

## **Integration with Social Media applications.**

Android allows your application to connect to facebook and share data or any kind of updates on facebook. This chapter is about integrating facebook into your application.

There are two ways through which you can integrate facebook and share something from your application. These ways are listed below –

- Facebook SDK
- Intent Share

### **Integrating Facebook SDK**

This is the first way of connecting with facebook. You have to register your application and then receive some Application Id , and then you have to download the facebook SDK and add it to your project. The steps are listed below:

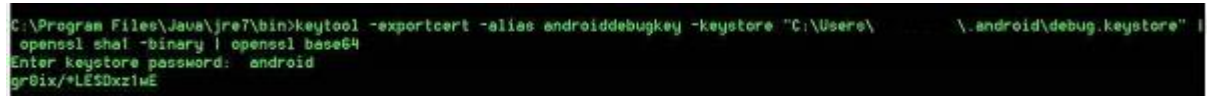
#### **Generating application signature**

You have to generate a key signature, but before you generate it, make sure you have SSL installed, otherwise you have to download SSL. It can be downloaded here.

Now open command prompt and redirect to your java jre folder. Once you reach there, type this command exactly. You have to replace the path in the inverted commas with your keystore path which you can found in eclipse by selecting the window tab and selecting the preferences tab and then selecting the build option under android from left side.

```
keytool -exportcert -alias androiddebugkey -keystore "your path"  
| openssl sha1 -binary | openssl base64
```

Once you enter it, you will be prompt for password. Give android as the password and then copy the key that is given to you. It is shown in the image below –



```
C:\Program Files\Java\jre7\bin>keytool -exportcert -alias androiddebugkey -keystore "C:\Users\  
.android\debug.keystore" |  
openssl sha1 -binary | openssl base64  
Enter keystore password: android  
gR6ix/*LE5Dxz1ME
```

#### **Registering your application**

Now create a new facebook application at developers.facebook.com/apps and fill all the information. It is shown below –

**Create New App**

App Name: [?] TutorialsPoint Valid

App Namespace: [?] sampletutorialspoint

App Category: [?] Education Choose a sub-category

By proceeding, you agree to the Facebook Platform Policies

Continue Cancel

Now move to the native android app section and fill in your project and class name and paste the hash that you copied in step 1. It is shown below –

**Native Android App**

Package Name: [?] com.example.facebook

Class Name: [?] com.example.facebook.MainActivity

Key Hashes: [?] gr0ix/+LESDxz1wE

Facebook Login: [?]  Enabled  Disabled

Deep Linking: [?]  Enabled  Disabled

If everything works fine, you will receive an application ID with the secret. Just copy the application id and save it somewhere. It is shown in the image below –

**TutorialsPoint**

App ID: 577787852259051

App Secret: 57ab9abdbf2f9a5fb1f081ba8a4eb093 (reset)

**This app is in Sandbox Mode** (Only visible to Admins, Developers and Testers)

### Downloading SDK and integrating it

Download facebook sdk here. Import this into eclipse. Once imported, right click on your facebook project and click on properties. Click on android, click on add button and select facebook sdk as the project. Click ok.

### Creating facebook login application

Once everything is complete , you can run the samples, that comes with SDK or create your own application. In order to login, you need to call **openActiveSession** method and implements its callback. Its syntax is given below –

```
// start Facebook LoginSession.openActiveSession(this, true, new
Session.StatusCallback() {
```

```
// callback when session changes state
public void call(Session session, SessionState state, Exception exception) {
    if (session.isOpened()) {
        // make request to;2 the /me API
        Request.executeMeRequestAsync(session, new Request.GraphUserCallback() {

            // callback after Graph API response with user object
            @Override
            public void onComplete(GraphUser user, Response response) {
                if (user != null) {
                    TextView welcome = (TextView) findViewById(R.id.welcome);
                    welcome.setText("Hello " + user.getName() + "!");
                }
            }
        });
    }
}
```

## Intent share

Intent share is used to share data between applications. In this strategy, we will not handle the SDK stuff, but let the facebook application handles it. We will simply call the facebook application and pass the data to share. This way, we can share something on facebook.

