

# Android SDK v1.0 beta 4

Release date: **2011.01.10**

Security classification: **PUBLIC**

## Getting Started

### About Example

Launched in July 2010, Example is the **first** dedicated Android app store for the gaming industry and the premier destination for players discover, download and manage the best gaming apps. Example provides a trusted and secure branded app store that makes it easy for operators to distribute and promote their real money gaming apps, and for players to discover, download and play all of the games they love. Once players install Example and download their favorite real-money apps, management, updates and deposits are seamless. Example was founded to create the first Android ecosystem specifically for gaming, and Example's executive team has deep roots in gaming and mobile industries. The company is privately held with offices in London, UK and Silicon Valley, USA.

### Purpose

This Example Android SDK provides the necessary instructions, code examples, and tools enabling Partner developers to integrate Example technologies into their Android gaming apps.

### Audience

This SDK is intended for use by Example Partner developers, engineers, and integrators who work with the Google Android APK application build process.

### System architecture overview

The following figure shows a high-level view of Example's system architecture and work flow. For your reference, several key components described below are numbered in the figure.

#### Back-end services:

##### 1 - Example Core System

The Example Core System is the main controller managing user activities and provides the integration with the casino's back-end servers.

##### 2 - Example Transaction System

The Example Transaction System is responsible for all payment-related activities, including management of payment processing interactions and user payment methods.

#### Client distributables:

##### 3 - Example Store Application

The Example Store Application (.apk) will be installed by consumers onto their Android devices. It allows a consumer to browse, search for, install, and update casino gaming applications. It requires the consumer to create a *user account* which is registered with the Example Core System, and provide a *payment method* which is registered with the Example Transaction System. The payment method is validated through the Bank System and stored in the Transaction System.

##### 4 - Example Android SDK

The Example Android SDK is integrated into your individual casino games. It provides the integration with the Example Store Application and payment services for your applications.

# System Architecture

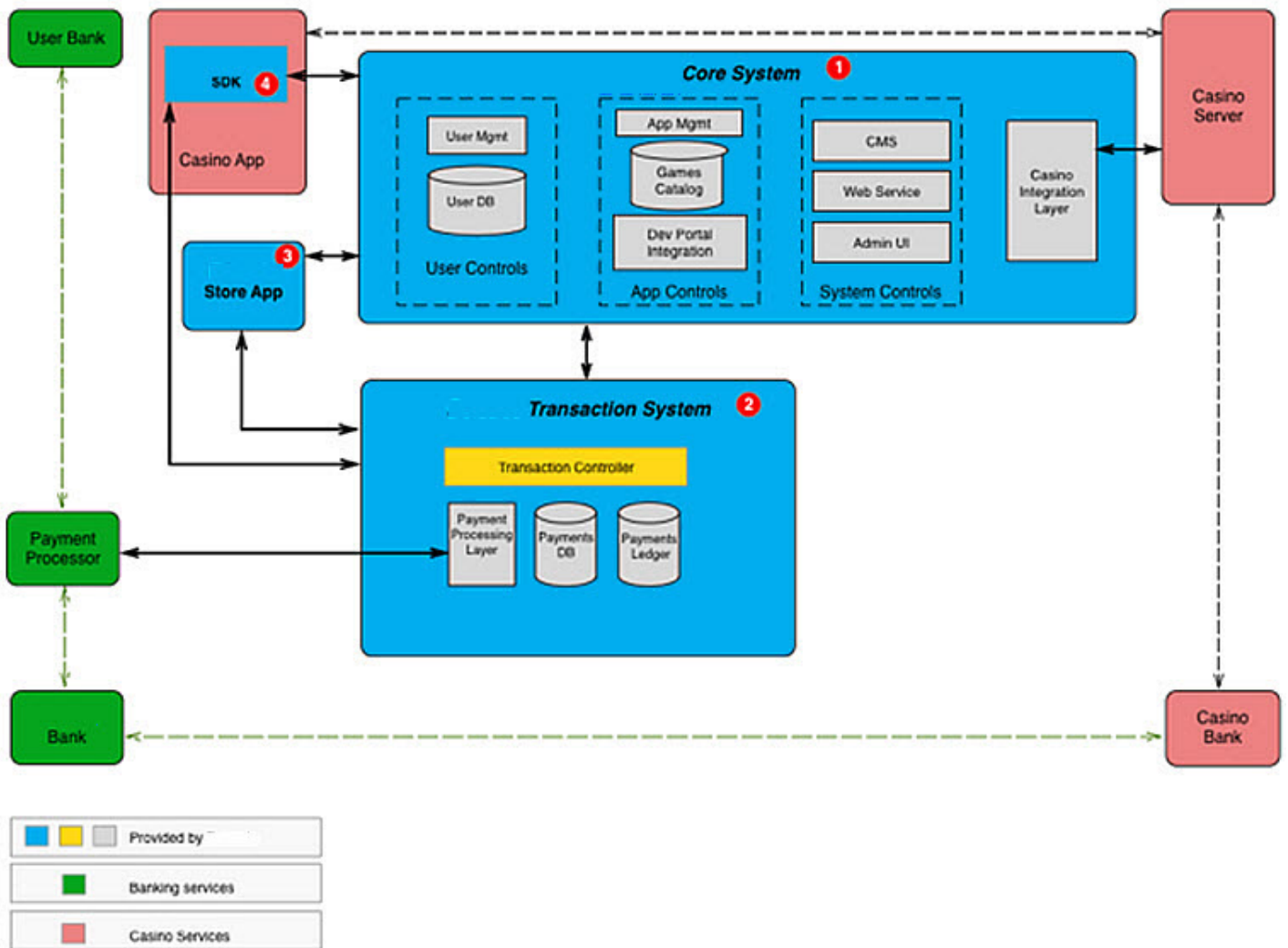


Figure: A high-level view of the Example system architecture

## Integration using Android Studio

Example provides SDK integration instructions in this document collection for developers using the Google Android Studio IDE.

- Please see the SDK Integration page in this SDK.
- Download Google's Android Studio IDE (<http://developer.android.com/sdk/index.html>)

## Android build reference

For your reference, extensive information regarding Google's Android build process can be found here:

- Getting Started with Android Studio (<http://developer.android.com/develop/index.html>)
- Android APK Workflow (<http://developer.android.com/tools/workflow/index.html>)
- Android User Interface (UI) (<http://developer.android.com/guide/topics/ui/overview.html>)

## SDK services

The **Example** Java Class along with its accompanying Interfaces and Methods provides service data and information for the SDK. Please see these documented in the Java Class Library page.

## Android operating systems support

The Example Android SDK provides support for the following Android OS versions:

### Minimum supported version:

Android 4.0 - Ice Cream Sandwich (API level 14)

Reference: Android Codenames, Tags, and Build Numbers (<https://source.android.com/source/build-numbers.html>)

## User permissions

The Example Android SDK requires specific user permissions to operate effectively. Android applications should add the following permissions to their user permission model.

### Geolocation (<http://developer.android.com/reference/android/webkit/GeolocationPermissions.html>)

Example requires geolocation to provide location-relevant search results.

### Internet (<http://developer.android.com/reference/android/Manifest.permission.html#INTERNET>)

Required to open network sockets.

### Access Network State ([http://developer.android.com/reference/android/Manifest.permission.html#ACCESS\\_NETWORK\\_STATE](http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_NETWORK_STATE))

Required to access information about networks.

### Access Wifi State ([http://developer.android.com/reference/android/Manifest.permission.html#ACCESS\\_WIFI\\_STATE](http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_WIFI_STATE))

Required to access information about Wifi networks.

### Read Phone State ([http://developer.android.com/reference/android/Manifest.permission.html#READ\\_PHONE\\_STATE](http://developer.android.com/reference/android/Manifest.permission.html#READ_PHONE_STATE))

Required to get Device specific details such as device id, OS, etc.

### Read External Storage (Read/Write)

([http://developer.android.com/reference/android/Manifest.permission.html#READ\\_EXTERNAL\\_STORAGE](http://developer.android.com/reference/android/Manifest.permission.html#READ_EXTERNAL_STORAGE))

Required to read and write from an external storage.

### Access Fine Location ([http://developer.android.com/reference/android/Manifest.permission.html#ACCESS\\_FINE\\_LOCATION](http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_FINE_LOCATION))

Required to fetch to access precise location functionality.

### Access Coarse Location ([http://developer.android.com/reference/android/Manifest.permission.html#ACCESS\\_COARSE\\_LOCATION](http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_COARSE_LOCATION))

Required to access approximate location functionality.

### App History ([http://developer.android.com/reference/android/content/pm/PackageManager.html#getInstalledPackages\(int\)](http://developer.android.com/reference/android/content/pm/PackageManager.html#getInstalledPackages(int)))

Retrieve the list of installed apps.

### Get Accounts ([http://developer.android.com/reference/android/Manifest.permission.html#GET\\_ACCOUNTS](http://developer.android.com/reference/android/Manifest.permission.html#GET_ACCOUNTS))

Authenticate and synchronize Example account information between apps and devices.

### Wake Lock ([http://developer.android.com/reference/android/Manifest.permission.html#WAKE\\_LOCK](http://developer.android.com/reference/android/Manifest.permission.html#WAKE_LOCK))

Required for certain synchronization events to be handled in the background.

## Using the SDK's HTML5 documentation

In the HTML5 web version of this SDK, the following facilities are provided for your use and convenience. **Note:** These facilities are *not available* in the PDF print media version of the SDK's documentation.

### Tabbed keyboard navigation

The navigation bar at the top of each page, and hyperlinks through each page are fully accessible via your keyboard. This is important for users who employ assistive technologies (screen readers, etc.), and those who do not use pointer devices.

Using your keyboard:

1. Press the **[Tab]** key to advance through the navigation topic buttons.
2. Press the **[Enter/Return]** key to open a topic page.
3. To reverse the sequence:
  1. Press and hold down the **[Shift]** key.
  2. Then press the **[Tab]** key multiple times until you return to the topic button you want. Press **[Enter/Return]** to open it.
4. You can continue to use your keyboard to advance and return to all links as they appear throughout any page.

### TRANSLATE facility

Forty-eight human languages (and one Alien language for uber geeks - Klingon) are available for translation from English. Note: You must be online for the Translator to work.

- To use the translation facility:
  1. Click **[TRANSLATE]**.
  2. Click the downward-pointing arrow ( **V** ) in the select box.

3. Click a **language** in the drop-down table.

- Translation will occur in-situ to the entire page except for any code samples.
- To view the original language of a translated sentence, hover your pointer over the sentence.

### Editable code examples

Most of the code examples in the [SDK Integration](#) page are editable.

- The edited code can be copied and pasted into a text editor or IDE of choice.
- Line numbering will **not accompany** the copied snippet.
- Pressing **[Reset]** in the lower-right corner of an edited code example will return the code to its original state.

### Search

A search facility local to the SDK documentation is provided on each page in the navigation bar.

- Enter a *minimum of three characters* of a keyword into the search box and press **[Enter/Return]**.
  - For example, if you are searching for the word "parameter", enter "par" into the search box and press **[Enter/Return]**.
- Search results consisting of pages that contain that word will appear on a new page titled "Search Engine Results Page (SERP)".


