

Android Notepad チュートリアル

[[Android](#)][[Java](#)][[Eclipse](#)]

- ・ <http://developer.android.com/resources/tutorials/notepad/index.html>

準備

ダウンロードして展開

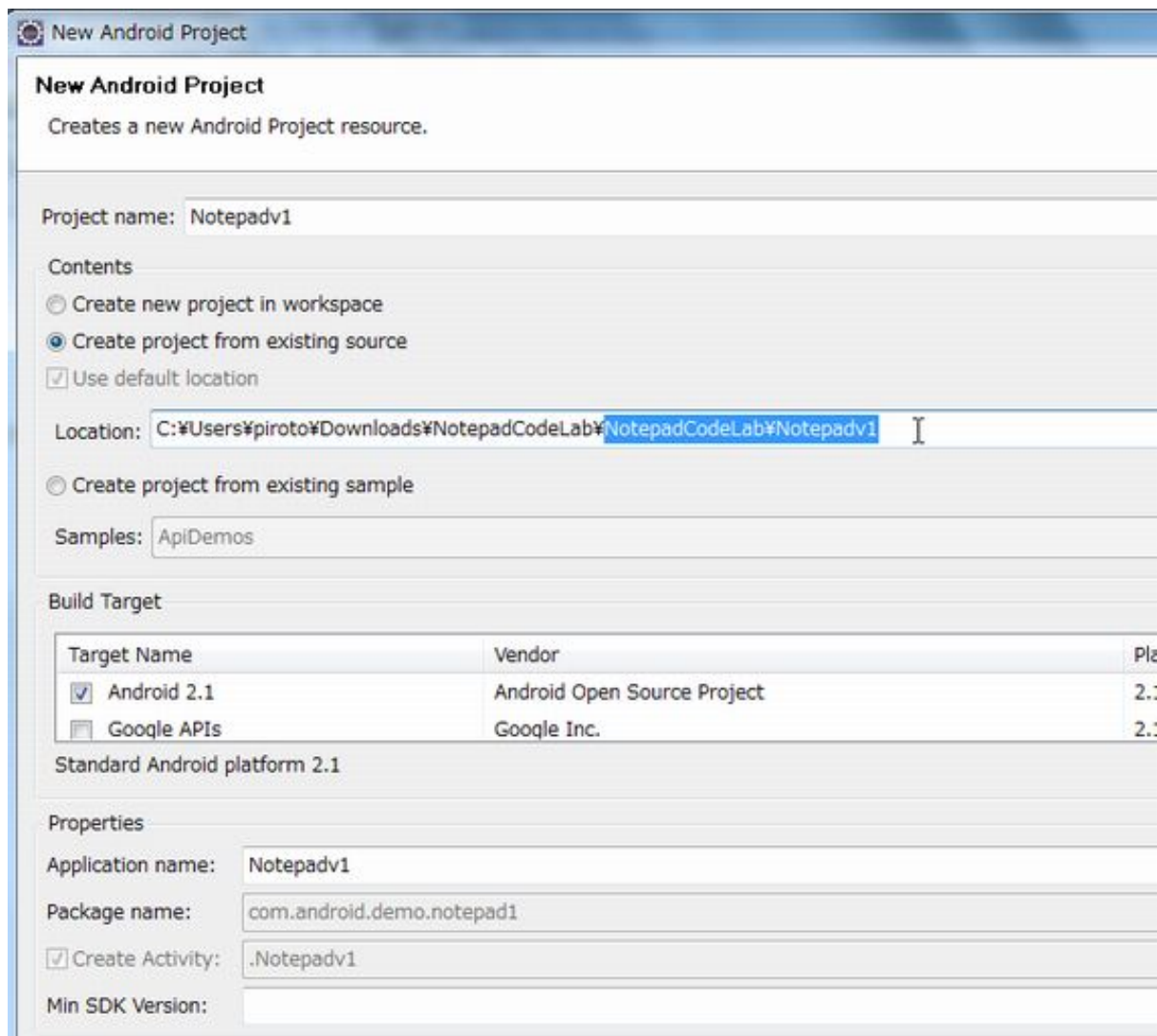
- ・ <http://developer.android.com/resources/tutorials/notepad/codelab/NotepadCodeLab.zip>

