



Advertisement

[Home](#) > [Code](#) > [Mobile Development](#) > [Android Development](#)

¿Qué Son Los Intents de Android?



Chinedu Izuchukwu

Aug 24, 2017 • 8 min read

Español ▾

Mobile Development

Android Development

() translation by (you can also [view the original English article](#))

Los Intents son un tema fundamental para desarrolladores Android. Es imposible construir aplicaciones Android sin entrar en contacto con Intents. En este tutorial, te enseñaré sobre Intents de la A a la Z.

¿Qué Son Intents?

En un partido de football, los miembros del equipo pasan la pelota alrededor del campo con la mira de enviarla a la meta de su oponente. La pelota es pasada del portero a sus defensas. Después, encuentra su camino a los medio campistas, y las cosas resultan según lo planeado, uno de los delanteros la



envía a la red del oponente. ¡Eso es asumiendo que el portero del otro equipo no pueda mantenerla lejos!

En Android, la habilidad de enviar mensajes es posible por el objeto `Intent`. Con la ayuda de los intents, los componentes de Android pueden solicitar funcionalidad de otros componentes Android. Cuando abres la aplicación Instagram en tu teléfono y la usas para tomar una foto, acabas de hacer uso de un intent. Los Intents también ayudan a comunicar entre partes de una app; el movimiento de una pantalla (actividad) a otra es posible mediante intents.

Míralo de esta forma: todos los componentes (aplicaciones y pantallas) del dispositivo Android están aisladas. La única manera de comunicarse entre ellas es a través de intents.

Comenzando Actividades Con Intents

Como mencioné anteriormente, puedes usar intents para comenzar diferentes componentes: actividades, servicios, y receptores de transmisión.

Para comenzar esta actividad, harás uso del método `startActivity(intent)`.

Aquí está un código que demuestra cómo comenzar otra actividad desde un intent.

```
1 | Intent numbersIntent = new Intent(MainActivity.this, NumbersActivity.class)
2 |
3 | startActivity(numbersIntent);
```

Primero, creamos un nuevo objeto `Intent` y lo pasamos a la clase `NumbersActivity`. Después comenzamos una nueva actividad usando ese intent.

Tipos de Intents



Android soporta dos tipos de intents: explícito e implícito. Cuando una aplicación define su componente meta en un intent, ese es un intent explícito. Cuando las aplicaciones no nombran un componente meta, ese es un intent implícito.

Ejemplo de Intent Explícito

El código de arriba es un ejemplo de un intent explícito. Échale un vistazo nuevamente.

```
1 | Intent numbersIntent = new Intent(MainActivity.this, NumbersActivity.class)
2 |
3 | startActivity(numbersIntent);
```

Aquí, `NumbersActivity` es el componente meta para nuestro `MainActivity`. Esto significa que `NumbersActivity` es el componente definido que será llamado por el sistema Android. Es importante notar (como en el ejemplo de arriba), que los intents explícitos son usados típicamente dentro de una aplicación, porque eso le da al desarrollador mayor control sobre cuál clase será lanzada.

Advertisement

Ejemplo de Intent Implícito

Aquí hay un intent implícito:

```
1 | Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("https://www.tu
```



```
2 | startActivity(intent);
```

Si tienes el código de arriba en tu base de código, tu aplicación puede comenzar un componente de navegador para una cierta URL vía un intent. ¿Pero cómo identifica el sistema Android los componentes que pueden reaccionar a un cierto intent?

Un componente puede ser registrado vía un *filtro intent* para una acción específica. Los filtros intent pueden ser registrados para componentes de manera estática en el **AndroidManifest.xml**. Aquí hay un ejemplo que registra un componente como un visor web:

```
1 | <activity android:name=".BrowserActivity"
2 |         android:label="@string/app_name">
3 |     <intent-filter>
4 |         <action android:name="android.intent.action.VIEW" />
5 |         <category android:name="android.intent.category.DEFAULT" />
6 |         <data android:scheme="http" />
7 |     </intent-filter>
8 | </activity>
```

Usando Intents en una App

Escribamos algo de código para ver cómo resulta. En esta sección, construirás una aplicación diminuta para probar ambos estilos de intent. La app tendrá un pequeño formulario para ingresar un primer nombre y apellido. Cuando se le da clic al botón **Enviar**, ambos valores ingresados serán pasados a otra actividad. Habrá también un botón para lanzar un navegador de tu elección. El navegador elegido abrirá <https://code.tutsplus.com>.

