

Android Prebe

Clase 2: Activity e Intents

Activity en Android:

Activity



Una Activity es un componente de la aplicación que contiene una pantalla con la que los usuarios pueden interactuar para realizar una acción, como marcar un número telefónico, tomar una foto, enviar un correo electrónico o ver un mapa. A cada actividad se le asigna una ventana en la que se puede dibujar su interfaz de usuario. La ventana generalmente abarca toda la pantalla, pero en ocasiones puede ser más pequeña que esta y quedar "flotando" encima de otras ventanas.

Cada vez que se inicia una actividad nueva, se detiene la actividad anterior, pero el sistema conserva la actividad en una pila (la "pila de actividades"). Cuando se inicia una actividad nueva, se la incluye en la pila de actividades y capta el foco del usuario. La pila de actividades cumple con el mecanismo de pila "el último en entrar es el primero en salir", por lo que, cuando el usuario termina de interactuar con la actividad actual y presiona el botón *Atrás*, se quita de la pila (y se destruye) y se reanuda la actividad anterior.

Cuando se detiene una actividad porque se inicia otra, se notifica el cambio de estado a través de los métodos callback del ciclo de vida de la actividad. Existen diferentes métodos callback que podría recibir una actividad como consecuencia de un cambio de estado (ya sea que el sistema la esté creando, deteniendo, reanudando o destruyendo) y cada callback te da la oportunidad de realizar una tarea específica que resulta adecuada para ese cambio de estado.

Crear una actividad

Para crear una actividad, debes crear una subclase de **Activity** (o una subclase existente de ella). En tu subclase debes implementar métodos callback a los que el sistema invoca cuando la actividad alterna entre diferentes estados de su ciclo de vida, como cuando se crea, se detiene, se reanuda o se destruye. Los dos métodos callback más importantes son los siguientes: **onCreate()** y **onPause()**

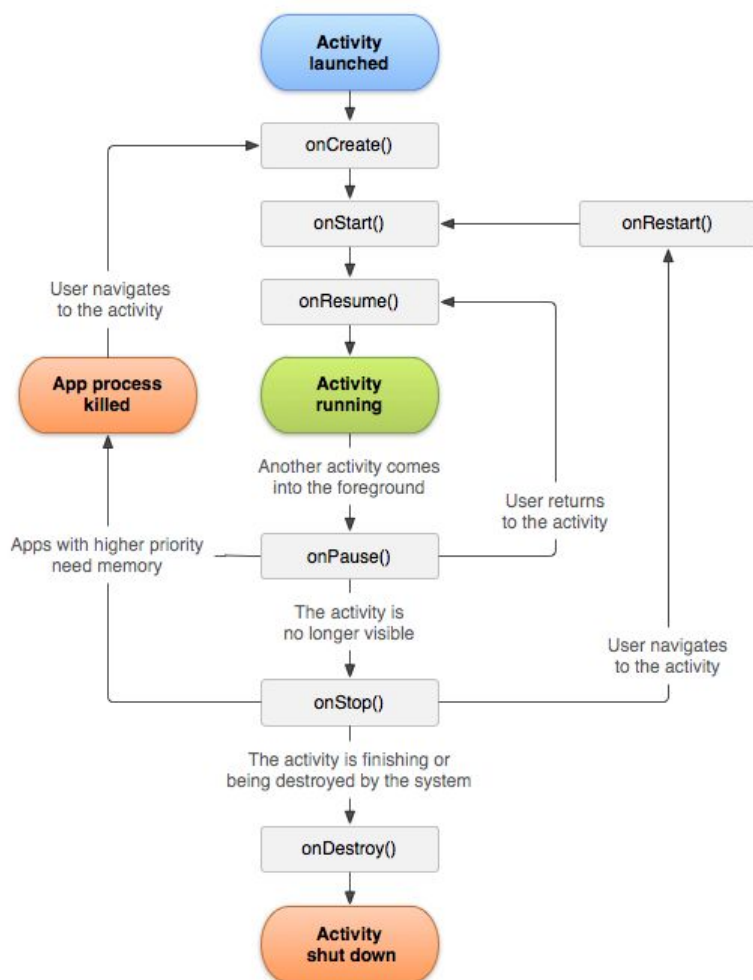
Declarar la actividad en el manifiesto.

