DESARROLLO DE APLICACIONES EN ANDROID

SAMSUNG

Abraham Gutiérrez Rodríguez

Samsung TECH INSTITUTE

Android: Conceptos Básicos
Componentes básicos

Samsung TECH INSTITUTE

Android: Conceptos Básicos
La clase Intent

- Un Intent es un mensaje que podemos utilizar para solicitar una acción determinada a otra componente de una aplicación. Se trata de una clase que nos facilita la comunicación entre componentes, principalmente en uno de estos casos básicos (mecanismos diferentes según el tipo de componente que queramos lanzar):
  - Empezar una **actividad**: Con un Intent podemos empezar una nueva instancia de una Actividad, gracias al método startActivity. En el Intent introduciremos la información de la actividad a empezar, así como cualquier otro dato que la actividad necesite.
  - Empezar un **servicio**: Totalmente similar al anterior, pero utilizando el método startService. Pero también hay que destacar, que si el servicio está diseñado con una interfaz cliente-servidor, podremos utilizar el método bindService.
  - Enviar un **broadcast**: Con un Intent y uno de los métodos sendBroadcast, sendOrderedBroadcast o sendStickyBroadcast, podremos enviar mensajes broadcast al sistema operativo de nuestro dispositivo.

Samsung TECH INSTITUTE

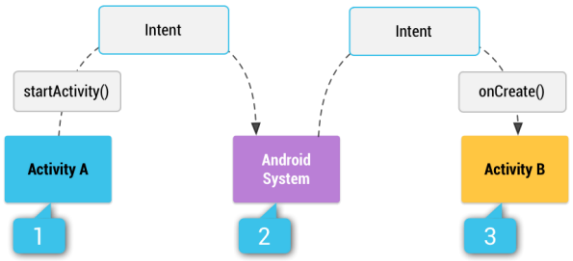
Android: Conceptos Básicos
La clase Intent

- Tipos de Intents: hay dos tipos de intenciones
  - Explícitos: Este tipo de intención es aquella en el que conocemos exactamente qué componente lanzar, el cual especificaremos a través del nombre de componente (el nombre de la clase completamente cualificado). Lo utilizaremos cuando queramos lanzar un servicio o actividad, típicamente de nuestra propia aplicación.
  - Implícitos: En este caso, no conocemos el nombre de componente (estará vacío), pero sabemos la acción que queremos ejecutar. De esta forma, le pediremos al sistema operativo que busque por nosotros qué aplicación podría realizar dicha acción y, en el caso de que haya varias, nos dará a elegir cuál utilizar.

Samsung TECH INSTITUTE

Android: Conceptos Básicos
La clase Intent

- Cuando se crea una intención explícita para arrancar un servicio o una actividad, el sistema inicia automáticamente.
- Cuando se crea una intención implícita el sistema Android busca el componente apropiado en el sistema para dar servicio a la petición.



Samsung TECH INSTITUTE

Android: Conceptos Básicos
La clase Intent

Samsung TECH INSTITUTE

Android: Conceptos Básicos
La clase Intent

¿Qué información contiene un Intent?

- Básicamente, portará información que el sistema operativo utilizará para determinar qué componente lanzar, así como cualquier información adicional que la componente pueda necesitar para poder llevar a cabo su acción.
- La información principal que podemos encontrar en un Intent es la siguiente:
  - Component Name (Nombre de componente): En este campo podremos indicar el nombre del componente que queremos lanzar. Es opcional, pues no siempre conoceremos el componente a lanzar (sin nombre la intención es implícita). Sólo será obligatorio en el caso de un servicio.
  - Action (Acción): Se trata de una cadena de texto que nos permite especificar una acción a ejecutar (ya sea una disponible en Android o la nuestra propia). En el caso de los mensajes broadcast, este campo será esencial, pues será en base al que lanzaremos los diferentes mensajes.

