Android SDK Documentation

Posit

In this document

1. Overview

About Posit API

The Posit API allows video-on-demand platforms to annotate videos and display relevant and contextual information to video consumers. The API can detect the following objects from videos -

1. Clothing
    a. T-shirts
    b. Shirts
    c. Dresses
    d. Trousers
    e. Jackets
    f. Ethnic wear
2. Fashion Accessories
    a. Eyewear
    b. Jewelry
    c. Bags
    d. Watches
    e. Shoes

The API uses the latest catalog uploaded by e-commerce businesses to select and display products that match closely to the ones available on the scene. By displaying relevant information to users while they pause during a video, platforms can make additional revenue by displaying relevant products from the scene in a non-interruptive manner. For more information please contact business@posit.tech

About Posit Android SDK

The Posit Android SDK is an open source library that uses Posit API to help implement the Posit Ad network on Android applications.

Current Stable Version - 1.0.0
Minimum Android SDK Version - API 16, Android 4.1

1. **Quick Integration:** Integration is very light weight and easy (takes only 10 mins).
2. **No sharing of source code**. Neither from your end nor from ours.
3. **We respect your DRM:** We do not take any content to our server. Everything is processed at your end. Your content is not shared with us for pre-processing.
4. **Seamless operations:** Works independent of your existing advertisement module. It does not affect your existing operations.

2. Installation

SDK Setup

Include the dependency in your project using Gradle

```
implementation 'tech.posit.android:posit:0.2.0'

repositories {
    maven {
        url  "https://dl.bintray.com/posit/posit-android-sdk"
    }
}
```

To ensure successful compilation, add the following lines in the build.gradle file under app module -

```
android.useAndroidX=true
android.enableJetifier=true
```

Ensure Exoplayer, AWS SDK and Jwt dependencies are added in your build.gradle file

```
implementation 'com.amazonaws:aws-android-sdk-s3:2.13.+'
implementation ('com.amazonaws:aws-android-sdk-mobile-client:2.13.+@aar') {
transitive = true }

implementation 'com.google.android.exoplayer:exoplayer:2.10.0'

api 'io.jsonwebtoken:jjwt-api:0.10.5'
runtimeOnly 'io.jsonwebtoken:jjwt-impl:0.10.5'
runtimeOnly('io.jsonwebtoken:jjwt-orgjson:0.10.5') {
  exclude group: 'org.json', module: 'json' //provided by Android natively
}
```

Add the following service under the application tag in your manifest file

```
<service
android:name="com.amazonaws.mobileconnectors.s3.transferutility.TransferSer
vice" android:enabled="true" />
```

Permissions

The Posit Android SDK needs the following permissions in order to work. Please add the following lines in your AndroidManifest.xml file. This permission allows the app to connect to network services.

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

Initialization

The SDK has to be initialized with your application keys before it can be used. Please add the relevant lines to your app's base Activity class.

```
import android.app.Application
import tech.posit.android.Posit

class JpApplication : Application() {
    override fun onCreate() {
        super.onCreate()
        val app = this
        // Required initialization logic here!
        val clientId = "jp-tv-007"
        val accessKey = "ae2aff8c-bd9c-11e9-bb3d-560d0e73f093"
        val secretKey = "ae2b01a8-bd9c-11e9-bb3d-560d0e73f093"
        val apiPath = "https://api.posit.tech/testing/"

        Posit.Manager.build(app, clientId, accessKey, secretKey, apiPath)
    }
}
```

Add the following attribute to your ExoPlayerView UI element

```
<com.google.android.exoplayer2.ui.SimpleExoPlayerView
      ...
app:surface_type="texture_view"
      ...
/>
```

