TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Persistieren mittels SQLite

Lukas Diem

WS 2014

Übersicht

Persistente Datenspeicherung

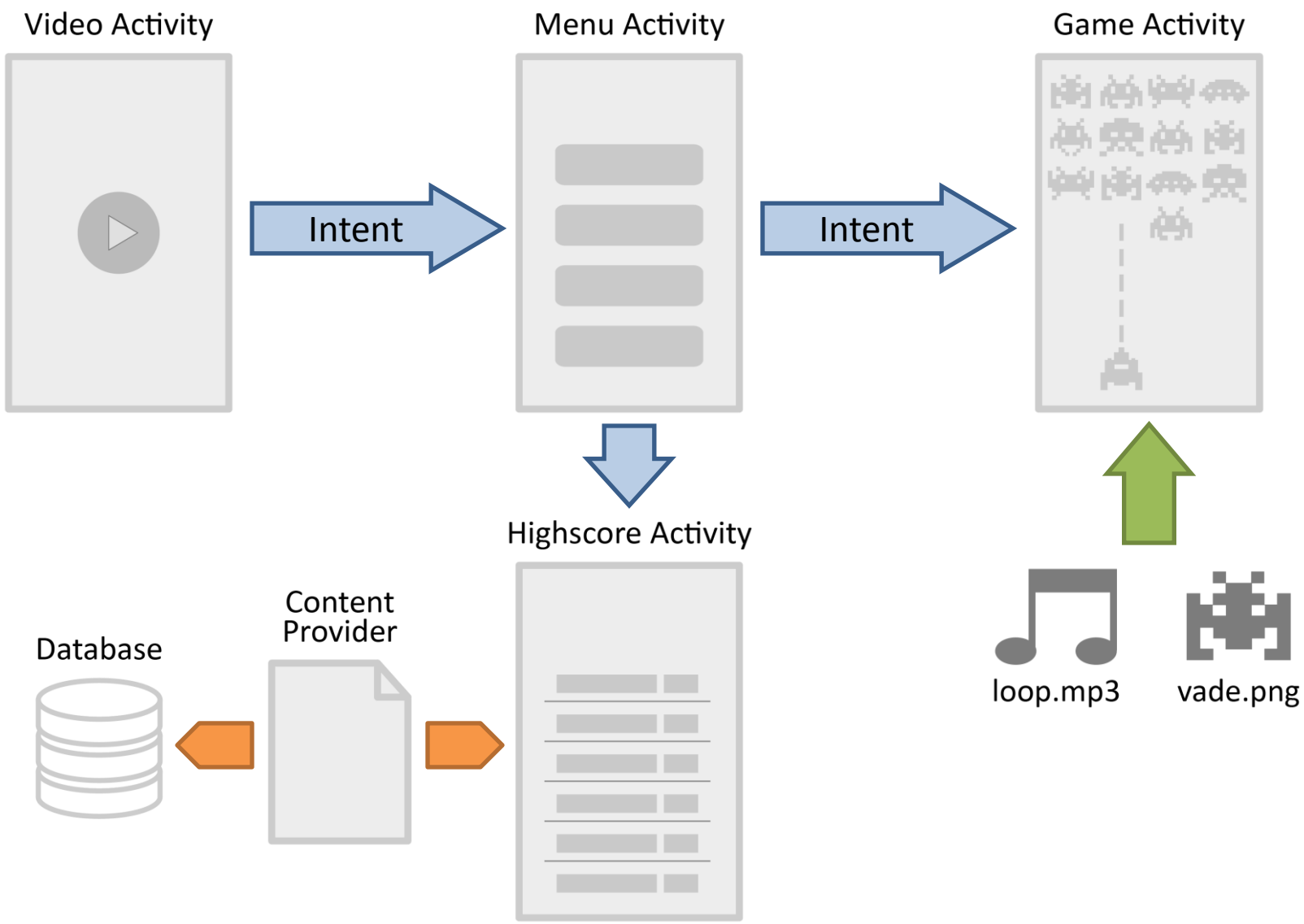
SQLite

Programmbeispiel

- OpenHelper
- ContentProvider
- LoaderManager
- Eintrag schreiben/auslesen

Android Support Library

Applikation – Übersicht



Übersicht

Persistente Datenspeicherung

SQLite

Programmbeispiel

- OpenHelper
- ContentProvider
- LoaderManager
- Eintrag schreiben/auslesen

Android Support Library

Persistente Datenspeicherung

= Permanentes Speichern von Daten

In der Übung:

