



DESARROLLO DE APLICACIONES V

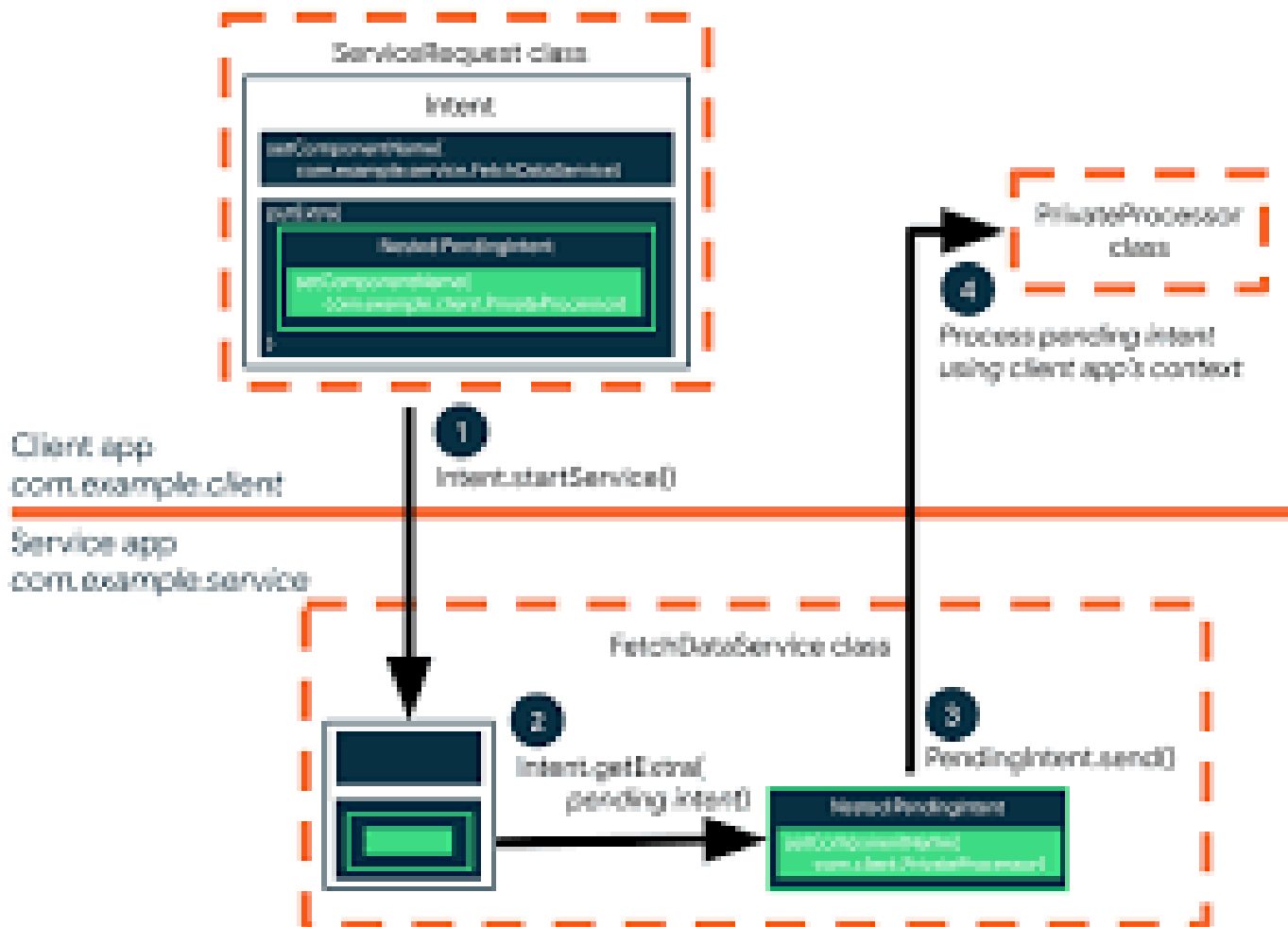
23/09/2024

Cómo enviar al usuario a otra app

Guía Completa sobre Intents en Android

Los **Intents** son una característica fundamental de la plataforma Android que permite la comunicación entre diferentes componentes de una aplicación. Mediante el uso de **Intents**, los desarrolladores pueden iniciar actividades, servicios y enviar broadcasts. Comprender cómo funcionan los **Intents** es esencial para crear aplicaciones interactivas y funcionales. Esta guía abarca los tipos de **Intents**, el manejo de resultados, los filtros de **Intents** y las limitaciones y consideraciones de seguridad.

Los **Intents** en Android permiten la comunicación entre componentes de una app o entre diferentes aplicaciones. Se dividen en **Intents explícitos**, que especifican directamente la actividad o servicio, e **Intents implícitos**, que permiten que otras apps manejen la acción. Los Intents también facilitan el envío de datos entre actividades con `putExtra()`. Para recibir resultados entre actividades, se recomienda usar `ActivityResultLauncher` en lugar del método antiguo `startActivityForResult()`.



