



Android - Sending Email

Advertisements

⬅ Previous Page

Next Page ➡

Email is messages distributed by electronic means from one system user to one or more recipients via a network.

Before starting Email Activity, You must know Email functionality with intent, Intent is carrying data from one component to another component with-in the application or outside the application.

To send an email from your application, you don't have to implement an email client from the beginning, but you can use an existing one like the default Email app provided from Android, Gmail, Outlook, K-9 Mail etc. For this purpose, we need to write an Activity that launches an email client, using an implicit Intent with the right action and data. In this example, we are going to send an email from our app by using an Intent object that launches existing email clients.

Following section explains different parts of our Intent object required to send an email.

Intent Object - Action to send Email

You will use **ACTION_SEND** action to launch an email client installed on your Android device. Following is simple syntax to create an intent with ACTION_SEND action.

```
Intent emailIntent = new Intent(Intent.ACTION_SEND);
```

We use cookies to provide and improve our services. By using our site, you consent to our Cookies Policy.

Accept

[Learn more](#)

```
emailIntent.setData(Uri.parse("mailto:"));  
emailIntent.setType("text/plain");
```

Intent Object - Extra to send Email

Android has built-in support to add TO, SUBJECT, CC, TEXT etc. fields which can be attached to the intent before sending the intent to a target email client. You can use following extra fields in your email –

Sr.No.	Extra Data & Description
1	EXTRA_BCC A String[] holding e-mail addresses that should be blind carbon copied.
2	EXTRA_CC A String[] holding e-mail addresses that should be carbon copied.
3	EXTRA_EMAIL A String[] holding e-mail addresses that should be delivered to.
4	EXTRA_HTML_TEXT A constant String that is associated with the Intent, used with ACTION_SEND to supply an alternative to EXTRA_TEXT as HTML formatted text.
5	EXTRA_SUBJECT A constant string holding the desired subject line of a message.
6	EXTRA_TEXT A constant CharSequence that is associated with the Intent, used with ACTION_SEND to supply the literal data to be sent.
7	EXTRA_TITLE

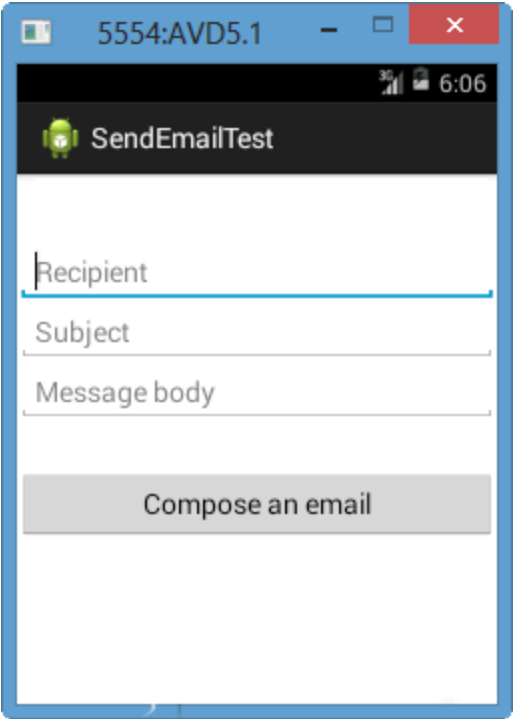
We use cookies to provide and improve our services. By using our site, you consent to our Cookies Policy.

[Accept](#)

[Learn more](#)

```
emailIntent.putExtra(Intent.EXTRA_EMAIL  , new String[]{"Recipient"});
emailIntent.putExtra(Intent.EXTRA_SUBJECT, "subject");
emailIntent.putExtra(Intent.EXTRA_TEXT   , "Message Body");
```

The out-put of above code is as below shown an image



EMAIL EXAMPLE

Example

Following example shows you in practical how to use Intent object to launch Email client to send an Email to the given recipients.

To Email experiment with this example, you will need actual Mobile device equipped with latest Android OS, otherwise you might get struggle with emulator which may not work properly. Second you will need to have an Email client like GMail(By default every android version having Gmail client App) or K9mail installed on your device.

Step	Description
1	You will use Android studio to create an Android application and name it as <i>Tutorialspoint</i> under a package <i>com.example.tutorialspoint</i> .

We use cookies to provide and improve our services. By using our site, you consent to our Cookies Policy.

Accept

Learn more

5	Modify <i>AndroidManifest.xml</i> as shown below
6	Run the application to launch Android emulator and verify the result of the changes done in the application.

Following is the content of the modified main activity file **src/com.example.Tutorialspoint/MainActivity.java**.

```
package com.example.tutorialspoint;

import android.net.Uri;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button startBtn = (Button) findViewById(R.id.sendEmail);
        startBtn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                sendEmail();
            }
        });
    }

    protected void sendEmail() {
        Log.i("Send email", "");
        String[] TO = {""};
        String[] CC = {""};
        Intent emailIntent = new Intent(Intent.ACTION_SEND);

        emailIntent.setData(Uri.parse("mailto:"));
        emailIntent.setType("text/plain");
        emailIntent.putExtra(Intent.EXTRA_EMAIL, TO);
        emailIntent.putExtra(Intent.EXTRA_CC, CC);
        emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Your subject");
        emailIntent.putExtra(Intent.EXTRA_TEXT, "Email message goes here");

        try {
```

We use cookies to provide and improve our services. By using our site, you consent to our Cookies Policy.