Android provides intent library to share data between activities and applications. In order to use it as share intent , we have to specify the type of the share intent to **ACTION\_SEND**. Its syntax is given below –

```
Intent shareIntent = new Intent();
shareIntent.setAction(Intent.ACTION_SEND);
```

Next thing you need to is to define the type of data to pass , and then pass the data. Its syntax is given below –

```
shareIntent.setType("text/plain");
```

```
shareIntent.putExtra(Intent.EXTRA_TEXT, "Hello, from tutorialspoint");  
startActivity(Intent.createChooser(shareIntent, "Share your thoughts"));
```

Apart from the these methods, there are other methods available that allows intent handling. They are listed below –

Sr.No	Method & description
1	<b>addCategory(String category)</b> This method add a new category to the intent.
2	<b>createChooser(Intent target, CharSequence title)</b> Convenience function for creating a ACTION_CHOOSER Intent
3	<b>getAction()</b> This method retrieve the general action to be performed, such as ACTION_VIEW
4	<b>getCategories()</b> This method return the set of all categories in the intent and the current scaling event
5	<b>putExtra(String name, int value)</b> This method add extended data to the intent.
6	<b>toString()</b> This method returns a string containing a concise, human-readable description of this object

### Example

Here is an example demonstrating the use of IntentShare to share data on facebook. It creates a basic application that allows you to share some text on facebook.

To experiment with this example, you can run this on an actual device or in an emulator.

Steps	Description
1	You will use Android studio to create an Android application under a package com.example.sairamkrishna.myapplication.
2	Modify src/MainActivity.java file to add necessary code.
3	Modify the res/layout/activity_main to add respective XML components.
4	Run the application and choose a running android device and install the application on it and verify the results.

Following is the content of the modified main activity file **MainActivity.java**.

```
package com.example.sairamkrishna.myapplication;
import android.content.Intent;import android.net.Uri;import android.os.Bundle;import
android.support.v7.app.AppCompatActivity;import android.view.View;
import android.widget.Button;import android.widget.ImageView;
import java.io.FileNotFoundException;import java.io.InputStream;
public class MainActivity extends AppCompatActivity {
    private ImageView img;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        img=(ImageView)findViewById(R.id.imageView);
        Button b1=(Button)findViewById(R.id.button);

        b1.setOnClickListener(new View.OnClickListener() {
            @Override
```

```

public void onClick(View v) {
    Intent sharingIntent = new Intent(Intent.ACTION_SEND);
    Uri screenshotUri = Uri.parse("android.
resource://comexample.sairamkrishna.myapplication/*");

    try {
        InputStream stream =
getContentResolver().openInputStream(screenshotUri);
    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    sharingIntent.setType("image/jpeg");
    sharingIntent.putExtra(Intent.EXTRA_STREAM, screenshotUri);
    startActivity(Intent.createChooser(sharingIntent, "Share image using"));
}
});
}}

```

Following is the modified content of the xml **res/layout/activity\_main.xml**.

In the below code **abc** indicates the logo of tutorialspoint.com

```

<?xml version="1.0" encoding="utf-8"?><RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".MainActivity">

```

<TextView

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/textView"  
    android:layout_alignParentTop="true"  
    android:layout_centerHorizontal="true"  
    android:textSize="30dp"  
    android:text="Facebook share " />
```

<TextView

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Tutorials Point"  
    android:id="@+id/textView2"  
    android:layout_below="@+id/textView"  
    android:layout_centerHorizontal="true"  
    android:textSize="35dp"  
    android:textColor="#ff16ff01" />
```

<ImageView

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/imageView"  
    android:layout_below="@+id/textView2"  
    android:layout_centerHorizontal="true"  
    android:src="@drawable/abc"/>
```

<Button

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Share"  
    android:id="@+id/button"  
    android:layout_marginTop="61dp"
```

```
android:layout_below="@+id/imageView"
android:layout_centerHorizontal="true" />
</RelativeLayout>
```