Debes declarar tu actividad en el archivo de manifiesto para que pueda estar accesible para el sistema. Para declarar tu actividad, abre el archivo de manifiesto y agrega un elemento **<activity>** como campo secundario del elemento **<application>**. Por ejemplo:

```
<manifest ... >
  <application ... >
    <activity android:name=".ExampleActivity" />
  </application ... >
</manifest >
```

Ciclo de vida de una Activity

El ciclo de vida del Activity en Android



onCreate

Este método se llama al crear una Actividad. Una Actividad debe emplear este método para configurar su pantalla principal y realizar cualquier otra configuración inicial, como la creación de vistas, el enlace de datos a las listas, etc. Este método proporciona un parámetro Bundle, que es un diccionario para almacenar y transmitir información de estado y de los objetos entre las actividades.

onStart

Este método se llama cuando la Actividad está a punto de ser visible para el usuario. Las Actividades deben emplear este método si necesitan realizar cualquier tarea específica justo antes de que se hagan visibles, como por ejemplo refrescar valores actuales de los valores de las vistas dentro de la Actividad, o de puente para frame rates (una tarea común en la construcción de juegos).

onPause

Este método se llama cuando el sistema está a punto de poner la Actividad en background. Las Actividades deben emplear este método si necesitan confirmar los cambios no guardados a los datos persistentes, destruir o limpiar objetos para liberar los recursos que consumen, etc. Las implementaciones de este método deben finalizar lo antes posible, ya que ninguna Actividad sucesiva se reanudará hasta que este método devuelve un valor.

onResume

Este método se llama cuando la Actividad va a empezar a interactuar con el usuario después de estar en un estado de pausa. Cuando este método se llama, la Actividad se está moviendo a la parte superior de la pila y está recibiendo actividad del usuario. Las Actividades pueden emplear este método si lo necesitan para realizar cualquier tarea después de que la Activity haya aceptado la interacción del usuario.

onStop

Este método se llama cuando la Actividad ya no es visible para el usuario, ya que otra Actividad se ha reanudado o iniciado y está eclipsando a esta. Esto puede suceder porque la Actividad ha finalizado (el método de finalización fue llamado), porque el sistema está destruyendo a esta instancia de Actividad para ahorrar recursos, o porque un cambio de orientación se ha producido en el dispositivo.

onRestart

Este método se llama después de que una Actividad se haya detenido, antes de que se inicie de nuevo. Este método siempre es seguido por onStart. Una Activity debe pasar por onRestart si es necesario para realizar las tareas inmediatamente antes de llamar a onStart. Por ejemplo, si la actividad previamente ha sido enviada al background y onStop ha sido llamada, pero el proceso de la Actividad aún no ha sido destruido por el sistema operativo, entonces el método onRestart debe ser anulado.

Un buen ejemplo de esto sería cuando el usuario presiona el botón de inicio cuando aún había una Activity en la aplicación. El onPause, y a continuación el onStop, son llamados, pero la actividad no se destruye. Si el usuario a continuación restaurara la aplicación mediante el administrador de tareas o una aplicación similar, el método onRestart de la actividad sería llamado por el sistema operativo, durante la reactivación de actividades.

onDestroy

Este es el último método que se llama en una Actividad antes de que sea destruido. Después de llamar a este método, su Actividad se matará y será purgado de los grupos de recursos del dispositivo. El sistema operativo se destruirán permanentemente los datos de estado de una Actividad después de que este método se haya ejecutado, por lo que una Actividad debe reemplazar este método como medio final para salvar los datos del estado.

onSaveInstanceState

Este método es proporcionado por el ciclo de vida de Android para dar a una Actividad la oportunidad de guardar los datos cuando se produce un cambio, por ejemplo un cambio de orientación de pantalla.

Implementación de callbacks del ciclo de vida

Cuando una actividad entra y sale de los diferentes estados que se describieron más arriba, esto se notifica a través de diferentes métodos callback. Todos los métodos callback son enlaces que puedes invalidar para realizar las tareas correspondientes cuando cambia el estado de la actividad. La siguiente actividad estructural incluye todos los métodos fundamentales del ciclo de vida:

```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart(); // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume(); // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause(); // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop(); // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy(); // The activity is about to be destroyed.
    }
}
```

Intents y filtros de intents

Una **Intent** es un objeto de acción que puedes usar para solicitar una acción de otro **componente de la aplicación**. Aunque las intents facilitan la comunicación entre los componentes de muchas maneras, existen tres casos de uso fundamentales:

Para comenzar una actividad:

Una Activity representa una única pantalla en una aplicación. Puedes iniciar una nueva instancia de una Activity pasando una Intent a `startActivity()`. La Intent describe la actividad que se debe iniciar y contiene los datos necesarios para ello.

Si deseas recibir un resultado de la actividad cuando finalice, llama a `startActivityForResult()`. La actividad recibe el resultado como un objeto Intent separado en el callback de `onActivityResult()` de la actividad.

Para iniciar un servicio:

Un Service es un componente que realiza operaciones en segundo plano sin una interfaz de usuario. Puede iniciar un servicio para realizar una operación única pasando una Intent a `startService()`. La Intent describe el servicio que se debe iniciar y contiene los datos necesarios para ello.