- Highscore
- Spielstände

Möglichkeiten unter Android

Shared Preferences

- Key-Value Paare speichern

SQLite Datenbank

- Strukturierte Daten

Interner/Externer Speicher

- Intern → private Daten, Extern → öffentliche Daten

Netzwerkbasiert

- Daten auf eigenem Netzwerkserver

Übersicht

Persistente Datenspeicherung

SQLite

Programmbeispiel

- OpenHelper
- ContentProvider
- LoaderManager
- Eintrag schreiben/auslesen

Android Support Library

SQLite unter Android

Relationale Datenbank

Etwas reduzierte Befehlssatz im Vergleich zu SQL

Benötigt keinen Server

Zugriff ohne *ContentProvider* nur durch eigene App

Übersicht

Persistente Datenspeicherung

SQLite

Programmbeispiel

- OpenHelper
- ContentProvider
- LoaderManager
- Eintrag schreiben/auslesen

Android Support Library

Beispiel – Überblick

Kleine Highscoreliste

Features

- Hinzufügen von Highscores
- Anzeigen der Highscores
- Löschen von Highscores

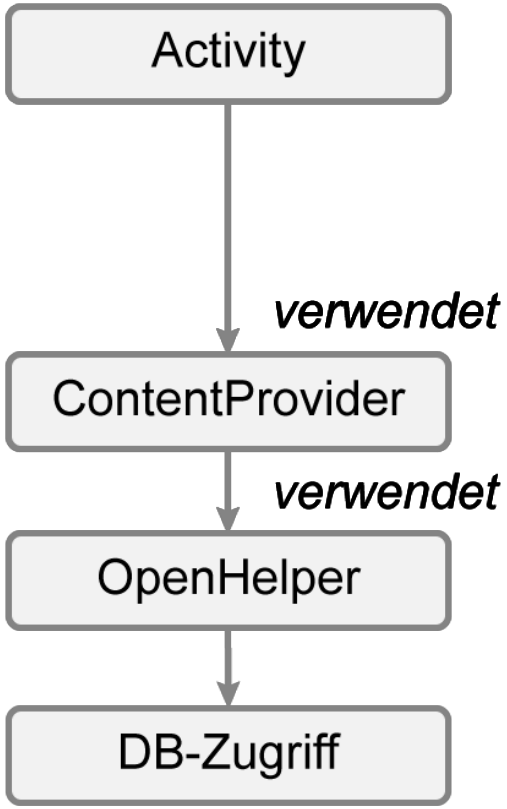
Beispiel – Datenbankstruktur

Minimalbeispiel besteht aus folgender Tabelle

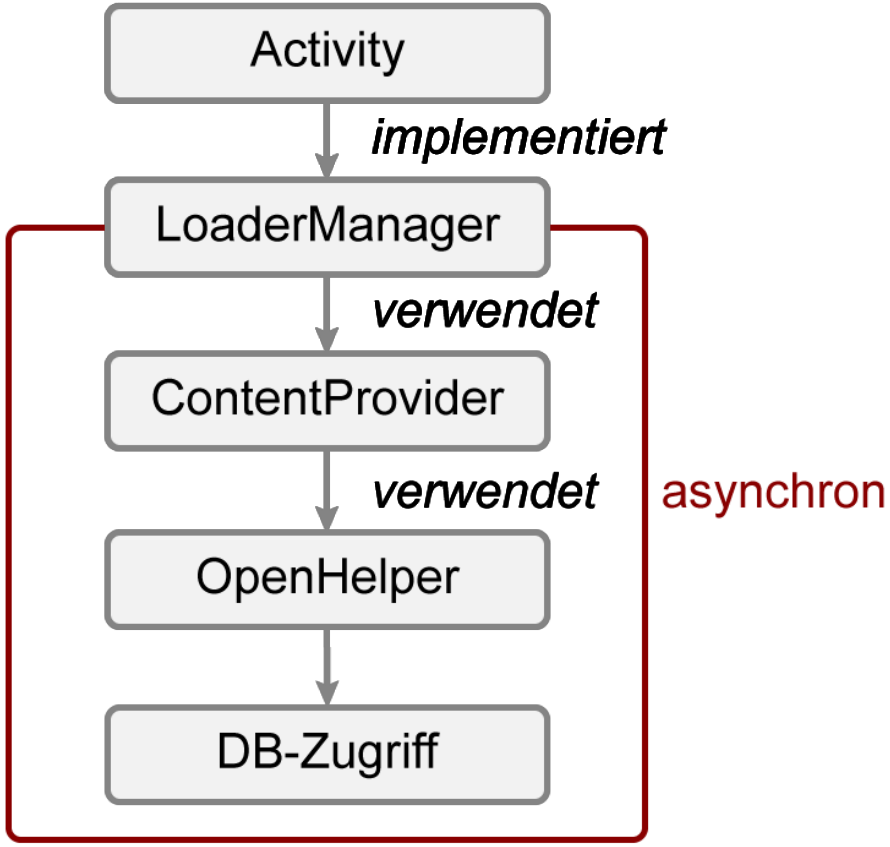
ID (key, auto increment)	Name	Score
1	Josef	10
2	Luise	8
3	Adam	42
…	…	…

