CONTENT PROVIDER, BROADCAST RECEIVER, SERVICE

David Bilík

ANDROID DEVELOPER @ ACKEE

#CODECAMPCZ

CO NÁS DNES ČEKÁ

- Intenty
- Content Provider + Loader
- Broadcast Receiver
- Service/Intent Service

INTENTY

- Slouží ke spouštění různých komponent ať vaší aplikace nebo cizích aplikací
- nastavují se mu dodatečné příznaky, ať systém ví, kterou komponentu spustit
- Explicitní
  - spouští konkrétní třídu (komponentu)
    - startActivity(new Intent(this, MainActivity.class))
  - komponenta může být Activity, Service, BroadcastReceiver
  - Používají se pro spouštění komponent vaší aplikace, protože je třeba definovat přesný název třídy, který často neznáte

INTENTY

- Implicitní
  - nemají definovaný cíl
  - pokud existuje více adresátů, je uživateli nabídnut dialog a zvolí si aplikaci, přes kterou chce akci dokončit
  - Intentu se může nastavit akce a data a dodatečně kategori a extras
    - Akce - definuje akci, pro kterou se má komponenta spustit (např. ACTION_SEND, ACTION_CALL)
    - Data - definuje data (nebo typ dat) na kterých se má akce provést. Data jsou předána formou Uri
    - Kategorie - Dodatečná informace o tom, jaká komponenta má zpracovat intent
    - Extras - Bundle s daty

INTENT FILTERS

- Pokud chceme, aby se naše aktivita nabízela pro nějakou konkrétní akci/kategorii, je třeba k aktivitě přidat do AndroidManifestu IntentFilter

```
<intent-filter>
    <action android:name=" android.intent.action.GET_CONTENT " />
    <category android:name=" android.intent.category.DEFAULT " />
    <data android:mimeType="vnd.android.cursor.item/vnd.google.note" />
</intent-filter>
```

http://developer.android.com/reference/android/content/Intent.html

CONTENT PROVIDER

- Prostředek pro poskytování dat - naší aplikaci ale i cizím aplikacím
- Data vrácena v podobě cursoru, ale nemusí se nacházet pouze v SQLiteDB ale třeba i v souboru nebo na webu
- Pro přístup k datům od ContentProvider slouží ContentResolver, který je přístupný z Context. mCtx.getContentResolver()
- Každý content provider definuje URI, která popisuje data, které poskytuje
- Existují systémové Content Providery na kontakty, SMS, zmeškané volání, záložky, multimédia, …

URI

- URI - Uniform Resource Identifier - jednotný identifikátor zdroje
- Skládá se ze 4 částí (povinné jen scheme a authority):
- scheme://authority/data_type/identifier
  - content://cz.codecamp.example/users/123
- schéma - standartní předpona indikující zpracování Content Providerem
- autorita - jedinečný identifikátor content provideru, nejčastěji se jedná o package name aplikace, musí být definován v android manifestu
- data_type - definuje typ dat, o která žádáme
- identifier - specifikuje konkrétní záznam v kolekci

IMPLEMENTACE

- Je třeba podědit systémovou třídu ContentProvider a přepsat 6 metod:
  - onCreate - zavoláno po vytvoření, slouží např. k inicializaci databáze
  - getType - vrací MIME typ ke konkrétní URI
  - query - dotaz na data
  - insert - vkládání dat
  - delete - mazání dat
  - upadte - aktualizace dat
- Definovat ContentProvider v AndroidManifestu společně s authority
  - příznak exported true/false - je přístupné z jiných aplikací

URI MATCHER

- Pokud náš ContentProvider poskytuje více URI, je vhodné si nadefinovat UriMatcher, který nám pro předpis URI dokáže jednoduše rozhodnout, o který typ jde
- příklad

CONTENT RESOLVER

- Na naše data se ptáme pomocí ContentResolver třídy
  - mCtx.getContentResolver().query(URI, projection, selection, selectionArgs, order)
- obdobné API jako Sqlite databáze, stejně tak pro insert/update/delete
- Hlavní rozdíl je v tom, že se ptáme na URI a ne na jméno tabulky

LOADERY

- Loader - další asynchronní komponenta systému
- Stará se automaticky o připojování/odpojování k fragmentu/aktivitě při změně konfigurace
- Klient musí implementovat callbacky na vytvoření loaderu, doručení výsledku a reset loaderu
- Nejčastější využití je implementace CursorLoader
- Asynchronně načte data z dané URI a doručí výsledný cursor
- Stará se o zavírání cursoru při finishování aktivity/ fragmentu

LOADERY

- Cursoru můžeme dát tzv. NotificationUri
- Pokud proběhne nějaká změna na této uri, CursorLoader automaticky nahraje nová data
  - není tedy třeba po změnách v databázi volat ručně donačtení aktuálních dat
- častý usecase - stahování dat na pozadí a zobrazení aktuálních dat v UI aplikace.

