AdsonUI Plug-in APIs

[1. Architecture Overview 1](#_Toc491946746)

[2. UOI Lookup table 1](#_Toc491946747)

[3. enableAdsOnUI() 2](#_Toc491946748)

[4. startAd() 3](#_Toc491946749)

[5.stopAd() 4](#_Toc491946750)

The java swing/AWT application will communicate with the AdsonUI plug-in module only through the following 3 APIs

### Architecture Overview

**JAVA 3rd Party Application**

* Creating JFrame
* Accepting the rendered JPanel from AdsonUI plug-in
* Calling APIs to AdsonUI as per this user manual
* User Navigations, User key acceptance
* Subscriber profile Handling

**APIs (enable AdsonUI, startAd, stopAd)**

**Ads On UI Plug-in**

* Server communication
* Fetching Ad Sequence , Ads
* Caching & storing sequence & Ads
* Rendering Ads on JPanels
* Handling user clicks and displaying AdCompanion

### UOI Lookup table

| **Ad Banner PetName** | **Client\_type** | **Width** | **Height** | **UniqueOppId** |
| --- | --- | --- | --- | --- |
| chList\_Right\_Vertical | PC | 384 | 1026 | 1 |
| EPG | PC | 1900 | 280 | 2 |
| Small\_H\_Banner | Mobile | 320 | 50 | 3 |
| Small\_Logo | Mobile | 300 | 250 | 4 |
| infoBanner | PC | 480 | 243 | 5 |
| programBanner | PC | 250 | 145 | 6 |
| HD\_FullScreen | PC | 1920 | 1080 | 7 |
| game | STB | 1920 | 1080 | 7 |
| Web\_stamp | WebClient | 180 | 280 | 8 |
| MedSize\_Stamp | Mobile | 320 | 480 | 9 |
| VertI\_ChalListSTB | STB | 417 | 540 | 10 |
| VertI\_ChalLockSTB | STB | 417 | 540 | 11 |
| Med\_infoBannerSTB | STB | 365 | 145 | 12 |
| Small\_progrBannerSTB | STB | 296 | 150 | 13 |
| Rect\_EPG\_STB | STB | 580 | 368 | 14 |
| TV\_Logo\_banner | TV | 350 | 280 | 15 |
| TV\_Small\_Stamp | TV | 250 | 250 | 16 |
| MedSize\_TV\_Stamp | TV | 950 | 500 | 17 |
| VerticalSize\_TV\_Banner | TV | 430 | 900 | 18 |
| LShape | WatchTV | 1921 | 1080 | 41 |
| Horizontal | WatchTV | 1900 | 280 | 42 |
| Vertical | WatchTV | 400 | 900 | 43 |

### enableAdsOnUI()

#### 1.1 Prototype

**public** **void** **enableAdsOnUI** (int[] oppIdsArray)

#### Purpose

At the time of launch, the application pass all the Unique Opportunity Ids to the Ads On UI plug-in through this API enableAdsOnUI, so that it can fetch the Ad Sequence from the Server in advance, before user navigates to the UI/App screen.

#### Input Parameter

* **oppIdsArray**

OppIdsArray is an integer array to share all the unique opportunity ids in the UI or App screens , where the app developer wants to insert Ads. The unique Ids are selected by the App developer based on UOI lookup-table provided as part of this API user manual.

#### API Usage

enableAdsOnUI API should be called by the application in the following way:-

static int[] oppIdsArray = { 1, 2, 5, 6, 7 }; // All unique Opp Ids of the Application

AdsOnUIManager. enableAdsOnUI (oppIdsArray);

### 4. startAd()

#### 2.1 Prototype

public Component **startAd**

(int pos\_x,

int pos\_y,

String screenName,

int uniqueOppId )

#### Purpose

Application needs to call *startAd* API in each of the screens where BannerAds need to be displayed. when the App/UI screen becomes visible.

#### Input Parameters

* **pos\_x:** x- co-ordinate position in pixels in the UI/App screen where Ad needs to be displayed
* **pos\_y:** y- co-ordinate position in pixels in the UI/App screen where Ad needs to be displayed
* **screenName:** Name of the App/UI screen where Ad needs to be displayed
* **uniqueOppId:**  opportunity Id of that screen where we have to show ad Banner, based on the UOI lookup table provided as part of this API user manual.

#### Return Value

* The AdsonUi plugin after receiving the ***startAd***  API from the application will create a Jlabel at the position pos\_x,pos\_y of width & height derived from the unique opp id. After creating Jlabel the Ad is displayed in the Jlabel by the AdsonUI plugin. This displayed Ad is returned as a **JLabel component,** which gets added as part of the Jframe by the application.

#### API Usage

*startAd API need to be called separately for each of the opportunities present in the same App /UI Screen .*