(1) 簡単なノートリストの作成

- ・ 新しいノートを追加する（まだ編集はできない）
- ・ `ListActivity` を試してみる
- ・ `SQLite` を試してみる

プロジェクトの作成

- ・ Create project from existing source を選択
- ・ 解凍したフォルダから、NotepadCodeLab/Notepadv1 を選択
- ・ Finish



NotesDbAdapter

用途

- NotesDbAdapter は SQLite へのデータアクセスをカプセル化する。

宣言

- クラスの前頭で、テーブルのフィールドおよびデータベースが存在しない場合に生成する文字列を定義
- データベースは data という名前で、notes という _id、title、body の 3 つのフィールドをもつテーブルを持つ。

コンストラクタ

- コンストラクタは、Context パラメータを取り、それにより Android OS との通信を許可。これは Android にアクセスする一般的な方法
- Activity クラスは Context を実装しているため、Context が必要な時に渡すことができる。

メソッド

open()

- ・ open() メソッドは、DatabaseHelper のインスタンスを呼び出す。
- ・ DatabaseHelper は SQLiteOpenHelper のローカル実装。
- ・ getWritableDatabase() で、データベースを作成、オープン

close()

- ・ データベースを閉じてリリースを解放

createNote()

- ・ 新しい note のためのタイトルと本文の文字列を引数に取り、note をデータベースに作成する
- ・ 作成が成功した場合、_id を返す。

deleteNote()

- ・ 行 ID を引数に取り、note をデータベースから削除

fetchAllNotes()

- ・ クエリーを発行して、カーソルを得る
- ・ query() には、データベーステーブル名、取得したいカラムリスト、残りのフィールドは順に、selection、selectionArgs、groupBy、having、orderBy
- ・ fetchNote() は、fetchAllNotes() と同様だが、行 ID で 1 つの note を取得する

updateNote()

- ・ 行 ID、タイトル、本文を引数に取り、ContentValues を使って note を更新する。

レイアウト

LinearLayout の設定

すべての Android の XML レイアウトのヘッダーは以下で始まらなければならない

```
<?xml version="1.0" encoding="utf-8"?>.
```

次に、たいていの場合、以下の様なレイアウトを置く

```
LinearLayout
```

トップレベルのコンポーネントまたはレイアウトで Android のネームスペースを宣言する

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

LinearLayout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <ListView android:id="@android:id/list"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <TextView android:id="@android:id/empty"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/no_notes"/>
</LinearLayout>
```

行のレイアウト

notes_row.xml

- res/layout に notes_row.xml を作成

```
<?xml version="1.0" encoding="utf-8"?>
<TextView android:id="@+id/text1"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

Notepadv1 クラスの編集

オーバーライドメソッド

onCreate()

- Activity が開始されたときに呼び出される。
- リソースの設定、ステートの設定を実施

onCreateOptionsMenu()

- Activity メニューの有効化
- ユーザーがメニューボタンをたいた時に呼び出される

onOptionsItemSelected()

- 他の半数のメニューと等しい
- メニューから生成されるイベントを制御

基底クラスの変更

- Activity を ListActivity に

```
public class Notepadv1 extends ListActivity
```

onCreate()

```

private NotesDbAdapter mDbHelper;

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.notepad_list);
    mDbHelper = new NotesDbAdapter(this);
    mDbHelper.open();
    fillData();
}

```

・ fillData() はまだ未作成

onCreateOptionsMenu()

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Notepad v1</string>
    <string name="no_notes">No Notes Yet</string>
    <string name="menu_insert">Add Item</string>
</resources>

public class Notepadv1 extends ListActivity {
    public static final int INSERT_ID = Menu.FIRST;
    ;