Beispiel – Java Komponenten

Schreiben



Lesen



Übersicht

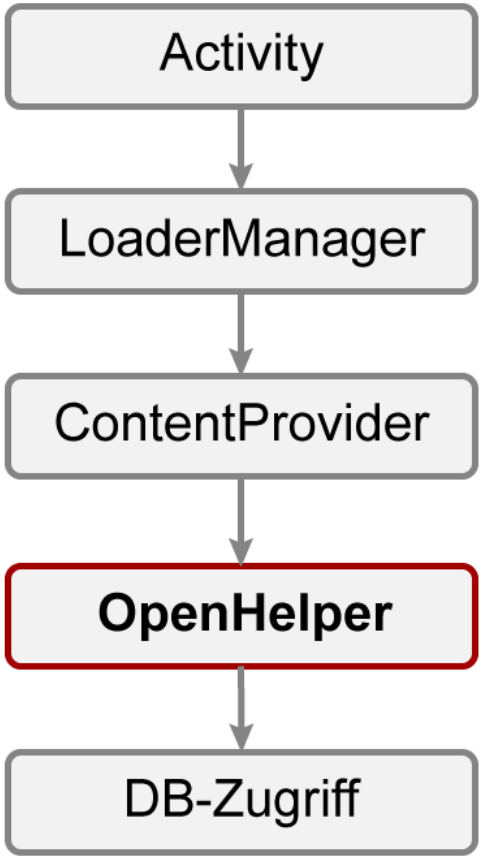
Persistente Datenspeicherung

SQLite

Programmbeispiel

- OpenHelper
- ContentProvider
- LoaderManager
- Eintrag schreiben/auslesen

Android Support Library



Beispiel – HighscoreOpenHelper

- Erzeugen der Tabellen
- Updaten der Tabellen
- Zugriff auf die Datenbank

Empfehlung: Eine Klasse pro Tabelle und diese dann im `OpenHelper` zusammenfassen

Damit das Zusammenspiel zwischen `Cursor`, `ListView` und `ContentProvider` funktioniert **muss** jede Tabelle die Spalte **_ID** besitzen

Beispiel – HighscoreTable (1/2)

```
public class HighscoreTable {
    public static final String HIGHSCORE_TABLE_NAME = "highscore";

    public static final String COLUMN_ID = "_id";
    public static final String COLUMN_NAME = "name";
    public static final String COLUMN_SCORE = "score";

    // Define the create command
    private static final String HIGHSCORE_TABLE_CREATE =
        "CREATE TABLE " + HIGHSCORE_TABLE_NAME + " (" +
         COLUMN_ID + " integer primary key autoincrement, " +
         COLUMN_NAME + " TEXT not null, " +
         COLUMN_SCORE + " integer);";

    ...
```

CREATE TABLE **highscore** (
_id INTEGER PRIMARY KEY AUTOINCREMENT,
name TEXT NOT NULL,
score INTEGER);

Beispiel – HighscoreTable (2/2)

```
...

public static void onCreate(SQLiteDatabase db) {
    // Create the highscore table
    db.execSQL(HIGHSCORE_TABLE_CREATE);
}

public static void onUpgrade(SQLiteDatabase db,
                             int oldVersion, int newVersion) {

    // Delete the Table if it already exists
    db.execSQL("DROP TABLE IF EXISTS " + HIGHSCORE_TABLE_NAME);
    // Create the new table
    onCreate(db);
}
}
```

Beispiel - HighscoreOpenHelper

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

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

    public void onCreate(SQLiteDatabase db) {
        HighscoreTable.onCreate(db);
    }