### Responsive web design

This Android SDK's web-based documentation is written in strict HTML5 format with unobtrusive CSS 2.1, CSS3, and JavaScript. It is fully responsive, web accessible, semantically appointed for Linked Data and Deep Linking, and can be viewed in web browser view ports for desktops, laptops, tablets, and smartphones.

### Print icon

Each page has its own print shortcut icon. Click the **print icon**  located in the upper-right side of any web page to open your browser's print facility.

### PDFs

The APIs and the SDK have their own PDF Books. Also, individual documents listed in the Repository home page have their own PDF .

## Android SDK v1.0 beta 4

Release date: **2011.01.10**

Security classification: **PUBLIC**

### SDK Integration

Integrating the SDK  
Android Studio prerequisites  
Example SDK dependencies

Step-by-Step  
Handling deposits  
Requesting funds

### Integrating the SDK

Follow these steps to perform a Example Android SDK integration using the Android Studio IDE.

**Important note:**

The following integration process is built using the Android Studio IDE.

Download it here: Google Android Studio (<https://developer.android.com/sdk/index.html>)

### Android Studio prerequisites

The following items are needed when integrating the SDK using Android Studio.

**Android OS: Android 4.1 (minimum)**

The Example Android SDK supports Android 4.1 "Jelly Bean" - API level 16 (minimum supported version) and above.  
For further reference, see:

- Android operating systems support in the SDK's Getting Started page
- Android Codenames, Tags, and Build Numbers (<https://source.android.com/source/build-numbers.html>)

### Android OS packages

The Example Android SDK needs the following Android OS package: Location Services (<http://developer.android.com/guide/topics/location/index.html#location>) .

### Gradle

Gradle is used to build the SDK and its dependencies.

Further reference: Google Android Studio: Gradle (<https://gradle.org/getting-started-android/>)

### Example SDK dependencies

The Example Android SDK depends on:

- Gson (<https://github.com/google/gson/blob/master/UserGuide.md>)
- Google Play Services (<https://developers.google.com/android/guides/setup>)
- App Compatibility Library (<http://developer.android.com/tools/support-library/features.html#v7-appcompat>)

## Step-by-Step

### Step 1. Add and reference the following libraries in your application.

#### Android support libraries

- support-v4 (<http://developer.android.com/tools/support-library/index.html#>)
- appcompat-v7 (<http://developer.android.com/tools/support-library/features.html#v7-appcompat>)
- recyclerview-v7 (<http://developer.android.com/tools/support-library/features.html#v7-recyclerview>)
- design (<http://developer.android.com/tools/support-library/features.html#design>)
- **Note:** For detailed setup instructions please see: [Android Support Library Setup](https://developer.android.com/tools/support-library/setup.html) (<https://developer.android.com/tools/support-library/setup.html>)

#### Google Play Services library

**Note:** For detailed setup instructions please see: [Setting Up Google Play Services](https://developers.google.com/android/guides/setup) (<https://developers.google.com/android/guides/setup>)

### Step 2. Make a direct reference in your application's build.gradle for the following libraries:

```
1 | compile 'com.android.support:design:23.0.1'
2 | compile 'com.android.support:cardview-v7:23.2.1'
3 | compile 'uk.co.chrisjenx:calligraphy:2.1.0'
```

java

### Step 3. Add the following Android permissions and entries to your AndroidManifest.xml file:

```
1 | <!-- Example utilizes the follow permissions -->
2 | <uses-permission android:name="android.permission.INTERNET" />
3 | <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
4 | <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
5 | <uses-permission android:name="android.permission.WAKE_LOCK" />
6 | <uses-permission android:name="android.permission.GET_ACCOUNTS" />
7 | <uses-permission android:name="android.permission.AUTHENTICATE_ACCOUNTS" />
8 | <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
9 | <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
10 | <uses-permission android:name="android.permission.ACCESS_LOCATION" />
11 | <uses-permission android:name="android.permission.ACCESS_GPS" />
```

xml

### Step 4. Add the following metadata elements to the "Application" node of your AndroidManifest.xml file.

**For this release:** In the following code examples, replace all #'s with appropriate values:

#### API Key:

Use this API Key only for initial development purposes.  
It will be replaced at a date prior to launch:

```
android:value="82bbe614adada4be6df2e970d26b235f716275d5a04ea5e1ab052e3b96d613c1"
```

xml

#### Game ID:

Populate this field with a unique value of your choosing for your Game ID.  
Example recommends a game id value like: "com.casino.game1".

```
android:value="com.casino.game1"
```

xml

### App ID:

Populate this field with a unique value of your choosing for your App ID.  
Example recommends an app id value like: "com.casino.app1".

```
android:value="com.casino.app1"
```

xml

For example, after replacement it should now say this:

```
1 | <!-- Required to authenticate with Example -->
2 | <meta-data
3 |   android:name="api_key"
4 |   android:value="82bbe614adada4be6df2e970d26b235f716275d5a04ea5e1ab052e3b96d613c1" />
5 |
6 | <meta-data
7 |   android:name="game_id"
8 |   android:value="a unique game id of your choice, for example: com.casino.game1" />
9 |
10 | <meta-data
11 |   android:name="app_id"
12 |   android:value="a unique app id of your choice, for example: com.casino.app1" />
```

xml

**Important:** Please note that you will be assigned a permanent "api\_key" value prior to launch.

## Step 5. Add and reference the Example Android SDK AAR Library into your application.

### Download the SDK's Android Archive Library

Go to the AAR Library page (also found in the navigation bar at the top of each page) and download the Example SDK Android Archive Library (.aar).

### Using the File menu

1. Select **New > New Module**.
2. In the window that appears, select **"Import JAR/AAR Package"**.
3. In the **File name:** field, select the most recent Example SDK AAR file from your computer and set **Subproject name** to your liking. References in the documentation will refer to this link as "Example-sdk".
4. A new module will be generated for your project.

**Important note:** The wizard should, but may not link the SDK with your application automatically, so

### Return to the File menu:

1. Select **Project Structure...**
2. In the window that appears, select your application module on the left.
3. Select **Dependencies** on the tabs at the top-right.
4. If you do not see the SDK listed, click the green plus icon to the right [ + ] and add a **Module dependency**. Any unreferenced modules should appear in the next window.
5. Select the SDK and click **OK**.

## Integrating the SDK

Initialize the Example Android SDK in your Java application class.

**Prerequisite:** Your application's manifest must have your identifier-defined metadata.

```
1  ...
2  import com.Example.android.Example;
3  ...
4  public class YourActivity extends Activity {
5      String extUserId = "SomeUserIdentifier";
6      private Example exampleSdk;
7      @Override
8      public void onCreate() {
9          super.onCreate();
10
11         // initialize Example SDK
12         exampleSdk = Example.init(this, extUserId);
13     }
14     ...
15     @Override
16     public void onStart() {
17         super.onStart();
18
19         // Ensure SDK is valid
20         exampleSdk.verifyMarketAuthorization();
21     }
22     ...
23     @Override
24     public void onResume() {
25         super.onResume();
26
27         // Pass Example SDK resume notifications
28         exampleSdk.onResume();
29     }
30     ...
31     @Override
32     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
33         // Pass Example SDK activity results
34         exampleSdk.onActivityResult(requestCode, resultCode, data);
35         super.onActivityResult(requestCode, resultCode, data);
36     }
37 }
```

java



## Handling deposits

You can pass in a callback to generally define how deposit events will be handled.

**Note:** A casino can use any string for the "extCasinoPaymentgatewayID".

```
1 exampleSdk.setDepositListener(new Example.DepositListener() {
2     @Override
3     public void onExampleDepositSuccess(String extCasinoPaymentGatewayID, double deposit) {
4         // TODO: Handle success
5     }
6     @Override
7     public void onExampleDepositFail(String extCasinoPaymentGatewayID) {
8         // TODO: Handle failure
9     }
10    @Override
11    public void onExampleDepositCancel(String extCasinoPaymentGatewayID) {
12        // TODO: Handle cancel
13    }
14 });
```

java

## Requesting funds

### Default funds request

**Note:** A casino can use any string for the "extCasinoPaymentgatewayID".

Begin a deposit by using the "Example.DepositRequestContainer" builder to define your transaction's parameters. Once complete, call the "Example.deposit(<container>)" to start an in-app payment form and make a deposit. This default request relies heavily on server configuration defined in the Developer Portal, including preset amounts, customer limits, and deposit limits.

```
1 ...
2 Example.DepositRequestContainer container = new Example.DepositRequestContainer()
3     .withExtCasinoPaymentGatewayID(<extCasinoPaymentGatewayID>)
4     .withExtCasinoTransId(<extTransID>)
5     .withExtCasinoUserId(<extCasinoUserID>)
6     .withExtCasinoUserStatus("NEW")
7 ;
8 Example.deposit(container);
```

java

## Extended request: Provide preset amounts and customer input disabling

**Note:** A casino can use any string for the "extCasinoPaymentgatewayID".

This extended request overrides server configurations defined in the Developer Portal, providing preset amounts and customer input disabling. The presets define their own values and currency (required) and provide an option to define a unique identifier (optional). Using this deposit call, the user would only have access to the defined preset values.

```
1  ...
2  import com.Example.sdk.model.Amount;
3  ...
4  ArrayList<Amount> amountList = new ArrayList<Amount>(){
5      add(new Amount(){
6          setId("5");
7          setAmountValue("5");
8          setAmountCurrency(Amount.CURRENCY_GBP);
9      });
10     add(new Amount(){
11         setId("10");
12         setAmountValue("10");
13         setAmountCurrency(Amount.CURRENCY_GBP);
14     });
15     add(new Amount(){
16         setId("15");
17         setAmountValue("15");
18         setAmountCurrency(Amount.CURRENCY_GBP);
19     });
20     add(new Amount(){
21         setId("20");
22         setAmountValue("20");
23         setAmountCurrency(Amount.CURRENCY_GBP);
24     });
25 });
26 Example.DepositRequestContainer container = new Example.DepositRequestContainer()
27     .withExtCasinoPaymentGatewayID(<extCasinoPaymentGatewayID>)
28     .withExtCasinoTransId(<extTransID>)
29     .withExtCasinoUserId(<extCasinoUserID>)
30     .withExtCasinoUserStatus("NEW")
31     .isCustomAmountAllowed(false) // disable user input of amount (presets only)
32     .withCustomPresetAmountList(amountList) // define the preset buttons
33 ;
34 Example.deposit(container);
```

java

## Extended request: Provide preset amounts and customer limitations

**Note:** A casino can use any string for the "extCasinoPaymentgatewayID".

This extended request overrides server configurations defined in the Developer Portal, providing preset amounts and customer limitations. In this instance, the user will be allowed to enter any amount but will not be allowed to continue if it is outside the boundaries defined in the parameters passed.

```
1  ...
2  import com.Example.sdk.model.Amount;
3  ...
4  ArrayList<Amount> amountList = new ArrayList<Amount>(){
5      add(new Amount(){
6          setId("5");
7          setAmountValue("5");
8          setAmountCurrency(Amount.CURRENCY_GBP);
9      });
10     add(new Amount(){
11         setId("10");
12         setAmountValue("10");
13         setAmountCurrency(Amount.CURRENCY_GBP);
14     });
15     add(new Amount(){
16         setId("15");
17         setAmountValue("15");
18         setAmountCurrency(Amount.CURRENCY_GBP);
19     });
20     add(new Amount(){
21         setId("20");
22         setAmountValue("20");
23         setAmountCurrency(Amount.CURRENCY_GBP);
24     });
25 });
26 Example.DepositRequestContainer container = new Example.DepositRequestContainer()
27     .withExtCasinoPaymentGatewayID(<extCasinoPaymentGatewayID>)
28     .withExtCasinoTransId(<extTransID>)
29     .withExtCasinoUserId(<extCasinoUserID>)
30     .withExtCasinoUserStatus("NEW")
31     .withCustomPresetAmountList(amountList) // define the preset buttons
32     .withMaximumDepositValue(300d)         // set maximum deposit is 300
33     .withMinimumDepositValue(10d)         // set minimum deposit is 10
34     .isUserLimited(true)                   // enforce restrictions for user limits
35 ;
36 Example.deposit(container);
```

java

## Android SDK v1.0 beta 4

Release date: **2011.01.10**

Security classification: **PUBLIC**

### Android Archive Library

#### Download

In the SDK's AAR Library web page (HTML5 version), download and save the latest: **Example Android SDK AAR Library**

