



# **AT&T Advertising SDK Developer Guide for Android**

Publication Date: May 28<sup>th</sup> 2014



## Legal Disclaimer

This document and the information contained herein (collectively, the "**Information**") is provided to you (both the individual receiving this document and any legal entity on behalf of which such individual is acting) ("**You**" and "**Your**") by AT&T, on behalf of itself and its affiliates ("**AT&T**") for informational purposes only. AT&T is providing the Information to You because AT&T believes the Information may be useful to You. The Information is provided to You solely on the basis that You will be responsible for making Your own assessments of the Information and are advised to verify all representations, statements and information before using or relying upon any of the Information. Although AT&T has exercised reasonable care in providing the Information to You, AT&T does not warrant the accuracy of the Information and is not responsible for any damages arising from Your use of or reliance upon the Information. You further understand and agree that AT&T in no way represents, and You in no way rely on a belief, that AT&T is providing the Information in accordance with any standard or service (routine, customary or otherwise) related to the consulting, services, hardware or software industries.

AT&T DOES NOT WARRANT THAT THE INFORMATION IS ERROR-FREE. AT&T IS PROVIDING THE INFORMATION TO YOU "AS IS" AND "WITH ALL FAULTS." AT&T DOES NOT WARRANT, BY VIRTUE OF THIS DOCUMENT, OR BY ANY COURSE OF PERFORMANCE, COURSE OF DEALING, USAGE OF TRADE OR ANY COLLATERAL DOCUMENT HEREUNDER OR OTHERWISE, AND HEREBY EXPRESSLY DISCLAIMS, ANY REPRESENTATION OR WARRANTY OF ANY KIND WITH RESPECT TO THE INFORMATION, INCLUDING, WITHOUT LIMITATION, ANY REPRESENTATION OR WARRANTY OF DESIGN, PERFORMANCE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, OR ANY REPRESENTATION OR WARRANTY THAT THE INFORMATION IS APPLICABLE TO OR INTEROPERABLE WITH ANY SYSTEM, DATA, HARDWARE OR SOFTWARE OF ANY KIND. AT&T DISCLAIMS AND IN NO EVENT SHALL BE LIABLE FOR ANY LOSSES OR DAMAGES OF ANY KIND, WHETHER DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, SPECIAL OR EXEMPLARY, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, LOSS OF GOODWILL, COVER, TORTIOUS CONDUCT OR OTHER PECUNIARY LOSS, ARISING OUT OF OR IN ANY WAY RELATED TO THE PROVISION, NON-PROVISION, USE OR NON-USE OF THE INFORMATION, EVEN IF AT&T HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES OR DAMAGES.



# Table of Contents

## Contents

1	Introduction .....	1
2	About the Advertising API.....	2
2.1	Prerequisites .....	2
2.2	Setting up your Android Project .....	2
2.3	Using the ATTAdView Class.....	4
2.3.1	Configuring ATTAdView Properties.....	4
2.3.2	Initializing the ATTAdView Object.....	6
2.3.3	Implementing Success and Failure Callback Methods.....	8
2.3.4	Advertising API Work Flow.....	9
3	Advertising API Reference.....	10
3.1	ATTAdView Methods .....	10
3.2	ATTAdView Properties .....	12
3.2.1	Category Property Values .....	14
3.3	ATTAdViewListener Callbacks .....	16
3.4	ATTAdView JSON Response .....	17
3.5	ATTAdView Error Codes .....	19



## Table of Figures

Figure 2-1: Package Explorer.....	3
Figure 2-2: Advertising API Work Flow .....	9



## Table of Tables

Table 3-1: ATTAdView Methods .....	11
Table 3-2: ATTAdView Properties .....	14
Table 3-3: ATTAdViewDelegate Callbacks.....	16
Table 3-4: JSON Response Parameters .....	17
Table 3-5: ATTAdView Error Codes.....	19



## Table of Examples

Example 2-1: viewDidLoad Method.....	6
Example 2-2: ATTAdViewDelegate Callback Methods.....	8



# 1 Introduction

The AT&T Advertising SDK for Android provides a library, documentation, and sample code that assist developers in building mobile apps using the AT&T Advertising API. The library contains classes and methods for handling authentication and authorization, requests for advertisement content, and rendering the advertising component in the designated frame.

