StreamhubAnalytics Swift plugin for Kaltura iOS applications

21th of Sept 2020	1.0	- Programs tracking - Ads tracking - Sample app - Samples logs

This document describes the steps required to add and configure the StreamhubAnalytics plugin for Kaltura on iOS and tvOS applications.

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

The downloadable .zip file contains:

- Plugin-files: the 4 plugin files to be dropped in your iOS or tvOS app.
- Sample.logs: contains an http.log file that features API requests sent from the test app with the plugin configured.
- Test-app: contains a test app that shows an example of plugin integration and configuration.

Prerequisites

The plugin is written in Swift 4.2. If your application uses a more recent version of the Swift language, then make sure the **Use Legacy Swift Language Version** setting is set to Yes under **Build Settings.**

The plugin has no external dependencies. Just add the plugin files to your app and start collecting logs immediately.

Getting started

The integration is very straightforward, as the complexity of interacting with our Backend REST API is being abstracted by the plugin.

The plugin is written on top on the Kaltura iOS SDK
https://developer.kaltura.com/player/ios/getting-started-ios

Add the plugin files to your application

Drag and drop the 4 files from the uncompressed .zip archive into your Xcode application.

Initialize the StreamhubAnalytics plugin

1. In your application controller where the Kaltura player is initialized, create a property of type StreamhubAnalyticsKalturaBridge to hold an instance of this class. For example;

```
var shBridge: StreamhubAnalyticsKalturaBridge?
```

2. Next, you will need to create an instance of StreamhubAnalyticsKalturaBridge, and supply initialisation parameters to it

```
self.shBridge = StreamhubAnalyticsKalturaBridge(partnerId: PARTNER_ID,
                                  endPoint: "",
                                  playerId: PLAYER_ID,
                                  isLive: IS_LIVE,
                                  userId: USER_ID,
                                  analyticsId: ANALYTICS_ID,
                                  userAgent: USER_AGENT,
                                  playerTitle: PLAYER_TITLE);
```

The table below gives a clarification the parameters:

Parameter name	**Type**	**Description**

partnerId	String	“kaltura”
endPoint	String	You can choose to point to a testing development server during the integration testing phase. Otherwise, leave this field empty and the plugin will setup `http://stats.streamhub.io` which is the production endPoint for submitting stats.
isLive	Boolean	Specify if the video content is either a Live or a VoD program. Default is **false**.
playerId	String	Unique identifier of the player playing the video. It is a unique identifier for the player in both your system and ours. You can generate your own unique identifier or contact your Streamhub account manager to get one generated for you.
userId	String	If your application users are logged, you can use this field to provide any user identifier. This will allow you to get analytics detailed at the user level of your audience.
analyticsId	String	The main tracking code that has been provided to you by your Streamhub account manager.
userAgent	String	Hardcoded user-agent string used to attribute the views to a device category. Please check the sample app for an example of user-agent.
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.

Your Streamhub account manager can help you with the initialisation parameters, if needed.

3. Finally, register the Kaltura player with the StreamhubAnalytics plugin bridge.

```
self.shBridge!.registerPlayer(player: self.player);
```

4. Build and test your application. We recommend that you capture the logs HTTP requests in a similar format as the ones in the **/sample.logs** folder and send those to us for review.

Notes regarding the userAgent parameters

At the time of writing, in native apps you will not be able to retrieve a dedicated user-agent String programmatically.
Therefore, we recommend that you derive the user-agent from one of these examples, depending on which device your are targeting:

AppleTV
com.example.apple-samplecode.TVMLCatalog/1.0 iOS/9.0 model/AppleTV5,3 build/13T5365h.

iPad
iPad; U; CPU OS 9_2 like Mac OS X; en-us; iPad5,3

iPhone
iPhone; CPU iPhone OS 7_0 like Mac OS X