Telerik Academy

Working with local data

SQLite, Content Providers

Antony Jekov

Android course

Telerik Software Academy

http://academy.telerik.com

SQLite in Android

- SQLite is an Open Source database.
- SQLite supports standard relational database features like SQL syntax, transactions and prepared statements.
- SQLite is embedded into every Android device – no need to setup or administer the database – all done by the Android platform.

SQLite in Android

- Access to an SQLite database involves accessing the file system - slow, make it async.
- Default creation place for the database - DATA/data/APP_NAME/databases/FILENAME.

SQLite in Android

- To create and upgrade a database in your Android application you create a subclass of the SQLiteOpenHelper class.
  - onCreate() - called by the framework, if the database is accessed but not yet created.
  - onUpgrade() - called, if the database version is increased in your application code. You can update drop and recreate.

SQLite in Android

- The SQLiteOpenHelper class provides the getReadableDatabase() and getWriteableDatabase() methods to get access to an SQLiteDatabase object in either read or write mode.
- The database tables should use the identifier _id for the primary key of the table. Several Android functions rely on this standard.
- It is a good practice to have a seperate class for each table.

SQLite in Android

- **SQLiteDatabase is the base class for working with a SQLite database in Android, provides methods to open, query, update and close the database as well as insert(), update() and delete().**
- **In addition it provides the execSQL() method - allows to execute an SQL statement directly.**

Cursor

- **A Cursor represents the result of a query and points to one row of the query result. This way Android can buffer the query results efficiently, as it does not have to load all data into memory.**

```
Cursor cursor = getReadableDatabase().
  rawQuery("select * from todo where _id = ?", new String[] { id });
```

SQLite

Live Demo

Content provider

- A SQLite database is private to the application which creates it. If you want to share data with other applications you can use a content provider.
- The access to a content provider is done via an URI. The basis for the URI is defined in the declaration of the provider in the AndroidManifest.xml file via the android:authorities attribute.

Content provider

- A content provider can be accessed from several programs at the same time – concurrency problem.
- Use synchronized methods to prevent that or put the android:multiprocess=true attribute to the provider definition in the manifest. It will cause the provider to be initialized seperately for each client.

Content Provider

Live Demo

Loader

- **Loader class allows you to load data asynchronously in an activity or a fragment.**
- **Introduced in version 3.0 of the Android platform, available down to 1.6 using compatibility.**
- **Implementing a loader:**
  - **getLoaderManager().initLoader(0, null, this);**
    - **Unique ID**
    - **Bundle with additional info**
    - **The loader callbacks interface**

Loader

- **CursorLoader class - default implementation for handling SQlite database connections.**
- **This Loader performs the database query in a background thread so that the application is not blocked.**

Resources

- Working with local data tutorial

Working with local data

Questions?

Exercises

1. Create an application, that works with SQLite database and keeps some data. Open a content provider, that is available for other applications.
2. Create an application, that connects to the content provider of the firs application and requests and displays the info retrieved by first application.