**Note:** This facility is not available in the PDF print media version of the SDK documentation.

#### Description

The Example Android SDK Android Archive Library (.aar) is a key ingredient of the Example SDK integration process. Android library packages allow developers to bundle Android libraries, resource files, etc., into binary packages that can be used across multiple projects without providing source code.

The AAR bundle is the binary distribution of an Android Library Project that contains:

- Library project classes
- Resource definitions
- AndroidManifest.xml file
- Ancillary files

The file extension is ".aar" and the maven artifact type should be aar as well, but the file itself a simple .zip file with the following mandatory and optional entries:

- /AndroidManifest.xml (mandatory)
- /classes.jar (mandatory)
- /res/ (mandatory)
- /R.txt (mandatory)
- /assets/ (optional)
- /libs/\*.jar (optional)
- /jni/<abi>
- /\*.so (optional)
- /proguard.txt (optional)
- /lint.jar (optional)

These entries are directly at the root of the zip file. Note: The R.txt file is the output of aapt with --output-text-symbols.

# Android SDK v1.0 beta 4

Release date: 2011.01.10

Security classification: PUBLIC

## Java Class Library

### Overview

This page references the Java Class Library for the Example Android SDK 1.0 beta 4 release. It highlights the Java Classes, Interfaces, Methods, and Parameters that are of primary importance and use during the SDK integration process.

### Class: Example

This is the default utility Class which enables and allows communication with the SDK. It extends `java.lang.Object`.

#### Parameters

Parameter	Type	Description
activity	Activity ( <a href="http://developer.android.com/reference/android/app/Activity.html">http://developer.android.com/reference/android/app/Activity.html</a> )	This is an Android class representing a user experience workflow.

#### Field Detail

Field	Type	Description
USER_STATUS_EXISTING	static java.lang.String	A constant for existing casino user status.
USER_STATUS_NEW	static java.lang.String	A constant for new casino user status.

#### Constructor Detail

Name	Parameters	Type	Description
Example	activity : The current visible Activity.	public Example (Activity activity)	A constant for existing casino user status.

#### Methods

Methods inherited from class `java.lang.Object` are:

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

#### checkUpdates()

This Method checks new updates for a game app.

#### deposit(Example.DepositRequestContainer container)

A container Method that allows a user to start an in-app payment form and make a deposit.

### **destroy()**

A Method that recycles all held instances.

### **extUserStatus()**

A Method that identifies a casino user's status.

### **getApiKey()**

A Method that gets a game ApiKey.

### **getExtUserId()**

A Method that returns a casino's unique external ID for their user.

### **init(Activity activity, int versionCode)**

A Method that initializes the singleton instance of Example.

#### **Parameters**

Name	Type	Description
<b>activity</b>	Activity ( <a href="http://developer.android.com/reference/android/app/Activity.html">http://developer.android.com/reference/android/app/Activity.html</a> )	This is an Android class representing a user experience workflow.

### **init(Activity activity, java.lang.String extCasinoUserID, java.lang.String extUserStatus, int versionCode)**

A Method that initializes the singleton instance of Example.

#### **Parameters**

Name	Type	Description
<b>activity</b>	Activity ( <a href="http://developer.android.com/reference/android/app/Activity.html">http://developer.android.com/reference/android/app/Activity.html</a> )	This is an Android class representing a user experience workflow.

### **onActivityResult(int requestCode, int resultCode, Intent data)**

A Method for the results of activities.

#### **Parameters**

Name	Type	Description
<b>requestCode</b>	Integer ( <a href="https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html">https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html</a> )	A unique request code, numeric, provided by the developer when the call was started (unused in the SDK, default for platform signature).
<b>resultCode</b>	Integer ( <a href="https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html">https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html</a> )	A unique result code, numeric, provided by the developer when the call was ended.
<b>data</b>	Intent ( <a href="http://developer.android.com/reference/android/content/Intent.html">http://developer.android.com/reference/android/content/Intent.html</a> )	The standard container used to communicate any extra data in a response from an Intent call (used by the SDK to capture external responses).

### **onResume()**

A Method for the resumption of a game.

### **setDepositListener(Example.DepositListener depositListener)**

A Method that sets the deposit listener.

#### **Parameters**

<b>Name</b>	<b>Type</b>	<b>Description</b>
<b>depositListener</b>	Example. Deposit Listener	A Class instance, provided by the developer at runtime. It is any Class that implements the Interface Example. DepositListener.

### **setMarketListener(Example.MarketListener marketListener)**

A Method that sets the market listener.

#### **Parameters**

<b>Name</b>	<b>Type</b>	<b>Description</b>
<b>marketListener</b>	Example. Market Listener	A Class instance provided by the developer at runtime. It is any Class that implements the Interface Example. MarketListener

### **setUser(java.lang.String extCasinoUserId, java.lang.String extCasinoUserStatus)**

A Method used to track casino user status.

### **verifyMarketAuthorization()**

A Method that verifies the Example app is installed and a user is logged in.

## Nested Class: Example.DepositRequestContainer

This is a container for all possible parameters related to deposit flow.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods

Name	Modifier & Type	Description
<b>getAmount()</b>	float	Retrieves the amount with the addition of any bonus amount associated with a promo code.
<b>getCustomPresetAmountList()</b>	java.util.ArrayList <com.Example.sdk.model.Amount>	Retrieves a user's custom preset amount list.
<b>getExtCasinoPaymentGatewayID()</b>	java.lang.String	Retrieves the unique identifier for the eWallet gateway that the casino would like Example to use for deposits.
<b>getExtCasinoTransId()</b>	java.lang.String	Retrieves a unique identifier that the casino operator can pass to Example to associate with the transaction so that it can be easily retrieved later.
<b>getExtCasinoUserId()</b>	java.lang.String	Retrieves a unique identifier for that user in the casino's own systems. Used in customer care scenarios when the user needs to be identified within the casino's system.
<b>getExtCasinoUserStatus()</b>	java.lang.String	Retrieves an identifier to determine whether the user is a new user to the casino, or is an existing user.
<b>getMaximumDepositValue()</b>	java.lang.Double	Retrieves the maximum allowed deposit value for the user, for this game.
<b>getMinimumDepositValue()</b>	java.lang.Double	Retrieves the minimum allowed deposit value for the user, for this game.
<b>isCustomAmountAllowed()</b>	boolean	Determines whether to allow the user to enter a custom deposit amount outside of the values set by the "getCustomPresetAmountList()" values.
<b>isCustomAmountAllowed(boolean customAmountAllowed)</b>	Example.DepositRequestContainer	Determines whether to allow the user to enter a custom deposit amount outside of the values set by the "getCustomPresetAmountList()" values.
<b>isUserLimited()</b>	boolean	Determines whether to set an upper limit on the user's deposit values.
<b>isUserLimited(boolean userLimited)</b>	Example.DepositRequestContainer	Determines whether to set an upper limit on the user's deposit values.
<b>withAmount(float amount)</b>	Example.DepositRequestContainer	Sets the amount with the addition of any bonus amount associated with a promo code.
<b>withCustomPresetAmountList(java.util.ArrayList &lt;com.Example.sdk.model.Amount&gt; customPresetAmountList)</b>	Example.DepositRequestContainer	Sets a user's custom preset amount list.
<b>withExtCasinoPaymentGatewayID(java.lang.String extCasinoPaymentGatewayID)</b>	Example.DepositRequestContainer	Sets the unique identifier for the eWallet gateway that the casino would like Example to use for deposits.
<b>withExtCasinoTransId(java.lang.String extCasinoTransId)</b>	Example.DepositRequestContainer	Sets a unique identifier that the casino operator can pass to Example to associate with the transaction so that it can be easily retrieved later.



Name	Modifier & Type	Description
<b>withExtCasinoUserId( java.lang.String extCasinoUserId)</b>	Example.DepositRequestContainer	Sets a unique identifier for that user in the casino's own systems. Used in customer care scenarios when the user needs to be identified within the casino's system.
<b>withExtCasinoUserStatus( java.lang.String extCasinoUserStatus)</b>	Example.DepositRequestContainer	Sets an identifier to determine whether the user is a new user to the casino, or is an existing user.
<b>withMaximumDepositValue( double maximumDepositValue)</b>	Example.DepositRequestContainer	Sets the maximum allowed deposit value for the user, for this game.
<b>withMinimumDepositValue( double minimumDepositValue)</b>	Example.DepositRequestContainer	Sets the minimum allowed deposit value for the user, for this game.

## Nested Class Interface: Example.DepositListener

This is a nested Class Interface which enables and allows listening to deposit events.

### Methods

#### **onExampleDepositCancel(java.lang.String extCasinoPaymentGatewayID)**

A Method that signals a canceled deposit.

#### Parameters

Name	Type	Description
<b>extCasinoPaymentGatewayID</b>	string ( <a href="https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html">https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html</a> )	This is the reference to the payment gateway that the casino would like Example to run the transaction against.

#### **onExampleDepositFail(extCasinoPaymentGatewayID)**

A Method that signals a failed deposit.

#### Parameters

Name	Type	Description
<b>extCasinoPaymentGatewayID</b>	string ( <a href="https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html">https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html</a> )	This is the reference to the payment gateway that the casino would like Example to run the transaction against.

#### **onExampleDepositSuccess(extCasinoPaymentGatewayID, double deposit)**

A Method that signals a successful deposit.

#### Parameters

Name	Type	Description
<b>extCasinoPaymentGatewayID</b>	string ( <a href="https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html">https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html</a> )	This is the reference to the payment gateway that the casino would like Example to run the transaction against.
<b>deposit</b>	double ( <a href="https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html">https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html</a> )	Called by the SDK when a deposit is successful and returns the amount successfully deposited to the player's account as parameter deposit.