Accept

[Learn more](#)

Following will be the content of **res/layout/activity_main.xml** file –

Here abc indicates about tutorialspoint logo

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sending Mail Example"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:textSize="30dp" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tutorials point "
        android:textColor="#ff87ff09"
        android:textSize="30dp"
        android:layout_above="@+id/imageButton"
        android:layout_alignRight="@+id/imageButton"
        android:layout_alignEnd="@+id/imageButton" />

    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/imageButton"
        android:src="@drawable/abc"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

    <Button
        android:id="@+id/sendEmail"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/compose_email"/>

</LinearLayout>
```

Following will be the content of **res/values/strings.xml** to define two new constants –

```
<?xml version="1.0" encoding="utf-8"?>
```

We use cookies to provide and improve our services. By using our site, you consent to our Cookies Policy.

Accept

Learn more

```
package="com.example.Tutorialspoint" >


<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >

    <activity
        android:name="com.example.tutorialspoint.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 **tutorialspoint** 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 installer 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 following screen –

We use cookies to provide and improve our services. By using our site, you consent to our Cookies Policy.

Accept

[Learn more](#)

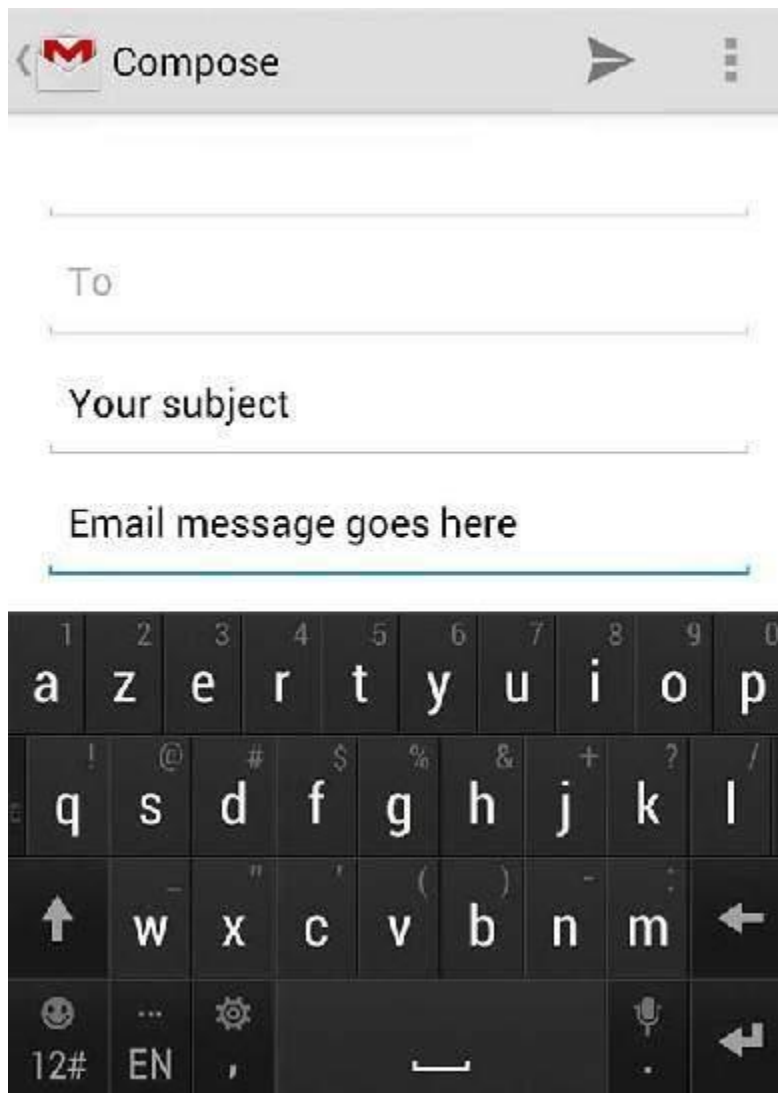


Now use **Compose Email** button to list down all the installed email clients. From the list, you can choose one of email clients to send your email. I'm going to use Gmail client to send my email which will have all the provided defaults fields available as shown below. Here **From:** will be default email ID you have registered for your Android device.

We use cookies to provide and improve our services. By using our site, you consent to our Cookies Policy.

Accept

Learn more



You can modify either of the given default fields and finally use send email button to send your email to the mentioned recipients.

[⬅ Previous Page](#)[Next Page ➡](#)

Advertisements

We use cookies to provide and improve our services. By using our site, you consent to our Cookies Policy.

Accept

[Learn more](#)



[Privacy Policy](#) [Cookies Policy](#) [Contact](#)

© Copyright 2019. All Rights Reserved.

We use cookies to provide and improve our services. By using our site, you consent to our Cookies Policy.

[Accept](#)

[Learn more](#)