13. getSponsored():
    Returns String for Sponsored text
14. getLikes():
    Returns String for count of likes.
15. getSalePrice():
    Returns String for Sale Price
16. getPhone():
    Returns String for phone
17. getAddress():
    Returns String for Address
18. getDisplayUrl():
    Returns String for display url
19. getPrice():
    Returns String for price
20. getRating():
    Returns String for ratings
21. isNativeVideoAd():
    Returns oolean value stating if native ad contains native video ad or not
22. getVideo():
    Returns back String of Vast XML if ad is native video
23. getVideoData():
    Returns back JioAd.VideoAd object for native video. Then using APIs available for VideoAd, data can be fetched to show native video.

24. getImpressionTrackers():
    Returns List<String> containing list of trackers to be fired to track impression
25. getClickTrackers():
    Returns List<String> containing list of trackers to be fired to track ad click

APIs available with VideoAd

1. getAdSystem():
   Returns String for ad system
2. getId():
   Returns String for Vast Ad Id
3. getTitle():
   Returns String for title of video ad
4. getDescription():
   Returns String for description
5. getDuration():
   Returns String for duration in seconds
6. getSkipOffset():
   Returns String for skip offset of Video
7. getClickThroughUrl():
   Returns String for click through url
8. getMedia():
   Returns back VideoAd. Media object selected on basis of device/network type & bitrate.
   Following APIs are available with media object to access data:
   a. *getMediaUrl()*: Returns string for url of selected video media
   b. *getMediaType()*: Returns string for type of selected video ad
   c. *getDuration()*: Returns Long for duration of video ad in seconds
   d. *getBitrate()*: Returns Int for selected bitrate
   e. *getHeight()*: Returns Int for height of selected video
   f. *getWidth()*: Returns Int for width of selected video
9. getImpressionTrackers():
   Returns List<String> containing urls to be fired for tracking impressions for selected video ad.
10. getClickTrackers():
    Returns List<String> containing urls to be fired for tracking click for selected video ad.
11. getErrorTrackers():
    Returns List<String> containing urls to be fired for tracking error for selected video ad.
12. getTrackingEventsUrls():
    Returns HashMap<String,List<String>> containing list of tracking urls mapped with corresponding event type for all available events.
    keys for events will be like below:

```
interface VastTrackingEvents {
    companion object {
        const val EVENT_EXIT_FULLSCREEN = “exitFullscreen”
        const val EVENT_IMPRESSION = “impression”
        const val EVENT_START = “start”
        const val EVENT_FIRST_QUARTILE = “firstQuartile”
        const val EVENT_MID_QUARTILE = “midpoint”
        const val EVENT_THIRD_QUARTILE = “thirdQuartile”
        const val EVENT_COMPLETE = “complete”
        const val EVENT_MUTE = “mute”
        const val EVENT_UNMUTE = “unmute”
        const val EVENT_PAUSE = “pause”
        const val EVENT_RESUME = “resume”
        const val EVENT_FULLSCREEN = “fullscreen”
        const val EVENT_CLOSE = “close”
        const val EVENT_SKIP = “skip”
    }
}
```

13. getTrackingEventUrls(String <VastTrackingEvents>):
    Returns List<String> containing tracking urls for event passed as argument.

APIs available with AdEventTracker

1. trackImpression():
   This API will fire impression tracking urls. Should be called when ad is first visible.

Companion Ads

JIO Ads SDK provides facility to implement Companion ad experience for Instream Video/Audio ads. For Instream video ads, companion can be rendered for Finite and infinite ad pod variants too.

Companion ads can be integrated in two ways

Sync companion experience: This is default behaviour which SDK provides where companion ad will be in Sync with Primary ad(Instream ad) i.e companion will be rendered on video/audio start & it will be closed as soon as video/audio ends.

Async companion experience: SDK provides facility to override behaviour for companion ad rendition. SDK provides an interface **CompanionEventReceiver.kt** which provides all the required callbacks to implement necessary behavior.