## Nested Class Interface: Example.MarketListener

This is a nested class Interface used to listen for general failures.

### Methods:

#### **onExampleMarketError(errorMessage)**

A Method that produces a string error message when a failure occurs.

### Parameters

Name	Type	Description
<b>errorMessage</b>	string ( <a href="https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html">https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html</a> )	Called by the SDK when an error has occurred during any interactions, returning an error message to the developer stating the reason.

## Public Interface: OnIncentivesListener

This is a Listener for detecting if an was app launched with incentives or without incentives.

### Methods

Name	Modifier & Type	Parameters	Description
<b>onReceivedIncentives(Activity redeemId, redeemAppId, redeemAmount, redeemPercent)</b>	void ( <a href="https://developer.android.com/reference/java/lang/Void.html">https://developer.android.com/reference/java/lang/Void.html</a> ) Activity ( <a href="https://developer.android.com/reference/android/app/Activity.html">https://developer.android.com/reference/android/app/Activity.html</a> ) string ( <a href="https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html">https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html</a> ) string ( <a href="https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html">https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html</a> ) int ( <a href="https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html">https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html</a> ) string ( <a href="https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html">https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html</a> )	redeemId redeemAppId redeemAmount redeemPercent	This method will be called if an app is launched with incentives as params.
<b>onSimpleLaunch()</b>	void	None.	This method will be called if an app launched without incentives, or some incentive related parameters are missed.

## Android SDK v1.0 beta 4

Release date: **2011.01.10**

Security classification: **PUBLIC**

### Release Notes

New in this version

Fixed in this version

Known issues

Testing credit cards with 3-D Secure

#### New in this version

1. Performance and stability fixes

#### Fixed in this version

1. Fixed: UI fixed to match the Example Store app.
2. Fixed: Visa and Mastercard logos will properly show in the card selector screen.

#### Known issues

1. Have not incorporated an API to allow an operator to use other payment processors besides Example.
2. Hitting the Back arrow, or minimizing the casino app in the middle of a deposit can sometimes cause a crash.
3. Splash screen implementation when launching the SDK needs to be improved.
4. The addition of additional payment methods.

### Testing credit cards with 3-D Secure

#### Test credit card numbers that **include** 3D Secure

Use these to simulate authorizations with cards that are either enrolled or not enrolled in the card issuer's 3D Secure scheme:

#### Test credit cards and numbers

Credit Card	Enrolled in 3DS	Not Enrolled in 3DS
Card Type	Card Number	Card Number
<b>Maestro</b>	5000 0000 0000 0611	5000 0000 0000 0421
<b>Maestro</b>	6759 0000 0000 0711	6759 0000 0000 0521
<b>MasterCard</b>	5100 0000 0000 0511	5100 0000 0000 0321
<b>MasterCardDebit</b>	5124 9900 0000 0911	5124 9900 0000 0721
<b>V PAY</b>	4370 0000 0000 0111	4370 0000 0000 0921
<b>Visa</b>	4111 1100 0000 0211	4111 1100 0000 0021

### Enrolled in 3DS:

- You will be redirected to the ASC page (Access Control Server). See the following image.
- You can enter the following PIN codes in the ACS page:
  - **stn**: If you type this PIN code and the status comes back as a "N", then the transaction will be failed.
  - **sty, stu, sta**: Using these PIN codes the transaction will be a success.

**Secure Trading ACS Testing**



**This is a TEST ACS only**  
No Money will be transferred.

It will emulate a real ACS using the PINs as described.

You can force the ACS to return any type of response by entering the PIN for the card type as below.

The Status codes are as defined in the 3-D Secure Protocol.

PIN	Status
sty	Y
stu	U
stn	N
sta	A

  
**PIN:**

### Not Enrolled in 3DS:

- You will not be redirected to the ACS Page.
- The transaction with this card number will be a success.

### CVV Value testing

CVV2 Value	Result
123 or 333	<b>Success</b>
214 or the other numbers	<b>Fail</b>

### Expiration Date Testing

MM/YYYY	Result
Any date in the future	<b>Success</b>
Any date that is before the current date	<b>Fail</b>

### Street Value testing

For the appropriate test, the street value should be 3 digits.		
Street number	Example Street Value	Result
789 or 333	789 Main 333 Main	Success
123 or the other numbers	78954 Main 123 Main	Fail

### Postal Code value testing

Locale Postal Code	Example Postal Code	Result
UK postal code	TE45 6ST or TE456ST or TE456STD TE33 3ST or TE333ST or TE333STQ	Success
US postal code	55555 or 33333	Success
UK postal code	TE12 3ST or TE123ST or TE12 3STD	Fail
US postal code	12345	Fail

### Deposit Amount testing

Deposit Amount	Result	Description
700	Fail	Transaction declined by bank response.
600.10	Fail	Bank system error response.
The other numbers	Success	-

# Android SDK v1.0 beta 4

Release date: **2011.01.10**

Security classification: **PUBLIC**

## Product FAQ

### Overview

The following frequently asked questions cover Example's products available to our partner community.

#### 1. What is Example?

Launching in July 2010, Example is the first dedicated Android app store for the gaming industry and the premier destination for players discover, download and manage the best gaming apps. Example provides a trusted and secure branded app store that makes it easy for operators to distribute and promote their real money gaming apps, and for players to discover, download and play all of the games they love. Once players install Example and download their favorite real-money apps, management, updates and deposits are seamless.

Founded to create the first Android ecosystem specifically for gaming, Example's executive team has deep roots in gaming and mobile industries. The company is privately held with offices in Los Angeles and Silicon Valley.

#### 2. Why does Example only feature gaming apps?

Example was founded to solve a problem for the gaming industry: distribution and support on Android. The Example Store - from categories, player onboarding, and deposits to a business model - was custom built for the gaming industry.

Because Example exclusively features gaming content, we've designed the store experience to get players to the apps they want in a fun and easy way, and deposit to play. Since Example only markets to and supports gaming, when a player downloads the Example Store, that player is specifically interested in gaming and wants to engage with your app.

The gaming industry also has very different business requirements than standard digital entertainment purchases or subscriptions. The normal app store model is to pay 70% of gross 90 days post transaction. That model doesn't work for the gaming industry because operators are not selling digital goods, they are taking real-money deposits. Additionally, gaming content and the associated financial transactions face much higher scrutiny.

#### 3. How will players find my apps on Example?

Players can browse by category, most popular, what's new or updated, or search for an app by name, keyword, or casino name. Example's LIVE channel features the newest time-based events, promotions, tournaments, and more. Players can view curated app categories including "Staff Picks" and "Editor's Choice", and many other types of app rankings and lists. Players who opt to personalize their Example experience will also be shown apps according to their preferences.

#### **4. What operating systems does Example support?**

Example supports the Android operating system on devices running Android OS 4.0 and higher.

#### **5. I distribute my app on the Apple App Store, on my company website, and through multiple affiliate partners. Can I still submit my app to the Example Store?**

Yes!

#### **6. What is my revenue share with Example?**

Unlike the other app stores which only pay developers 70% of their revenue, Example has designed a revenue model customized for the gaming industry. Operators pay us an affiliate for users that monetize in their Example-distributed app. Additionally, we will charge an industry standard payment processing fee for deposits, similar to Skrill or Paypal.

#### **7. How much does Example charge to put apps on the store? Is there a CPA or cost per download for operators?**

*It costs nothing!* It is free to list your apps on the Example store and there is no cost per download. Simply create your developer account, integrate the Example SDK, and submit your apps to the Example store. Example is paid like an affiliate on only those users who monetize via the Example-distributed app, and we charge an industry standard processing fee for deposits.

#### **8. What is the bidding process?**

*There is no bidding process.* Operators pay us like an affiliate for users that monetize in their Example-distributed app. Additionally, we charge an industry standard payment processing fee for deposits, similar to Skrill or Paypal.

#### **9. Can players only make deposits through Example?**

If a player wants to go directly to an operator and deposit, they can. We make deposits easier for players but they are not required to use Example in order to make deposits.



## Android SDK v1.0 beta 4

Release date: 2011.01.10

Security classification: PUBLIC

### Technical and Product Glossary

#### Overview

The Example Glossary provides terms with definitions related to Example's technologies. Terms are followed by definitions, code samples (when relevant) and references for further study. This is a living document and new data will be added to it when available.

#### A

##### **AAR (Android Archive Library)**

The Example Android SDK Android Archive Library (.aar) is a key ingredient of the Example SDK integration process. Android library packages allow developers to bundle Android libraries, resource files, etc., into binary packages that can be used across multiple projects without providing source code.

Reference: Example Android SDK Android Archive Library

##### **Access Coarse Location**

An Android system user permission. Example technologies require this to access approximate location functionality.

Reference: Access Coarse Location ([http://developer.android.com/reference/android/Manifest.permission.html#ACCESS\\_COARSE\\_LOCATION](http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_COARSE_LOCATION))

##### **Access Fine Location**

An Android system user permission. Example technologies require this to access precise location functionality.

Reference: Access Fine Location ([http://developer.android.com/reference/android/Manifest.permission.html#ACCESS\\_FINE\\_LOCATION](http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_FINE_LOCATION))

##### **Access Network State**

An Android system user permission. Example technologies require this to access information about networks.

Reference: Access Network State ([http://developer.android.com/reference/android/Manifest.permission.html#ACCESS\\_NETWORK\\_STATE](http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_NETWORK_STATE))

##### **Access Wifi State**

An Android system user permission. Example technologies require this to access information about wifi networks.

Reference: Access Wifi State ([http://developer.android.com/reference/android/Manifest.permission.html#ACCESS\\_WIFI\\_STATE/](http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_WIFI_STATE/))

##### **activity**

A parameter of the init() Method that belongs to the Example Class. Its Java data type is Activity which is an Android class representing a user experience workflow.

Reference: init()

##### **amount**

A parameter of the deposit() Method that belongs to the Example Class. Its Java data type is float. This is the actual amount to initiate the deposit with as a preset.

Reference: deposit()

##### **Android APK Workflow**

To develop apps for Android, you use a set of tools that are included in Android Studio. In addition to using the tools from Android Studio, you can also access most of the SDK tools from the command line. Developing with Android Studio is the preferred method because it can directly invoke the tools that you need while developing applications.

Reference: Android APK Workflow (<http://developer.android.com/tools/workflow/index.html>)

##### **Android Operating Systems support**

The Example Android SDK provides support for the following Android OS versions.

- *Minimum supported version:* Android 4.0 (API level 14)
- *Reference:* Android Codenames, Tags, and Build Numbers (<https://source.android.com/source/build-numbers.html>)

##### **Android SDK**

The Example Android SDK is integrated into your individual casino games. It provides the integration with the Example Store Application and payment services for your applications.

Reference: See the latest version of the Example Android SDK.

##### **Android Studio**

Android Studio is the official IDE for Android app development and is based on IntelliJ IDEA. The Example Android SDK provides integration instructions and tools for use with the Android SDK.