Following is the content of **AndroidManifest.xml** file.

```
<?xml version="1.0" encoding="utf-8"?><manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.sairamkrishna.myapplication" >
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:theme="@style/AppTheme" >

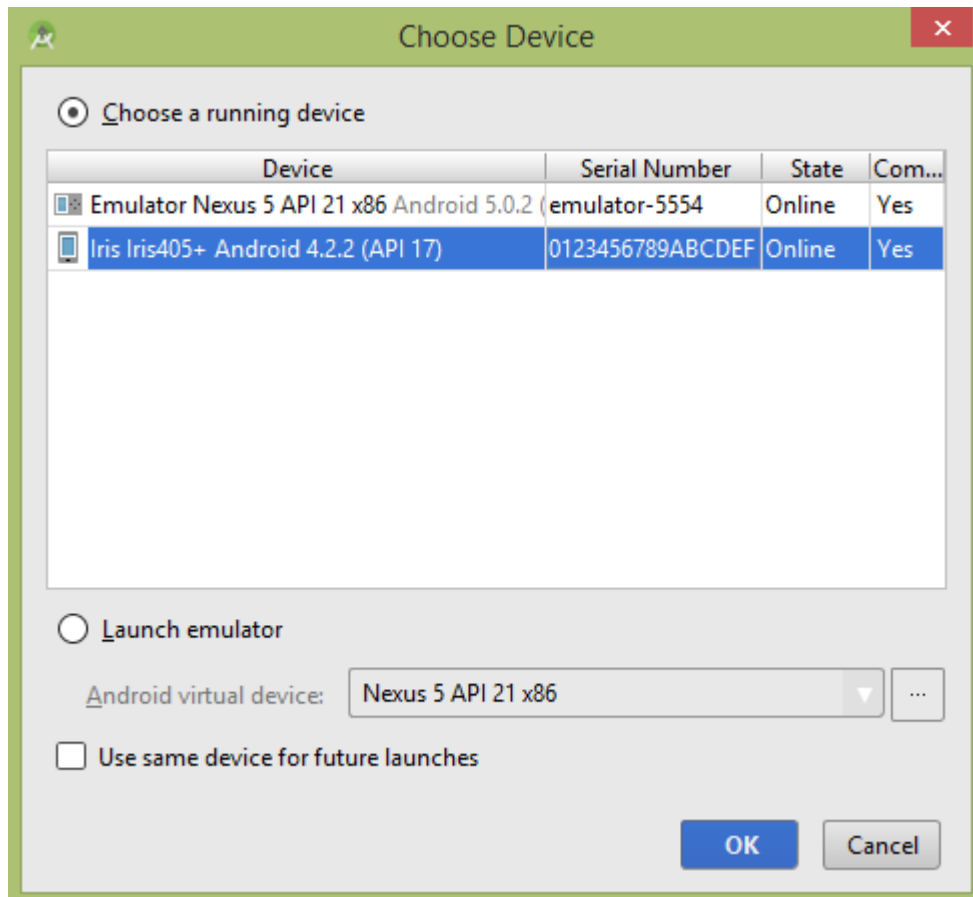
<activity
android:name=".MainActivity"
android:label="@string/app_name" >

<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>

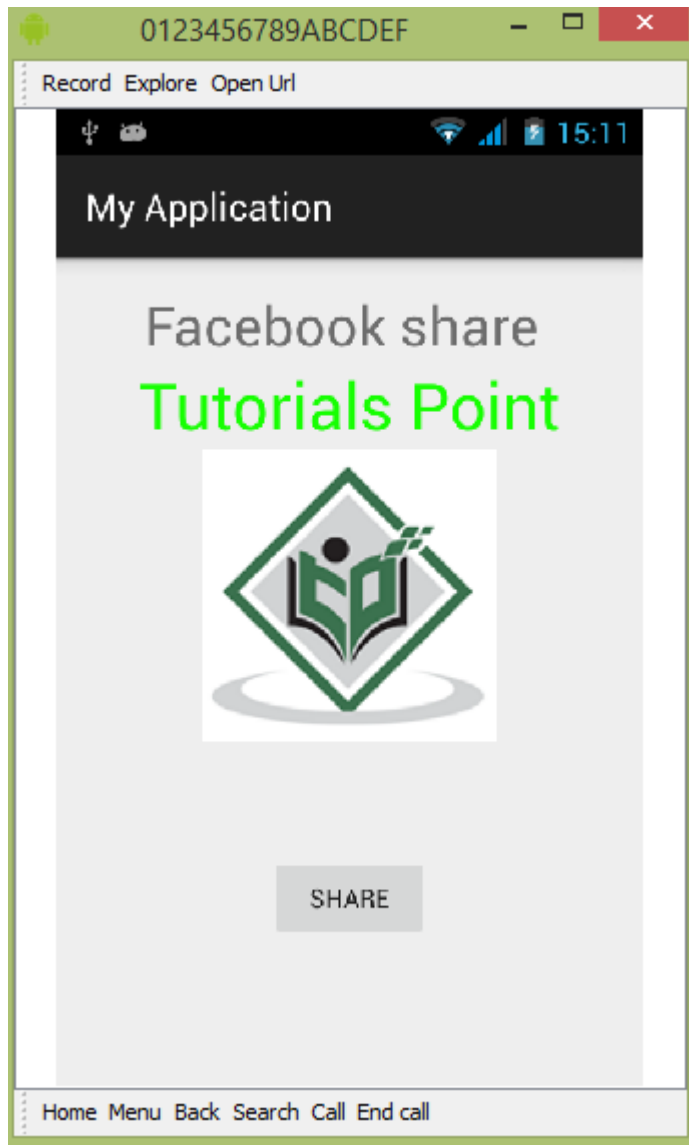
</activity>
</application></manifest>
```

