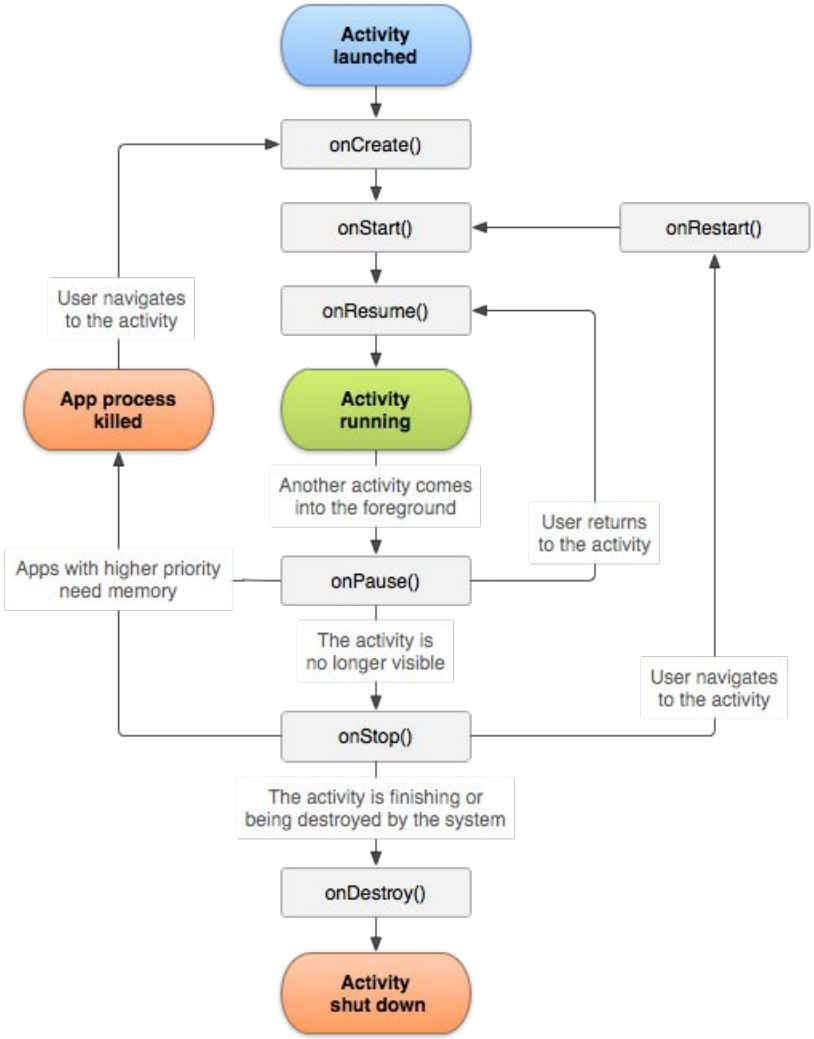
Desarrollo de aplicaciones móviles con Android

Babel Learning

Activity. Ciclo de vida.



Intent

Un intent es una descripción abstracta de una operación que se realizará. Se puede usar con startActivity para iniciar una actividad, broadcastIntent para enviarlo a cualquier componente de BroadcastReceiver interesado, y startService (Intent) o bindService (Intent, ServiceConnection, int) para comunicarse con un servicio en segundo plano.

Un Intent proporciona una función para realizar el enlace de tiempo de ejecución tardío entre el código en diferentes aplicaciones. Su uso más significativo es en el lanzamiento de actividades, donde se puede considerar como el pegamento entre las actividades.

Básicamente es una estructura de datos pasiva que contiene una descripción abstracta de una acción a realizar.

Estructura de los Intent

Los componentes principales de información en un intento son:

1. Acción: la acción general que se realizará, como ACTION_VIEW, ACTION_EDIT, ACTION_MAIN, etc.

2. Datos: los datos para operar, como un registro de persona en la base de datos de contactos, expresado como URI.

Algunos ejemplos de pares de acción / datos son:

ACTION_VIEW ***tel:123*** -- Muestre el marcador del teléfono con el número proporcionado. Observe cómo la acción VIEW hace lo que se considera lo más razonable para un URI en particular.