Abre Android Studio y genera tu `MainActivity`. Puedes establecer el nombre del paquete a **com.tutsplus.code.android.droidintent**.

Tu `MainActivity` comenzará con algunos imports y la declaración de clase:

```
1 | package com.tutsplus.code.android.droidintent;
2 |
3 | import android.content.Intent;
4 | import android.net.Uri;
5 | import android.support.v7.app.AppCompatActivity;
```



```
6 | import android.os.Bundle;
7 | import android.view.View;
8 | import android.widget.Button;
9 | import android.widget.EditText;
10 |
11 | public class MainActivity extends AppCompatActivity {
12 |
13 |
14 | }
```

Después anularás el método `onCreate()` para iniciar la actividad con cualquier estado guardado y el diseño de la actividad (crearemos este después).

```
1 | @Override
2 | protected void onCreate(Bundle savedInstanceState) {
3 |     super.onCreate(savedInstanceState);
4 |     setContentView(R.layout.activity_main);
5 |
6 |     // button handlers go here
7 |
8 | }
```

Después obtendrás referencias para cada uno de los botones definidos en el diseño y agregarás un escuchador de clic a cada uno de ellos.

```
1 | final Button submitButton = (Button) findViewById(R.id.submit_button);
2 |
3 | submitButton.setOnClickListener(new View.OnClickListener() {
4 |     @Override
5 |     public void onClick(View v) {
6 |         EditText firstName = (EditText) findViewById(R.id.first_name_text);
7 |         EditText lastName = (EditText) findViewById(R.id.last_name_text);
8 |
9 |         String firstNameString = firstName.getText().toString();
10 |        String lastNameString = lastName.getText().toString();
11 |
12 |        Intent submitIntent = new Intent(MainActivity.this, ShowActivity.cl
13 |        submitIntent.putExtra("firstNameString", firstNameString);
14 |        submitIntent.putExtra("lastNameString", lastNameString);
15 |        startActivity(submitIntent);
16 |    }
17 | });
18 |
```

Para el botón **Enviar**, estableces un `OnClickListener` para disparar una acción siempre que el botón es clicado. Cuando ocurre un clic, tomamos el nombre y apellido de la vista, y los enviamos a la siguiente actividad: `ShowActivity`. El componente meta es definido de manera explícita en el intent, haciendo esto un ejemplo de intent explícito.



```
1 final Button browserButton = (Button) findViewById(R.id.browser_button)
2
3 browserButton.setOnClickListener(new View.OnClickListener() {
4     @Override
5     public void onClick(View v) {
6
7         Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("https://c
8         startActivity(intent);
9     }
10 });
```

Para el navegador del botón, el `OnClickListener` creará un nuevo intent para lanzar cualquier aplicación que encaje con el filtro: un `ACTION_VIEW` que debería manejar una URL web. En otras palabras, lanza un navegador web. Si tienes más de una aplicación de navegador instalada en tu dispositivo, se te pedirá seleccionar uno para desempeñar la acción de abrir el sitio web. Este es un ejemplo de un intent explícito.

Diseño `MainActivity`

El diseño para `MainActivity` será muy simple para el propósito de este tutorial.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:orientation="vertical"
7     android:layout_margin="16dp"
8     android:layout_height="match_parent"
9     tools:context="com.tutsplus.code.android.droidintent.MainActivity">
10
11     <TextView
12         android:id="@+id/first_name_view"
13         android:layout_width="wrap_content"
14         android:layout_height="wrap_content"
15         android:text="Enter first name" />
16
17     <EditText
18         android:id="@+id/first_name_text"
19         android:padding="8dp"
20         android:layout_width="200dp"
21         android:layout_height="wrap_content" />
22
23     <TextView
24         android:id="@+id/last_name_view"
25         android:layout_width="wrap_content"
26         android:layout_height="wrap_content"
27         android:text="Enter last name" />
```