Si el servicio está diseñado con una interfaz cliente-servidor, puedes establecer un enlace con el servicio de otro componente pasando una Intent a `bindService()`.

Para entregar un mensaje:

Un mensaje es un aviso que cualquier aplicación puede recibir. El sistema entrega varios mensajes de eventos del sistema, como cuando el sistema arranca o el dispositivo comienza a cargarse. Puedes enviar un mensaje a otras apps pasando una Intent a `sendBroadcast()`, `sendOrderedBroadcast()` o `sendStickyBroadcast()`.

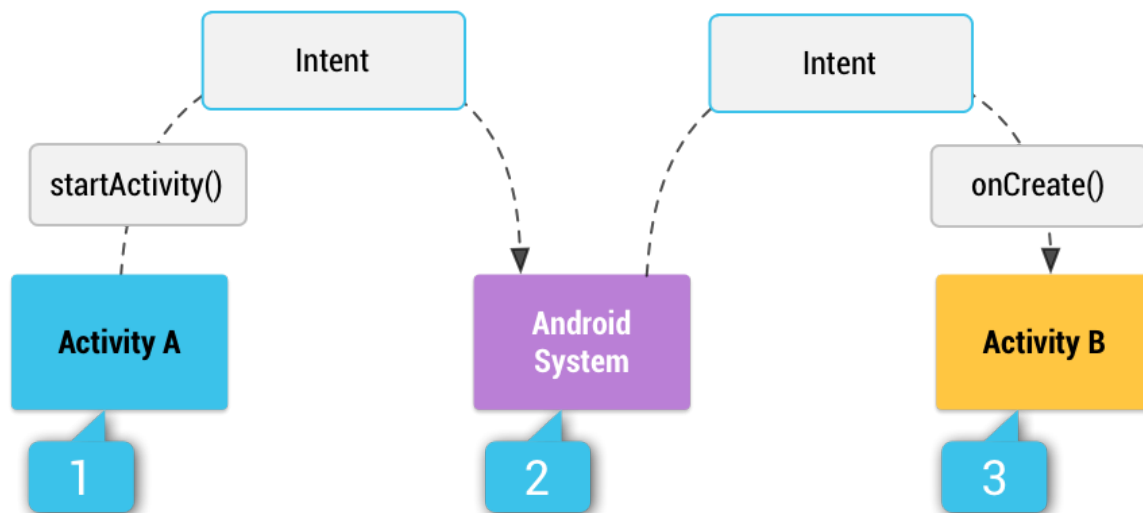
Tipos de intents

Existen dos tipos de intents:

Intents explícitas: especifican qué componente se debe iniciar mediante su nombre (el nombre de clase completamente calificado). Usualmente, el usuario usa una intent explícita para iniciar un componente en su propia aplicación porque conoce el nombre de clase de la actividad o el servicio que desea iniciar. Por ejemplo, puede utilizarla para iniciar una actividad nueva en respuesta a una acción del usuario o iniciar un servicio para descargar un archivo en segundo plano.

Intents implícitas: no se nombra el componente específicamente; pero, en cambio, se declara una acción general para realizar, lo que permite que un componente de otra aplicación la maneje. Por ejemplo, si desea mostrar al usuario una ubicación en un mapa, puede usar una intent implícita para solicitar que otra aplicación capaz muestre una ubicación específica en un mapa.

Cuando crea una **intent explícita** para iniciar una actividad o un servicio, la aplicación inicia inmediatamente el componente de la aplicación especificado en el objeto Intent.



[1] La *actividad A* crea una **Intent** con una descripción de acción y la pasa a **startActivity()**.

[2] El sistema Android busca en todas las apps un filtro de intents que coincida con la intent. Cuando se encuentra una coincidencia,

[3] El sistema inicia la actividad coincidente (*actividad B*) invocando su método onCreate() y pasándolo a la Intent.

Creación de una intent

Un objeto Intent tiene información que el sistema Android usa para determinar qué componente debe iniciar (como el nombre exacto del componente o la categoría del componente que debe recibir la intent), además de información que el componente receptor usa para realizar la acción correctamente (por ejemplo, la acción que debe efectuar y los datos en los que debe actuar).

La información principal que contiene una **Intent** es la siguiente:

Nombre del componente

El nombre del componente que se debe iniciar.

Esto es opcional, pero es la información clave que hace que una intent sea **explícita**, sin un nombre de componente, la intent es **implícita** y el sistema decide qué componente debe recibir la intent conforme la información restante que esta contiene (como la acción, los datos y la categoría, que se describen a continuación).

Acción

Una string que especifica la acción genérica que se debe realizar (como *ver* o *elegir*).