    public void onUpgrade(SQLiteDatabase db,
                          int oldVersion, int newVersion) {
        HighscoreTable.onUpgrade(db, oldVersion, newVersion);
    }
}
```

Übersicht

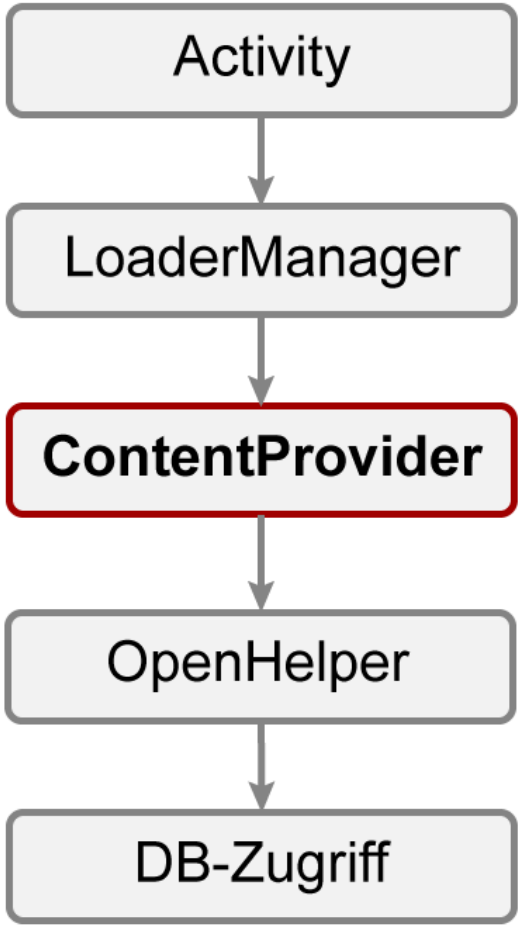
Persistente Datenspeicherung

SQLite

Programmbeispiel

- OpenHelper
- ContentProvider
- LoaderManager
- Eintrag schreiben/auslesen

Android Support Library



Beispiel – ContentProvider

Trennt Applikationscode von Datenspeicherung

Zugriff abstrahiert durch URIs

Ermöglicht Zugriff auf Daten

- anderer Apps
- für andere Apps

Eintrag in *AndroidManifest.xml* nicht vergessen!

Beispiel – Aufbau der URI

content://AUTHORITY/BASE_PATH

- AUTHORITY … Package Name + Provider Name
- BASE_PATH … Tabellenname
- z.B.: content://at.ac.tuwien.ims.hs.provider/highscore

UriMatcher – Uri hinzufügen

- `myUriMatcher.add(AUTHORITY, BASE_PATH, CODE);`

Beispiel – URI Matching

content://ims.provider/level/2/highscore

Uri	Codename	Code
content://ims.provider/level	LEVEL	1
content://ims.provider/level/#	LEVEL_ID	2
content://ims.provider/level/#/highscore	LEVEL_HS	3
content://ims.provider/level/#/highscore/#	LEVEL_HS_ID	4
content://ims.provider/savegame	SAVEGAME	5
content://ims.provider/savegame/#	SAVEGAME_ID	6

ContentProvider/SQL Query

Query() argument	SQL Keyword, Parameter	Anmerkung
Uri	FROM table_name	Tabelle auswählen
projection	col, col, col, …	Spalten auswählen, die im Ergebnis sind
selection	WHERE col = value	Datenbankeinträge auf bestimmtes Kriterium prüfen
selectionArgs	-	Keine direkte Übereinstimmung
sortOrder	ORDER BY col, col, …	Bestimmt die Sortierung der Zeilen im retournierten Cursor

Beispiel – HighscoreContentProvider

```
public class HighscoreContentProvider extends ContentProvider {
    // Database helper
    HighscoreOpenHelper dbHelper;

    // TODO: Initialize URI Constants and URI Matcher

    @Override
    public boolean onCreate() {
        dbHelper = new HighscoreOpenHelper(getContext());
        return true;
    }
...
```

Beispiel – HighscoreContentProvider

```
public Cursor query(Uri uri, String[] projection,
    String selection, String[] selectionArgs, String sortOrder) {

    checkColumns(projection);
    SQLiteQueryBuilder queryBuilder = new SQLiteQueryBuilder();
    queryBuilder.setTables(HighscoreTable.HIGHSCORE_TABLE_NAME);

    switch (sUriMatcher.match(uri)) {
        case HIGHSCORE:
            break;
        case HIGHSCORE_ID:
            queryBuilder.appendWhere(HighscoreTable.COLUMN_ID +
                "=" + uri.getLastPathSegment());
            break;
        default: // throw error
    }
...
```

Beispiel – HighscoreContentProvider

```
...
    // open up the connection to the db
    SQLiteDatabase db = dbHelper.getWritableDatabase();

    Cursor cursor = queryBuilder.query(
                    db, projection, selection, selectionArgs,
                    null, null, sortOrder);

    // make sure that potential listeners are getting notified
    cursor.setNotificationUri(
         getContext().getContentResolver(), uri);

    return cursor;
}
...
```

Beispiel – HighscoreContentProvider

```
...
    public int delete(Uri uri, String selection,
        String[] selectionArgs) { }

    public Uri insert(Uri uri, ContentValues values) { }

    public int update(Uri uri, ContentValues values,
        String selection, String[] selectionArgs) { }

    public String getType(Uri uri) { // TODO }

    private void checkColumns(String[] projection) { // TODO }
}
```

Beispiel – HighscoreContentProvider

Eintrag im *AdroidManifest.xml*