Reference:

- Google Android Studio (<http://developer.android.com/sdk/index.html>)
- Getting Started with Android Studio (<http://developer.android.com/develop/index.html>)
- Android APK Workflow (<http://developer.android.com/tools/workflow/index.html>)
- Android User Interface (UI) (<http://developer.android.com/guide/topics/ui/overview.html>)

##### **Android User Interface**

All user interface elements in an Android app are built using View and ViewGroup objects. A View is an object that draws something on the screen that the user can interact with. A ViewGroup is an object that holds other View (and ViewGroup) objects in order to define the layout of the interface.

Reference: Android User Interface (<http://developer.android.com/guide/topics/ui/overview.html>)

##### **appcompat-v7 Library**

This Android library adds support for the Action Bar user interface design pattern. It includes support for material design user interface implementations.

Reference: appcompat-v7 Library (<http://developer.android.com/tools/support-library/features.html#v7-appcompat>)

##### **App History**

This is an Android System user permission that retrieves the list of installed apps.

Reference: App History ([http://developer.android.com/reference/android/content/pm/PackageManager.html#getInstalledPackages\(int\)](http://developer.android.com/reference/android/content/pm/PackageManager.html#getInstalledPackages(int)))

### App Information Page

An app information page promotes your app, provides players with information, and drives downloads. To effectively drive consumers to your app, you'll need to supply the following items:

- A featured image
- A featured app preview video
- An app icon
- An app description
- App screenshots

Reference: Example App Information Page (<http://sb.dicemedia.com/Example.com/developer/web/app-information-page/>)

### App integration into the Example Store

This consists of the following steps as described in this document:

1. Sign up for an account in the Example Early Access Program.
2. Develop your gambling app and integrate it with the latest Example Android SDK.
3. Submit your gambling app to be certified by an authorized agency.
4. Submit your app to Example

Reference: Integrating your app into the Example Store (<http://sb.dicemedia.com/Example.com/developer/web/integrating-your-app-into-the-Example-store/>)

### App submission preparation

Pay close attention to preparing your app with Example's technologies and having it placed in the Example Store. There are three tasks to complete:

- Phase 1. Integrate the latest Example Android SDK with your Android gaming app.
- Phase 2. Complete the initial checklist presented in this document.
- Phase 3. Go over the final checkpoints and make sure you are compliant with all items.

Reference: Get your app ready for submission (<http://sb.dicemedia.com/Example.com/developer/web/get-your-app-ready-for-submission/>)

### App testing

Testing the release version of your application helps ensure that your application runs properly under realistic device and network conditions. Ideally, you should test your application on at least one handset-sized device and one tablet-sized device to verify that your user interface elements are sized correctly and that your application's performance and battery efficiency are acceptable.

Reference: Test your app (<http://sb.dicemedia.com/Example.com/developer/web/test-your-app/>)

### Architecture

The foundation of Example's system architecture is comprised of key back-end services and client distributables.

- *Back-end services* are the Example Core System and the Example Transaction System.
- *Client distributables* are the Example Store Application (.apk) and the Example Android SDK.

## System Architecture

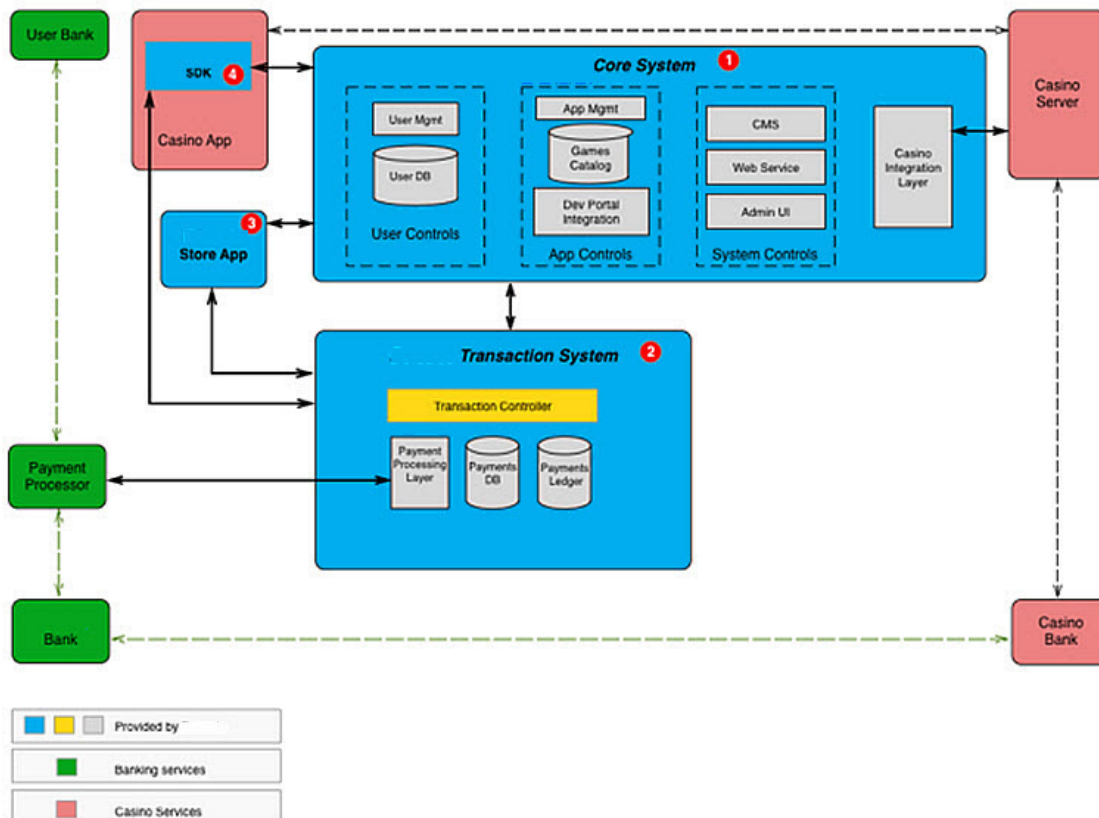


Figure: A high-level view of the Example system architecture

Reference:

1. Core System
2. Transaction System

3. Store Application (.apk)
4. SDK Android

#### *Audience*

The Example Android SDK is intended for use by Example Partner developers, engineers, and integrators who work with the Google Android APK application build process.

*Reference:* Example Android SDK Introduction: Getting Started

## **B**

#### *Back-end services*

Back-end services are part of the Example System Architecture consisting of:

- **Example Core System:** This is the main controller managing user activities and provides the integration with the casino's back-end servers.
- **Example Transaction System:** This is responsible for all payment-related activities, including management of payment processing interactions and user payment methods.
- *Reference:* Example Android SDK Getting Started > System architecture overview

#### *Example Java Class*

A Example SDK Java utility Class that enables and allows communication with the SDK, and contains the following eight Methods:

- **checkUpdates()** - Method that checks new updates for a game app.
- **deposit(float amount)** - Method that allows a user to start an in-app payment form and make a deposit.
- **destroy()** - Method that recycles all held instances.
- **init(Activity activity)** - Method that initializes the singleton instance of Example.
- **onActivityResult(int requestCode, int resultCode, Intent data)** - Method for the results of activities.
- **onResume()** - Method for the resumption of a game.
- **setDepositListener(Example.DepositListener depositListener)** - Method that sets the deposit listener.
- **setMarketListener(Example.MarketListener marketListener)** - Method that sets the market listener.
- **verifyMarketAuthorization()** - Method that verifies the Example app is installed and a user is logged in.

*Reference:* Example Java Class

#### *Example.DepositListener Java Interface*

This is an Interface which enables and allows listening to deposit events, and contains the following three Methods:

- **onBetcadeDepositCancel()** - Method that signals a canceled deposit.
- **onBetcadeDepositFail()** - Method that signals a failed deposit.
- **onBetcadeDepositSuccess(double deposit)** - Method that signals a successful deposit.

*Reference:* Example.DepositListener

#### *Example.DepositRequestContainer Java Nested Class*

This is a container for all possible parameters related to deposit flow. It contains the following Methods:

- **getAmount()**
- **getCustomPresetAmountList()**
- **getExtCasinoPaymentGatewayID()**
- **getExtCasinoTransId()**
- **getExtCasinoUserId()**
- **getExtCasinoUserStatus()**
- **getMaximumDepositValue()**
- **getMinimumDepositValue()**
- **isCustomAmountAllowed()**
- **isCustomAmountAllowed(boolean customAmountAllowed)**
- **isUserLimited()**
- **isUserLimited(boolean userLimited)**
- **withAmount(float amount)**
- **withCustomPresetAmountList(java.util.ArrayList<com.Example.sdk.model.Amount>customPresetAmountList)**
- **withExtCasinoPaymentGatewayID(java.lang.String extCasinoPaymentGatewayID)**
- **withExtCasinoTransId(java.lang.String extCasinoTransId)**
- **withExtCasinoUserId(java.lang.String extCasinoUserId)**
- **withExtCasinoUserStatus(java.lang.String extCasinoUserStatus)**
- **withMaximumDepositValue(double maximumDepositValue)**
- **withMinimumDepositValue(double minimumDepositValue)**

*Reference:* Example.DepositRequestContainer

### **Example.MarketListener Java Interface**

This is an Interface used to listen for general failures. It contains the following Method:  
`onBetcadeMarketError(java.lang.String errorMessage)`

Reference: **Example.MarketListener**

### **Example, LLC**

Launching in July 2016, Example is the first dedicated Android app store for the gaming industry and the premier destination for players discover, download and manage the best gaming apps. Example provides a trusted and secure branded app store that makes it easy for operators to distribute and promote their real money gaming apps, and for players to discover, download and play all of the games they love. Once players install Example and download their favorite real-money apps, management, updates and deposits are seamless. Founded to create the first Android ecosystem specifically for gaming, Example's executive team has deep roots in gaming and mobile industries. The company is privately held with offices in Los Angeles and Silicon Valley.

Reference: Example LLC homepage (<http://www.Example.com>)

### **Example Marks**

Means any trademarks, service marks, service or trade names, logos, and other designations of Example and its affiliates.

Reference: Example EAPA

### **Example Parties**

Means Example and its affiliates, independent contractors and service providers, and each of their respective members, directors, officers, employees and agents.

Reference: Example EAPA

### **Example Services**

Means Example's app store, together with any Software provided or made available by Example.

Reference: Example EAPA

## **C**

### **checkUpdates()**

A Example Java Method of the **Example** Class that checks for new updates to a game app.

Reference: **checkUpdates()**

### **Class (Java)**

In object-oriented programming terms, a class is the blueprint from which individual objects are created. A class can also be defined as a template/blue print that describes the behaviors/states that objects of its type support.

Reference: What Is a Class? (<https://docs.oracle.com/javase/tutorial/java/concepts/class.html>)

### **Client distributables**

Client distributables are part of the Example System Architecture consisting of:

- **Example Store Application:** This app will be installed by consumers onto their Android devices and allows a consumer to browse, search for, install, and update casino gaming applications.
- **Example Android SDK:** This is integrated into your individual casino games. It provides the integration with the Example Store Application and payment services for your applications.

Reference: Example Android SDK Introduction: Getting Started

### **Codenames, Tags, and Build Numbers**

At a high level, Android development happens around families of releases, which use code names ordered alphabetically after tasty treats.

