МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федерально автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

кафедра Информационных систем

Куркчи Ариф Эрнестович

Институт информационных технологий и управления в технических системах

курс 3 группа ИС/б-31-о

09.03.02 Информационные системы и технологии (уровень бакалавриата)

ОТЧЕТ

по лабораторной работе №6

по дисциплине «Мобильные ИТ»

на тему «Взаимодействие с сервером»

Отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

cт.преподаватель Шишкевич В.Е.

(должность) (подпись) (инициалы, фамилия)

Севастополь 2017

1. ЦЕЛЬ РАБОТЫ

Изучить работу с потоками. Научиться работать с мультимедиа файлами. Изучить работу с классом AsyncTask.

1. ИСХОДНЫЙ КОД

MainActivity.java

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1 | package ru.justnero.sevsu.s3.mit.e6; | | 2 |  | | 3 | import *android.content.ContentValues*; | | 4 | import *android.database.Cursor*; | | 5 | import *android.database.sqlite.SQLiteDatabase*; | | 6 | import *android.media.AudioManager*; | | 7 | import *android.media.MediaPlayer*; | | 8 | import *android.os.AsyncTask*; | | 9 | import *android.os.Bundle*; | | 10 | import *android.support.v7.app.AppCompatActivity*; | | 11 | import *android.util.Log*; | | 12 | import *android.util.Pair*; | | 13 | import *android.view.View*; | | 14 | import *android.widget.ArrayAdapter*; | | 15 | import *android.widget.Button*; | | 16 | import *android.widget.ListView*; | | 17 |  | | 18 | import *org.json.JSONException*; | | 19 | import *org.json.JSONObject*; | | 20 |  | | 21 | import *java.io.IOException*; | | 22 | import *java.util.ArrayList*; | | 23 | import *java.util.Collections*; | | 24 |  | | 25 | import *okhttp3.OkHttpClient*; | | 26 | import *okhttp3.Request*; | | 27 | import *okhttp3.Response*; | | 28 |  | | 29 | public class MainActivity extends *AppCompatActivity* implements *View*.*OnClickListener*, *MediaPlayer*.*OnPreparedListener*, | | 30 | *MediaPlayer*.*OnCompletionListener* { | | 31 | private *MainActivity* activity; | | 32 | private *OkHttpClient* client = new *OkHttpClient*(); | | 33 | final *String* musicSource = "http://icecast.radiomaximum.cdnvideo.ru/maximum.mp3"; | | 34 | final *String* dataSource = "http://maximum.ru/currenttrack.aspx?station=maximum"; | | 35 | *MediaPlayer* mediaPlayer; | | 36 | *AudioManager* am; | | 37 | *Button* btnPlay, btnInfo; | | 38 | *ListView* listView; | | 39 |  | | 40 | *@Override* | | 41 | protected *void* onCreate(*Bundle* *savedInstanceState*) { | | 42 | super.onCreate(savedInstanceState); | | 43 | activity = this; | | 44 | setContentView(*R*.layout.activity\_main); | | 45 | am = (*AudioManager*) getSystemService(AUDIO\_SERVICE); | | 46 | btnPlay = (*Button*) findViewById(*R*.id.btnStopResume); | | 47 | btnInfo = (*Button*) findViewById(*R*.id.btnInfo); | | 48 | btnPlay.setOnClickListener(this); | | 49 | btnInfo.setOnClickListener(this); | | 50 | listView = (*ListView*) findViewById(*R*.id.listView); | | 51 | refillList(); | | 52 | } | | 53 |  | | 54 | private *void* refillList() { | | 55 | *DataBase* db = new *DataBase*(this); | | 56 | *SQLiteDatabase* table = db.getWritableDatabase(); | | 57 | *Cursor* cursor = table.query(*DataBase*.TABLE, null, null, null, null, null, null); | | 58 | *ArrayList<String>* arrayList = new *ArrayList<>*(); | | 59 | *String* song; | | 60 | *String* artist; | | 61 | if (cursor.moveToFirst()) { | | 62 | do { | | 63 | song = cursor.getString(cursor.getColumnIndex(*DataBase*.COLUMN\_SONG)); | | 64 | artist = cursor.getString(cursor.getColumnIndex(*DataBase*.COLUMN\_ARTIST)); | | 65 | arrayList.add(artist.isEmpty() ? song : artist + " - " + song); | | 66 | } while (cursor.moveToNext()); | | 67 | } | | 68 | cursor.close(); | | 69 | *Collections*.reverse(arrayList); | | 70 | *ArrayAdapter<String>* adapter = new *ArrayAdapter<>*(this, *android.R*.layout.simple\_list\_item\_1, arrayList.toArray(new *String*[arrayList.size()])); | | 71 | listView.setAdapter(adapter); | | 72 | } | | 73 |  | | 74 | private *void* releaseMP() { | | 75 | if (mediaPlayer != null) { | | 76 | try { | | 77 | mediaPlayer.stop(); | | 78 | mediaPlayer.release(); | | 79 | mediaPlayer = null; | | 80 | } catch (*Exception* e) { | | 81 | e.printStackTrace(); | | 82 | } | | 83 | } | | 84 | } | | 85 |  | | 86 | *@Override* | | 87 | public *void* onClick(*View* *v*) { | | 88 | switch (v.getId()) { | | 89 | case *R*.id.btnStopResume: | | 90 | try { | | 91 | if (mediaPlayer == null) { | | 92 | mediaPlayer = new *MediaPlayer*(); | | 93 | mediaPlayer.setDataSource(musicSource); | | 94 | mediaPlayer.setAudioStreamType(*AudioManager*.STREAM\_MUSIC); | | 95 | mediaPlayer.setOnPreparedListener(this); | | 96 | mediaPlayer.prepareAsync(); | | 97 | } else { | | 98 | releaseMP(); | | 99 | } | | 100 | } catch (*Exception* e) { | | 101 | e.printStackTrace(); | | 102 | } | | 103 | break; | | 104 | case *R*.id.btnInfo: | | 105 | new *RetrieveSongInfoTask*().execute(); | | 106 | break; | | 107 | } | | 108 |  | | 109 | } | | 110 |  | | 111 | *@Override* | | 112 | public *void* onCompletion(*MediaPlayer* *mp*) {} | | 113 |  | | 114 | *@Override* | | 115 | public *void* onPrepared(*MediaPlayer* *mp*) { | | 116 | mp.start(); | | 117 | } | | 118 |  | | 119 | *@Override* | | 120 | protected *void* onDestroy() { | | 121 | super.onDestroy(); | | 122 | releaseMP(); | | 123 | } | | 124 |  | | 125 | private class RetrieveSongInfoTask extends *AsyncTask<Void, Void, Pair<String, String>>* { | | 126 | private *Exception* exception; | | 127 |  | | 128 | protected *Pair<String, String>* doInBackground(*Void*... *urls*) { | | 129 | *String* songArtist = "Unknown"; | | 130 | *String* songTitle = "Unknown"; | | 131 | try { | | 132 | *Request* request = new *Request*.*Builder*() | | 133 | .url(activity.dataSource) | | 134 | .build(); | | 135 | *Response* response = client.newCall(request).execute(); | | 136 | *String* jsonData = response.body().string(); | | 137 | *JSONObject* songInfo = new *JSONObject*(jsonData).getJSONObject("Current"); | | 138 | songArtist = songInfo.getString("Artist"); | | 139 | songTitle = songInfo.getString("Song"); | | 140 | } catch (*IOException* | *JSONException* e) { | | 141 | e.printStackTrace(); | | 142 | exception = e; | | 143 | } | | 144 | return new *Pair<>*(songArtist, songTitle); | | 145 | } | | 146 |  | | 147 | protected *void* onPostExecute(*Pair<String, String>* *data*) { | | 148 | *DataBase* dataBase = new *DataBase*(activity); | | 149 | *SQLiteDatabase* table = dataBase.getWritableDatabase(); | | 150 | *ContentValues* cv = new *ContentValues*(); | | 151 | cv.put(*DataBase*.COLUMN\_ARTIST, data.first); | | 152 | cv.put(*DataBase*.COLUMN\_SONG, data.second); | | 153 | table.insert(*DataBase*.TABLE, null, cv); | | 154 | activity.refillList(); | | 155 | } | | 156 | } | | 157 | } | |  |