```
<provider
    android:name=".sql.HighscoreContentProvider"
    android:authorities=
         "at.ac.tuwien.ims.highscoresqlite.highscoreprovider"
    android:exported="false" >
</provider>
```

Privater Content Provider
(andere Apps können diesen nicht verwenden!)

Übersicht

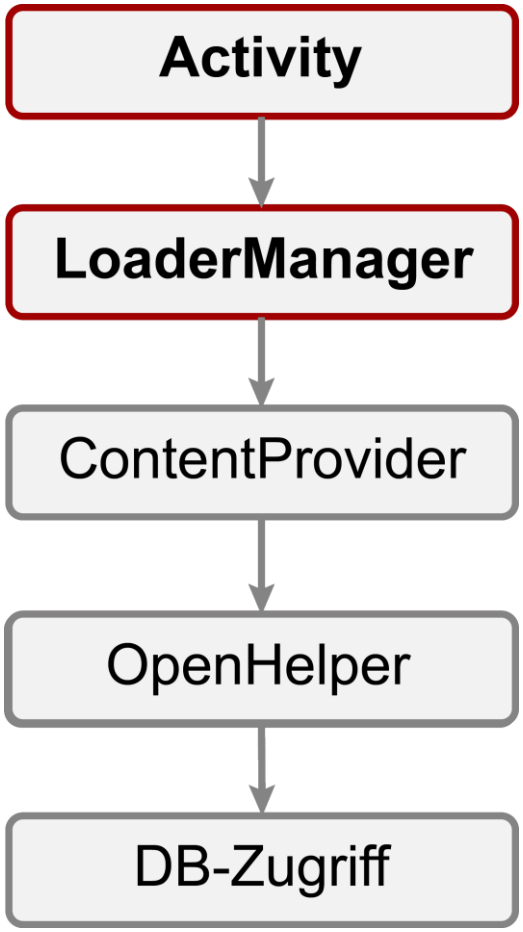
Persistente Datenspeicherung

SQLite

Programmbeispiel

- OpenHelper
- ContentProvider
- LoaderManager
- Eintrag schreiben/auslesen

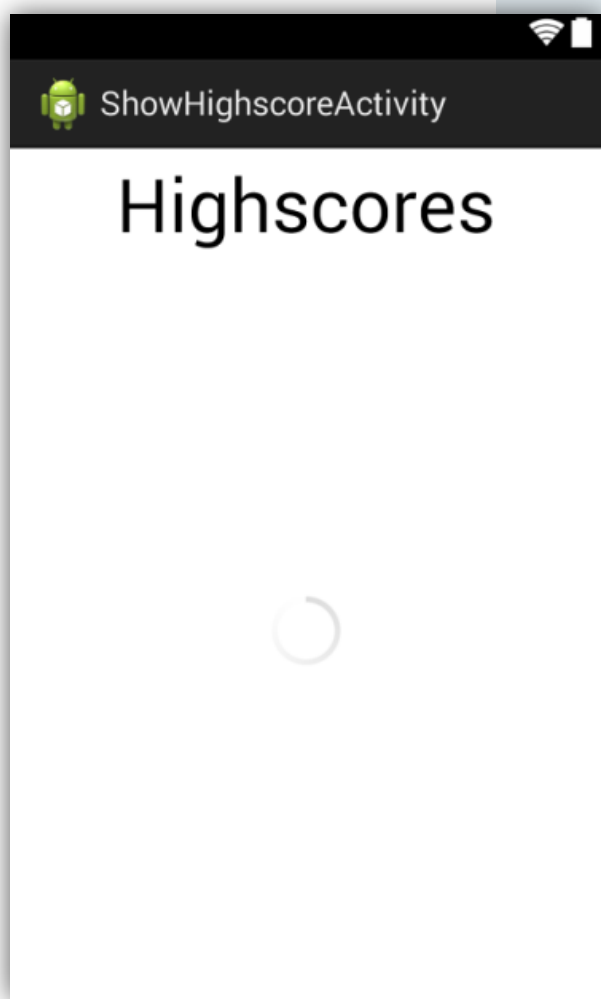
Android Support Library



activity_show_highscore.xml

```
<LinearLayout ...>
    <TextView
        android:id="@+id/textShowHighscoreCaption"
        android:text="@string/lblHighscores"
        ... />
    <LinearLayout
        android:id="@+id/pbHighscore"
        ... >
        <ProgressBar
            ... />
    </LinearLayout>
    <!-- Show if list is empty -->
    <TextView
        android:id="@+id/emptyHighscoreList"
        android:text="@string/lblHighscoreEmpty"
        android:visibility="gone"
        ... >
    </TextView>