BROADCAST RECEIVER

- Broadcast reciever poslouchá na “některé” události specifikované Intent Filterem
- Může se jednat buď o systémové události (změna typu připojení, low battery, incoming message) a nebo o naše vlastní události, které si zašleme
- Můžeme si takto posílat zprávy mezi nesouvisejícími komponentami aplikace (služba na pozadí mi něco stahuje a notifikuje aktivitu o progressu)

BROADCAST RECEIVER

- Dvě možnosti poslouchání - vydefinované v Manifestu a nebo ruční registrace v kódu
  - pokud děláme ruční registraci, je třeba vždy receiver odpojit, jinak poslouchá “navždy”
- Pro implementaci je třeba podědit třídu BroadcastReceiver a přepsat metodu onReceive(context, intent)
- běží na hlavním vlákně, pro delší operace je třeba spustit nějakou jinou službu (např. pokaždé když se připojím k internetu, chci stáhnout nová data)

BROADCAST RECEIVER

- Odesílání zpráv pomocí intentů - vydefinujeme akci (a případně další součásti) a receivery, kteří na tuto akci poslouchají dostanou zprávu
- Dvě možnosti odeslání zprávy
  - non-ordered broadcast - všichni receiveři dostanou zprávu v nedefinovaném pořadí - Context.sendBroadcast()
  - orderer broadcast - je doručen postupně receiverům podle priority definované v manifestu
    - může se rozhodnout, zda daný broadcast pošle dál a nebo ho zruší

SERVICE

- Poslední ze 4 základních komponent systému (Activity, Content Provider, Broadcast Receiver, Service)
- Je to služba, která běží nezávisle na UI aplikace
- Hodí se tedy pro operace, které UI nevyžadují
  - stahování dat na pozadí
  - přehrávání hudby
  - periodicky se opakující akce (odeslání GPS polohy na server každých 30s)

SERVICE

- Běží v UI vlákne aplikace. Pro náročnější operace je tedy třeba spustit vedlejší vlákno a na něm je vykonat
- Jsou spouštěny z jiných komponent aplikace (BroadcastReceiver, Activity)
- Je menší pravděpodobnost, že ji systém zabije
- Pokud ji přeci jen zabije, existuje mechanismus pro znovuspuštění dané service
- Je nutné ji (stejně jako ostatní komponenty) registrovat v AndroidManifestu
- Běží vždy jen jedna instance

LIFECYCLE

- Stejně jako Aktivity a Fragmenty má i Service svůj životní cyklus (naštěstí ne tak složitý)
- onCreate() - zavoláno při prvním vytvoření instance
- onStartCommand() - zavoláno, pokud klient zavolal Context.startService()
- onBind() - klient se připojil k service pomocí Context.bindService()
- onDestroy() - zavoláno při zrušení service

ZABITÍ SYSTÉMEM

- onStartCommand() vrací konstanty, které určují jak se má Service chovat při zabití systémem
  - START_NOT_STICKY - pokud systém zabil service po tom co byla spuštěna a žádný další požadavek na její spuštění není, služba nebude znovu spuštěna
  - START_STICKY - služba je spuštěna znovu, ale Intent který je předán do metody onStartCommand je null. Vhodné pro Servicy, které nevykonávají nějaké příkazy ale běží nezávisle - například přehrávač hudby
  - START_REDELIVER_INTENT - služba je znovu spuštěna s posledním zavolaným intentem. Vhodné tehdy, když chcete například navázat na stahování souboru

BOUNDED SERVICES

- Services mohou být buď "started" a nebo "bounded". Started servica se startuje/ukončuje pomocí metod startService() stopService().
  - Vhodné tehdy, když nechceme udržovat se service spojení. Servicu je nutné stopnout buď z aktivity a nebo ze servicy pomocí stopSelf() metody
- Bounded servicy slouží k navázání dlouhodobého spojení mezi aktivitou a service. Využívá se metoda bindService/ unbindService
  - není třeba explicitně ukončovat service, ukončí se s posledním odpojeným binderem

INTENT SERVICE

- Service, která běží ve vedlejším vlákně
- Jediná metoda, kterou je třeba přepsat je onHandleIntent(Intent), která běží na vedlejším vlákně
- Intenty přicházející do onStartCommand jsou řazeny do fronty a zpracovávány metodou onHandleIntent. Pokud se fronta vyprázdní, service je ukončena
- Nedá se k takovéto service nabindovat

DĚKUJI ZA POZORNOST!

OTÁZKY?