Programmation avec android Cours 2

Hadjila Fethallah

Maître de Conférences au Département d'Informatique

F_hadjila@mail.univ-tlemcen.dz

recapitulation

- Composants de base
  - activités
  - Broadcast receivers
  - services
  - Content providers
- Communication intercomposants
  - les intents avec :
  - Invocation implicite et explicite

Les intents (intentions)

- C’est une classe qui represente soit:
  - Une description abstraite d’une operation
  - une specification d’une activité cible à lancer
  - Un evenement qui occure dans android
- Exemples:
  - description d’une operation, ex: ACTION_SENDTO
  - evenement, ex :Intent.ACTION_BOOT_COMPLETED, Intent.ACTION_BATTERY_LOW
- Les Intents permettent d'interagir avec des composants de la même application ou des composants appartenant à d'autres applications.
- Ils peuvent démarrer un service ou un broadcast receiver

Types d'intents

- **Les Intents explicites** définissent explicitement le composant qui doit être appelé par Android
- Exemple (ActivityOne.java):
  - **Intent i = new Intent(ActivityOne.this, ActivityTwo.class);**
  - **i.putExtra("id1", "val1 "); i.putExtra("id2", "val2");**
- id1/id2: representent les clés (de type chaines de caractères, mais les valeurs peuvent être string ou des types primitifs)

Types d’intents

- Traitement effectué dans l’activité cible
- Bundle extras = getIntent().getExtras();
- if (extras != null) {

String value1 = extras.getString(“id1”);

if (value1 != null) { // faire qcq chose }}

Types d’intents

- Les Intents implicites précisent l'action qui devrait être exécutée avec éventuellement des arguments
- Ex: pour afficher une page Web on met:
- Intent i = new Intent(Intent.ACTION_VIEW, Uri.parse("http://....."));startActivity(i);
- Tous les navigateurs Web installés doivent être enregistrés avec l'Intent correspondant dans le fichier manifest.xml.

Attributs d’un intent

Une intention est définie par:

- son **action**,par ex: **ACTION_SENDTO, ACTION_DIAL, ACTION_VIEW**
- ses **données** (data):
  - Uri.parse("geo:33.7749,-1.4192");
  - Uri.parse(“tel:+213550102030");
- sa **catégorie** par ex:
  - CATEGORY_LAUNCHER, CATEGORY_BROWSABLE
- Composant (Component) :nom de l’activité cible
- Son **Type**, ex: "text/plain","text/html","image/png"," image/jpg"
- Extra: paires clés/valeurs
  - it.putExtra(Intent.EXTRA_EMAIL, new String[] {" et1@gmail.com ", "et2@yahoo.com"});
- Flags ex:
  - FLAG_ACTIVITY_CLEAR_TASK
  - FLAG_ACTIVITY_NO_HISTORY

Recuperation du résultat d'une activité cible

- **code de l’activité source (premiere partie):**
- public void onClick(View view) { Intent i = new Intent(ActivityOne.this, ActivityTwo.class); i.putExtra(“id1", “val1"); i.putExtra(“id2", “val2");
- // initialiser request_code à une valeur entiere arbitraire
- startActivityForResult(i, REQUEST_CODE); }

Recuperation du résultat d'une activité cible

- **Code de l’activité cible:**
- public void finish() { // Preparer data intent Intent data = new Intent(); data.putExtra("id1", "L3");
- data.putExtra("id2", "android"); // RESULT_OK veut dire une fin normale de l’activité cible
- setResult(RESULT_OK, data); super.finish(); }

Récupération du résultat d'une activité cible

- dès que la l'activité cible est terminée, la méthode onActivityResult() de **l'activité** source est exécutée.
- protected void onActivityResult(int requestCode, int resultCode, Intent data) { if (resultCode == RESULT_OK && requestCode == REQUEST_CODE) { if (data.hasExtra("id1")) { Toast.makeText(this, data.getExtras().getString(" id1"), Toast.LENGTH_SHORT).show(); } } }

Broadcast receivers

- C’est une classe sans UI qui réagit aux evenements qui se produisent dans android
- ex:
  - fin de démarrage d’android
  - Lancement d’un SMS/MMS
  - La batterie atteint un certain niveau d’energie
  - ….
- Elle fonctionne selon le modele publier souscrire

Workflow typique

- Enregistrement du brodacast receiver
  - Statique (AndroidManifest.xml) . Ceci est fait après demarrage d'android ou après installation du apk
  - ou dynamique (Context.registerReceiver())
- Une autre classe crée et diffuse un intent
- Android délivre le message (intent) aux composants à son ecoute et lance leurs respectives onreceive() methode + l'intent comme argument
- L'évenement est géré dans le code de onReceive()
- Apres l'éxecution de onReceive(), Android a le droit d'éliminer le broadcastreceiver.

broadcast receiver (BR)

- Le BR ne contient que la méthode onReceive
- Un composant peut définir des permissions sur les BR appelés, de même un BR peut définir des permissions sur les activités appelantes.
- il n'a pas le droit de lancer une fenêtre de dialogue/ activité (sa durée de vie est limitée)
- Puisque un BR s'éxecute dans le main thread, il est préférable de lancer un service dans onReceive pour les longues tâches
- Les BR enregistrés dynamiquement ne réagissent pas aux intents diffusés après l'arrêt de l'application
- Un BR peut arrêter la diffusion de l'intent au reste des BR concernés

Exemple complet

- Etape1 :enregistrement statique
- <application android:icon="@drawable/ic_launcher" android:label="@string/app_name" android:theme="@style/AppTheme" > <receiver android:name="Receiver1"> <intent-filter android:priority="100" > <action android:name=" CUSTOM_INTENT "> </action> </intent-filter> </receiver> </application>

Exemple complet

- Etape2 : création et diffusion de l'intent dans l'activité source (dans la fonction oncreate)
- ……
- Intent intent = new Intent(); intent.setAction("CUSTOM_INTENT"); sendBroadcast(intent);
- ……..

Exemple complet

- Etape 3:Reaction du BroadcastRceiver
- public class Receiver1 extends BroadcastReceiver {
- public void onReceive(Context context, Intent intent) { Toast.makeText(context, "Intent intercepté.", Toast.LENGTH_LONG).show(); } }

Caracteristique des intents diffusés

- Portée Locale/globale
- Normal/ordered
- Sticky/non Sticky
- Sans / avec permission

FIN du Cours2