헬로, 안드로이드

13주차 – SQL 활용하기 (2)

강대기
동서대학교 컴퓨터정보공학부

학습 목표

- 데이터 바인딩을 통해 데이터 소스에 해당하는 데이터 베이스와 뷰에 해당하는 액티비티를 연결한 데이터베이스 응용 프로그램을 작성할 수 있다.
- 안드로이드 내의 다른 어플리케이션의 데이터에 접근하기 위해 제공되는 ContentProvider를 사용할 수 있다.
- 자신의 어플리케이션에서 다른 어플리케이션으로의 데이터 제공을 위한 ContentProvider를 구현할 수 있다.

차례

- 데이터 바인딩
- ContentProvider 사용하기
- ContentProvider 구현하기
- 요약
- 퀴즈
- 연습문제

데이터 바인딩

- 몇 줄의 코드로 데이터(또는 모델)과 뷰를 연결
  - 모델은 상황에 따라 다르게 해석되나, (예를 들어 패션 모델, 모델 하우스, 성능 평가 모델, 기계 학습 모델 등) 데이터베이스에서는 데이터 모델, 즉 스키마와 비슷하게 해석됨
- 데이터 바인딩을 위해, Events 예제를 수정하여 데이터베이스 쿼리 결과에 엮인 ListView 를 사용하도록 함

Cursor-Adapter-Activity 간의 연결에 대한 개인적인 견해

- ASP.NET이나 C#.NET의 데이터베이스 응용을 보면, DataTable 방식과 DataSet 방식이 있는 데, 지난 시간의 예제 코드는 DataTable 방식에 가깝고, 이번 시간의 ContentProvider를 사용하는 예제 코드는 DataSet 방식에 가까움

- 데이터-컨트롤-뷰의 3-tier 구조와 비슷하게, 뷰에 해당하는 Activity에 대해 데이터를 의미하는 Cursor를 지정해 주는 이러한 방식은, 이미, 예를 들어, ASP.NET 등에서 활발히 사용되고 있음

- 본 안드로이드 예제에서 소개된 방식은 오히려 진정한 3-tier 분할이 안되고 있음. ASP.NET 의 방식이 더 편하고 진보된 방식임.
- ASP.NET에선 ASP를 나타내는 .aspx 파일(뷰)과 C#을 나타내는 .cs 파일(콘트롤)이 연결되어 있으며, 데이터 소스를 .aspx에서 직접 지정할 수 있음 – 차기 안드로이드 설계에 반영할 필요가 있음

- 또한, 차기 안드로이드 설계에서는, 이런 경우 Activity를 확장해서 ListActivity로 사용하는 것보다 View를 확장해서 ListView를 사용하게 하는 게 더 이치에 맞을 수도 있음. (그러나 ListView는 이미 가로 방향 스크롤되는 뷰로 사용되고 있는 문제가 있기는 함)
- 이에 따라, Java 소스에서 뿐만 아니라, 뷰를 의미하는 XML 파일에서 직접 데이터 소스를 지정할 수 있도록 하는 게 좋음

Events.showEvents()를 수정

- Events 는 Activity를 확장한 ListActivity 로 선언
- 데이터베이스 테이블에는, Cursor 클래스를 통해 레코드들에 접근함
- Cursor를 받아서, 사용자가 볼 수 있도록 출력함
- ListActivity에는 임의의 데이터 소스를 Adapter를 통해 연결해 줄 수 있음.
  - 이를 위해 setListAdapter() 메서드 사용
  - 여기서는 Adapter를 확장한 SimpleCursorAdapter 사용

SimpleCursorAdapter

- 어댑터는 뷰와 소스를 연결하는 중간 다리 역할
- 웹 서비스 이용하기의 Translate.setAdapters() 에서 이미 사용했었음 – 그 예제에서 데이터 소스는 XML 배열이므로 ArrayAdapter 를 사용했음
- 여기서는 데이터베이스 쿼리에서 나온 Cursor 객체이므로 SimpleCursorAdapter 사용
- 매개 변수
  - context – 현재 Activity의 참조
  - layout – 하나의 목록 아이템을 정의하는 리소스 (layout/item.xml)
  - cursor – 데이터 집합 커서
  - from – 데이터가 나오는 열 이름 목록
  - to – 데이터가 들어갈 뷰 목록