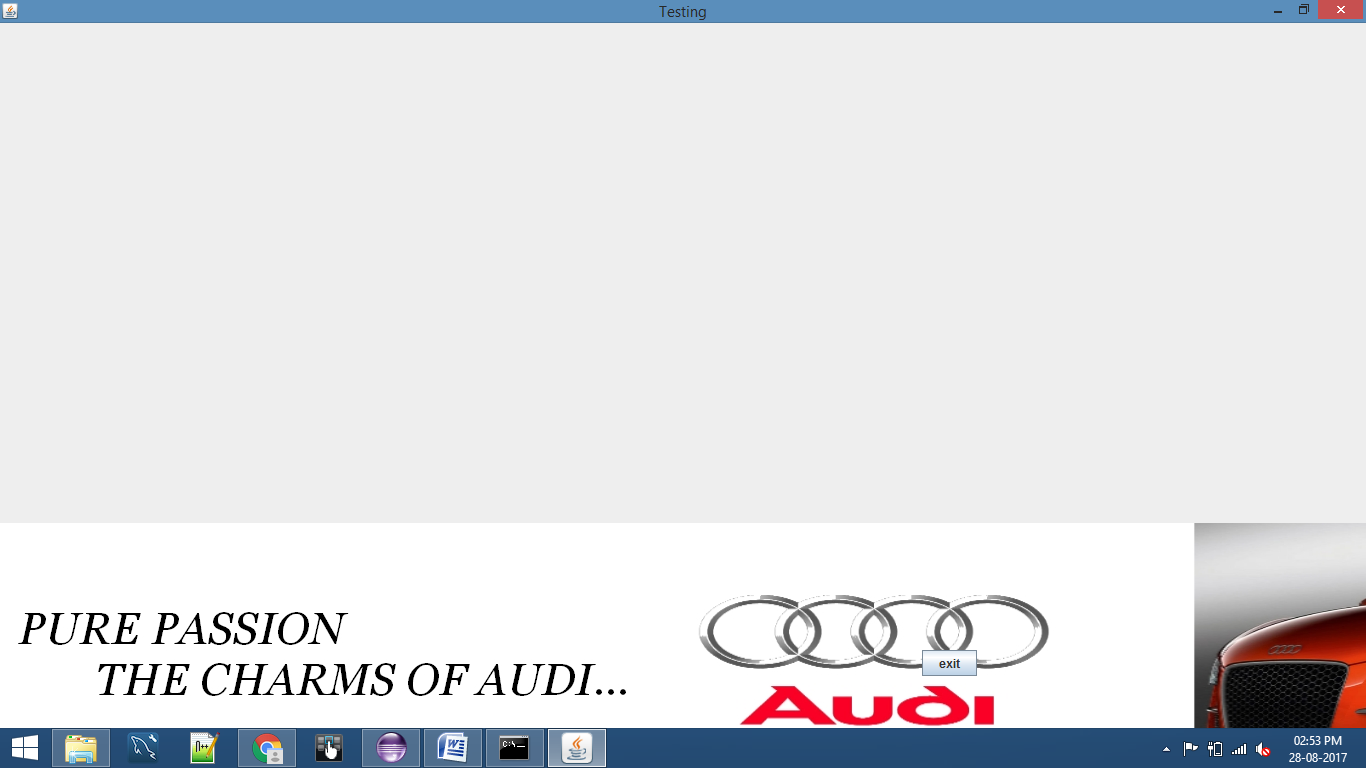
*startAd API should be called by the application in the following way for each opportunity:-*

*ManageAdsOnUI uoiScreenobj=new ManageAdsOnUI();*

frame.add(*uoiScreenobj* .startAd(0,500, “epg\_screenName”,5) );

Here frame is the keyword used for JFrame.

After calling startAd API, ad banner will appear in the EPG Screen as follows:-



*NOTE:-*

In one screen If you want to show more than one ad Banner for different unique Opportunity Ids then you have to call startAd(-,-,-,-) API using diffent object of ManageAdsOnUI class as follows:

*ManageAdsOnUI uoiScreenobj1=new ManageAdsOnUI(); // for first Unique Opp Id of Channel List*

*ManageAdsOnUI uoiScreenobj2=new ManageAdsOnUI(); // for the 2nd Unique Opp Id of Channel List*

add(*uoiScreenobj1*.startAd(1110, 0,"Channel List ",1) );

add(*uoiScreenobj2* .startAd(0,0, “Channel List”,6) );



*uoiScreenobj2*

*uoiScreenobj1*

**Channel List**

**Displaying Ads on Ui component on videos:**

manageAdsOnUI=**new** ManageAdsOnUI();

//Adding the advertisementPanel to Program Banner panel and starting ad

// *advertisementPanel\_XPos* & *advertisementPanel\_YPos* are the Relative Ad panel position from Program Banner panel add(manageAdsOnUI.startAd(ProgramBannerConstants.*advertisementPanel\_XPos*,ProgramBannerConstants.*advertisementPanel\_YPos*, "Program Banner", 6) );



Note:The Ad JLabel component should be added as part of the UI Panel displayed on top of video.

### 5.stopAd()

#### 3.1 Prototype

public void **stopAd** (boolean isScreenVisible);

#### Purpose

Application needs to call stopAd () API when the user navigates away from the current App/UI Screen to another screen

or

when the user quits the App

or

when the user invokes the Adcompanion.

#### Input Parameters

This API should be called with one parameter named as isScreenVisible .this parameter type is Boolean(true/false).

* **isScreenVisible:** Boolean type shared to indicate whether screen is visible or not.

#### Return Value:

* + None.

#### API Usage:

Used to stop the current adBanner and deregister the current *advertisementListener object .*

*stopAd API should be called by the application in the following way:-*

Please use the same UOI object by name uoiScreenobj created at the time of startAd() API call.

*uoiScreenobj. stopAd(false);*

**NOTE**:-

If the user navigates away from a screen which has more than one Ad banner displayed (ex: Channel List), then the application developer needs to call StopAd() twice with visibility parameter as false, using the two specific UOI objects as follows:

Please use the UOI objects uoiScreenobj1 & uoiScreenobj2 created at the time of startAd() API call.

*uoiScreenobj1. stopAd(false);*

*uoiScreenobj2. stopAd(false);*

**Stopping Ads on Ui component from videos when navigating to a new screen:**

//calling stopAd() API from manageAdsOnUI object using program Banner class

programBanner.manageAdsOnUI.stopAd(**false**);

// after that make other UI elements and video Invisible.