```
28
29     <EditText
30         android:id="@+id/last_name_text"
31         android:layout_height="wrap_content"
32         android:layout_width="200dp"
33         android:padding="8dp" />
34
35     <Button
36         android:id="@+id/submit_button"
37         android:layout_width="wrap_content"
38         android:layout_height="wrap_content"
39         android:text="Submit"/>
40
41     <Button
42         android:id="@+id/browser_button"
43         android:layout_width="wrap_content"
44         android:layout_height="wrap_content"
45         android:layout_marginTop="50dp"
46         android:layout_marginLeft="100dp"
47         android:text="Open Browser"/>
48
49 </LinearLayout>
```

Aquí tienes dos `TextView` y dos `EditText` indicando Primer Nombre y Apellido respectivamente. También hay un botón para enviar los nombres, y otro para lanzar tu navegador.

Advertisement

Crea el `ShowActivity`

Para completar nuestra app, necesitamos crear una actividad para manejar el intent explícito definido arriba. Crea una nueva actividad llamada `ShowActivity`. Esta es la actividad en donde el resultado de ingresar nombre y apellido será mostrado. Aquí está cómo debería lucir:



```

1 package com.tutsplus.code.android.droidintent;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.widget.TextView;
6
7 public class ShowActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_show);
13
14        Bundle extras = getIntent().getExtras();
15        String inputFirstName = extras.getString("firstNameString");
16        String inputLastName = extras.getString("lastNameString");
17
18        TextView showFirstName = (TextView) findViewById(R.id.show_first_name);
19        TextView showLastName = (TextView) findViewById(R.id.show_last_name);
20
21        showFirstName.setText(inputFirstName);
22        showLastName.setText(inputLastName);
23    }
24 }

```

En esta actividad, comenzarás obteniendo las cadenas pasadas desde el `MainActivity`. Puedes obtener una referencia al intent que disparó el lanzamiento de esta actividad con la función `getIntent()`. Después puedes acceder a las cadenas que fueron pasadas a este usando `getExtras().getString()`. Finalmente, después de obtener las instancias `TextView` del diseño, muestras los valores que obtuviste.

Diseño ShowActivity

El diseño para esta actividad es simple:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:layout_margin="16dp"
8     android:orientation="vertical"
9     tools:context="com.tutsplus.code.android.droidintent.ShowActivity">
10
11     <TextView
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="First Name:"/>
15

```




```
16     <TextView
17         android:id="@+id/show_first_name"
18         android:layout_width="wrap_content"
19         android:layout_height="wrap_content" />
20
21     <TextView
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"
24         android:text="Last Name:" />
25
26     <TextView
27         android:id="@+id/show_last_name"
28         android:layout_width="wrap_content"
29         android:layout_height="wrap_content" />
30
31 </LinearLayout>
```

Probando la App

¡Ahora puedes construir tu app y probarla en tu dispositivo Android!

Pasando Datos Usando Bundles

También puedes hacer uso de Bundles cuando pasas datos vía intents.

La clase `Bundle` te permite almacenar datos complejos y soporta tipos de datos tales como cadenas, caracteres, booleanos, enteros y así por el estilo. Aquí hay un ejemplo de cómo pararte de **MainActivity.java** luciría si usaras `Bundle`.

```
1 // Creating Bundle object
2 Bundle b = new Bundle();
3
4 // Storing data into bundle
5 b.putString("firstNameString", firstNameString);
6 b.putString("lastNameString", lastNameString);
7
8 // Creating Intent object
9 Intent submitIntent = new Intent(MainActivity.this, ShowActivity.class);
10
11 // Storing bundle object into intent
12 submitIntent.putExtra(b);
13 startActivity(submitIntent);
```

Conclusión



En este tutorial, tuvimos una breve introducción para usar intents para crear actividades en Android. Vimos la diferencia entre intents explícitos e implícitos, y codificamos un ejemplo simple que usó cada tipo.

Puedes leer más sobre intents en la [documentación Android](#). Entender cómo funcionan es muy importante. Mientras construyes más apps, encontrarás muchos tipos diferentes de Intents y maneras de usarlos.

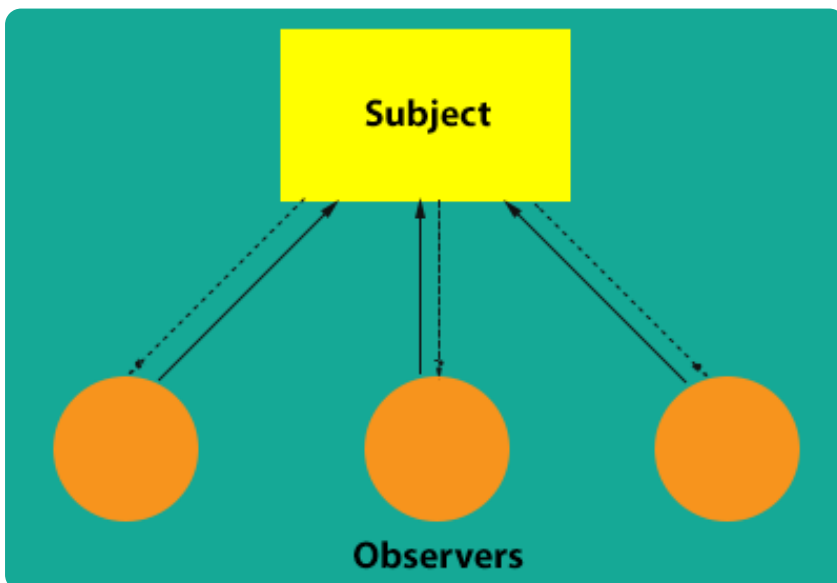
Y mientras tanto, ¡revisa algunos de nuestras otras publicaciones sobre desarrollo de apps Android!