Let's try to run your Application. I assume you have connected your actual Android Mobile device with your computer. To run the app from Android studio, open one of your project's activity files and click Run  icon from the toolbar. Before starting your application, Android studio will display following window to select an option where you want to run your Android application.

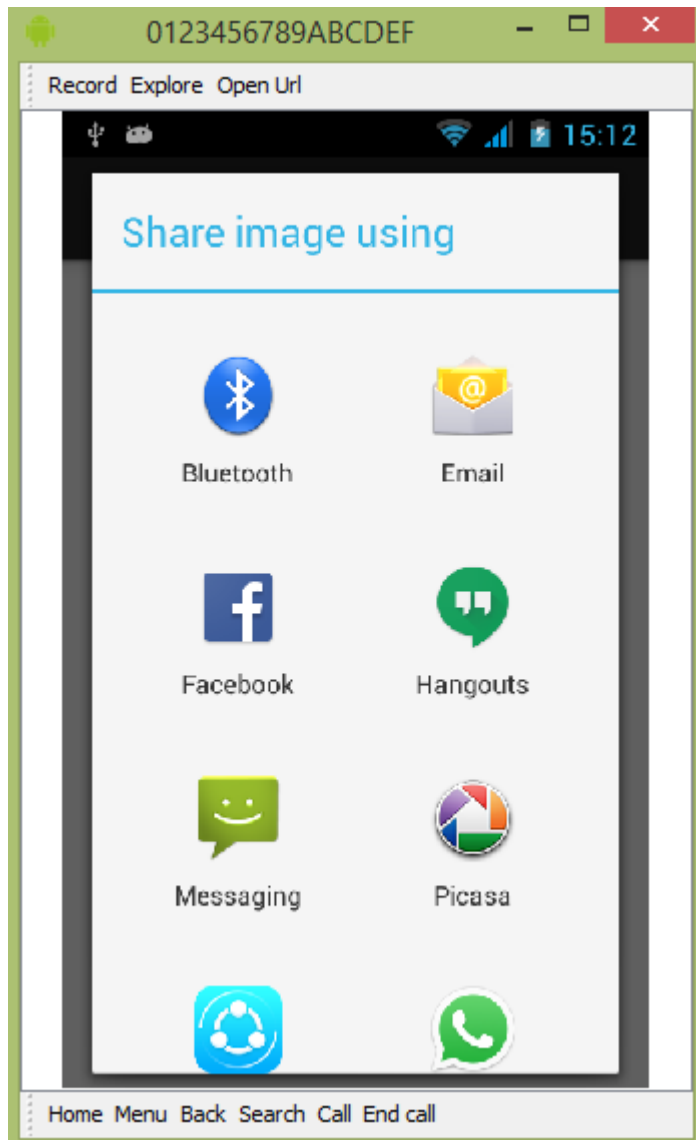




Select your mobile device as an option and then check your mobile device which will display your default screen –



Now just tap on the button and you will see a list of share providers.



Now just select facebook from that list and then write any message. It is shown in the image below –

