

Session 5 - Exercises

v1.0

JUAN RONDON

2016

Table of Contents

[Android Note taking application 2](#_Toc447691229)

[Database Implementation (ORM with Realm) 2](#_Toc447691230)

# Android Note taking application

## Database Implementation (ORM with Realm)

In this lab, you will be adding the code for deleting a note and the use of a persistent database using REALM ORM so every time the application is closed your existing notes won’t be destroyed.

1. Install Realm into your android app.

#### Installation

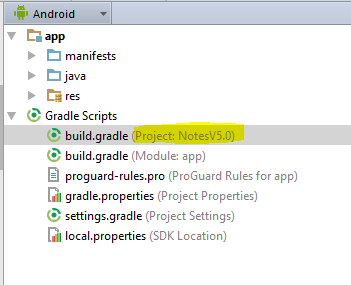
Realm is installed as a Gradle plugin.

Installing Realm as a Gradle plugin is a two-step process.

1. Add the following class path dependency to the **project level build.gradle file**.



The project level build.gradle file is located here:

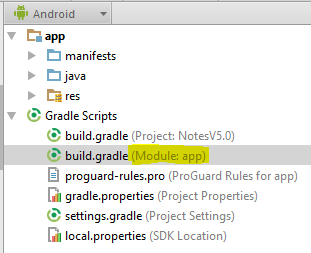




1. Apply the realm-android plugin to the **top of application level build.gradle file**.



The application level build.gradle is located here:





Once these two changes are made, simply refresh your gradle dependencies.

Continues in next page

1. Open **Note.java** file and make it a subclass of **RealmObject** class so a database table will be automatically created for our notes. Next you will be adding some annotations (@PrimaryKey, @Required) to some of the properties of the class. And finally you will be removing the static property that was used to create the id for the note.

**package** androidcourse.notes.Models;  
**import** java.text.SimpleDateFormat;  
**import** java.util.Date;  
**import** io.realm.RealmObject;  
**import** io.realm.annotations.PrimaryKey;  
**import** io.realm.annotations.Required;  
  
**public class** Note **extends** RealmObject {  
 @PrimaryKey  
 **private int mId**;  
 @Required  
 **private** String **mTitle**;  
 @Required  
 **private** String **mContent**;  
 @Required  
 **private** Date **mLastModified**;

**private** String **mPassword**;  
   
 **public** Note(String title, String content) {  
 **mTitle** = title;  
 **mContent** = content;  
 **mLastModified** = **new** Date();  
 }  
 **public** Note(String title, String content, String password) {  
 **this**(title, content);  
 **mPassword** = password;  
 **mLastModified** = **new** Date();  
 }  
 **public int** getmId() {  
 **return mId**;  
 }  
 **public void** setmId(**int** mId) {  
 **this**.**mId** = mId;  
 }  
 **public** String getmTitle() {  
 **return mTitle**;  
 }  
 **public** String getmContent() {  
 **return mContent**;  
 }  
 **public** Date getmLastModified() {  
 **return mLastModified**;  
 }  
 **public** String getmPassword() {  
 **return mPassword**;  
 }  
 **public void** setmTitle(String mTitle) {  
 **this**.**mTitle** = mTitle;  
 }  
 **public void** setmContent(String mContent) {  
 **this**.**mContent** = mContent;  
 }  
 **public void** setmLastModified(Date mLastModified) {  
 **this**.**mLastModified** = mLastModified;  
 }  
 **public void** setmPassword(String mPassword) {  
 **this**.**mPassword** = mPassword;  
 }  
 **public** String dateFormatted() {  
 SimpleDateFormat sdf = **new** SimpleDateFormat(**"MMM dd - HH:mm"**);  
 **return "Last edited on: "** + sdf.format(**mLastModified**);  
 }  
}

Note that password property is **not** set as required otherwise Realm won’t allow us to create a note without password.

1. You will need to modify **NotesAdapter** class in order to work with a **List<Note>** objects instead of **ArrayList<Note>**. When querying Realm for all the rows, it returns a list of **RealmResults;** this collection is not compatible with **ArrayList**.



1. Once the adapter is fixed, open **NotesList** java file. You will need to modify some of the code from this file. Start by removing both **request codes** at the top. For this app, when you use a database, you no longer require to start an activity for result, as soon as you create, edit or remove a note from the database the changes will be reflected in **NotesList**.