[SDK Android](#)

[Introducción a Componentes de Arquitectura Android](#)

[Tin Megali](#)



[Java](#)

[Patrones de Diseño Android: El Patrón Observador](#)



Chike Mgbemena



Android

Android Desde Cero: Construyendo Tu Primera Aplicación Android

Gianluca Segato

Did you find this post useful?



Yes



No

Want a weekly email summary?

Subscribe below and we'll send you a weekly email summary of all new Code tutorials. Never miss out on learning about the next big thing.

Sign up



Chinedu Izuchukwu



Lagos, Nigeria

I enjoy writing code and passing knowledge. When not doing either of them, I watch movies.

[View on GitHub](#)

Advertisement

LOOKING FOR SOMETHING TO HELP KICK START YOUR NEXT PROJECT?

Envato Market has a range of items for sale

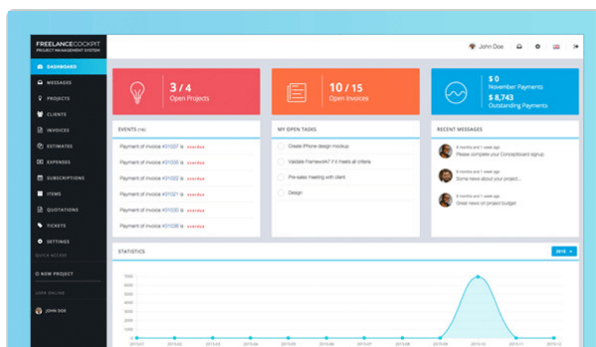


to help get you started.



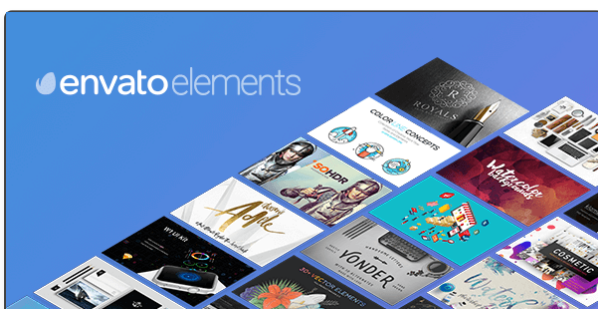
WordPress Plugins

From \$5



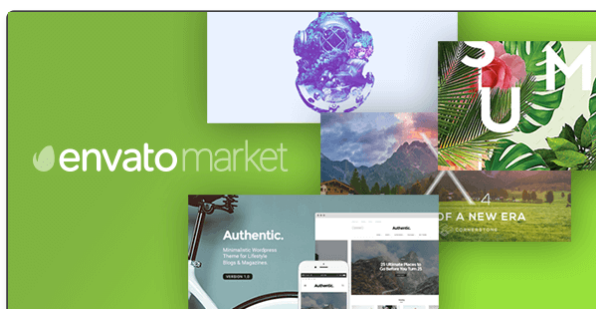
PHP Scripts

From \$5



Unlimited Downloads From \$16.50/month

Get access to over one million creative assets on Envato Elements.



Over 9 Million Digital Assets

Everything you need for your next creative project.

QUICK LINKS - Explore popular categories

ENVATO TUTS+

About Envato Tuts+
Terms of Use
Advertise

HELP

FAQ
Help Center





tuts+

30,338
Tutorials

553
Courses

42,531
Translations



[Envato](#) [Envato Elements](#) [Envato Market](#) [Placeit by Envato](#) [All products](#) [Careers](#) [Sitemap](#)

© 2023 Envato Pty Ltd. Trademarks and brands are the property of their respective owners.