ACTION_VIEW

Usa esta acción en una intent con startActivity() cuando tengas información que la actividad puede mostrar al usuario, como una foto para ver en una app de galería o una dirección para ver en una app de mapa.

ACTION_SEND

También se conoce como la intent de “compartir” y debes usarla en una intent con startActivity() cuando tengas información que el usuario puede compartir mediante otra app, como una app de correo electrónico o intercambio social. Puedes especificar la acción de una intent con setAction() o un constructor Intent.

`static final String ACTION_TIMETRAVEL = "com.example.action.TIMETRAVEL";`

Datos

El URI (un objeto Uri) que hace referencia a los datos en los que se debe realizar la acción o el tipo de MIME de esos datos. El tipo de datos suministrado está generalmente determinado por la acción de la intent. Por ejemplo, si la acción es ACTION_EDIT, los datos deben contener el URI del documento que se debe editar.

Cuando se crea una intent, suele ser importante especificar el tipo de datos (su tipo de MIME), además del URI. Por ejemplo, una actividad que puede mostrar imágenes, probablemente no podrá reproducir un archivo de video, aunque los formatos del URI podrían ser similares. Por lo tanto, especificar el tipo de MIME de tus datos ayuda al sistema Android a encontrar el mejor componente para recibir la intent.

Categoría

Una string que contiene información adicional sobre el tipo de componente que la intent debe manejar. En una intent, se puede incluir la cantidad deseada de descripciones de categorías, pero la mayoría de las intents no requieren una categoría. Estas son algunas categorías comunes:

CATEGORY_BROWSABLE

La actividad de destino permite que la inicie un navegador web para mostrar datos a los que hace referencia un vínculo, como una imagen o un mensaje de correo electrónico.

CATEGORY_LAUNCHER

La actividad es la actividad inicial de una tarea y está enumerada en el lanzador de la aplicación del sistema. Para conocer la lista completa de categorías, consulte la descripción de la clase Intent. También puedes especificar una categoría con addCategory().

Extras

Pares de valores clave que tienen información adicional necesaria para lograr la acción solicitada.

Puedes agregar datos de campos extra con varios métodos putExtra(), cada uno de los cuales acepta dos parámetros: el nombre clave y el valor. También puedes crear un objeto Bundle con todos los datos de campos extra y luego insertar el Bundle en el Intent con putExtras().

Por ejemplo, al crear una intent para enviar un correo electrónico con ACTION_SEND, puedes especificar el destinatario "para" con la clave EXTRA_EMAIL y especificar el "asunto" con la clave EXTRA_SUBJECT.

Indicadores

Indicadores definidos en la clase Intent que funcionan como metadatos para la intent. Los indicadores pueden indicar al sistema Android cómo iniciar una actividad (por ejemplo, a qué tarea debe pertenecer la actividad) y cómo tratarla después del inicio

Ejemplo de una intent explícita

Una intent explícita es una intent que se usa para iniciar un componente específico de la aplicación, como una actividad o un servicio particular en la aplicación. Para crear una intent explícita, define el nombre de componente para el objeto Intent, todas las otras propiedades de la intent son opcionales.

Por ejemplo, si creaste un servicio en tu app denominado Download Service, diseñado para descargar un archivo de la Web, puedes iniciarlo con el siguiente código:

```
// The fileUrl is a string URL, such as "http://www.example.com/image.png"
Intent downloadIntent = new Intent(this, DownloadService.class);
downloadIntent.setData(Uri.parse(fileUrl));
startService(downloadIntent);
```

Ejemplo de una intent implícita

Una intent implícita especifica una acción que puede invocar cualquier aplicación en el dispositivo que pueda realizar la acción. El uso de una intent implícita es útil cuando la aplicación no puede realizar la acción pero otras aplicaciones probablemente sí, y tu desearas que el usuario eligiera qué aplicación usar.

Por ejemplo, si tiene contenido que deseas que el usuario comparta con otras personas, crea una intent con la acción `ACTION_SEND` y agrega extras que especifiquen el contenido para compartir. Cuando llama a `startActivity()` con esa intent, el usuario puede elegir una aplicación mediante la cual compartir el contenido.

```
// Create the text message with a string
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, textMessage);
sendIntent.setType("text/plain");
// Verify that the intent will resolve to an activity
if (sendIntent.resolveActivity(getPackageManager()) != null) {
    startActivity(sendIntent);
}
```

Referencias:

<https://developer.android.com/guide/components/activities.html?hl=es-419>

<http://www.androidcurso.com/index.php/tutoriales-android/37-unidad-6-multimedia-y-ciclo-de-vida/158-ciclo-de-vida-de-una-actividad>

<https://stories.devacademy.la/android-ciclo-de-vida-de-un-activity-b2a68c506961>

<https://developer.android.com/guide/components/intents-filters.html?hl=es-419>