layout/item.xml

```
<RelativeLayout    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
  android:layout_height="fill_parent"    android:orientation="horizontal"
    android:padding="10sp">
  <TextView      android:id="@+id/rowid"      android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
  <TextView      android:id="@+id/rowidcolon"      android:layout_width="wrap_content"
    android:layout_height="wrap_content"      android:text=": "
    android:layout_toRightOf="@id/rowid" />
  <TextView      android:id="@+id/time"      android:layout_width="wrap_content"
    android:layout_height="wrap_content"      android:layout_toRightOf="@id/rowidcolon"
    />
  <TextView      android:id="@+id/timecolon"      android:layout_width="wrap_content"
    android:layout_height="wrap_content"      android:text=": "
    android:layout_toRightOf="@id/time" />
  <TextView      android:id="@+id/title"      android:layout_width="fill_parent"
    android:layout_height="wrap_content"      android:ellipsize="end"
    android:singleLine="true"      android:textStyle="italic"
    android:layout_toRightOf="@id/timecolon" />
</RelativeLayout>
```

layout/main.xml

- list 와 empty 라는 built-in ID 사용
- 목록에 아이템이 있다면, @android:id/list 뷰가 보이고, 아니면 @android:id/empty 뷰가 보임
- 데이터 바인딩을 사용하는 본 예제에서 이벤트 데이터베이스에 데이터를 추가하거나, 데이터베이스를 직접 보는 방법은? – ContentProvider 를 사용하는 것

layout/main.xml

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/
  res/android"   android:layout_width="fill_parent"
  android:layout_height="fill_parent">
<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/empty" />
</LinearLayout>
```

ContentProvider 사용하기

- 안드로이드 보안 모델로 인해, 원칙적으로 한 어플리케이션이 작성한 파일은 다른 어플리케이션에 의해 읽고 쓸 수 없음
- 각 프로그램은 개별 리눅스 사용자 ID와 데이터 디렉토리(/data/data/<package name>)와 보안된 메모리 공간을 가짐
- 안드로이드 프로그램들은 다음 두가지로 소통함
  - Inter-Process Communication – Android Interface Definition Language (AIDL) 와 IBinder 인터페이스
  - ContentProvider – 프로세스는 시스템에 자신이 어떤 종류 데이터의 제공자인지 등록함. 그 정보가 요청되면 안드로이드는 적합한 방식으로 컨텐츠를 쿼리 또는 수정하기 위해 고정 API를 통해 호출함

ContentProvider URI

- ContentProvider에 의해 관리되는 정보는 다음과 같은 URI를 통해 나타내짐
  - content://<authority>/<path>/<id>
  - content
  - authority
  - path
  - id
- 안드로이드는 다음과 같은 내장된 ContentProvider가 있음
  - content://browser
  - content://contacts
  - content://media
  - content://settings
- 본 예제의 Event 제공자로는 다음의 URI를 사용
  - content://org.example.events/events/3 – _id=3 인 단일 이벤트
  - content://org.example.events/events/ – 모든 이벤트

Constants.java

```
public interface Constants extends BaseColumns {
  public static final String TABLE_NAME =
  "events";

  public static final String AUTHORITY =
  "org.example.events";
  public static final Uri CONTENT_URI =
  Uri.parse("content://"
      + AUTHORITY + "/" + TABLE_NAME);


  // Columns in the Events database
  public static final String TIME = "time";
  public static final String TITLE = "title";
}
```

Event.onCreate()

- 추적할 데이터베이스 객체가 없어짐
- try catch 블록 필요 없으며, EventData 참조 없어짐

```
@Override
 public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.main);
   addEvent("Hello, Android!");
   Cursor cursor = getEvents();
   showEvents(cursor);
 }
```

행 추가하기 (addEvent())

- addEvent() 메서드를 통해 데이터베이스에 새로운 행을 추가함
- 시간은 현재 시간
- 제목은 주어진 string 변수
- CONTENT_URI 는 Constants.java 에 정의되어 있음

- getWritableDatabase() 제거됨
- insertOrThrow() 호출은 getContentResolver().insert()로 대체됨

쿼리 실행하기 (getEvents())

- Cursor를 채워주기 위한(populate) Activity.managedQuery() 실행으로 충분함

- 데이터베이스에 대한 모든 참조를 제거함으로써, Events 클라이언트를 Events 데이터 제공자로부터 분리함

Events.java

