StreamhubAnalytics Android plugin for Kaltura Android applications

28th of September 2020	1.0	- Programs tracking - Ads tracking - Sample app - Sample logs

This document describes the steps required to add and configure the StreamhubAnalytics plugin for ThePlatform Android applications.

Download the plugin and sample application at
https://streamhub-static-content.s3-eu-west-1.amazonaws.com/plugins/kaltura/kaltura-android-plugin.zip

The plugin is composed of the following files and components:

1. **StreamhubAnalyticsKalturaBridge.java** is the bridge between a Kaltura android player and StreamhubAnalytics data tracking abstraction layer
2. **StreamhubAnalytics.java** is the data tracking abstraction layer
3. **NetworkUI.java** is handling Http REST requests and responses

The downloadable .zip file contains:

1. The plugin files
2. A sample Android Kaltura player project that integrates with the StreamhubAnalytics plugin and shows how to initialise it.

Getting Started

Integrate the plugin in your own application

The integration process is straightforward, because the complexity of interacting with our Backend REST API has been abstracted in the plugin logic.

Add the dependency for HTTP I/O requests

We will assume you are using Android studio for this document. If another IDE is being used please just adapt the steps.

1. Open ***build.gradle*** in the Module level
2. Add the following to the “dependencies”

```
implementation 'com.squareup.retrofit2:retrofit:2.9.0'
implementation 'com.squareup.retrofit2:converter-scalars:2.9.0'
```

3. Go to the Android Studio menu and click on File > Sync Project with Gradle Files, and sync the Gradle file

Add the plugin files

Locate the 3 plugin files StreamhubAnalyticsKalturaBridge.java, StreamhubAnalytics.java and NetworkUI.java in the archive and copy them into your Android project.

Initialize the StreamhubAnalytics plugin

In your application code java class, create a class member to hold an instance of the StreamhubAnalyticsTPBridge class.

Then, in an appropriate location of your code, initialize it:

Java

```
new StreamhubAnalyticsKalturaBridge(player, this, "my-player-id", isLive, "logged-user-id",
"my-analytics-id", "my-player-id");
```

Kotlin

```
StreamhubAnalyticsKalturaBridge(
  player,
  this,
  "my-player-id",
  isLive,
  "logged-user-id",
  "my-analytics-id",
  "my-player-id"
)
```

Parameters details

name	type	comments
player	com.kaltura.playkit.Player	An instance of a class that implement the com.kaltura.playkit.Player interface.
context	Context	Activity context.
playerId	String	A unique identifier for your player in your system.
isLive	Boolean	Whether the stream is a live or recorded stream.
userId	String	Optional. If you want to track user, provide the user logged identifier here. Otherwise, pass an empty string “”.
analyticsId	String	Your analyticsId identifier provided by your Streamhub account manager.
playerTitle	String	A user-friendly name associated with the playerId. This information might have already been shared with your account manager during the pre-integration phase.