Tipos de Intents

Intents Explícitos:

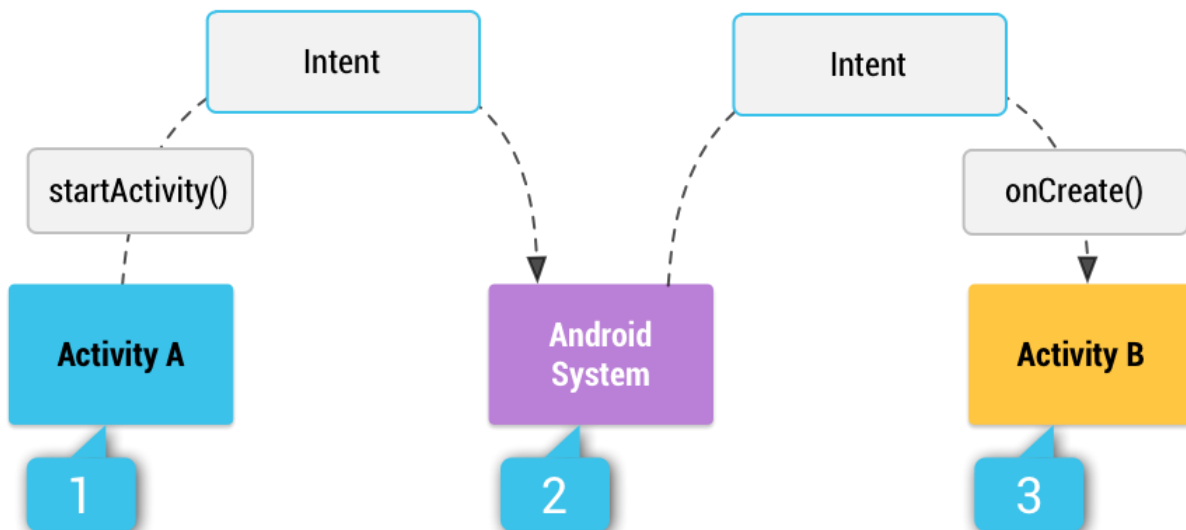
- Se utilizan para iniciar una actividad o servicio específico dentro de la misma aplicación.
- Especifican qué componente de qué aplicación satisfará el intent, ya que especifican un `ComponentName` completo. Deberás normalmente usar un intent explícito para iniciar un componente en tu propia app, porque conoces el nombre de clase de la actividad o el servicio que quieres iniciar. Para ejemplo, puedes iniciar una nueva actividad dentro de la aplicación en respuesta a una acción del usuario, o iniciar un servicio que descargue un archivo en segundo plano.

```
// Executed in an Activity, so 'this' is the Context
```

Intents Implícitos:

- Se utilizan para solicitar una acción que puede ser manejada por cualquier aplicación disponible en el dispositivo.
- Se definen mediante acciones estándar como `ACTION_VIEW`, `ACTION_SEND`, etc
- No nombran un componente específico, sino que declaran una acción general. lo que permite que un componente de otra app lo controle. Por ejemplo, si quieres mostrar al usuario una ubicación en un mapa, puedes usar un intent implícito para solicitar que se cree una app muestra una ubicación específica en un mapa.

```
val intent = Intent(Intent.ACTION_VIEW)  
  
intent.data = Uri.parse("http://www.ejemplo.com")
```



La figura muestra cómo se usa una intent para iniciar una actividad. Cuando El objeto Intent nombra un componente de actividad específico de forma explícita, el sistema inicia de inmediato ese componente.

Filtro de intents

Es una expresión en el archivo de manifiesto de una app que especifica el tipo de intents que el componente podría recibir. Por ejemplo, al declarar un filtro de intents para una actividad, permites que otras apps inicien tu actividad directamente con un tipo determinado de intent. Del mismo modo, si no declaras ningún filtro de intents para una actividad, esta solo se puede iniciar con un intent explícito.

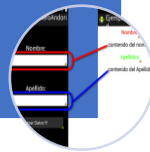
- Para crear un Intent, se necesita especificar el contexto y el componente que se desea iniciar (en el caso de los Intents explícitos) o la acción que se desea realizar (en el caso de los Intents implícitos)

CREAR UN INTENTS



- Se pueden agregar datos adicionales al Intent utilizando el método `putExtra`, lo que permite pasar información entre componentes.
- `intent.putExtra("clave", "valor")`

Agregar Datos Extras



- Una vez creado el Intent y agregados los datos necesarios, se inicia la actividad usando `startActivity`.
- `startActivity(intent)`

Iniciar la Actividad



Un objeto Intent contiene información que usa el sistema Android para determinar qué componente debe comenzar (como el nombre exacto del componente o el componente) que debe recibir el intent), además de información que el componente receptor utiliza en para realizar la acción correctamente (como la acción que se debe realizar y los datos sobre los que debe actuar). La información principal que se incluye en un Intent es la siguiente:

- **Nombre del componente**
- **Acción**
- **Datos**
- **Categoría**
- **Adicional**
- **Marcas**

Manejo de Resultados de Intents

Enviar un Resultado desde una Actividad

- **Establecer el Resultado**

```
val intent = Intent()

intent.putExtra("clave", "valor")

setResult(Activity.RESULT_OK, intent)
```

- **Finalizar la Actividad:** Se llama al método `finish()` para cerrar la actividad.