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // TODO Auto-generated method stub
        boolean result = super.onCreateOptionsMenu(menu);
        menu.add(0, INSERT_ID, 0, R.string.menu_insert);
        return result;
    }
}

```

onOptionsItemSelected()

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // TODO Auto-generated method stub
    switch (item.getItemId()) {
        case INSERT_ID:
            createNote();
            return true;
    }
    return super.onOptionsItemSelected(item);
}

```

createNote()

```

public void createNote() {
    String noteName = "Note " + mNoteNumber++;
    mDbHelper.createNote(noteName, "");
    fillData();
}

```

fillData()

```

private void fillData() {
    Cursor c = mDbHelper.fetchAllNotes();
    startManagingCursor(c);

    String[] from = new String[] { NotesDbAdapter.KEY_TITLE };
    int[] to = new int[] { R.id.text1 };

    SimpleCursorAdapter notes =
        new SimpleCursorAdapter(this, R.layout.notes_row, c, from, to);
}

```

```
        setListAdapter(notes);  
    }
```

ここまでで、起動



(2) Activity の追加、高度なレイアウトの使用など

- <http://developer.android.com/resources/tutorials/notepad/notepad-ex2.html>
- 2 つめの Activity を作成しマニフェストに追加
- startActivityForResult() を使って非同期で別の Activity を呼び出す
- Bundle オブジェクトで Activity 間でデータを受け渡す
- 他のさらに高度なレイアウトの使用
- コンテキストメニューの作成

プロジェクトの作成

- 上記 Notepadv1 同様に、NotepadCodeLab/Notepadv2 からプロジェクトを作成する。
- /res/values/strings.xml に今回追加する機能の文字列が追加されている。
- Notepadv2 クラス には、カーソルを使うための mNotesCursor フィールドとともに、いくつかの定数が定義されている。
- fillData() では、mNotesCursor カーソルフィールドを使うようになっている。

- onCreate() は変更なし。
- いくつかのオーバーライドメソッドが記述されている (onCreateContextMenu(), onContextItemSelected(), onListItemClick() and onActivityResult())

表示されている note を削除するためのコンテキストメニューを追加する

registerForContextMenu()

- ListView のアイテムそれぞれに順にコンテキストメニューに registerForContextMenu() を呼び出して登録
- onCreate() の最後に以下を追加

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.notes_list);
    mDbHelper = new NotesDbAdapter(this);
    mDbHelper.open();
    fillData();

    registerForContextMenu(getListView());
}
```

onCreateContextMenu()

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
    ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);

    // TODO: fill in rest of method
    super.onCreateContextMenu(menu, v, menuInfo);
    menu.add(0, DELETE_ID, 0, R.string.menu_delete);
}
```

onContextItemSelected()

```
@Override
public boolean onContextItemSelected(MenuItem item) {

    // TODO: fill in rest of method
    switch(item.getItemId()) {
        case DELETE_ID:
            AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();
            mDbHelper.deleteNote(info.id);
            fillData();
            return true;
    }
    return super.onContextItemSelected(item);
}
```

onActivityResult()

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
    // TODO Auto-generated method stub
    super.onActivityResult(requestCode, resultCode, intent);
    Bundle extras = intent.getExtras();

    switch(requestCode) {
        case ACTIVITY_CREATE:
            String title = extras.getString(NotesDbAdapter.KEY_TITLE);
            String body = extras.getString(NotesDbAdapter.KEY_BODY);
            mDbHelper.createNote(title, body);
            fillData();
            break;
        case ACTIVITY_EDIT:
            Long mRowId = extras.getLong(NotesDbAdapter.KEY_ROWID);
    }
```

```

        if (mRowId != null) {
            String editTitle = extras.getString(NotesDbAdapter.KEY_TITLE);
            String editBody = extras.getString(NotesDbAdapter.KEY_BODY);
            mDbHelper.updateNote(mRowId, editTitle, editBody);
        }
        fillData();
        break;
    }
}

```

createNote()

```

private void createNote() {
    // TODO: fill in implementation
    Intent i = new Intent(this, NoteEdit.class);
    startActivityForResult(i, ACTIVITY_CREATE);
}

```

onListItemClick()