ACTION_VIEW ***content://contacts/people/1*** -- Muestra información sobre la persona cuyo identificador es "1".

ACTION_DIAL ***content://contacts/people/1*** -- Muestra el marcador del teléfono con la persona completada.

Intents

ACTION_VIEW ***content://contacts/people/*** -- Muestra una lista de personas a través de las cuales el usuario puede navegar. Este ejemplo es una entrada típica de nivel superior en la aplicación Contactos, que le muestra la lista de personas. Seleccionar a una persona en particular para verlo daría como resultado un nuevo Intent.

new Intent { ACTION_VIEW ***content://contacts/people/N*** } siendo utilizada para iniciar una actividad para mostrar a esa persona.

ACTION_DIAL ***tel:123*** -- Muestra el marcador del teléfono con el número proporcionado.

ACTION_EDIT ***content://contacts/people/1*** -- Editar información sobre la persona cuyo identificador es "1".

```
Intent callIntent = new
Intent(Intent.ACTION_CALL,
Uri.parse(uri));
```

```
startActivity(callIntent);
```

SharedPreferences

Interfaz para acceder y modificar los datos de preferencia devueltos por getSharedPreferences (String, int). Para cualquier conjunto particular de preferencias, hay una sola instancia de esta clase que todos los clientes comparten. Las modificaciones a las preferencias deben pasar por un objeto SharedPreferences.

Editor para garantizar que los valores de preferencia permanezcan en un estado constante y controlados cuando se comprometan con el almacenamiento.

Los objetos que se devuelven de los diversos métodos get deben ser tratados como inmutables por la aplicación.

Nota: Esta clase no admite el uso en múltiples procesos.

SharedPreferences. Métodos Públicos

Verifica si las preferencias contienen algo.

boolean contains (String key)

GetBoolean.

boolean getBoolean (String key,
boolean defValue)

Obtiene el valor de un booleano desde las preferencias.

Editar.

SharedPreferences.Editor edit ()

Cree un nuevo Editor para estas preferencias, a través del cual puede hacer modificaciones a los datos en las preferencias y volver a cometer esos cambios atómicamente al objeto SharedPreferences.

Tenga en cuenta que debe llamar a commit () para que los cambios que realice en el Editor realmente aparezcan en SharedPreferences.

SharedPreferences. Métodos Públicos

getAll.

Map<String, ?> getAll ()

Recupera todos los valores de las preferencias.

Tenga en cuenta que no debe modificar la colección devuelta por este método ni alterar ninguno de sus contenidos. La consistencia de sus datos almacenados no está garantizada si lo hace.

GetBoolean.

boolean getBoolean (String key,
boolean defValue)

Obtiene el valor de un booleano desde las preferencias.

getFloat.

float getFloat (String key,
float defValue)

Obtiene el valor de un float desde las preferencias.

SharedPreferences. Métodos Públicos

getInt.

```
int getInt (String key,
            int defValue)
```

Obtiene el valor de un entero desde las preferencias.

getLong.

```
long getLong (String key,
              long defValue)
```

Obtiene el valor de un Long desde las preferencias.

getString.

```
String getString (String key,
                  String defValue)
```

Obtiene el valor de un Long desde las preferencias.

SharedPreferences. Métodos Públicos

getStringSet.

Set<String> getStringSet (String key,
Set<String> defValues)

Recupere un conjunto de valores de cadena de las preferencias.

Tenga en cuenta que no debe modificar la instancia establecida devuelta por esta llamada. La consistencia de los datos almacenados no está garantizada si lo hace, ni su capacidad para modificar la instancia en absoluto.

```
SharedPreferences preferences =
getSharedPreferences("MyPreferences",
Context.MODE_PRIVATE);
```

```
String stringListUser =
prefs.getString("ListUsers", null);
```

Ejercicios.

#1. Crear una aplicación que haga una llamada, utilizando el intent correspondiente.

Nota. Deben usar “tel:” y el Intent de ActionCall.

#2. Crear una aplicación que guarde los datos del usuario en el las preferencias.

Nota. En caso de que quieran almacenar un listado de usuarios. Pueden utilizar el paquete para deserializarlo como json.

```
compile
'com.google.code.gson:gson:2.8.0'
```