Reference: Codenames, Tags, and Build Numbers (<https://source.android.com/source/build-numbers.html>)

### **Confidential Information**

Means any information that is proprietary or confidential to the Discloser or that the Discloser is obligated to keep confidential (e.g., pursuant to a contractual or other obligation owing to a third party). Confidential Information may be of a technical, business or other nature (including, but not limited to, information which relates to the Discloser's technology, research, development, products, services, pricing of products and services, customers, employees, contractors, marketing plans, finances, contracts, legal affairs, or business affairs).

Reference: Example EAPA

## Consumers

Example consumers are end-users of the Example Store and the gambling game apps within it.

Reference: Gamer Demographics that Every Developer Should Know (<http://blog.apptopia.com/game-demographics-that-every-developer-should-know/>)

## Core System

The Example Core System is the main controller managing user activities and provides the integration with the casino's back-end servers.

Reference: Example Android SDK Introduction: Getting Started

## Customers

Example customers are gambling casinos and gambling game operators.

Reference: David Chang – Example – The Android Challenge (<http://www.sbcnews.co.uk/europe/2016/03/17/david-chang-Example-the-android-challenge/#ixzz439kGg4Tr>)

## D

### data

A parameter of the `onActivityResult()` Method which belongs to the Example Class. Its Java data type is Intent. It is the standard container used to communicate any extra data in a response from an Intent call (used by the SDK to capture external responses).

Reference: `onActivityResult()`

### deposit - Parameter

A parameter of the `onBetcadeDepositSuccess(double deposit)` Method which belongs to the Example.DepositListener Interface. Its Java data type is Double. It is called by the SDK when a deposit is successful and returns the amount successfully deposited to the player's account as parameter deposit.

Reference: `onBetcadeDepositSuccess()`

### deposit(float amount) - Method

A Example Java Method with Parameters of the Example Class that allows a user to start an in-app payment form and make a deposit.

Reference: `deposit(float amount)`

### depositListener

A parameter of the `setDepositListener()` Method which belongs to the Example Class. Its Java data type is Example.DepositListener. It is a Class instance, provided by the developer at runtime. It is any Class that implements the Interface Example.DepositListener.

Reference: `setDepositListener()`

## Design Support Library

The Android Design Library package provides APIs to support adding material design components and patterns to your apps. It adds support for various material design components and patterns for app developers to build upon, such as navigation drawers, floating action buttons (FAB), snackbars, and tabs.

Reference: Design Support Library (<http://developer.android.com/tools/support-library/features.html#design>)

### destroy()

A Example SDK Java Method of the Example Class that recycles all held instances.

Reference: `destroy()`

## Discloser

Means a Party that discloses any of its Confidential Information to the other Party.

Reference: Example EAPA

## Documentation

Means the user manuals, technical manuals, specifications and other documentation relating to any Example Services furnished by Example to Customer under the Example EAPA Agreement.

Reference: Example EAPA

## E

### errorMessage

A parameter of the `onBetcadeMarketError()` Method which belongs to the Example.MarketListener Interface. Its Java data type is String. It is called by the SDK when an error has occurred during any interactions, returning an error message to the developer stating the reason.

Reference: `onBetcadeMarketError()`

### Evaluation Period

Means the period commencing on the date the Example Services are made available to Customer and ending December 30, 2016.

Reference: Example EAPA

## F

### Feedback

Means information and feedback (including, without limitation, questions, comments, suggestions, or the like) regarding the performance, features, functionality and your overall experience using the Example Services.

Reference: Example EAPA

## G

### Geolocation

An Android system user permission. Example requires geolocation to provide location-relevant search results.

Reference: Geolocation (<http://developer.android.com/reference/android/webkit/GeolocationPermissions.html>)

### Get Accounts

An Android system user permission. This authenticates and synchronizes Example account information between apps and devices.

Reference: Get Accounts ([http://developer.android.com/reference/android/Manifest.permission.html#GET\\_ACCOUNTS](http://developer.android.com/reference/android/Manifest.permission.html#GET_ACCOUNTS))

### Google Play Services

Google Play services enables your app to take advantage of the latest, Google-powered features such as Maps, Google+, etc. with automatic platform updates distributed as an APK through the Google Play store.

Reference: Overview: Google Play Services (<https://developers.google.com/android/guides/overview>)

### Gradle

Gradle is the official build system for Android Studio.

Reference: Getting Started with Gradle (<https://gradle.org/getting-started-android/>)

### Gson

Gson is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object.

Reference: Gson User Guide (<https://github.com/google/gson/blob/master/UserGuide.md#TOC-Overview>)

## H

## I

### *init(Activity activity)*

A Example SDK Java Method with Parameters of the **Example** Class that initializes the singleton instance of Example.

*Reference:* **init(Activity activity)**

### *Integration (SDK)*

This Example SDK document provides steps to perform a Example Android SDK integration using the Android Studio IDE.

*Reference:* SDK Integration using Android Studio

### *Intellectual Property Rights*

Means any patent, copyright, trademark, service mark, trade name, trade secret, know-how, moral right or other intellectual property right under the laws of any jurisdiction, whether registered, unregistered, statutory, common law or otherwise (including any rights to sue, recover damages or obtain relief for any past infringement, and any rights under any application, assignment, license, legal opinion or search).

*Reference:* Example EAPA

### *Interface (Java)*

An interface is a reference type in Java, it is similar to class and is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface. Along with abstract methods an interface may also contain constants, default methods, static methods, and nested types.

*Reference:* What Is an Interface? (<https://docs.oracle.com/javase/tutorial/java/concepts/interface.html>)

### *Internet*

An Android system user permission. Example technologies require this to open network sockets.

*Reference:* Internet (<http://developer.android.com/reference/android/Manifest.permission.html#INTERNET>)

## J

### *Java Class Library*

The Example Java Class and its Interfaces, Methods, and Parameters are of primary importance and use during the integration process. They are documented in this page.

*Reference:* Example Android SDK Java Class Library

## K

## L

### *Location Services*

The Example Android SDK requires the Android OS Location Services package. Android gives your applications access to the location services supported by the device through classes in the **android.location** package. The central component of the location framework is the **LocationManager** system service, which provides APIs to determine location and bearing of the underlying device (if available).

*Reference:* Location Services (<http://developer.android.com/guide/topics/location/index.html#location>)

## M

### **marketListener**

A parameter of the `setMarketListener()` Method which belongs to the Example Class. Its Java data type is `Example.MarketListener`. It is a Class instance provided by the developer at runtime. It is any Class that implements the Interface `Example.MarketListener`

Reference: `setMarketListener()`

### **Materials**

Means content of any data or information that Customer provides to the Example Services.

Reference: Example EAPA

### **Method (Java)**

A method is a set of code which is referred to by name and can be called (invoked) at any point in a program simply by utilizing the method's name. Think of a method as a subprogram that acts on data and often returns a value. Each method has its own name. When that name is encountered in a program, the execution of the program branches to the body of that method. When the method is finished, execution returns to the area of the program code from which it was called, and the program continues on to the next line of code.

Reference: Defining Methods (<https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html>)

## N

## O

### **onActivityResult(int requestCode, int resultCode, Intent data)**

A Example SDK Method of the `Example` Class with Parameters used for the results of activities.

Reference: `onActivityResult(int requestCode, int resultCode, Intent data)`

### **onBetcadeDepositCancel()**

A `Example.DepositListener` Method that signals a canceled deposit.

Reference: `onBetcadeDepositCancel()`

### **onBetcadeDepositFail()**

A `Example.DepositListener` Method that signals a failed deposit.

Reference: `onBetcadeDepositFail()`

### **onBetcadeDepositSuccess(double deposit)**

A `Example.DepositListener` Method that signals a successful deposit.

Reference: `onBetcadeDepositSuccess(double deposit)`

### **OnIncentiveListener - Public Interface**

This is a Listener for detecting if an app was launched with incentives or without incentives. It contains two Methods:

- `onReceivedIncentives(Activity, redeemId, redeemAppId, redeemAmount, redeemPercent)`
- `onSimpleLaunch()`

Reference: `OnIncentiveListener`

### **onResume()**

A Example Java Method of the `Example` Class for the resumption of a game.

Reference: `onResume()`

### **Operator information**

The operator information page promotes your brand and all of the gaming apps you're offering on the Example store. It should consist of the following items:

- Your featured image
- Your operator logo
- Your operator description
- Your operator apps

Reference: Example Operator Information (<http://sb.dicemedia.com/Example.com/developer/web/operator-info-page/>)

## P

### **Parameter (Java)**

In Java there are two types of parameters, implicit parameters and explicit parameters. Explicit parameters are the arguments passed into a method. The implicit parameter of a method is the instance that the method is called from. Arguments are simply one of the two types of parameters.

Reference: Passing Information to a Method or a Constructor (<https://docs.oracle.com/javase/tutorial/java/javaOO/arguments.html>)

### **Partners**

Example partners are gambling game app developers and operators.

Reference: How To Make Money With Casino Apps ([http://www.bluecloudsolutions.com/source-codes/party\\_slots\\_b/#](http://www.bluecloudsolutions.com/source-codes/party_slots_b/#))

### **Party**

Means Example or Customer.

Reference: Example EAPA

### **Policies**

Means Example's acceptable use policy and Example's privacy policy as found on the Site, all restrictions described on the Site, and any other policy or terms incorporated by reference into this Example EAPA Agreement.

Reference: Example EAPA

## Q

## R

### **Read External Storage (Read/Write)**

This is an Android user permission that is required to read and write from an external storage.

*Reference:* Read External Storage (Read/Write) ([http://developer.android.com/reference/android/Manifest.permission.html#READ\\_EXTERNAL\\_STORAGE](http://developer.android.com/reference/android/Manifest.permission.html#READ_EXTERNAL_STORAGE))

### **Read Phone State**

This is an Android System user permission that is required to get Device specific details such as device id, OS, etc.

*Reference:* Read Phone State ([http://developer.android.com/reference/android/Manifest.permission.html#READ\\_PHONE\\_STATE](http://developer.android.com/reference/android/Manifest.permission.html#READ_PHONE_STATE))

### **Recipient**

Means a Party that receives any Confidential Information of the other Party.

*Reference:* Example EAPA

### **recyclerview-v7 Library**

The recyclerview library adds the RecyclerView class. This class provides support for the RecyclerView widget, a view for efficiently displaying large data sets by providing a limited window of data items.

*Reference:* recyclerview-v7 Library (<http://developer.android.com/tools/support-library/features.html#v7-recyclerview>)

### **requestCode**

A parameter of the onActivityResult() Method which belongs to the Example Class. Its Java data type is integer. It is a unique request code, numeric, provided by the developer when the call was started (unused in the SDK, default for platform signature).

*Reference:* onActivityResult()

### **resultCode**

A parameter of the onActivityResult() Method which belongs to the Example Class. Its Java data type is integer. It is a unique result code returned to the developer to indicate the operation status (used by the SDK to capture external responses).

*Reference:* onActivityResult()

## S

### **SDK (Android)**

The Example Android SDK is integrated into your individual casino games. It provides the integration with the Example Store Application and payment services for your applications.

*Reference:* See the latest version of the Example Android SDK.