Database.java

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 1 | package ru.justnero.sevsu.s3.mit.e6; | | 2 |  | | 3 | import *android.content.Context*; | | 4 | import *android.database.sqlite.SQLiteDatabase*; | | 5 | import *android.database.sqlite.SQLiteOpenHelper*; | | 6 |  | | 7 | class DataBase extends *SQLiteOpenHelper* { | | 8 | final static *String* DATABASE = "lab06"; | | 9 | final static *String* TABLE = "music\_history"; | | 10 | final static *String* COLUMN\_ID = "id"; | | 11 | final static *String* COLUMN\_ARTIST = "author"; | | 12 | final static *String* COLUMN\_SONG = "song"; | | 13 | final static *String* COLUMN\_TIME = "time"; | | 14 |  | | 15 | DataBase(*Context* *context*) { | | 16 | super(context, DATABASE, null, 1); | | 17 | } | | 18 |  | | 19 | *@Override* | | 20 | public *void* onCreate(*SQLiteDatabase* *db*) { | | 21 | db.execSQL("CREATE TABLE " + TABLE + "(" + | | 22 | COLUMN\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " + | | 23 | COLUMN\_ARTIST + " TEXT, " + | | 24 | COLUMN\_SONG + " TEXT, " + | | 25 | COLUMN\_TIME + " TEXT " + ");"); | | 26 | } | | 27 |  | | 28 | *@Override* | | 29 | public *void* onUpgrade(*SQLiteDatabase* *db*, *int* *oldVersion*, *int* *newVersion*) {} | | 30 | } | |  |

ВЫВОДЫ

В ходе лабораторной работы были изучены принципы работы с потоками, изучен класс AsyncTask, создано приложение, взаимодействующее с мультимедиа файлами и потоками.