

# Documentation Android External DCB\_SDK

## Direct Carrier Billing Android

### Description

Documentation and example of DOCOMO Digital Direct Carrier Billing (from now on DCB) SDK for Android (Including Android TV)

Following this guide your app will be technically enabled to be promoted and integrated with Docomo Digital mobile payment.

After the integration of the sdk the resultant app can check if the user coming from docomo acquisition (optionally with explicit login with msdisdn+pin) and can check expiration date and several informations.

This integration can be used in apps already published on Google Play/App Store or only in case of Android environment, outside Google Play.

### Requirements

**Android Studio** <sup>™</sup> with Gradle 2.x+

Java/Kotlin

### Getting Started with Android java integration

We will provide the aar to integrate in you application

#### Step 1 / 5 - Add Maven URL

In the project build.gradle file add necessary maven url

**Note:** the build.gradle placed on the root of the android project

```
....

allprojects {
    repositories {
        jcenter()

        /* Used by docomo dcb sdk */
        maven { url "http://static.
newton.pm/android" }

    }
}

....
```

#### Step 2 / 5 - Add gradle dependencies

In the app build.gradle file add the new two dependencies

```

apply plugin: 'com.android.application'
....
    dependencies {
        ....
        /* The okhttp dependency is needed for docomo sdk.*/
        implementation 'com.squareup.okhttp3:okhttp:
3.10.0'
        implementation 'com.squareup.okhttp3:okhttp-
urlconnection:3.10.0'
        implementation 'com.buongiorno:newton:3.0.0
@aar'
        implementation "com.google.code.gson:gson:
2.8.6"
        implementation "org.jetbrains.kotlin:kotlin-
stdlib-jdk7:1.3.61"

    }

....

```

## Step 3 / 5 - Add manifest permissions

In your `AndroidManifest.xml` add (if is absent) one new permissions `INTERNET` and check which is your main `android.intent.category.LAUNCHER` Activity class, also be sure to extend `Application` class (here named `App`)

```

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

```

remember to add that line, to give the possibility to have call http and not only https (from android 10 is necessary)

```

<application ....

    android:
usesCleartextTraffic
="true"

....

```

the `distribution_group` is provided by Docomo Digital.

## Step 4 / 5 - Add the library bootstrap

In your APPLICATION add the call to

`DcbExternal.dcbWith(...)` just after the `super.onCreate()`

```
....  
  
import com.docomodigital.sdk.  
DcbExternal;  
  
public class Application extends  
android.app.Application {  
  
    @Override  
    public void onCreate() {  
        super.onCreate();  
  
        DcbExternal.dcbWith(this,  
"apikey", "product-id",  
"namespace", "us", "host", null,  
false);  
  
    ....  
}
```

All configuration parameters are provided by Docomo Digital. The last two arguments are "fingerprint host" and "isTest" and should be leaved as shown if not specified

## Step 5 / 5 - Call the "DcbRecognise" method to get the data

In your LAUNCHER Activity retrieve a Dcb instance and try to call recognise method

`DcbExternal.dcbRecognise(...)`

Just after the `super.onCreate(savedInstanceState)`

```

....

DcbExternal.dcbRecognise(new DcbCallback() {
    @Override
    public void onSuccess(DocomoUser dcbUser) {
        runOnUiThread() -> {
            //User is recognised as Docomo Acquisition
            ((TextView) findViewById(R.id.text)).setText("subscribed: " + dcbUser.isSubscribed);

            //SAVE dcbUser.utcExpirationUnixTime IN THE PREFERENCE TO CHECK THE SUBSCRIPTION IN THE
            APP
            if (dcbUser.utcExpirationUnixTime > System.currentTimeMillis() / 1000){
                //user is premium
                ((TextView) findViewById(R.id.text)).setText("subscribed: " + dcbUser.isSubscribed +
                    "; expireday unix: " + dcbUser.utcExpirationUnixTime +
                    "; expire date: " + new Date(dcbUser.utcExpirationUnixTime * 1000).
toString());
            }

            else{
                //user is expired
                ((TextView) findViewById(R.id.text)).setText("subscribed: " + dcbUser.isSubscribed +
                    "; expireday unix: " + dcbUser.utcExpirationUnixTime +
                    "; expire date: " + new Date(dcbUser.utcExpirationUnixTime * 1000).
toString());
            }

            if (dcbUser.isSubscribed) {
                //User is subscribed, goto ahead with product

                ((TextView) findViewById(R.id.text)).setText("subscribed: " + dcbUser.isSubscribed +
                    "; expireday unix: " + dcbUser.utcExpirationUnixTime +
                    "; expire date: " + new Date(dcbUser.utcExpirationUnixTime * 1000).
toString());
            } else {
                //User expired, not subscribed
                //user must pay again to access the product
                dcbUser.openAcquisitionPage(MainActivity.this);
                finish();
            }
        }
    }
});

@Override
public void onFailure(NewtonError docomoError) {
    runOnUiThread() -> {

        //Error informations are stored in newtonError Object.
        Log.d("", "failed recognise" + docomoError.getInfo());
        ((TextView) findViewById(R.id.text)).setText("subscription failed");

        //you're here because the user discovered the app on Google Play
        //go on with your code

    }
}
}
....

```

Hooray! You have finished the integration.

Do you have the expected result? No?

Get in touch with Native apps team [appsdocomodigital@gmail.com](mailto:appsdocomodigital@gmail.com)

Best from  
Native App Team