Iniciar una Actividad para Obtener un Resultado

Para iniciar una actividad y esperar un resultado, se utiliza el método `startActivityForResult`

```
val intent = Intent(this, NombreDeLaActividad::class.java)

startActivityForResult(intent, CODIGO_SOLICITUD)finish()
```

Manejar el Resultado en la Actividad Llamante

Para recibir el resultado, se debe sobrescribir el método `onActivityResult`.

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {

    super.onActivityResult(requestCode, resultCode, data)

    if (requestCode == CODIGO_SOLICITUD && resultCode == Activity.RESULT_OK) {
```

Ejemplo de una intent explícita

Un intent explícito es aquel que se usa para iniciar un componente específico de la aplicación, como una actividad o un servicio particular en tu app. Para crear un intent explícito, define el nombre del componente para el objeto Intent, otras propiedades del intent son opcionales.

Por ejemplo, si compilaste un servicio en tu app, llamado DownloadService, diseñado para descargar un archivo de la Web, puedes comenzar con el siguiente código:

```
Kotlin    Java

// Executed in an Activity, so 'this' is the Context
// The fileUrl is a string URL, such as "http://www.example.co
val downloadIntent = Intent(this, DownloadService::class.java)
    data = Uri.parse(fileUrl)
}
startService(downloadIntent)
```

El Intent(Context, Class) este constructor proporciona la app Context y el componente, un objeto Class. Por lo tanto, este intent inicia de manera explícita la clase DownloadService en la app.

Ejemplo de una intent implícita

Un intent implícito especifica una acción que puede invocar cualquier aplicación en el dispositivo que pueda para realizar la acción. El uso de un intent implícito es útil cuando la app no puede realizar la acción, pero otras apps probablemente sí, y quieres que el usuario elija qué app usar.

Por ejemplo, si tienes contenido que quieres que el usuario comparta con otras personas, crear un intent con la acción ACTION_SEND y agregar extras que especifiquen el contenido que se compartirá. Cuando llames startActivity() con ese intent, el usuario puede elegir una app mediante la cual compartir el contenido.

```
Kotlin    Java

// Create the text message with a string.
val sendIntent = Intent().apply {
    action = Intent.ACTION_SEND
    putExtra(Intent.EXTRA_TEXT, textMessage)
    type = "text/plain"
}

// Try to invoke the intent.
try {
    startActivity(sendIntent)
} catch (e: ActivityNotFoundException) {
    // Define what your app should do if no activity can handle
}
```



Cómo forzar un selector de apps

Cuando más de una app responde a tu intent implícito, el usuario puede seleccionar qué app se debe usar y establecerla como la opción predeterminada para la acción. La capacidad de seleccionar un valor predeterminado es útil cuando se realiza una acción para la que el usuario probablemente quiera usar siempre la misma aplicación, por ejemplo, al abrir una página web (los usuarios a menudo prefieren un solo navegador web).

Para mostrar el selector, crea un Intent con `createChooser()` y pásalo a `startActivity()`, como se muestra en el siguiente ejemplo. En este ejemplo, se muestra un diálogo con una lista de apps que responden al intent pasado al método `createChooser()`, con el texto proporcionado como título del diálogo.

Kotlin

Java

```
val sendIntent = Intent(Intent.ACTION_SEND)
...

// Always use string resources for UI text.
// This says something like "Share this photo with"
val title: String = resources.getString(R.string.chooser_title)
// Create intent to show the chooser dialog
val chooser: Intent = Intent.createChooser(sendIntent, title)

// Verify the original intent will resolve to at least one activity
if (sendIntent.resolveActivity(packageManager) != null) {
    startActivity(chooser)
}
```

Detecta lanzamientos de intents no seguros

Tu app podría iniciar intents para navegar entre componentes dentro de ella, o para realizar una acción en nombre de otra app. Para mejorar la seguridad de la plataforma, Android 12 (nivel de API 31) y las versiones posteriores proporcionan una función de depuración que te advierte si tu app realiza un inicio no seguro de un intent. Por ejemplo, tu app podría realizar un lanzamiento no seguro de un intent anidado, que es un intent que se pasa como un extra en otro intent.

Si tu app realiza las siguientes dos acciones, el sistema detecta un lanzamiento de intent inseguro y se produce un incumplimiento de `StrictMode`:

La app descomprime un intent anidado desde los valores adicionales de un intent entregado.

La app inicia inmediatamente un componente de la app con ese intent anidado, como cuando se pasa el intent a `startActivity()`, `startService()` o `bindService()`.