    <ListView
        android:id="@+id/listViewHighscore"
        ... />
</LinearLayout>
```

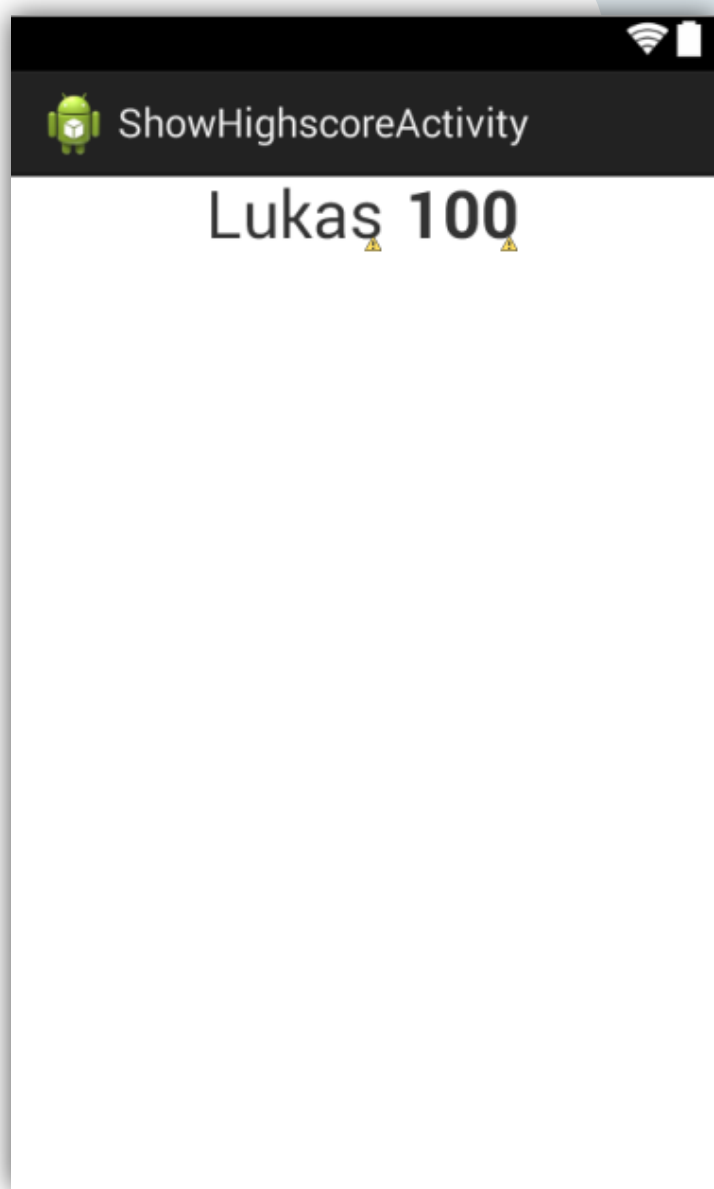


highscore_entry_layout.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="..."
    ... >

    <TextView
        android:id="@+id/lblName"
        ... >
    </TextView>

    <TextView
        android:id="@+id/lblScore"
        ... >
    </TextView>
</LinearLayout>
```



Beispiel – Highscores auslesen

```
public class ShowHighscoreActivity extends ListActivity
    implements LoaderManager.LoaderCallbacks<Cursor> {

    private SimpleCursorAdapter dbAdapter;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_show_highscore);

        // fetch data from the db and show it in the list
        showData();
    }
...
```

Beispiel – Highscores auslesen

```
...
    public void showData() {
        // Initialize to load the data in a background task
        getLoaderManager().initLoader(0, null, this);

        String[] from = new String[] {
            HighscoreTable.COLUMN_NAME,
            HighscoreTable.COLUMN_SCORE };

        int[] to = new int[] { R.id.lblName, R.id.lblScore };

        dbAdapter = new SimpleCursorAdapter(this,
            R.layout.highscore_entry_layout, null, from, to, 0);

        setListAdapter(dbAdapter);
    }
...
```

Beispiel – Highscores auslesen

```
...
    @Override
    public Loader<Cursor> onCreateLoader(int id, Bundle args) {
        // Set the columns to load
        String[] projection = { HighscoreTable.COLUMN_ID,
            HighscoreTable.COLUMN_NAME,
            HighscoreTable.COLUMN_SCORE };

        CursorLoader cursorLoader = new CursorLoader(this,
            HighscoreContentProvider.CONTENT_URI,
            projection, null, null,
            HighscoreTable.COLUMN_SCORE + " DESC");

        return cursorLoader;
    }
...
```

Beispiel – Highscores auslesen

```
...
    @Override
    public void onLoadFinished(Loader<Cursor> loader,
        Cursor data) {
        // Show the data from the loader
        dbAdapter.swapCursor(data);
    }

    @Override
    public void onLoaderReset(Loader<Cursor> loader) {
        // data is not available anymore, delete reference
        dbAdapter.swapCursor(null);
    }
}
```