Integration steps:

Step 1: Add below progurad rules:

```
-keep public class com.jio.jioads.adinterfaces.JioCompanionListener {
   public <fields>;
   public <methods>;
}

-keep public class com.jio.jioads.adinterfaces.CompanionEventReceiver {
   public <fields>;
   public <methods>;
}

-keep public class com.jio.jioads.companionads.JioCompanionCache{
public <fields>;
   public <methods>;
}

-keep public class com.jio.jioads.companionads.CompanionManager{
public <fields>;
   public <methods>;
}
```

Step 2: Registering companion Ad Slots:

Companion slot can be registered basis on what approach developer has chosen. The approach is different for sync & async experience.
Registering companion for Sync experience

1) Create a JioAdView object with appropriate adspot id. Please note this AdSpot id will be same as what is linked as AdSlot Id with Primary Ad (Instream video/audio).

```
JioAdView companionAdView
=JioAdView(this,"ADSPOT",JioAdView.AD_TYPE.DYNAMIC_DISPLAY
);
```

2) Use below API to declare JioAdView as companion slot

```
companionAdView. setAsCompanion(true);
```

3) Implement JioCompanionListener to listen to Companion ad events as shown below

```
comapnionAdview.setCompanionAdListener(object : JioCompanionListener {

   override fun onCompanionRender() {

      companionContainer.addView(comapnionAdview)

    }

   override fun onCompanionClose() {

   companionContainer.removeAllViews()
   }
})
```

Registering companion for Async experience

1) Implement CompanionEventReceiver interface as and implement its methods

```
public class CloseAfterDelayCompanion implements CompanionEventReceiver
{

void doShow(String masterAdViewId, String adSlotId){}
 void doClose(String masterAdViewId, String adSlotId){}
}
```

2) Create an object of above class and register it with CompanionManager of JIO Ads SDK as shown below

```
val companionContainer =
findViewById<FrameLayout>(R.id.companion_Ad_container)

val closeAfterDelayCompanion =
CloseAfterDelayCompanion(companionContainer, this)
CompanionManager.instance?.registerCompanionView(
   companionAdspot,
   closeAfterDelayCompanion
)
```

Sample implementation of Class implementing CompanionEventReceiver interface

```
class CloseAfterDelayCompanion(private val viewGroup: ViewGroup, private val
context: Context) :
   CompanionEventReceiver {


   override fun showCompanionAd(masterAdViewId: String?, adSlotId: String?) {

      if (masterAdViewId != null) {
         if (adSlotId != null) {
            CompanionManager.instance?.attachCompanionAd(
               context,
               viewGroup, masterAdViewId, adSlotId
            )
         }
      }
   }

   override fun onCompanionClosed() {

         }
```

Step 3: Registering primary Ad Slot:

Developer is supposed to set below API to their respective primary JioAdView object to enable Companion ad feature.

```
val jioAdView = JioAdView(this, primaryAdspot, adformat)
jioAdView.setAsPrimary(true)
```

Step 4: Implement JioAdListener for Primary adspot & call cacheAd() and loadAd() as shown below

```
jioAdView.setAdListener(object : JioAdListener() {
   override fun onAdFailedToLoad(jioAdView: JioAdView?, jioAdError: JioAdError?)
{}
   override fun onAdClosed(jioAdView: JioAdView?,
      isVideoCompleted: Boolean,
      isEligibleForReward: Boolean) {

   }

   fun onAdClosed(jioAdView: JioAdView?, isVideoCompleted: Boolean) {

   }

   override fun onAdPrepared(jioAdView: JioAdView?) {
      jioAdView!!.loadAd()
   }

   override fun onAdRender(jioAdView: JioAdView?) {

   }

   override fun onAdMediaEnd(jioAdView: JioAdView?) {

   }
})
      jioAdView.cacheAd()
```

Creating Hybrid AdSlot for companion ads

For Sync companion implementation, JIO Ads Sdk provides facility to declare Companion slot as Hybrid, Where same companion slot will be used to display regular dynamic banner/billboard ads with configured refresh rate. Here companion ad will be shown till the time primary ad is playing otherwise SDK will render regular dynamic ads. Please note SDK provides this Facility for Sync companion experience.

