



Cuelinks SDK

Integration Guide

Version: 1.0.0

Integration Guide for Android SDK

Date: 2016-06-07

All rights reserved

Important notice: The content in this document is for the use of the intended recipient only. If the the recipient is not the intended recipient, please notify us immediately and remove/delete this document irretrievably from your physical and/ or electronic records.

The content of this document including design, text, graphic and selection and arrangement thereof, is the property of Parity Cube Pvt Ltd. The content in this document is copyrighted by Parity Cube Pvt Ltd (© Parity Cube Pvt Ltd 2016, all rights reserved) and the information described herein may be protected by one or more of Parity Cube's Intellectual Property registrations or pending applications.. Except as specifically provided otherwise in writing , no part of this document may be reproduced in any form by any means without prior written authorization of Parity Cube. All rights not expressly granted herein are reserved. Unauthorized use of any content in this document may violate copyright, trademark and other applicable laws, including confidentiality obligations and could result in criminal or civil penalties. All copyright, trademark, and other intellectual property and proprietary rights in this document and in the software, text, graphics, images, and all other materials originated or used by Parity Cube at document are the exclusive property of Parity Cube and its licensors. Content in this document is solely for informational purpose ; it is not intended as and does not constitute legal or tax advice. Any use of this document and the information described herein will be governed by these terms and conditions. By viewing, downloading or otherwise copying this document, you agree that you have read, understood, and will comply with all the terms and conditions set forth herein.

www.cuelinks.com



Table of Contents

1. Installation Steps	3
2. Additional Settings	7
a. Passing a SubId	7
b. Adding Internal Deeplink URLs	8
c. Overriding the default behaviour	8
3. Demo Project	9



1. Installation Steps

Step 1

Add the following in your app gradle file : compile

```
compile 'com.cuelinks.sdk:link-kit:1.0.+'
```

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 23
    buildToolsVersion "23.0.2"

    defaultConfig {
        applicationId "com.cuelinksdemo.demo"
        minSdkVersion 15
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.4.0'
    compile 'com.cuelinks.sdk:link-kit:1.0.+'
```



Step 2

Configure your Publisher Id (mandatory) in the AndroidManifest File.

Provide your `publisherId` (mandatory) in AndroidManifest file as :

```
<meta-data android:name="com.cuelinks.publisherId"
android:value="XXXXCL10XXX"/>
```

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.cuelinksdemo.demo">

  <application
    android:name=".CuelinksApplication"
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Demo"
    android:label="@string/app_name"
    android:theme="@style/AppTheme">

    <meta-data android:name="com.cuelinks.publisherId" android:value="11767CL10691"/>
    <meta-data android:name="com.cuelinks.internalUrl" android:value="desidime.com"/>
    <meta-data android:name="com.cuelinks.subId" android:value="appLevel"/>

    <activity android:name=".MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```



Step 3

Initialise the sdk in your app Application file as

```
@Override
public void onCreate() {
    super.onCreate();
    Cuelinks.initialize(this);
}
```

```
package com.cuelinksdemo.demo;

import android.app.Application;
import android.util.Log;

import com.cuelinks.Cuelinks;

/**
 * Created by tasneem on 25/05/16.
 */
public class CuelinksApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        Cuelinks.initialize(this);
        Log.i("CuelinksApplication", "publisher id :"+ Cuelinks.getPublisherId());
    }
}
```



Step 4

Now you are ready to use the SDK , wherever you want to convert your link you can use the followings in your TextView, default behaviour is if you want to redirect all the links to browser.

Adding affiliate links in normal text:

In the activity layout file add

```
android:autoLink="all"
```

```
<TextView  
    android:layout_marginTop="10dp"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:autoLink="all"  
    android:textColor="@color/colorPrimary"  
    android:textColorLink="@color/colorPrimary"  
    android:textStyle="bold"/>
```

```
textView.setMovementMethod(CuelinkMovementMethod.getInstance(MainActivity.this));
```

Adding affiliate links in html text :

```
textView.setText(CuelinkSpan.affiliateHrefUrlsFromHtml(HtmlText, textView, MainActivity.this));
```



2. Additional Settings

1. Passing a SubId

Configuring subId in your affiliate links also possible with default and dynamic subId configurations. You can configure the default subId (optional) for in the AndroidManifest

```
<meta-data android:name="com.cuelinks.subId"
android:value="yoursubid"/>
```

pass it dynamically in with the links.

Dynamic subId in normal text

```
textView.setMovementMethod(CuelinkMovementMethod.getInstance(MainActivity.this, subId));
```

Dynamic subId in html text

```
textView.setText(CuelinkSpan.affiliateHrefUrlsFromHtml(HtmlText, textView, MainActivity.this, subId));
```



2. Adding Internal Deeplink URLs

Configure the base URL of your internal deeplink URL's so that the SDK doesn't affiliate those URLs. You can configure the `internalUrl` (optional) for in the AndroidManifest

```
<meta-data android:name="com.cuelinks.internalUrl"
android:value="example.com"/>
```

3. Overriding the default behaviour

If you wish to override the default method with a custom method:

Implement interface `CuelinkListener` and override method `openUrl` in it, as per your requirements. Note: Here the url will be affiliated already unless configured as `internalUrl` in Android Manifest file.

```
@Override
public void openUrl(String url, Context context) {
    Your code goes here..
}
```

And pass the reference of your listener as :

Overriding affiliate links in normal text

```
textView.setMovementMethod(CuelinkMovementMethod.getInstance(MainActivity.this, (CuelinkListener)MainActivity.this));
```

Overriding affiliate links in html text

```
textView.setText(CuelinkSpan.affiliateHrefUrlsFromHtml(HtmlText, textView, MainActivity.this, (CuelinkListener)MainActivity.this));
```




3. Demo Project

For a better understanding of the installation and working of Cuelinks SDK, you can refer the demo project mentioned below:

1. Clone project as ' `git clone git@github.com:TasneemBohra/cuelinkSDKDemo.git` '
2. Change publisherId, internalUrl and subId in Android Manifest file and observe the affiliated urls.