Übersicht

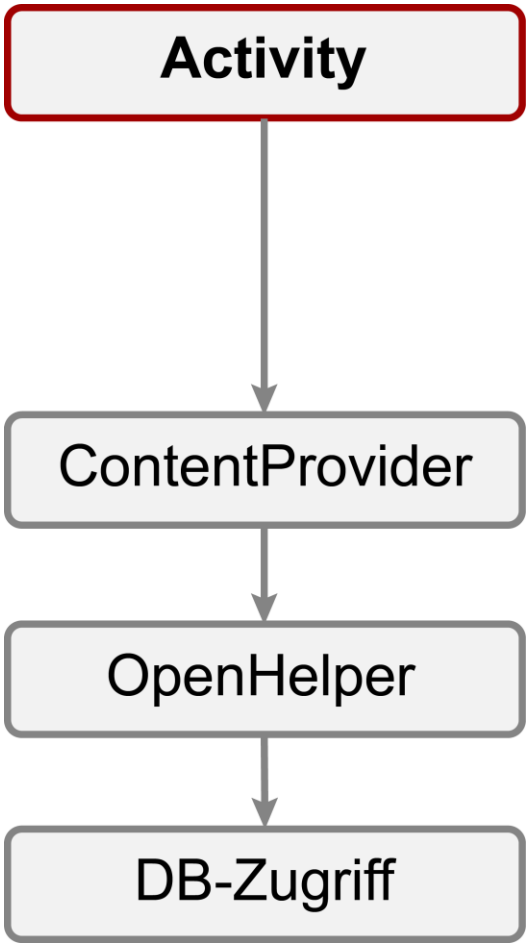
Persistente Datenspeicherung

SQLite

Programmbeispiel

- OpenHelper
- ContentProvider
- LoaderManager
- Eintrag schreiben/auslesen

Android Support Library



Beispiel – Highscore eintragen

```
...

// Generate values readable by the content provider
ContentValues values = new ContentValues();
values.put(HighscoreTable.COLUMN_NAME, name);
values.put(HighscoreTable.COLUMN_SCORE, score);

// save the new score in the database
getContentResolver().insert(
    HighscoreContentProvider.CONTENT_URI, values);

...
```

Übersicht

Persistente Datenspeicherung

SQLite

Programmbeispiel

- OpenHelper
- ContentProvider
- LoaderManager
- Eintrag schreiben/auslesen

Android Support Library

Android Support Library (ASL)

Problem: LoaderManager und CursorLoader erst ab Android **API 11** (Honeycomb)

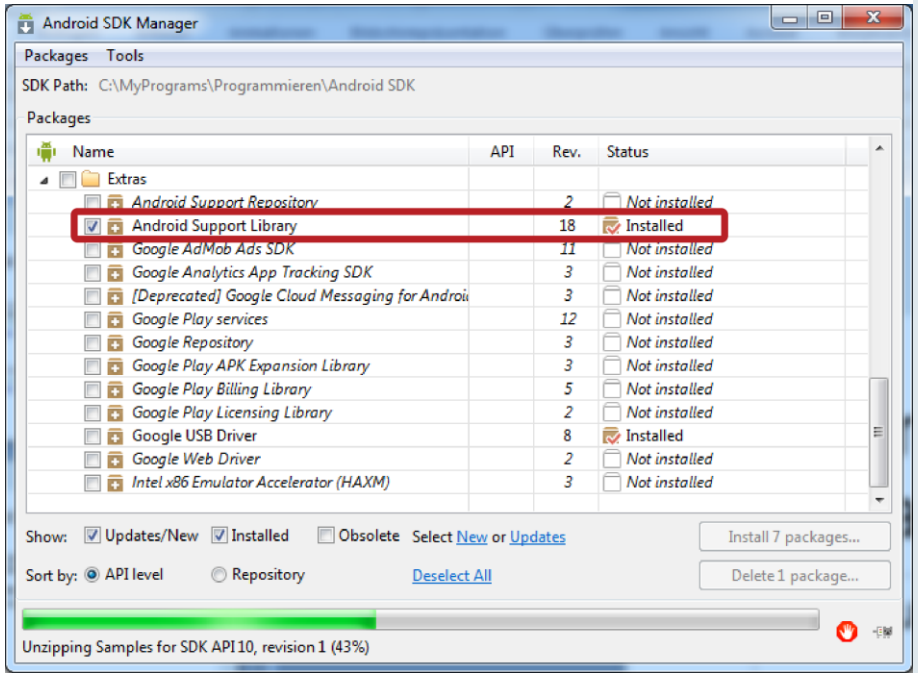
Lösung: Android Support Library (v4)