The Advertising API enables your app to support advertisements within the app. This allows the developer of the application to collect a revenue share of the advertisement. When users click the advertisement in the app, they are redirected to the web page for the advertisement.

Android is the software platform for Android Smart Phones, and Tablets. This SDK supports apps that are developed with the Android Software Development Kit. The Android SDK, which is available at <http://developer.android.com>, provides the tools and APIs necessary to develop apps on the Android platform using the Java programming language.

This SDK supports applications that target Android version 2.2 and higher.



## 2 About the Advertising API

The Advertising API is packaged in the static library ATTAdKit. This library exports the ATTAdView class, which contains methods that perform the following tasks.

- Handle authentication and authorization.
- Handle click events in the advertising component.
- Get the advertisement.
- Render the advertisement content to the designated area of your app.

The methods in the library contain parameters that allow you to create an advertising component based on filtering criteria and display it in the designated area of your app. The library components require the App key and secret key that are created when you register your app for the Advertisement API on the [AT&T Developer Program web site](#).

### 2.1 Prerequisites

To develop Advertising API apps for Android, you must have the following prerequisites.

- The Android SDK, version 2.2 or newer.
- A supported IDE, such as Eclipse.

To install the Android SDK and Eclipse IDE, follow the instructions on the Android Developer web site.

In addition to the software required to develop Android apps, the following prerequisites must be met to develop apps using the Advertising API for Android.

- Knowledge of how to program Android applications using Java. If you are not familiar with Android development, see the Android developer documentation.
- The Advertising SDK for Android from the AT&T Developer Program web site.
- The permissions and minimum SDK version must be updated in the manifest for the app.
- The project must contain at least one activity and layout containing the UI elements of the app.

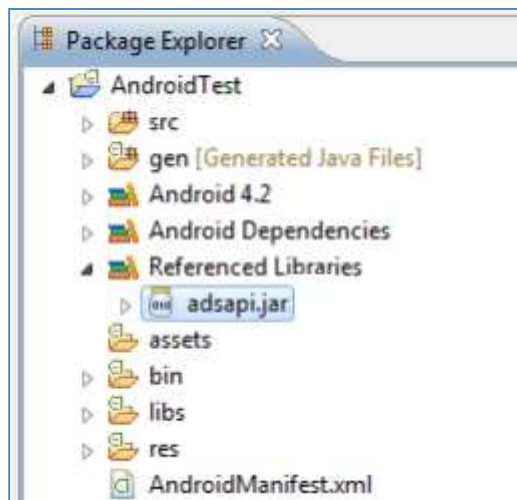
### 2.2 Setting up your Android Project

To incorporate the Advertising API into an Android project in Eclipse, perform the following procedure.





1. Open the Android project for your application in Package Explorer.
2. Select Project, Properties.
3. Select the Java Build Path entry in the navigation bar.
4. In the Libraries tab, click the Add External JARs button on the right side of the properties window.
5. Browse to the folder where you extracted the Advertising files and select the following library: adsapi.jar
6. Click OK. The library should appear in Package Explorer, as shown in the following illustration.



**Figure 2-1: Package Explorer**

7. Double-click AndroidManifest.xml and add the following permissions.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
  android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission
  android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
  android:name="android.permission.ACCESS_NETWORK_STATE" />
```
8. Open the activity and layout that contains the UI elements of your application.
9. Add functionality to your app as described in [Section 2.3](#) Using the ATAdView Class.
10. Build your app that includes the AT&T Advertising SDK.



## 2.3 Using the ATAdView Class

This section describes how to use the ATAdView class in your app, including implementing the listener callback methods.

### 2.3.1 Configuring ATAdView Properties

You must set values for the following properties of the ATAdView class to authenticate and authorize your app to use the Advertising API.

- appKey
- category
- secret
- udid

You can set additional properties in the ATAdView object to filter the advertisement content. These properties include keywords, city, and zip code. For a full list of properties, see [Section 0](#)



ATTAdView Properties.



## 2.3.2 Initializing the ATTAdView Object

The ATTAdView class is a member of the com.att.ads namespace; the ATTAdViewListener class is a member of the com.att.listener namespace. Add the following import statements to your app code to use these classes.

### Import Statements