Samsung TECH INSTITUTE

Android: Conceptos Básicos
La clase Intent

Ejemplos de acciones estándar

- ACTION_MAIN
- ACTION_VIEW
- ACTION_ATTACH_DATA
- ACTION_EDIT
- ACTION_PICK
- ACTION_CHOOSER
- ACTION_GET_CONTENT
- ACTION_DIAL
- ACTION_CALL
- ACTION_SEND
- ACTION_SENDTO
- ACTION_ANSWER
- ACTION_INSERT
- ACTION_DELETE
- ACTION_RUN
- ACTION_SYNC
- ACTION_PICK_ACTIVITY
- ACTION_SEARCH
- ACTION_WEB_SEARCH
- ACTION_FACTORY_TEST

Samsung TECH INSTITUTE

Android: Conceptos Básicos
La clase Intent

Ejemplos de intenciones implícitas

- `ACTION_VIEW content://contacts/people/1`
- `ACTION_DIAL content://contacts/people/1`
- `ACTION_VIEW tel:123`
- `ACTION_DIAL tel:123`
- `ACTION_EDIT content://contacts/people/1`
- `ACTION_VIEW content://contacts/people/`

Samsung TECH INSTITUTE

Android: Conceptos Básicos
La clase Intent

Ejemplos de intenciones implícitas

```
Uri number = Uri.parse("tel:5551234");
Intent callIntent = new Intent(Intent.ACTION_DIAL, number);
```

```
// Map point based on address
Uri location = Uri.parse("geo:0,0?q=1600+Amphitheatre+Parkway,+Mountain+View,+California");
// Or map point based on latitude/longitude
// Uri location = Uri.parse("geo:37.422219,-122.08364?z=14"); // z param is zoom level
Intent mapIntent = new Intent(Intent.ACTION_VIEW, location);
```

```
Uri webpage = Uri.parse("http://www.android.com");
Intent webIntent = new Intent(Intent.ACTION_VIEW, webpage);
```

Android: Conceptos Básicos
La clase Intent

¿Qué información contiene un Intent?

- A los atributos principales se les añaden unos atributos secundarios:
  - Data (Datos): Será una URI que haga referencia a los datos que queremos utilizar, así como el tipo de datos (tipo MIME).
  - Category (Categoría): Es una cadena de texto que ofrece información adicional sobre el tipo de componente que debería ser lanzado.
  - Extras (Extras): Aquí podremos meter tantos pares clave-valor como queramos como información adicional que necesitemos enviar al componente a lanzar.
  - Flags (Banderas): Son metadatos que ayudan a indicar al sistema cómo lanzar, por ejemplo, nuestra actividad.

Android: Conceptos Básicos
La clase Intent

Ejemplos de intenciones implícitas

```
Intent emailIntent = new Intent(Intent.ACTION_SEND);
// The intent does not have a URI, so declare the "text/plain" MIME type
emailIntent.setType(HTTP.PLAIN_TEXT_TYPE);
emailIntent.putExtra(Intent.EXTRA_EMAIL, new String[] {"jon@example.com"}); // recipients
emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Email subject");
emailIntent.putExtra(Intent.EXTRA_TEXT, "Email message text");
emailIntent.putExtra(Intent.EXTRA_STREAM, Uri.parse("content://path/to/email/attachment"));
// You can also attach multiple items by passing an ArrayList of Uris
```

```
Intent calendarIntent = new Intent(Intent.ACTION_INSERT, Events.CONTENT_URI);
Calendar beginTime = Calendar.getInstance().set(2012, 0, 19, 7, 30);
Calendar endTime = Calendar.getInstance().set(2012, 0, 19, 10, 30);
calendarIntent.putExtra(CalendarContract.EXTRA_EVENT_BEGIN_TIME, beginTime.getTimeInMillis());
calendarIntent.putExtra(CalendarContract.EXTRA_EVENT_END_TIME, endTime.getTimeInMillis());
calendarIntent.putExtra(Events.TITLE, "Ninja class");
calendarIntent.putExtra(Events.EVENT_LOCATION, "Secret dojo");
```

Android: Conceptos Básicos
La clase Intent

- Resolución de intenciones
  - Para encontrar el componente adecuado a una intención implícita, el sistema compara el contenido de la intención con los filtros de intención (intent filters) declarados en el archivo de manifiesto de todas las aplicaciones del dispositivo.
    - El mecanismo de resolución de intención básicamente gira en torno al emparejamiento de la intención contra todas las descripciones <intent-filter> de todos los paquetes de aplicaciones instalados.
  - Si la intención coincide con un filtro de intención, el sistema inicia ese componente y le entrega el objeto Intent. Si varios filtros de intención son compatibles, el sistema muestra un cuadro de diálogo para que el usuario puede elegir qué aplicación usar.
  - Si no se encuentra un componente adecuado el sistema genera un error.