Also remove the **ArrayList** of Notes, basically you will replace the Array List with a method that will return a list of all the notes from the database. (***getNotesList()***)



1. The next step is to remove all the unnecessary code from NotesList.
2. Remove the method **updateNotesList**
3. Remove the method **onActivityResult**
4. Modify **EditNoteIntent** so it will look like the following code:



You don’t need to send the entire note to **EditNote** activity, this time you’ll send only the id and then you can load that specific note by id from the database.

1. You are almost done with **NotesList** file; the las change is in **notesListView** listener.



1. The last step would be to refresh the adapter every time **NotesList** is loaded. For this you can override onStart method.



1. Completed **NotesList** java file:

**package** androidcourse.notes;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.graphics.Color;  
**import** android.support.v7.app.AlertDialog;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.text.Editable;  
**import** android.text.TextWatcher;  
**import** android.view.View;  
**import** android.widget.AdapterView;  
**import** android.widget.EditText;  
**import** android.widget.ImageView;  
**import** android.widget.ListView;  
**import** android.widget.TextView;  
**import** java.util.List;  
  
  
**import** androidcourse.notes.Adapters.NotesAdapter;  
**import** androidcourse.notes.Models.Note;  
**import** io.realm.Realm;  
**import** io.realm.Sort;  
  
**public class** NotesList **extends** AppCompatActivity {  
  
 **private** NotesAdapter **adapter**;  
 **private** Realm **realm**;  
  
 **private** List<Note> getNotesList() {  
 **return realm**.allObjectsSorted(Note.**class**,**"mTitle"**, Sort.***ASCENDING***);  
 }  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_notes\_list***);  
 **realm** = Realm.*getDefaultInstance*();  
 **final** ListView notesListView = (ListView) findViewById(R.id.***listView***);  
 **adapter** = **new** NotesAdapter(**this**, R.layout.***note\_row***, getNotesList());  
 notesListView.setAdapter(**adapter**);  
  
 notesListView.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {  
 *//get the selected note from the list* Note note = **adapter**.getItem(position);  
 *//if note is password protected* **if** (note.getmPassword() != **null**) {  
 displayPinPrompt(note);  
 } **else** {  
 editNoteIntent(note);  
 }  
 }  
 });  
 *//listener for add NoteImage* ImageView addNoteImg = (ImageView) findViewById(R.id.***addNoteImg***);  
 addNoteImg.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//create an intent* Intent addNoteIntent = **new** Intent(getBaseContext(), AddNote.**class**);  
 startActivity(addNoteIntent);  
 }  
 });  
 }  
  
 @Override  
 **protected void** onStart() {  
 **super**.onStart();  
 **adapter**.notifyDataSetChanged();  
 }  
  
 **private void** displayPinPrompt(**final** Note note) {  
 View layout = getLayoutInflater().inflate(R.layout.***pin\_prompt\_layout***, **null**);  
 **final** EditText password1 = (EditText) layout.findViewById(R.id.***pwd1***);  
 **final** TextView error = (TextView) layout.findViewById(R.id.***TextView\_PwdProblem***);  
  
 password1.addTextChangedListener(**new** TextWatcher() {  
 @Override  
 **public void** beforeTextChanged(CharSequence s, **int** start, **int** count, **int** after) {  
 }  
  
 @Override  
 **public void** onTextChanged(CharSequence s, **int** start, **int** before, **int** count) {  
 }  
  
 **public void** afterTextChanged(Editable s) {  
 String strPass1 = password1.getText().toString();  
 *//validate if password is correct* **if** (!strPass1.equals(note.getmPassword())) {  
 error.setText(**"Invalid Password"**);  
 error.setTextColor(Color.***RED***);  
 }  
 **else** {  
 error.setText(**"Valid Password"**);  
 error.setTextColor(Color.***GREEN***);  
 }  
 }  
 });  
  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(NotesList.**this**);  
 builder.setView(layout);  
 builder.setNegativeButton(android.R.string.***cancel***, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** whichButton) {  
 dialog.cancel();  
 }  
 });  
 builder.setPositiveButton(android.R.string.***ok***, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 String strPassword1 = password1.getText().toString();  
 **if** (strPassword1.equals(note.getmPassword())) {  
 editNoteIntent(note);  
 }  
 }  
 });  
 AlertDialog passwordDialog = builder.create();  
 passwordDialog.show();  
 }  
  