```
1 | import com.att.ads.*;
2 | import com.att.ads.listeners.*;
```

#### *Example 2-1: Import Statements*

To initialize an ATTAdView object, insert code such as the following in the onCreate method.

### onCreate Method

```
1 | // Create the ATTAdView component
2 | ATTAdView attAdView = new ATTAdView(this, appkey, secret, uidid,
   | category);
3 |
4 | // Optionally, set the properties, such as keywords, to
5 | // filter the ad content
6 | attAdView.setKeywords("some selected keywords");
7 |
8 | // Optionally, set the reload period. This is the interval,
9 | // in seconds, in which the Ad component will be refreshed.
10 | // The default is 2 minutes.
11 | attAdView.setAdReloadPeriod(60);
12 |
13 | //Add the ATT Ad View component to your Layout.
14 | adFrameLayout.addView(attAdView);
15 |
16 | // Set the layout properties for Ad view component
17 | attAdView.setLayoutParams(new
   | ViewGroup.LayoutParams(ViewGroup.LayoutParams.FILL_PARENT, 100));
18 |
19 | // Initialize authentication/authorization, the ad service will
20 | // call and draw the ad component based on the received response.
21 | // Also, this method starts the timer to reload the new Ad when
   | the setAdReloadPeriod interval elapses.
22 | attAdView.initOrRefresh();
```

#### *Example 2-2: onCreate Method*

The Reload timer, whose interval is set by the setAdReloadPeriod method and is initiated by the initOrRefresh method, works as a parallel thread to the activity of the AttAdView object. The Reload timer works only on foreground activity. To



support this functionality, the ATTAdView object overrides the onAttachedToWindow and onDetachedFromWindow methods in the parent android.webkit.WebView class.

This refresh model works for almost all cases. However, in some of the cases for some platforms or devices, the onAttachedToWindow and onDetachedFromWindow call back methods might not be reached. To ensure that the refresh timer works properly, override the onPause and onResume callback methods in the Activity, as shown in the following code example.

#### onPause and onResume Callback Methods