Android: Conceptos Básicos
La clase IntentFilter

- Resolución de intenciones
  - Hay tres piezas de información en la intención que se utilizan para la resolución: la acción, tipo y categoría. Usando esta información, se realiza una consulta en el PackageManager para encontrar un componente que puede manejar la intención.
  - El componente apropiado se determina basándose en la información de intención suministrada en el archivo AndroidManifest.xml y debe cumplir todos los requisitos de acción, categoría y datos.
    - El tipo, si no se ha suministrado, se recupera a partir de datos de la intención. Para los datos que no son contenidos (URI) y no se ha explicitado su tipo se tienen en cuenta los esquemas de los esquemas (tales como *http:* o *mailto:*).
    - La categoría DEFAULT permite iniciar la actividad por una intención implícita que no hay definido categoría.

Android: Conceptos Básicos
La clase IntentFilter

- Intents Filter (filtro de intenciones)
  - Un filtro de intención es una expresión en el archivo de manifiesto de una aplicación que especifica el tipo de intenciones que el componente le gustaría recibir.
  - Por ejemplo, al declarar un filtro de intención para una actividad, se hace posible que otras aplicaciones puedan iniciar directamente su actividad con un cierto tipo de intención. Del mismo modo, si no se declara ningún filtro de intención para una actividad, entonces dicha actividad sólo se puede iniciar sólo con la intención explícita.
  - Atención: Para asegurar que una aplicación es segura, utilizar siempre una intención explícita al iniciar un servicio y no declarar filtros intención para los servicios.
    - Usar una intención implícita para iniciar un servicio es un riesgo de seguridad porque no se puede estar seguro de lo que el servicio responde a la intención, y el usuario no puede saber que servicio se inicia.

Android: Conceptos Básicos
La clase IntentFilter

- Intents Filter (filtro de intenciones)

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

  - Este filtro proporciona una entrada de nivel superior en la aplicación: La acción estándar MAIN es el punto de entrada principal a la aplicación (no requiere ninguna otra información de la intención), y la categoría LAUNCHER indica que ese punto de entrada debe aparecer en el lanzador de aplicaciones.

Android: Conceptos Básicos
La clase IntentFilter

- Intents Filter (filtro de intenciones)

```
<intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <action android:name="android.intent.action.EDIT" />
    <action android:name="android.intent.action.PICK" />
    <category android:name="android.intent.category.DEFAULT" />
    <data mimeType:name="vnd.android.cursor.dir/vnd.google.note" />
</intent-filter>
```

  - La actividad permite al usuario ver o editar el directorio de datos (a través de las acciones VIEW y EDIT), o recoger una nota en particular y devolverlo al llamador (a través de la acción de PICK).
    - Se incluye la categoría DEFAULT para resolver la actividad cuando no se especifica explícitamente el nombre del componente.

Android: Conceptos Básicos
La clase IntentFilter

- PendingIntents (intenciones pendientes)
  - Un objeto PendingIntent es una envoltura alrededor de un objeto Intención. El propósito principal de una intención pendiente es conceder permiso a una aplicación externa para utilizar la intención contenida como si se hubiera ejecutado desde el propio proceso de la aplicación. Sus principales usos incluyen:
    - Declarar la intención para ser ejecutada cuando el usuario realiza una acción asociada a una notificación (el gestor NotificationManager del sistema Android ejecuta la Intención).
    - Declarar la intención de ser ejecutada cuando el usuario realiza una acción asociada con una App Widget (la pantalla de inicio de aplicaciones ejecuta la intención).
    - Declarar la intención de ser ejecutada en un momento determinado en el futuro (el gestor AlarmManager del sistema Android ejecuta la Intención).

Android: Conceptos Básicos
La clase IntentFilter

- PendingIntents (intenciones pendientes)
  - Debido a que cada objeto intención está diseñado para ser manejado por un tipo específico de componente de aplicación (ya sea una actividad, un servicio, o un receptor de difusión), una intención pendiente deber ser creada con la misma consideración.
  - Cuando se utiliza una intención pendiente, la aplicación no la ejecutará con una llamada del tipo startActivity(). Se debe declarar el tipo de componente deseado al crear el PendingIntent llamando al método creador respectivo:
    - PendingIntent.getActivity() para una intención que inicia una actividad.
    - PendingIntent.getService() para una intención que inicia un servicio.
    - PendingIntent.getBroadcast() para una intención que inicia un receptor de difusión.
  - A menos que la aplicación esté recibiendo intenciones pendientes de otras aplicaciones, los métodos anteriores para crear un PendingIntent son los únicos métodos que probablemente necesitará.