 **private void** editNoteIntent(Note note) {  
 Intent editNoteIntent = **new** Intent(NotesList.**this**, EditNote.**class**);  
 editNoteIntent.putExtra(**"note to edit"**, note.getmId());  
 startActivity(editNoteIntent);  
 }

@Override  
 **protected void** onDestroy() {  
 **super**.onDestroy();  
 **realm**.close();  
 }  
}

Close realm database connection

1. Open **addNote** java file. You will be adding the code required to save the notes in the database.

Create a method that will be used to save a note.

**private void** saveNote(Note note) {  
 *//set the id for the note* note.setmId(getNextNoteId());  
 *// persist your data* **realm**.beginTransaction();  
 **realm**.copyToRealm(note);  
 **realm**.commitTransaction();  
}  
*//used to generate the next note id***private int** getNextNoteId() {  
 **int** id = 1;  
 **if** (**realm**.allObjects(Note.**class**).size() > 0)  
 id = **realm**.where(Note.**class**).max(**"mId"**).intValue() + 1;  
 **return** id;  
}

1. Modify **onOptionsItemSelected** menu method in order to remove the intent and just call saveNote method.

**public boolean** onOptionsItemSelected(MenuItem item) {  
 **if** (item.getItemId() == R.id.***SaveNote***) {  
 String title = ((EditText) findViewById(R.id.***title\_add***)).getText().toString();  
 String contents = ((EditText) findViewById(R.id.***title\_add***)).getText().toString();  
 Note note;  
 *//check if the note has password set.* **if** (**mPassword** == **null**) {  
 note = **new** Note(title, contents);  
 } **else** {  
 note = **new** Note(title, contents, **mPassword**);  
 }  
 saveNote(note);  
 finish();  
 }  
 **return true**;  
}

1. Override **onDestroy** method to this activity so the realm connection is closed.

@Override  
**protected void** onDestroy() {  
 **super**.onDestroy();  
 **realm**.close();  
}

1. Completed code for **AddNote** java file:

**package** androidcourse.notes;  
**import** android.content.DialogInterface;  
**import** android.graphics.Color;  
**import** android.support.v7.app.AlertDialog;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.text.Editable;  
**import** android.text.TextWatcher;  
**import** android.view.Menu;  
**import** android.view.MenuInflater;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.CheckBox;  
**import** android.widget.EditText;  
**import** android.widget.TextView;  
**import** androidcourse.notes.Models.Note;  
**import** io.realm.Realm;  
  
**public class** AddNote **extends** AppCompatActivity {  
  
 **private** String **mPassword**;  
 **private** Realm **realm**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_add\_note***);  
 **realm** = Realm.*getDefaultInstance*();  
 **final** CheckBox password = (CheckBox) findViewById(R.id.***pwdCheckBox***);  
 **assert** password != **null**;  
 password.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 View layout = getLayoutInflater().inflate(R.layout.***pin\_layout***, **null**);  
 **final** EditText password1 = (EditText) layout.findViewById(R.id.***pwd1***);  
 **final** EditText password2 = (EditText) layout.findViewById(R.id.***pwd2***);  
 **final** TextView error = (TextView) layout.findViewById(R.id.***TextView\_PwdProblem***);  
 *//Listener for the password checkbox* password2.addTextChangedListener(**new** TextWatcher() {  
 @Override  
 **public void** beforeTextChanged(CharSequence s, **int** start, **int** count, **int** after) {  
 }  
  
 @Override  
 **public void** onTextChanged(CharSequence s, **int** start, **int** before, **int** count) {  
 }  
  