### **setDepositListener(Example.DepositListener depositListener)**

A Example SDK Java Method of the Example Class with an Interface and Parameter that sets the deposit listener.

*Reference:* `setDepositListener(Example.DepositListener depositListener)`

### **setMarketListener(Example.MarketListener marketListener)**

A Example SDK Java Method of the Example with an Interface and Parameter that sets the market listener.

*Reference:* `setMarketListener(Example.MarketListener marketListener)`

### **Site**

Means [www.Example.com](http://www.Example.com) (<http://www.Example.com>), and any successor or related site designated by Example.

*Reference:* Example EAPA

### **Software**

Means any down-loadable software made available by Example and required for use of the Example Services.

*Reference:* Example EAPA

### **Store Application**

The Example Store Application (.apk) will be installed by consumers onto their Android devices. It allows a consumer to browse, search for, install, and update casino gaming applications. It requires the consumer to create a user account which is registered with the Example Core System, and provide a payment method which is registered with the Example Transaction System. The payment method is validated through the Bank System and stored in the Transaction System.

*Reference:* See the Example Android SDK "Getting Started" page System Architecture overview section.

### **support-v4**

The Android Support Library package is a set of code libraries that provide backward-compatible versions of Android framework APIs as well as features that are only available through the library APIs.

*Reference:* Android Support Libraries (<http://developer.android.com/tools/support-library/index.html#>)

### **Support Library Setup**

How you setup the Android Support Libraries in your development project depends on what features you want to use and what range of Android platform versions you want to support with your application. This document guides you through downloading the Support Library package and adding libraries to your development environment.

*Reference:* Support Library Setup (<https://developer.android.com/tools/support-library/setup.html>)

## T

### **Trademarks**

As between Example and Customer, Example owns all right, title and interest in and to the Example Marks and any goodwill arising out of the use of the Example Marks will remain with and belong to Example and its licensors. The Example Marks may not be copied, imitated or used without the prior written consent of Example or the applicable trademark holder.

*Reference:* Example EAPA

### **Transaction System**

The Example Transaction System is responsible for all payment-related activities, including management of payment processing interactions and user payment methods.

*Reference:* Example Android SDK Getting Started > System architecture overview

## U

### **Unauthorized Use**

Means any use, reproduction, modification, distribution, disposition, possession, examination, inspection, viewing, disclosure or other activity involving the Example Services, Documentation or Confidential Information of Example that is not expressly authorized under the Agreement or otherwise in writing by Example.

*Reference:* Example EAPA

### **User Permissions**



The Example Android SDK requires specific user permissions to operate effectively. Android applications should add the following permissions to their user permission model.

- Geolocation (<http://developer.android.com/reference/android/webkit/GeolocationPermissions.html>) : Example requires geolocation to provide location-relevant search results.
- Internet (<http://developer.android.com/reference/android/Manifest.permission.html#INTERNET>) : Required to open network sockets.
- Access Network State ([http://developer.android.com/reference/android/Manifest.permission.html#ACCESS\\_NETWORK\\_STATE](http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_NETWORK_STATE)) : Required to access information about networks.
- Access Wifi State ([http://developer.android.com/reference/android/Manifest.permission.html#ACCESS\\_WIFI\\_STATE](http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_WIFI_STATE)) : Required to access information about Wifi networks.
- Read Phone State ([http://developer.android.com/reference/android/Manifest.permission.html#READ\\_PHONE\\_STATE](http://developer.android.com/reference/android/Manifest.permission.html#READ_PHONE_STATE)) : Required to get Device specific details such as device id, OS, etc.
- Read External Storage (Read/Write) ([http://developer.android.com/reference/android/Manifest.permission.html#READ\\_EXTERNAL\\_STORAGE](http://developer.android.com/reference/android/Manifest.permission.html#READ_EXTERNAL_STORAGE)) : Required to read and write from an external storage.
- Access Fine Location ([http://developer.android.com/reference/android/Manifest.permission.html#ACCESS\\_FINE\\_LOCATION](http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_FINE_LOCATION)) : Required to fetch to access precise location functionality.
- Access Coarse Location ([http://developer.android.com/reference/android/Manifest.permission.html#ACCESS\\_COARSE\\_LOCATION](http://developer.android.com/reference/android/Manifest.permission.html#ACCESS_COARSE_LOCATION)) : Required to access approximate location functionality.
- App History ([http://developer.android.com/reference/android/content/pm/PackageManager.html#getInstalledPackages\(int\)](http://developer.android.com/reference/android/content/pm/PackageManager.html#getInstalledPackages(int))) : Retrieve the list of installed apps.
- Get Accounts ([http://developer.android.com/reference/android/Manifest.permission.html#GET\\_ACCOUNTS](http://developer.android.com/reference/android/Manifest.permission.html#GET_ACCOUNTS)) : Authenticate and synchronize Example account information between apps and devices.
- Wake Lock ([http://developer.android.com/reference/android/Manifest.permission.html#WAKE\\_LOCK](http://developer.android.com/reference/android/Manifest.permission.html#WAKE_LOCK)) : Required for certain synchronization events to be handled in the background.

## V

### *verifyMarketAuthorization()*

A Example SDK Method of the **Example** Class that verifies the Example app is installed and a user is logged in.

Reference: **verifyMarketAuthorization()**

## W - Z

### *Wake Lock*

An Android system user permission. Example technologies require this for certain synchronization events to be handled in the background.

Reference: Wake Lock ([http://developer.android.com/reference/android/Manifest.permission.html#WAKE\\_LOCK](http://developer.android.com/reference/android/Manifest.permission.html#WAKE_LOCK))

## Android SDK v1.0 beta 4

Release date: 2011.01.10

Security classification: PUBLIC

### Early Adopter Program Agreement

**Note:** This is a copy of the Example Early Adopter Program Agreement.

The Early Adopter Program Agreement (the "Agreement") between Example, LLC ("Example") and the Customer.

Example and Customer agree as follows:

#### Section 1. Definitions

Whenever used in this Agreement with initial letters capitalized, the following terms will have the following specified meanings:

**"Example Marks"**

Means any trademarks, service marks, service or trade names, logos, and other designations of Example and its affiliates.

**"Example Parties"**

Means Example and its affiliates, independent contractors and service providers, and each of their respective members, directors, officers, employees and agents.

**"Example Services"**

Means Example's app store, together with any Software provided or made available by Example.

**"Confidential Information"**

Means any information that is proprietary or confidential to the Discloser or that the Discloser is obligated to keep confidential (e.g., pursuant to a contractual or other obligation owing to a third party). Confidential Information may be of a technical, business or other nature (including, but not limited to, information which relates to the Discloser's technology, research, development, products, services, pricing of products and services, customers, employees, contractors, marketing plans, finances, contracts, legal affairs, or business affairs). However, Confidential Information does not include any information that: (a) was known to the Recipient prior to receiving the same from the Discloser in connection with this Agreement; (b) is independently developed by the Recipient; (c) is acquired by the Recipient from another source including any end user of the Example Services without restriction as to use or disclosure; or (d) is or becomes part of the public domain through no fault or action of the Recipient.

**"Discloser"**

Means a Party that discloses any of its Confidential Information to the other Party.

**"Documentation"**

Means the user manuals, technical manuals, specifications and other documentation relating to any Example Services furnished by Example to Customer under this Agreement.

**"Evaluation Period"**

Means the period commencing on the date the Example Services are made available to Customer and ending December 30, 2016.

**"Feedback"**

Means information and feedback (including, without limitation, questions, comments, suggestions, or the like) regarding the performance, features, functionality and your overall experience using the Example Services.

**Example EARLY ADOPTER PROGRAM AGREEMENT**  
116161-0001/128648743.5

**"Intellectual Property Rights"**

Means any patent, copyright, trademark, service mark, trade name, trade secret, know-how, moral right or other intellectual property right under the laws of any jurisdiction, whether registered, unregistered, statutory, common law or otherwise (including any rights to sue, recover damages or obtain relief for any past infringement, and any rights under any application, assignment, license, legal opinion or search).

**"Materials"**

Means content of any data or information that Customer provides to the Example Services.

**"Party"**

Means Example or Customer.

**"Policies"**

Means Example's acceptable use policy and Example's privacy policy as found on the Site, all restrictions described on the Site, and any other policy or terms incorporated by reference into this Agreement.

**"Recipient"**

means a Party that receives any Confidential Information of the other Party.

**"Site"**

Means [www.Example.com](http://www.Example.com) (*http://www.Example.com*), and any successor or related site designated by Example.

**"Software"**

Means any downloadable software made available by Example and required for use of the Example Services.

**"Unauthorized Use"**

Means any use, reproduction, modification, distribution, disposition, possession, examination, inspection, viewing, disclosure or other activity involving the Example Services, Documentation or Confidential Information of Example that is not expressly authorized under the Agreement or otherwise in writing by Example.

**Section 2. License****2.1 License to Use Example Services.**

Subject to the terms of this Agreement, Example hereby grants to Customer a limited, nonexclusive, nontransferable, nonsublicensable, revocable, royalty free license to use the Example Services solely for Customer's own internal testing and evaluation of the Example Services for the Evaluation Period.

**2.2 Software.**

If Customer receives Software from Example, including, without limitation Software downloaded separate from this Agreement or from a third party marketplace, its use is governed in one of two ways: if Customer is presented with license terms that it must accept in order to use the Software, those terms apply; if no license is presented, this Agreement applies. Example reserves all other rights to the Software. Any Software is licensed, not sold. Unless Example notifies Customer otherwise, the Software license ends when Customer's Example Services end. Customer must then promptly uninstall the Software from all of Customer's computers or devices, or Example may disable it. Customer must not work around any technical limitations in the Software.

**2.3 Restrictions; Limitations.**

Customer may not use the Example Services in any manner or for any purpose other than as expressly permitted by this Agreement. Without limitation of the foregoing, the license granted under this Section 2 does not include or authorize: (a) publicly performing or publicly displaying any of the Example Services; (b) modifying or otherwise making any derivative use of any of the Example Services; (c) using any data mining, robots or similar data gathering or extraction methods; (d) reverse engineering or accessing any of the Example Services to build a competitive product or service; or (e) using any of the Example Services other than for their intended use. During and after the Evaluation Period, Customer will not assert, nor authorize, assist or encourage any third party to assert, against any of the Example Parties, any patent infringement or other intellectual property infringement claim regarding any Example Services Customer has used. The license granted under this Section 2 is conditioned on Customer's continued compliance this Agreement (including, without limitation, its compliance with the Policies), and will immediately and automatically terminate if Customer does not comply with any material term or condition of this Agreement.