```
1 | @Override
2 | protected void onPause() {
3 |     super.onPause();
4 |     Log.d(TAG, "onPause()");
5 |     if(null != attAdView){
6 |         //Stops the Reload timer and all listeners.
7 |         attAdView.stopRefresh();
8 |     }
9 | }
10 |
11 | @Override
12 | protected void onResume() {
13 |     super.onResume();
14 |     Log.d(TAG, "onResume()");
15 |     if(null != attAdView){
16 |         //Starts the Reload timer and all listeners.
17 |         attAdView.startRefresh();
18 |     }
19 | }
```

**Example 2-3: onPause and onResume Callback Methods**



### 2.3.3 Implementing Success and Failure Callback Methods

To handle the success or failure when your app requests an advertisement, override the `onSuccess` and `onFailure` methods of the `ATTAdViewListener` class, as shown in the following code example.

#### onSuccess and onFailure Methods

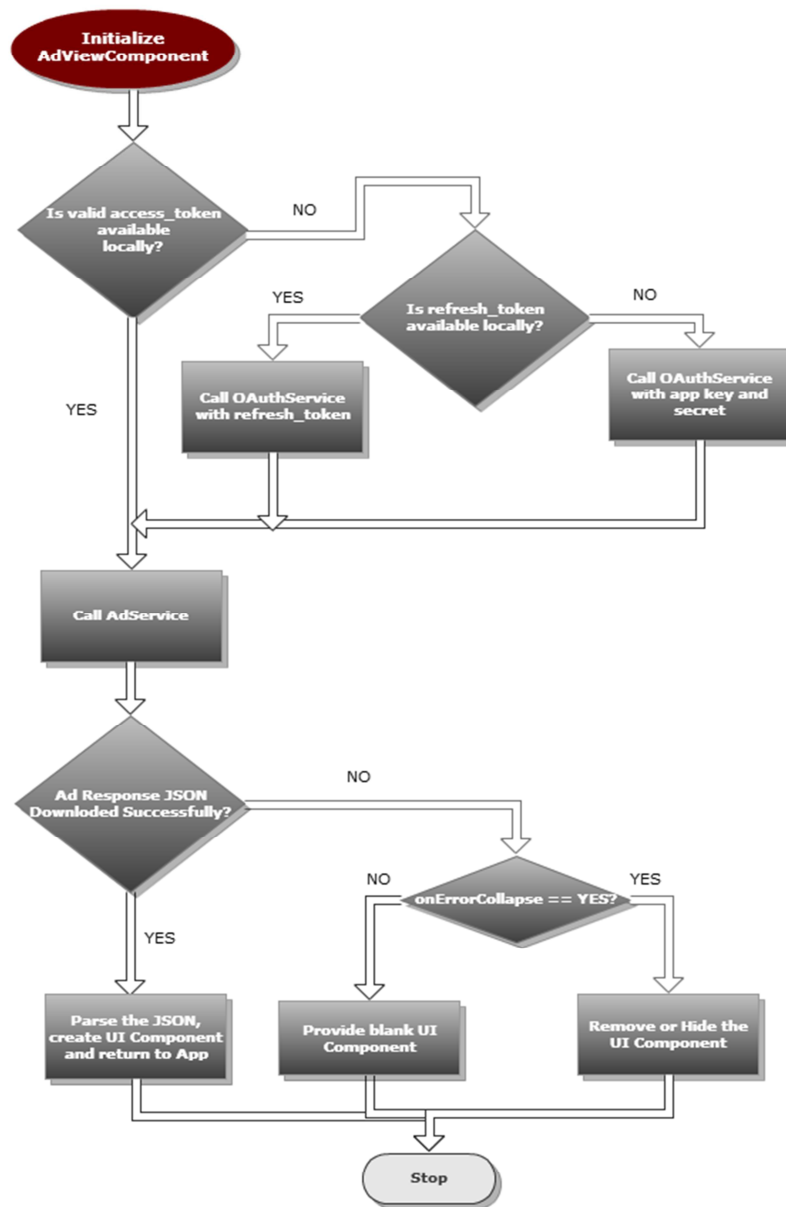
```
1 | adview.setAdViewListener(new ATTAdViewListener() {
2 |
3 |     public void onSuccess(String adViewResponse) {
4 |         Log.d(TAG, "onSuccess():"+adViewResponse);
5 |     }
6 |
7 |     public void onError(ATTAdViewError error) {
8 |         StringBuffer res = new StringBuffer();
9 |         res.append(error.getType());
10 |        res.append(": ");
11 |        res.append(error.getMessage());
12 |        Exception e = error.getException();
13 |        if(null != e){
14 |            res.append("\n Exception :\n ");
15 |            res.append(e);
16 |        }
17 |        Log.e(TAG, "onError(): "+res.toString());
18 |    }
19 | });
```

**Example 2-4: onSuccess and onFailure Methods**



## 2.3.4 Advertising API Work Flow

The following illustration shows the work flow of the Advertising API.



**Figure 2-2: Advertising API Work Flow**



## 3 Advertising API Reference

This section provides reference information on the `ATTAdView` and `ATTAdViewDelegate` classes.

### 3.1 `ATTAdView` Methods

The following table describes the methods in the `ATTAdView` class.

Method	Result Type	Required/Optional	Description
<code>ATTAdView(Context context, String appKey, String secret, String udid, String category)</code>	<code>ATTAdView</code>	Required	Creates and initializes the advertisement component using the following parameters: <ul style="list-style-type: none"><li>• A reference to the context of the Activity.</li><li>• The application key of the publisher.</li><li>• The application secret of the publisher.</li><li>• The udid of the publisher.</li><li>• The preferred advertising category of the publisher.</li></ul>
<code>initOrRefresh()</code>	<code>void</code>	Required	Initializes or refreshes the advertisement view content.
<code>setAdViewListener(ATTAdViewListener adViewListener)</code>	<code>void</code>	Optional	Registers an <code>AdViewListener</code> delegate allowing the app to monitor success or failure when an advertisement is loaded.
<code>setAdReloadPeriod(int adReloadPeriod)</code>	<code>void</code>	Optional	Sets the interval at which the Reload timer will refresh the advertisement content. The minimum reload time is 30 seconds.
<code>stopRefresh()</code>	<code>void</code>	Required	Stops the Reload timer, which stops the loading of advertisement content.
<code>startRefresh()</code>	<code>void</code>	Required	Starts the Reload timer, which triggers the refreshing of the advertisement content at the interval specified by the <code>adReloadPeriod</code> parameter in the <code>setAdReloadPeriod</code>





Method	Result Type	Required/ Optional	Description
			method.

***Table 3-1: ATAdView Methods***



## 3.2 ATTAdView Properties

The following table describes the properties in the ATTAdView class.

Property	Default Value	Type	Required/Optional	Description
Udid		String	Required	Specifies the UDID that is provided by the developer. Must be at least 30 characters in length. If it is less than 30 characters, an error will be returned.
