

CS 683 Emerging Technologies  
Fall Semester, 2008  
Doc 19 Android Data  
Nov 25 2008

## References

Google Android Documentation, <http://code.google.com/android/documentation.html>

# Data Topics

Preferences

Files

SQLite database

Content Providers

Network

# Preferences

Key value pairs for program

Key - string

Value

boolean

float

int

long

string

`getPreferences(int mode)`

For access in activity only

`getSharedPreferences(String name,int mode)`

To share preferences with other activities

mode

0 = `MODE_PRIVATE`

`MODE_WORLD_READABLE`

`MODE_WORLD_WRITEABLE`

Cannot share preferences across packages

# Example

```
public class Calc extends Activity {
public static final String PREFS_NAME = "MyPrefsFile";
protected void onCreate(Bundle state){
    super.onCreate(state);

    SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
    boolean silent = settings.getBoolean("silentMode", false);
    setSilent(silent);
}

protected void onStop(){
    super.onStop();
    SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
    SharedPreferences.Editor editor = settings.edit();
    editor.putBoolean("silentMode", mSilentMode);
    editor.commit();
}
}
```

# Files

Application can write/read files on phone

Cannot directly read files written by other application

Write a file

`FileOutputStream openFileOutput(String name, int mode)`

Creates file if it does not exist

mode

`0 = MODE_PRIVATE`

`MODE_APPEND`

`MODE_WORLD_READABLE`

`MODE_WORLD_WRITEABLE`

`FileInputStream openFileInput(String name)`

name can not contain path separators

# Static files

You can package static files with your application

Place file in `res/raw/<mydatafile>`

Generates resource id in R

Read file using

`Resources.openRawResource (R.raw.mydatafile)`

# Database

## SQLite

Embedded SQL database engine

Free

Source is in public domain

Transactions

File format is cross-platform

<http://www.sqlite.org/index.html>



# Key Android Database Classes

`android.database.sqlite.SQLiteOpenHelper`

Database creation

Version management

Database access

`android.database.sqlite.SQLiteDatabase`

Create, delete, execute SQL commands

`android.database.Cursor`

Read-write access to the result set

# SQLiteOpenHelper

synchronized void close()

Close any open database object.

synchronized SQLiteDatabase getReadableDatabase()

Create and/or open a database.

synchronized SQLiteDatabase getWritableDatabase()

Create and/or open a database that will be used for reading and writing.

abstract void onCreate(SQLiteDatabase db)

Called when the database is created for the first time.

void onOpen(SQLiteDatabase db)

Called when the database has been opened.

abstract void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)

Called when the database needs to be upgraded.

# Android Databases

Accessible to all classes in an application

Can't access directly databases from other applications

Database errors are logged

Can connect to database on phone from shell

<http://code.google.com/android/reference/adb.html#sqlite>

# Example

Show

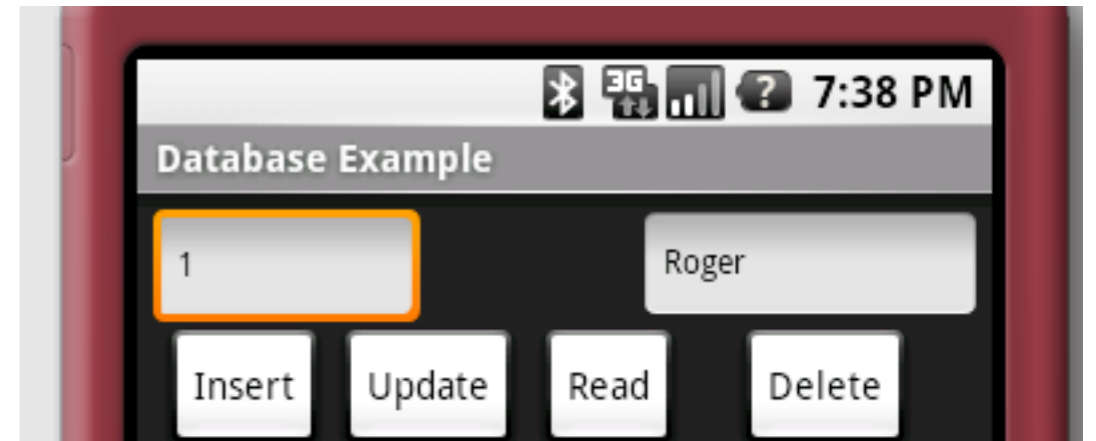
Creating database

Inserts

Update

Delete

Query



# DatabaseHelper

```
public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "name.db";
    private static final int DATABASE_VERSION = 1;

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    public void onCreate(SQLiteDatabase nameDb) {
        nameDb.execSQL("CREATE TABLE " + "NAMES" + " ("
            + "_ID" + " INTEGER PRIMARY KEY,"
            + "NAME" + " TEXT"
            + ");");
        nameDb.execSQL("INSERT INTO NAMES ( name) VALUES ('Roger' );");
    }

    public void onUpgrade(SQLiteDatabase arg0, int oldVersion, int newVersion) {
    }
}
```

# DatabaseExample.java - Main Class

```
public class DatabaseExample extends Activity implements View.OnClickListener {
    private EditText databaseldText;
    private EditText nameText;
    private DatabaseHelper namesHelper;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        int[] buttonIds = { R.id.delete, R.id.read, R.id.insert, R.id.update };
        for (int id : buttonIds) {
            Button button = (Button) findViewById(id);
            button.setOnClickListener(this);
        }
        databaseldText = (EditText) this.findViewById(R.id.databaseld);
        nameText = (EditText) this.findViewById(R.id.name);
        namesHelper = (new DatabaseHelper(this));
        displayDatabaseRecord(1);
    }
}
```

# DatabaseExample.java

```
private void displayDatabaseRecord(int id) {
    displayDatabaseRecord(String.valueOf(id));
}

private void displayDatabaseRecord(String id) {
    SQLiteDatabase nameDb = namesHelper.getWritableDatabase();
    Cursor result = nameDb.rawQuery("select * from NAMES where _ID = ?",
        new String[] { id });
    int rowCount = result.getCount();
    if (rowCount > 0) {
        result.moveToFirst();
        databaseldText.setText(String.valueOf(result.getInt(0)));
        nameText.setText(result.getString(1));
    }
}
```

# DatabaseExample.java

```
private String getName() {  
    return nameText.getText().toString();  
}  
  
private String getId() {  
    return databaseldText.getText().toString();  
}
```



# DatabaseExample.java

```
public void onClick(View clicked) {
    SQLiteDatabase db = namesHelper.getWritableDatabase();
    switch (clicked.getId()) {
        case R.id.read:
            displayDatabaseRecord(getId());
            break;
        case R.id.delete:
            db.delete("NAMES", "_ID = ?", new String[] { getId() });
            break;
    }
}
```

# onClick

```
case R.id.insert:
    ContentValues newName = new ContentValues(1);
    newName.put("NAME", getName());
    db.insert("NAMES", null, newName);
    break;
case R.id.update:
    ContentValues updateName = new ContentValues(1);
    updateName.put("NAME", getName());
    db.update("NAMES", updateName, "_ID = ?", new String[] { getId() });
    break;
}
}
}
```