3. Solution

Ad Network on Pause Layer

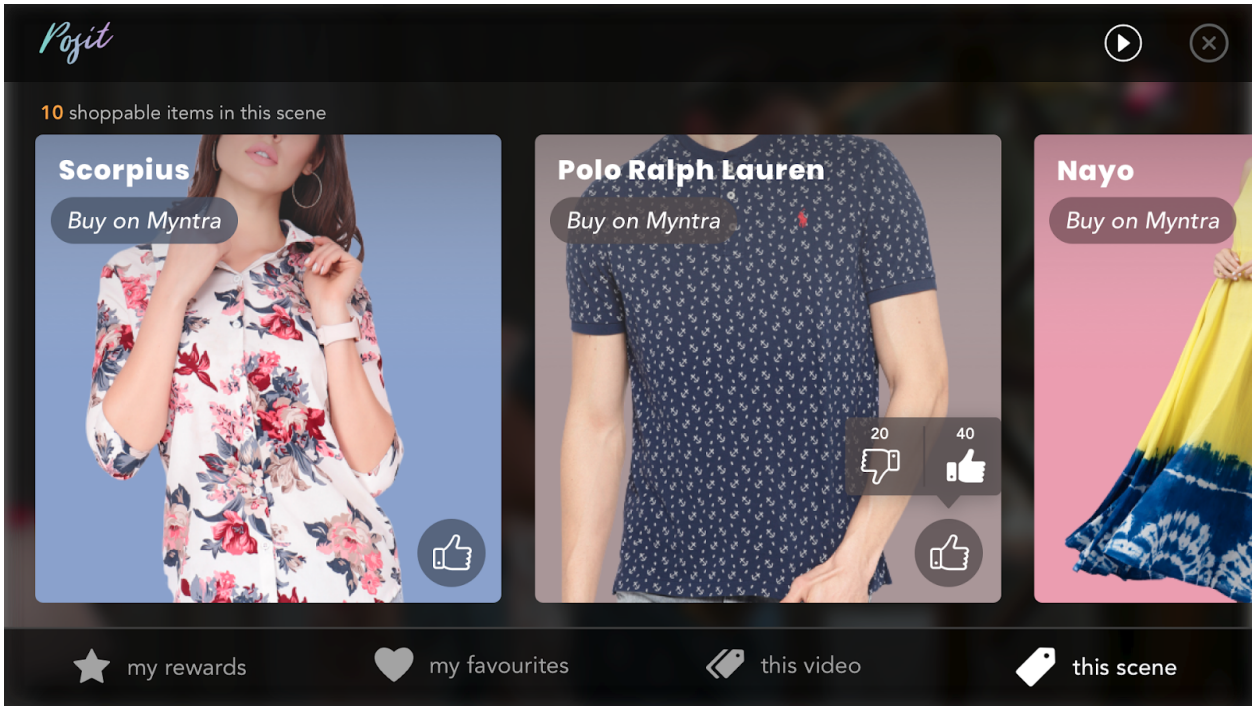

Figure 1: An information layer opens up when the user pauses a video. The layer contains information that is relevant to objects in the scene.

The default implementation displays an information layer on top of the video when the user pauses while watching a video. Figure 1 represents a typical information layer. The user can choose to explore products that are displayed or close the layer to continue watching the video. The current implementation only supports videos playing on “Landscape mode”.

The following actions are enabled by default for the user -

1. Upvote and Downvote products.
2. Browse all products in the current scene.
3. Browse all products in the current video.
4. Browse all the user’s favorite products across the platform.
5. Browse all the user’s rewards across the platform.
6. Redeem rewards.

About Rewards

Apart from being non-interruptive, Posit rewards a user for pausing while watching a video. The reward is based on probability and every pause activity lets a user win cash credits on e-commerce websites. The credits can instantly be redeemed by a user. The default implementation of Posit supports the rewards system. Please see Figure 2 that represents a reward being issued to a user.

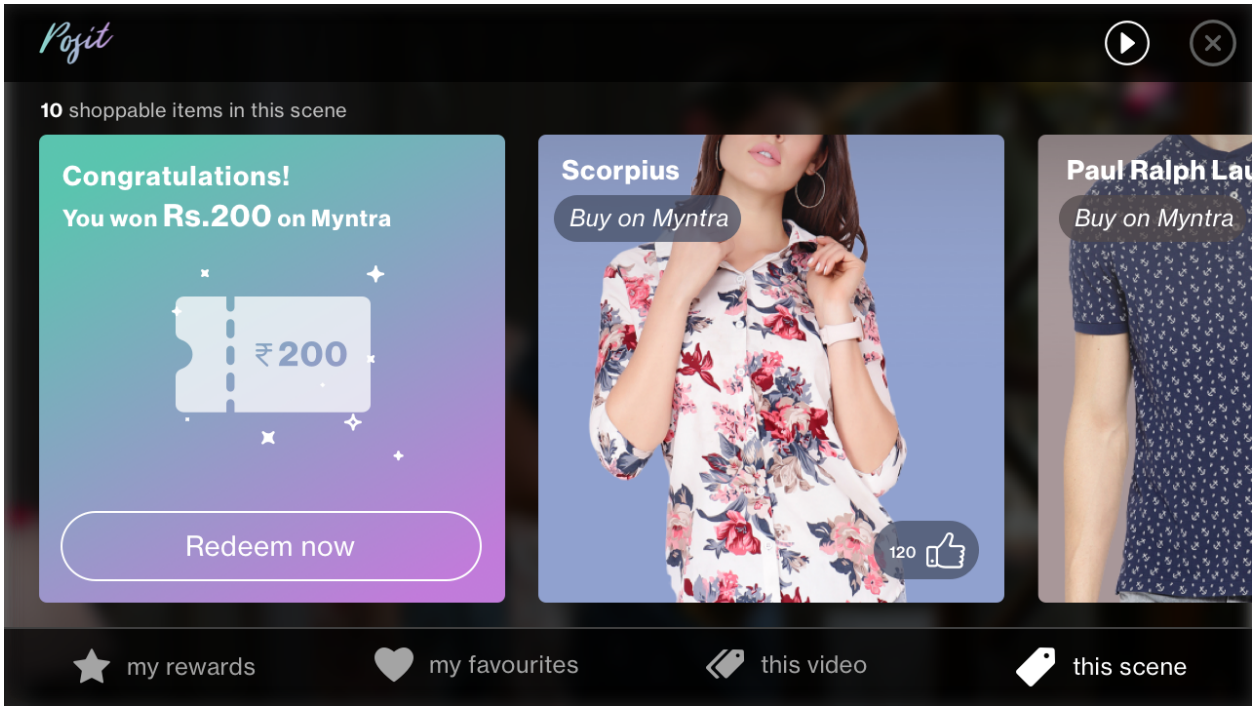

Figure 2: A coupon is awarded to lucky users when they pause. The coupons are stored and can be browsed or redeemed from the "my rewards" section.

Implementation

For the default implementation after creating the Exoplayer instance in your activity, register with Posit like this

```
Posit.register(this, playerView, videoId, fps)
```

videoId = the ID of the video in your database; fps = frames per second of the video

That's it! All your videos now display information when the user pauses. You can enable/disable annotations for certain videos from your developer dashboard.

Posit

Contact

business@posit.tech
Sudha KS - +917755078033