Category		NSString	Required	Specifies the category filter for the advertisement. The value of this property filters the advertisement content based on one of the following categories listed in <a href="#">Section 3.2.1 Category Property Values</a> .
Gender		NSString	Optional	Indicates the gender of the audience demographic. The supported values are M for male and F for female.
zipCode		NSInteger	Optional	Specifies the USA Zip code of the app user.
areaCode		Integer	Optional	Specifies the USA area code of app user.
City		String	Optional	Specifies the USA city and state of the user.
Country		String	Optional	Specifies the country code of the user. When this value is specified, it overrides the country detected by IP. Valid values are the ISO 3166 country codes.
Longitude		double	Optional	Specifies the current longitude, in degrees, of the geographical position for the mobile device.



Property	Default Value	Type	Required/ Optional	Description
Latitude		double	Optional	Specifies the current latitude, in degrees, of the geographical position for the mobile device.
maxHeight		Integer	Optional	Specifies the maximum height of the advertisement content to be displayed. The value of this property sets the maximum height of the advertisement content when the advertisement is requested from the server. The height of the content can be less than or equal to this value.
maxWidth		Integer	Optional	Specifies the maximum width of the advertisement content to be displayed. The value of this property sets the maximum width of the advertisement content when the advertisement is requested from the server. The width of the content may be less than or equal to this value.
minHeight		Integer	Optional	Specifies the minimal height of the advertisement content to be displayed. The value of this property sets the minimum height of the advertisement content when the advertisement is from the server. The height of the content may be greater than or equal to this value
minWidth		Integer	Optional	Specifies the minimum width of the advertisement content to be displayed. The value of this property sets the minimum width of the advertisement content when the



Property	Default Value	Type	Required/ Optional	Description
				advertisement is requested from the server. The width of the content may be greater than or equal to this value.
Timeout	1000	Integer	Optional	Specifies the amount of time, in milliseconds, that the user is willing to wait for a response. The maximum value is 3000ms (three seconds) and the default value is 1000ms (one second).
ageGroup		NSString	Optional	The age group of the demographic audience of the app. The defined values of this parameter are: <ul style="list-style-type: none"><li>• 1-13</li><li>• 14-25</li><li>• 26-35</li><li>• 36-55</li><li>• 55-100</li></ul>
Keywords		String	Option	Specifies the keywords that are used to filter the ads. The values are not case-sensitive and multiple values must be separated by commas. For example, to filter for ads about music, tv, or games, use "music,tv,games".
Type		String	Option	Specifies the type of advertisement being returned. The only acceptable value for this parameter is: <ul style="list-style-type: none"><li>• 3: Text or image.</li></ul> Example: Type=3

**Table 3-2: ATAdView Properties**

### 3.2.1 Category Property Values

The valid values for the Category property include the following.



- Auto
- Business
- Finance
- Chat
- Community
- Social
- Personals
- Communication
- Technology
- Games
- Health
- Medical
- Maps
- Local
- Entertainment
- Movies
- TV
- Music
- Photos
- Video
- News
- Weather
- Shopping
- Sports
- Tools
- Travel
- Other



### 3.3 ATTAdViewListener Callbacks

By default, the response to an advertisement request is handled by the listener methods of the `ATTAdViewListener` class. You must implement these methods to either handle the returned advertisement or handle an error.

The following table describes these methods.

Callback Method	Return	Required/ Optional	Description
