Android App Programming

Lecture 5: Databases
November 28, 2017

Thomas Nowak
Université Paris-Sud
Last lecture recap
Services

• for background tasks

• ex: downloads (that should continue after the app quit)

• ex: music player

• ex: polling an SMTP for new emails
Service Lifecycle

Unbounded service

Bounded service

Call to startService()
- onCreate()
- onStartCommand()

Service running
- The service is stopped by itself or a client
- onDestroy()

Service shut down

Call to bindService()
- onCreate()
- onBind()

Clients are bound to service
- All clients unbind by calling unbindService()
- onUnbind()
- onDestroy()

Service shut down
Databases
Saving State

- several possibilities:
  - passing bundles (ugh)
  - write into a file (uuuugh)
  - databases
  - preferences

```java
SharedPreferences prefs = getPreferences(MODE_PRIVATE)
```
Databases

- SQL databases:
  - relational data model
  - proven, robust, optimized

- NoSQL databases:
  - different data model, e.g., objects, graphs
  - less data translation necessary
Databases

- Physical location of data varies:
  - locally (on the phone)
  - remote (on one or more servers on the internet)
    - e.g., Firebase
SQLite

- DBMS integrated into Android: SQLite
- packaged into the app
  - runs in the same process
  - no connection setup necessary
- use the class `SQLiteDatabase` included in Android
SQLite in Android

- get instance of `SQLiteDatabase` via `openOrCreateDatabase`

- call `db.rawQuery` with SQL statement for results

- loop over result set with a `Cursor`

- for SQL statements without result, call `db.execSQL`

- there are specialized methods for some operations, e.g., `db.insert`
Coding demo
Lab 5

1. Redo your quiz app with the questions stored in an SQLite database
Questions