**Example EARLY ADOPTER PROGRAM AGREEMENT**

116161-0001/128648743.5

### **Section 3. Confidential Information**

Each Party reserves any and all right, title and interest (including, without limitation, any Intellectual Property Rights) that it may have in or to any Confidential Information that it may disclose to the other Party under this Agreement. The Recipient will protect Confidential Information of the Discloser against any unauthorized use or disclosure to the same extent that the Recipient protects its own Confidential Information of a similar nature against unauthorized use or disclosure, but in no event will use less than a reasonable standard of care to protect such Confidential Information; provided that the Confidential Information of the Discloser is conspicuously marked or otherwise identified as confidential or proprietary upon receipt by the Recipient or the Recipient otherwise knows or has reason to know that the same is Confidential Information of the Discloser. The Recipient will use any Confidential Information of the Discloser solely for the purposes for which it is provided by the Discloser. This paragraph will not be interpreted or construed to prohibit: (a) any use or disclosure which is necessary or appropriate in connection with the Recipient's performance of its obligations or exercise of its rights under this Agreement or any other agreement between the Parties; (b) any use or disclosure required by applicable law (e.g., pursuant to applicable securities laws or legal process), provided that the Recipient uses reasonable efforts to give the Discloser reasonable advance notice thereof (e.g., so as to afford the Discloser an opportunity to intervene and seek an order or other appropriate relief for the protection of its Confidential Information from any unauthorized use or disclosure); or (c) any use or disclosure made with the consent of the Discloser. In the event of any breach or threatened breach by the Recipient of its obligations under this paragraph, the Discloser will be entitled to injunctive and other equitable relief to enforce such obligations.

### **Section 4. Proprietary Rights**

#### **4.1 The Example Services.**

As between Example and Customer, Example owns all right, title, and interest in and to the Example Services. Except as otherwise specified in Sections 2.1 and 2.2 of this Agreement, Customer does not obtain any rights under this Agreement from Example to the Example Services, including any related Intellectual Property Rights.

#### **4.2 Feedback.**

Customer will at its risk and expense use reasonable efforts to provide Example with reasonable Feedback. Customer will not provide any such Feedback to any third party without Example's prior written consent in each instance. Customer hereby grants to Example an exclusive, royalty-free, irrevocable, perpetual worldwide right and license to reproduce, use, disclose, exhibit, display, transform, create derivative works and distribute any such Feedback without limitation. Further, Example will be free to use any ideas, concepts, know-how or techniques contained in such Feedback for any purpose whatsoever, including, without limitation, developing, making, marketing, distributing and selling products and services incorporating such Feedback. Customer should not expect any compensation of any kind from Example with respect to Feedback. Example will exclusively own any improvements or modifications to the Example services based on or derived from any Feedback, including all Intellectual Property Rights therein or thereto.

#### **4.3 Trademarks.**

As between Example and Customer, Example owns all right, title and interest in and to the Example Marks and any goodwill arising out of the use of the Example Marks will remain with and belong to Example and its licensors. The Example Marks may not be copied, imitated or used without the prior written consent of Example or the applicable trademark holder.

**Example EARLY ADOPTER PROGRAM AGREEMENT  
116161-0001/128648743.5**

#### **4.4 Additional Protection of Proprietary Rights.**

Customer will not infringe or violate, and will take appropriate steps and precautions for the protection of, the Example Services and related Intellectual Property Rights referred to in Section 4.1. Without limiting the generality of the foregoing, Customer will not: (a) make any Example Services or Documentation available to any third party without the prior written consent of Example; (b) reverse engineer, disassemble or decompile the Example Services or otherwise attempt to discover or recreate the source code to the Example Services; (c) interfere or attempt to interfere in any manner with the functionality or proper working of the Example Services; (d) except as permitted by Example, create, build, or otherwise develop a commercial website utilizing the Example Services; (e) remove, obscure or alter any notice of copyright or other Intellectual Property Right appearing in or as part of the Example Services; or (f) engage in or permit any Unauthorized Use. Customer will immediately notify Example of any Unauthorized Use that comes to Customer's attention. In the event of any Unauthorized Use relating to the activities of Customer, or any employees, agents, or representatives of Customer, Customer will take all steps reasonably necessary to terminate such Unauthorized Use. Customer will immediately notify Example of any legal proceeding initiated by Customer in connection with any such Unauthorized Use. Example may, at its option and expense, assume control of such proceeding. If Example assumes such control, Example will have exclusive control over the prosecution and settlement of the proceeding, and Customer will provide such assistance related to such proceeding as Example may reasonably request. Customer will assist Example in enforcing any settlement or order made in connection with such proceeding.

#### **Section 5. Termination.**

Either Party may terminate this Agreement at any time by written notice of termination to the other Party. Unless otherwise agreed to by the Parties in writing, upon the expiration of the Evaluation Period or the termination of this Agreement by either Party as provided above, the license granted to Customer in this Agreement will terminate. Sections 1, 2.3, 3, 4, 6, 7 and 8 shall survive the termination or expiration of this Agreement.

#### **Section 6. Limited Warranties and Remedies**

##### **6.1 DISCLAIMER.**

Customer acknowledges that the Example Services are a beta test version, are not ready for general commercial release and may contain bugs, errors and defects. Accordingly, the Example Services are provided by Example and accepted by Customer "AS IS" and "WITH ALL FAULTS, DEFECTS AND ERRORS." Example WILL HAVE NO LIABILITY FOR ANY ERROR, OMISSION OR DEFECT IN THE Example SERVICES, OR ANY INABILITY TO ACCESS OR USE THE Example SERVICES. Example MAKES NO, AND HEREBY DISCLAIMS, ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, ARISING BY LAW OR OTHERWISE, REGARDING THE Example SERVICES, ANY SERVICE PROVIDED IN CONNECTION WITH THE Example SERVICES OR ANY OTHER ITEMS, INFORMATION, MATERIALS OR DATA PROVIDED BY Example IN CONNECTION WITH THE EVALUATION OF THE Example SERVICES, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT.

#### **Section 7. Limitations of Liability**

IN NO EVENT SHALL ANY OF THE Example PARTIES BE LIABLE FOR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY OTHER DAMAGES OF ANY KIND, INCLUDING BUT NOT LIMITED TO LOSS OF USE, LOSS OF PROFITS OR LOSS OF DATA, WHETHER IN AN ACTION IN CONTRACT, TORT (INCLUDING BUT NOT LIMITED TO NEGLIGENCE) OR OTHERWISE, ARISING OUT OF OR IN ANY WAY CONNECTED WITH THE USE OF OR INABILITY TO USE THE Example SERVICES OR THE Example SITE,

[Example EARLY ADOPTER PROGRAM AGREEMENT](#)

116161-0001/128648743.5

INCLUDING THE INFORMATION, CONTENT AND MATERIALS CONTAINED THEREIN, INCLUDING WITHOUT LIMITATION ANY DAMAGES CAUSED BY OR RESULTING FROM RELIANCE BY CUSTOMER ON ANY INFORMATION OBTAINED FROM Example, OR THAT RESULT FROM MISTAKES, OMISSIONS, INTERRUPTIONS, DELETION OF FILES, ERRORS, DEFECTS, VIRUSES, DELAYS IN OPERATION OR TRANSMISSION OR ANY FAILURE OF PERFORMANCE, WHETHER OR NOT RESULTING FROM ACTS OF GOD, COMMUNICATIONS FAILURE, THEFT, DESTRUCTION OR UNAUTHORIZED ACCESS TO Example'S RECORDS, PROGRAMS, Example SERVICE OR THE Example SITE. IN NO EVENT SHALL THE AGGREGATE LIABILITY OF ANY OF THE Example PARTIES, WHETHER IN CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE, WHETHER ACTIVE, PASSIVE OR IMPUTED), PRODUCT LIABILITY, STRICT LIABILITY OR OTHER THEORY, ARISING OUT OF OR RELATING TO THIS AGREEMENT EXCEED \$100.

## **Section 8. Miscellaneous**

### **8.1 Independent Contractors.**

Each Party is an independent contractor and not a partner or agent of the other. This Agreement will not be interpreted or construed as creating or evidencing any partnership or agency between the Parties or as imposing any partnership or agency obligations or liability upon either Party. Further, neither Party is authorized to, and will not, enter into or incur any agreement, contract, commitment, obligation or liability in the name of or otherwise on behalf of the other Party.

### **8.2 Assignment.**

Customer may not assign this Agreement or any right, interest or benefit under this Agreement without prior written consent of Example. Any attempted assignment in violation of the foregoing will be void. Subject to the foregoing, this Agreement will be fully binding upon, inure to the benefit of and be enforceable by any permitted assignee.

### **8.3 Dispute Resolution Procedures.**

(a) The Parties will attempt to resolve through good faith discussion any dispute that arises under this Agreement. Any such dispute may at any time, at the election of either Party, be referred to a senior executive of each Party for discussion and possible resolution. If the senior executives are unable to resolve the dispute within sixty (60) days after delivery of written notice of the dispute, then either Party may, by notice to the other Party, demand mediation under the mediation rules of JAMS in Los Angeles, California. The Parties give up their right to litigate their disputes and may not proceed to arbitration without first attempting mediation, except that the Parties are NOT required to arbitrate any dispute in which either Party seeks equitable and other relief from the alleged unlawful use any Intellectual Property Rights by the other Party. Whether the dispute is heard in arbitration or in court, the Parties will not commence against the other a class action, class arbitration or other representative action or proceeding.

(b) If settlement is not reached within sixty (60) days after service of a written demand for mediation, any unresolved controversy or claim shall be resolved by arbitration in accordance with the rules of JAMS before a single arbitrator in Los Angeles, California. The language of all proceedings and filings shall be English. The arbitrator shall render a written opinion including findings of fact and law and the award and/or determination of the arbitrator shall be binding upon the Parties, and their respective administrators and assigns, and shall not be subject to appeal. Judgment may be entered upon the award of the arbitrator in any court of competent jurisdiction. The expenses of the arbitration shall be shared equally by the Parties unless the arbitration determines that the expenses shall be otherwise assessed and the prevailing Party may be awarded its attorneys' fees and expenses by the arbitrator. It is the intent of the Parties that, barring extraordinary circumstances, arbitration proceedings shall be concluded within ninety (90) days from the date the arbitrator is appointed. The arbitrator may extend this time limit only if failure to do so would unduly prejudice the rights of the Parties. Failure to adhere to this time limit shall not constitute a basis for challenging the award. Consistent with the expedited nature of arbitration, pre-hearing information exchange shall be limited to the reasonable production of relevant, non-privileged documents, carried out expeditiously.

**Example EARLY ADOPTER PROGRAM AGREEMENT**  
**116161-0001/128648743.5**

**8.4 Severability.**

If any provision of this Agreement shall be deemed unlawful, void or for any reason unenforceable, then that provision shall be deemed severable from this Agreement and shall not affect the validity and enforceability of any remaining provisions.

**8.5 Applicable Law.**

This Agreement will be interpreted, construed and enforced in all respects in accordance with the laws of the State of Delaware, U.S.A., without reference to its choice of law principles to the contrary. Subject to Section 8.3, Customer hereby consents to the jurisdiction and venue of the state and federal courts located in Los Angeles County, State of California, U.S.A. with respect to any claim arising under or by reason of this Agreement.

**8.6 Entire Agreement.**

This Agreement, together with any agreement, policy or guideline referenced in this Agreement, constitutes the complete and exclusive statement of all mutual understandings between the parties with respect to the subject matter hereof, superseding all prior or contemporaneous proposals, communications and understandings, oral or written.

[Example EARLY ADOPTER PROGRAM AGREEMENT](#)

116161-0001/128648743.5