Sample snippet of implementing Companion Ad view object as Hybrid Adslot

```
val companionContainer =
findViewById<FrameLayout>(R.id.companion_Ad_container)
```

```
val comapnionAdview = JioAdView(this, companionAdspot,
JioAdView.AD_TYPE.DYNAMIC_DISPLAY)
comapnionAdview.setAsCompanion(true)

comapnionAdview.setCompanionAdListener(object : JioCompanionListener {

   override fun onCompanionRender() {
      companionContainer.removeAllViews()
      companionContainer.addView(comapnionAdview)
      Log.e("merc", "onCompanionRender")
   }

   override fun onCompanionClose() {
      companionContainer.visibility = View.GONE
   }
})

comapnionAdview.setAdListener(object : JioAdListener() {
   override fun onAdFailedToLoad(jioAdView: JioAdView?, jioAdError: JioAdError?)
{

   }

   override fun onAdClosed( jioAdView: JioAdView?,
      isVideoCompleted: Boolean,isEligibleForReward: Boolean ) {

   }

   override fun onAdPrepared(jioAdView: JioAdView?) {
      jioAdView?.loadAd()
   }
   override fun onAdRender(jioAdView: JioAdView?) {

   }
   override fun onAdMediaEnd(jioAdView: JioAdView?) {

   }

});
var arrayData: ArrayList<Constants.DynamicDisplaySize> = ArrayList()
arrayData.add(Constants.DynamicDisplaySize.SIZE_160x600)
arrayData.add(Constants.DynamicDisplaySize.SIZE_300x250)
arrayData.add(Constants.DynamicDisplaySize.SIZE_320x50)

comapnionAdview.setDisplayAdSize(arrayData.toList())
comapnionAdview.cacheAd();
```

Requesting VMAP

The IAB Video Multiple Ad Playlist (VMAP) specification is an XML template that video content owners can use to describe the structure for ad inventory insertion. In order to effectively monetize video content with in-stream insertion advertising, video content owners must carefully manage the structure and use of ad inventory opportunities available within their content.

With VMAP, video content creators can define the following:

- Ad breaks within their content
- Timing for each break
- How many breaks are available

VMAP was designed to be used in conjunction with VAST and is well-suited for video content creators who have no control over the video player, but want to control the ad experience within the videos. This method of ad scheduling is supported via VAST and Google IMA plugins

Fetch VMAP URL

Use the below code snippet to fetch VMAP URL

```
private fun loadVmap(vmapId:String,packageNAme:String)
{
   val jioAdsLoader =JioAdsLoader(vmapId ,this //context,object
:JioAdsLoaderListener

   {
      override fun onAdsUrlLoaded(vmapUrl: String) {
         e("onAdsUrlLoaded:: "+vmapUrl)

         initializePlayer(vmapUrl)

      }

      override fun onAdsLoadingError(jioAdError: JioAdError?) {
      }

   })
   jioAdsLoader.contentVideoLength=100
   val cuePoints=ArrayList<Long>()
  cuePoints.add(10)
  cuePoints.add(20)
  cuePoints.add(40)
  cuePoints.add(60)
  jioAdsLoader.contentVideoCuePoints=cuePoints
   jioAdsLoader.getVmap()
}
```

AdMob Mediation

Step 1: The AdMob adapter for display ad is already a part of JioAds SDK you just need to add below Google Mobile ads library in your project.

```
Implementation 'com.google.android.gms:play-services-ads:16.0.0'
```

Step 2: Add the following permission in AndroidManifest.xml

```
<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

IMA Mediation

Step 1: Create a package com.jio.jioads.mediation.partners in your app and add the file GoogleIMA.java shared along with JioAds library.

Step 2 : Include the below dependency in your app level build.gradle file.

```
implementation 'com.google.ads.interactivemedia.v3:interactivemedia:3.24.0'
```

Step 3: Add all the proguard rules mentioned in page no 15 in your project proguard-rules.pro file.

PRISM Ads

Caching Prism Ad

Step 1: Create the object of the JioAdView Class and pass ad type as PRISM.