<code>onSuccess(String adViewResponse)</code>	void	Optional	This method is invoked when the advertisement content is received successfully and renders properly. The method will have a reference to the raw <code>JSONResponse</code> received from Advertisement service.
<code>onError(ATTAdViewError error)</code>	void	Optional	This method is invoked when the advertisement content fails to load or cannot be authenticated. The method will have a reference to the <code>ATTAdViewError</code> object that has the error information.

***Table 3-3: ATTAdViewDelegate Callbacks***



### 3.4 ATTAdView JSON Response

The following table describes the parameters returned in the JSON response to an ATTAdView method.

Parameter	Type	Required/ Optional	Description
AdsResponse	Object	Required	Container for the advertisements.
Ads	Object	Required	This JSON object contains the detailed information of the advertisement.
ClickUrl	String	Required	Specifies the web site to which the user is sent if they click the advertisement on their device. For SMS advertisements, the URL is shortened to between 35 and 40 characters.
Content	String	Optional	Contains the content of the advertisement from the third party.
Text	String	Optional	Contains any textual representation of the advertisement.
TrackUrl	String	Optional	Contains the pixel tracking URL.
Type	String	Required	Specifies the type of advertisement.

**Table 3-4: JSON Response Parameters**



A JSON response from the ATTAView method might look something like the following:

ATTAView Method JSON Response	
1	{
2	"AdsResponse": {
3	"Ads": {
4	"Type": "thirdparty",
5	"ClickUrl" :
	http://ads.advertising.bf.sl.attcompute.com/1/redirect/21707",
6	"TrackUrl": "http://bos-tapreq25.jumtap.com/a30/r/bos-
	tapreq25/13499/L",
7	"Text": "",
8	"Content": "<a
9	ref="http://ads.advertising.bf.sl.attcompute.com/1/redirect/6dea9/0/2
	21707">
10	</a>
12	
14	}
15	}
16	}

**Example 3-1: ATTAView Method JSON Response**





## 3.5 ATTAdView Error Codes

When the `adViewListener.onError` method is invoked, the app can determine what happened by examining the properties of the `ATTAdViewError` argument.

The `adViewListener.onError` method receives errors from a variety of sources, such as Advertising service downstream errors, OAuth downstream errors, network errors, HTTP error results, and errors reported by the operating system. The different types of errors are encapsulated into an `ATTAdViewError` object, so each source of error corresponds to a value of the `errorCode` property and each distinct error from a source corresponds to the error message of the `errorMessage` property.

The following table describes the possible values of the `errorCode` property that the `adViewListener.onError` method has to handle.

Error Code	Description
-10	Returned when the OAuth service is unable to make a successful response. The corresponding information for this error is returned in the <code>getMessage</code> and <code>getException</code> methods of the <code>ATTAdViewError</code> object.
-20	Returned when the Advertising API service is unable to make a successful response. The corresponding information for this error is returned in the <code>getMessage</code> and <code>getException</code> methods of the <code>ATTAdViewError</code> object.
-30	Returned when the device was unable to connect to the Internet to make service calls. The corresponding information for this error is returned in the <code>getMessage</code> method of the <code>ATTAdViewError</code> object.
-40	Returned when any of the parameter values was either not supplied or was invalid. The corresponding information for this error is returned in the <code>getMessage</code> method of the <code>ATTAdViewError</code> object.
-50	Returned when any other error occurs, such as unable to set or get the values from the <code>SharedPreferences</code> of the device. The corresponding information for this error is returned in the <code>getMessage</code> method of the <code>ATTAdViewError</code> object.

**Table 3-5: ATTAdView Error Codes**