```

@Override
protected void onListItemClick(ListView l, View v, int position, long id) {
    super.onListItemClick(l, v, position, id);

    // TODO: fill in rest of method
    Cursor c = mNotesCursor;
    c.moveToPosition(position);
    Intent i = new Intent(this, NoteEdit.class);
    i.putExtra(NotesDbAdapter.KEY_ROWID, id);
    i.putExtra(NotesDbAdapter.KEY_TITLE,
        c.getString(c.getColumnIndexOrThrow(NotesDbAdapter.KEY_TITLE)));
    i.putExtra(NotesDbAdapter.KEY_BODY,
        c.getString(c.getColumnIndexOrThrow(NotesDbAdapter.KEY_BODY)));

    startActivityForResult(i, ACTIVITY_EDIT);
}

```

onActivityResult()

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
    super.onActivityResult(requestCode, resultCode, intent);

    // TODO: fill in rest of method
    Bundle extras = intent.getExtras();

    switch(requestCode) {
        case ACTIVITY_CREATE:
            String title = extras.getString(NotesDbAdapter.KEY_TITLE);
            String body = extras.getString(NotesDbAdapter.KEY_BODY);
            mDbHelper.createNote(title, body);
            fillData();
            break;
        case ACTIVITY_EDIT:
            Long mRowId = extras.getLong(NotesDbAdapter.KEY_ROWID);
            if (mRowId != null) {
                String editTitle = extras.getString(NotesDbAdapter.KEY_TITLE);
                String editBody = extras.getString(NotesDbAdapter.KEY_BODY);
                mDbHelper.updateNote(mRowId, editTitle, editBody);
            }
            fillData();
            break;
    }
}

```

NoteEdit クラスの追加

```

public class NoteEdit extends Activity {
    private EditText mTitleText;
    private EditText mBodyText;
    private Long mRowId;
}

```



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    setContentView(R.layout.note_edit);
    mTitleText = (EditText) findViewById(R.id.title);
    mBodyText = (EditText) findViewById(R.id.body);
    Button confirmButton = (Button) findViewById(R.id.confirm);

    confirmButton.setOnClickListener(
        new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Bundle bundle = new Bundle();
                bundle.putString(NotesDbAdapter.KEY_TITLE, mTitleText.getText().toString());
                bundle.putString(NotesDbAdapter.KEY_BODY, mBodyText.getText().toString());
                if (mRowId != null) {
                    bundle.putLong(NotesDbAdapter.KEY_ROWID, mRowId);
                }
                Intent mIntent = new Intent();
                mIntent.putExtras(bundle);
                setResult(RESULT_OK, mIntent);
                finish();
            }
        }
    );

    mRowId = null;
    Bundle extras = getIntent().getExtras();
    if (extras != null) {
        String title = extras.getString(NotesDbAdapter.KEY_TITLE);
        String body = extras.getString(NotesDbAdapter.KEY_BODY);
        mRowId = extras.getLong(NotesDbAdapter.KEY_ROWID);