- Stellt neue Methoden für ältere Devices zur Verfügung

http://developer.android.com/tools/support-library/index.html

ASL – Installation

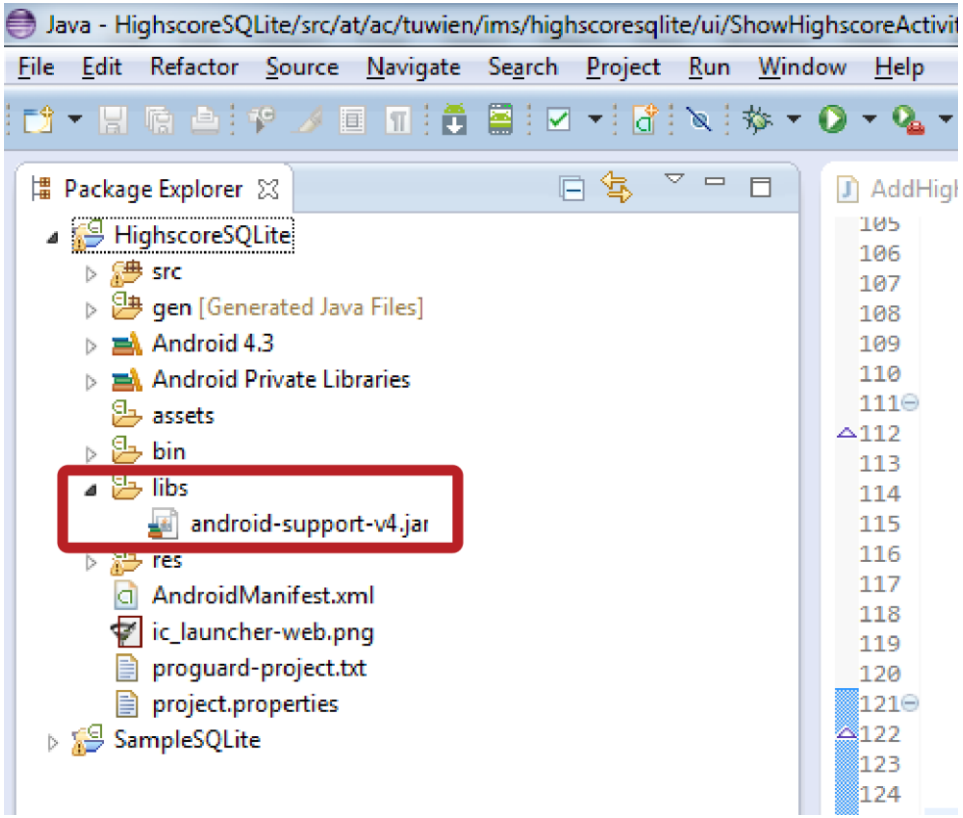
1. Android SDK Manager starten
2. Extras → Android Support Library
3. Install



ASL – Einbinden

1. **`libs/`** Ordner im Projekt erstellen
2. JAR-Datei vom SDK Ordner nach **`libs/`** kopieren
   - z.B.: **`<sdk>/extras/android/support/v4/android-support-v4.jar`** nach **`<your project>/libs`**
3. Rechtsklick auf die JAR-Datei → **Build Path → Add to Build Path**

ASL – Einbinden



ASL – Änderungen im Projekt

Folgende **imports** anpassen:

```
import android.app.ListActivity;
import android.app.LoaderManager;
import android.content.CursorLoader;
import android.content.Loader;
import android.widget.SimpleCursorAdapter;
```

```
import android.support.v4.app.FragmentActivity;
import android.support.v4.app.LoaderManager;
import android.support.v4.content.CursorLoader;
import android.support.v4.content.Loader;
import android.support.v4.widget.SimpleCursorAdapter;
```

ASL – Änderungen im Projekt

1. Ersetze **`extend ListActivity`** durch
   `extend FragmentActivity`

2. Ersetze **`listView = getListView();`** durch **`listView = (ListView) findViewById(R.id.<listViewId>);`**

3. Falls der **`onListItemClick()`** Listener überschrieben wurde muss dieser durch **`listView.setOnItemClick(this)`** in **`onCreate()`** eingebunden werden

4. Die empty View muss im **`onCreate()`** gesetzt werden:
   `listView.setEmptyView(findViewById(R.id.<emptyId>))`

Weiterführende Links

SQLite, ContentProvider und Loader:

- http://www.vogella.com/articles/AndroidSQLite/article.html

Content Provider:

- http://developer.android.com/guide/topics/providers/content-provider-basics.html

Loaders & LoaderManager:

- http://www.androiddesignpatterns.com/2012/07/loaders-and-loadermanager-background.html

Weiterführende Links

ListView mit eigenem Adapter:

- http://www.vogella.com/articles/AndroidListView/article.html

Mehrere Tabellen:

- http://www.androidhive.info/2013/09/android-sqlite-database-with-multiple-tables/

Android Developers – Android Design in Action:

- http://www.youtube.com/user/androiddevelopers

Danke für eure Aufmerksamkeit!

Grafik: http://www.nila.at/internet/mach-das-internet-zu-einem-schlauen-ort