```
JioAdView jioAdView = JioAdView(this, adspotId, JioAdView.AD_TYPE.PRISM)
```

Step 2: Assign the listener to the JioAdView Object.

```
jioAdView?.setAdListener(object : AdListener(this) {

   override fun seeAllAdReceived(jioAdView: JioAdView?) {
      super.seeAllAdReceived(jioAdView)
    }

   override fun onAdFailedToLoad(jioAdView: JioAdView?, jioAdError: JioAdError?) {
      super.onAdFailedToLoad(jioAdView, jioAdError)
    }

   override fun onAdPrepared(jioAdView: JioAdView?) {
      super.onAdPrepared(jioAdView)
   }

   override fun onAdClicked(jioAdView: JioAdView?) {
      super.onAdClicked(jioAdView)
   }

   override fun onAdRender(jioAdView: JioAdView?) {
      super.onAdRender(jioAdView)
   }

   override fun onAdRefresh(jioAdView: JioAdView?) {
      super.onAdRefresh(jioAdView)
   }

   override fun onAdMediaExpand(jioAdView: JioAdView?) {
      super.onAdMediaExpand(jioAdView)
   }

   override fun onAdMediaCollapse(jioAdView: JioAdView?) {
      super.onAdMediaCollapse(jioAdView)
   }

   override fun onAdMediaStart(jioAdView: JioAdView?) {
      super.onAdMediaStart(jioAdView)
```

```
  }

  override fun onAdReceived(jioAdView: JioAdView?) {
    super.onAdReceived(jioAdView)
  }

  override fun onAdSkippable(jioAdView: JioAdView?) {
    super.onAdSkippable(jioAdView)
  }

  override fun onMediaPlaybackChange(jioAdView: JioAdView?, mediaPlayBack:
JioAdView.MediaPlayBack?) {
    super.onMediaPlaybackChange(jioAdView, mediaPlayBack)
  }

  override fun onAdClosed(jioAdView: JioAdView?, isVideoCompleted: Boolean,
isEligibleForReward: Boolean) {
    super.onAdClosed(jioAdView, isVideoCompleted, isEligibleForReward)
  }

  override fun onAdMediaEnd(jioAdView: JioAdView?) {
    super.onAdMediaEnd(jioAdView)
  }
})
```

3. Set the image width and height to render.

Then call initPrism(contentId) - This method will download the slot information available for video, so better to call this api at the start of the video

```
jioAdView?.setCustomImageSize(width.toInt(), height.toInt())
jioAdView?.initPrism(contentId!!)
```

Shopping Icons Visibility

Use the below api to hide/show a shopping icon.

```
jioAdView?.addShoppingIconControlListener(simpleExoPlayer!!, object :
OnShoppingIconStateChangeListener {
    override fun onShoppingStateChanged(shouldShow: Boolean) {
        if (shouldShow) {
            iv_shooping_icon?.visibility = View.VISIBLE
        } else {
            iv_shooping_icon?.visibility = View.GONE
        }
    }
})
```

At the end of video or if your UI is destroying don't forget to remove listener

```
jioAdView?.removeShoppingIconControlListener()
```

Loading Prism Ad

Use below api to load prism ads and pass

1. customContainer - if you need custom card design
2. videoPausedTime - paused time of video
3. MediaCaching - pass true if caching is enabled
4. customData - metadata in hashmap

```
jioAdView?.loadPrismAds(customContiner, videoPausedTime, true, customData)
```

See All

Use below api to see all card in grid layout

```
jioAdView?.seeAllAds(R.id.see_all_container)
```

Use below api to dismiss grid layout

```
jioAdView?.dismissSeeAllXrayAds()
```

Callbacks

After calling load ad api you will get jioAdView object consisting of ads into onAdPrepared()

```
override fun onAdPrepared(jioAdView: JioAdView?) {
    super.onAdPrepared(jioAdView)

    if (adContainer != null) {
        adContainer?.removeAllViews()
        adContainer!!.visibility = View.VISIBLE
        adContainer?.addView(jioAdView)
    }
}
```

Close Prims Ad

Use below api to close the Prism

```
jioAdView?.closePrismAd()
```