        if (title != null) {
            mTitleText.setText(title);
        }
        if (body != null) {
            mBodyText.setText(body);
        }
    }
}

```

AndroidManifest.xml の編集

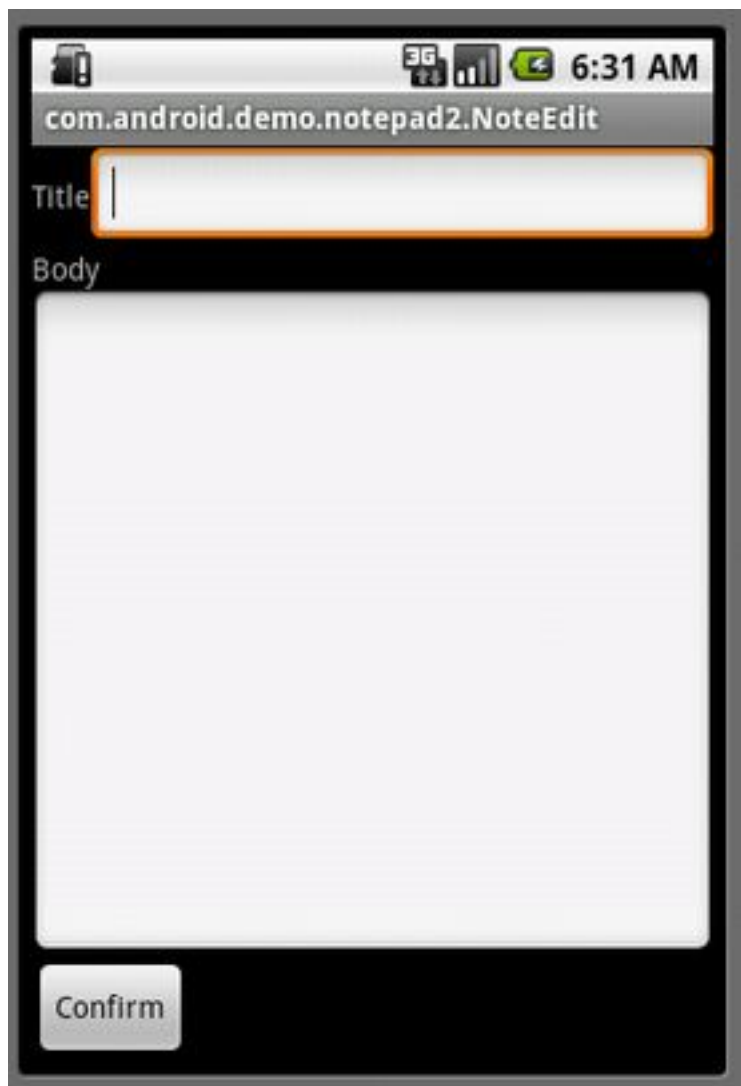
- <activity android:name=".NoteEdit"></activity> を追加

```

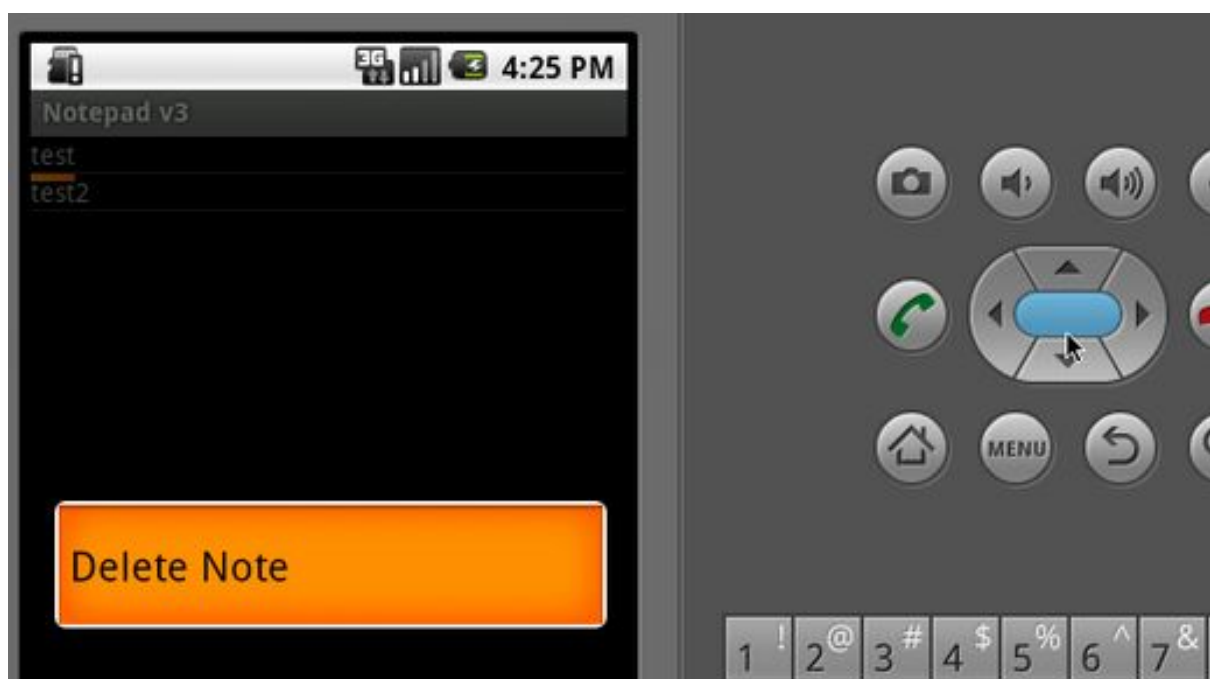
<activity android:name=".Notepadv2" android:label="@string/app_name">
    ...
</activity>
<activity android:name=".NoteEdit"></activity>

```

ここまでで起動



コンテキストメニューは、コントロールパッド長押し



(3)

- ・ <http://developer.android.com/resources/tutorials/notepad/notepad-ex3.html>

- ・ ライフサイクルイベントの利用
- ・ アプリケーション

プロジェクトの作成

- ・ 同様に、Notepadv3 からプロジェクトを作成。
- ・ 上記 Notepadv2 を完成させた状態と Notepadv3 は同じ

NoteEdit クラス

onCreate()

- ・ メンバ変数の宣言