```
public class Events extends ListActivity {
  private static String[] FROM = { _ID, TIME, TITLE, };
  private static int[] TO = { R.id.rowid, R.id.time, R.id.title, };
  private static String ORDER_BY = TIME + " DESC";

 @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    addEvent("Hello, Android!");
    Cursor cursor = getEvents();
    showEvents(cursor);
  }

  private void addEvent(String string) {
    ContentValues values = new ContentValues();
    values.put(TIME, System.currentTimeMillis());
    values.put(TITLE, string);
    getContentResolver().insert(CONTENT_URI, values);
  }

  private Cursor getEvents() {
    return managedQuery(CONTENT_URI, FROM, null, null, ORDER_BY);
  }

  private void showEvents(Cursor cursor) {
    // Set up data binding
    SimpleCursorAdapter adapter = new SimpleCursorAdapter(this,  R.layout.item, cursor, FROM, TO);
    setListAdapter(adapter);
  }
}
```

ContentProvider 구현하기

- AndroidManifest.xml 에서 ContentProvider 추가
- <provider android:name="*EventsProvider*" android:authorities="*org.example.events*" />
  - android:name – 클래스 이름
  - android:authorities – 컨텐츠 URI에 사용되는 문자열
- ContentProvider를 확장하는 EventsProvider 클래스 생성 – 관습에 따라 MIME 타입에 org.example 대신 vnd.example 사용
- EventsProvider는 두 종류의 데이터를 다룸
  - CONTENT_TYPE – 이벤트들의 디렉토리의 MIME 타입
  - CONTENT_ITEM_TYPE – 단일 이벤트의 MIME 타입, ID 수준까지 상세히 지정할 수 있음

EventsProvider.java

```
public class EventsProvider extends ContentProvider {
  private static final int EVENTS = 1;
  private static final int EVENTS_ID = 2;

  /** The MIME type of a directory of events */
  private static final String CONTENT_TYPE
    = "vnd.android.cursor.dir/vnd.example.event";

  /** The MIME type of a single event */
  private static final String CONTENT_ITEM_TYPE
    = "vnd.android.cursor.item/vnd.example.event";

  private EventsData events;
  private UriMatcher uriMatcher;
  // ...
```

요약

- 데이터 바인딩을 통해 데이터 소스에 해당하는 데이터 베이스와 뷰에 해당하는 액티비티를 연결한 데이터베이스 응용 프로그램을 작성해 보았다.
- 안드로이드 내의 다른 어플리케이션의 데이터에 접근하기 위해 제공되는 ContentProvider를 사용하는 방법에 대해 알아 보았다.
- 자신의 어플리케이션에서 다른 어플리케이션으로의 데이터 제공을 위한 ContentProvider를 구현해 보았다.

퀴즈

- 어댑터란 무엇인가? 그리고 커서란 무엇인가?
- SimpleCursorAdapter는 무엇을 위한 클래스인가? 무엇을 위한 어댑터인가? XML 입력을 위해서는 어떤 어댑터를 사용해야 하는가?
- 안드로이드에서 다른 어플리케이션이 사용하는 파일에 접근하려면 어떻게 해야 하는가?
- ContentProvider에서 사용하는 URI는 어떤 구조인가?
- 안드로이드가 기본적으로 제공하는 ContentProvider 용 URI는 무엇이 있는가?
- ContentProvider 를 구현하기 위해 해야 할 일들은 무엇이 있는가?
- MIME는 무슨 뜻인가?

연습문제

- 본 장의 예제를 확장하여 이벤트 데이터 레코드에 새로운 컬럼을 추가하고, 사용자가 이벤트를 선택하면, 이벤트에 대한 자세한 정보를 보여주는 뷰어를 여는 프로그램을 작성하라.
- 사용자가 이벤트를 선택하면, 그 이벤트 정보를 특정 이메일로 보내는 프로그램을 작성하라.
- 각각의 이벤트에 대해 사용자가 별점을 줄 수 있는 프로그램일 작성하라.
- 사용자가 이벤트를 삭제할 수 있는 프로그램을 작성하라.
- 안드로이드에서 데이터를 저장하는 또 다른 방법인 db4o 에 대해 알아보자.