```
private NotesDbAdapter mDbHelper;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    mDbHelper = new NotesDbAdapter(this);
    mDbHelper.open();

    setContentView(R.layout.note_edit);

    mTitleText = (EditText) findViewById(R.id.title);
    mBodyText = (EditText) findViewById(R.id.body);

    Button confirmButton = (Button) findViewById(R.id.confirm);

    mRowId = savedInstanceState != null ? savedInstanceState.getLong(NotesDbAdapter.KEY_ROWID)
                                         : null;

    if (mRowId == null) {
        Bundle extras = getIntent().getExtras();
        mRowId = extras != null ? extras.getLong(NotesDbAdapter.KEY_ROWID)
                                : null;
    }

    populateFields();
    confirmButton.setOnClickListener(new View.OnClickListener() {

        public void onClick(View view) {
            setResult(RESULT_OK);
            finish();
        }

    });
}
```

populateFields() を作成

```
private void populateFields() {
    if (mRowId != null) {
        Cursor note = mDbHelper.fetchNote(mRowId);
        startManagingCursor(note);
        mTitleText.setText(note.getString(note.getColumnIndexOrThrow(NotesDbAdapter.KEY_TITLE)));
        mBodyText.setText(note.getString(note.getColumnIndexOrThrow(NotesDbAdapter.KEY_BODY)));
    }
}
```

オーバーライドメソッドを実装

onSaveInstanceState()

```
@Override
protected void onSaveInstanceState(Bundle outState) {
    // TODO Auto-generated method stub
    super.onSaveInstanceState(outState);
    outState.putLong(NotesDbAdapter.KEY_ROWID, mRowId);
}
```

onPause()

- saveState() は後ほど作成

```
@Override
protected void onPause() {
    // TODO Auto-generated method stub
    super.onPause();
    saveState();
}
```

onResume()

```
@Override
protected void onResume() {
    // TODO Auto-generated method stub
    super.onResume();
    populateFields();
}
```

saveState() を追加

```
private void saveState() {
    String title = mTitleText.getText().toString();
    String body = mBodyText.getText().toString();

    if (mRowId == null) {
        long id = mDbHelper.createNote(title, body);
        if (id > 0) {
            mRowId = id;
        }
    } else {
        mDbHelper.updateNote(mRowId, title, body);
    }
}
```

Notepadv3

onActivityResult()

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
    super.onActivityResult(requestCode, resultCode, intent);
    fillData();
}
```

onListItemClick()

```
@Override
protected void onListItemClick(ListView l, View v, int position, long id) {
    super.onListItemClick(l, v, position, id);
}
```

```
        Intent i = new Intent(this, NoteEdit.class);  
        i.putExtra(NotesDbAdapter.KEY_ROWID, id);  
        startActivityForResult(i, ACTIVITY_EDIT);  
    }
```