 **public void** afterTextChanged(Editable s) {  
 String strPass1 = password1.getText().toString();  
 String strPass2 = password2.getText().toString();  
 *//validate if both passwords are the same* **if** (strPass1.equals(strPass2) && strPass2.length() == 4) {  
 error.setText(**"Passwords Match"**);  
 error.setTextColor(Color.***GREEN***);  
 } **else if** (strPass2.length() != 4) {  
 error.setText(**"Password must contain 4 digits"**);  
 error.setTextColor(Color.***RED***);  
 } **else** {  
 error.setText(**"Passwords do not Match"**);  
 error.setTextColor(Color.***RED***);  
 }  
 }  
 });  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(AddNote.**this**);  
 builder.setView(layout);  
 builder.setNegativeButton(android.R.string.***cancel***, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** whichButton) {  
 *//if password prompt is cancelled disable checkbox* password.setChecked(**false**);  
 **mPassword** = **null**;  
 }  
 });  
 builder.setPositiveButton(android.R.string.***ok***, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 String strPassword1 = password1.getText().toString();  
 String strPassword2 = password2.getText().toString();  
 **if** (strPassword1.equals(strPassword2)) {  
 **mPassword** = strPassword1;  
 }  
 }  
 });  
 AlertDialog passwordDialog = builder.create();  
 passwordDialog.show();  
 }  
 });  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.***add\_note\_menu***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **if** (item.getItemId() == R.id.***SaveNote***) {  
 String title = ((EditText) findViewById(R.id.***title\_add***)).getText().toString();  
 String contents = ((EditText) findViewById(R.id.***title\_add***)).getText().toString();  
 Note note;  
 *//check if the note has password set.* **if** (**mPassword** == **null**) {  
 note = **new** Note(title, contents);  
 } **else** {  
 note = **new** Note(title, contents, **mPassword**);  
 }  
 saveNote(note);  
 finish();  
 }  
 **return true**;  
 }  
  
 **private void** saveNote(Note note) {  
 *//set the id for the note* note.setmId(getNextNoteId());  
 *// persist your data* **realm**.beginTransaction();  
 **realm**.copyToRealm(note);  
 **realm**.commitTransaction();  
 }  
  
 *//used to generate the next note id* **private int** getNextNoteId() {  
 **int** id = 1;  
 **if** (**realm**.allObjects(Note.**class**).size() > 0)  
 id = **realm**.where(Note.**class**).max(**"mId"**).intValue() + 1;  
 **return** id;  
 }  
  
 @Override  
 **protected void** onDestroy() {  
 **super**.onDestroy();  
 **realm**.close();  
 }  
}

1. Open **EditNote** java file. You will be adding the code required to update the notes in the database and delete notes from the database.
2. The next step would be to create a method that will be used to search for a note by its id.

**private** Note findNote(**int** id) {  
 **return realm**.where(Note.**class**).equalTo(**"mId"**, id).findFirst();  
}

1. Create a new Realm property at the top of the class.

Inside onCreate method, get the default realm instance and assign it to the newly created property.

**package** androidcourse.notes;  
**import** android.content.Intent;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.Menu;  
**import** android.view.MenuInflater;  
**import** android.view.MenuItem;  
**import** android.widget.EditText;  
**import** androidcourse.notes.Models.Note;  
**import** io.realm.Realm;  
  
**public class** EditNote **extends** AppCompatActivity {  
  
 **private** Note **noteToEdit**;  
 **private** Realm **realm**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_edit\_note***);  
  
 **realm** = Realm.*getDefaultInstance*();  
  
 *//obtaining the note from notesList activity* Intent intent = getIntent();  
 **if** (intent.hasExtra(**"note to edit"**)) {  
 **int** noteId = intent.getIntExtra(**"note to edit"**, 0);  
 **noteToEdit** = findNote(noteId);  
 }  
 *//updating the UI with the note info* EditText title = (EditText) findViewById(R.id.***title\_edit***);  
 title.setText(**noteToEdit**.getmTitle());  
 EditText content = (EditText) findViewById(R.id.***note\_info\_edit***);  
 content.setText(**noteToEdit**.getmContent());  
 }

As seen from the above code, instead of receiving a Note object, the intent carries the id of the note that you want to edit.

Your app uses **findNote** method to retrieve the note from the database and then it populates the title and content widgets.

1. Create the CRUD functionality for Update and Delete a note.

**private void** updateNote(String title, String content) {  
 **realm**.beginTransaction();  
 **noteToEdit**.setmTitle(title);  
 **noteToEdit**.setmContent(content);  
 **noteToEdit**.setmLastModified(**new** Date());  
 **realm**.commitTransaction();  
}  
  
**private void** deleteNote() {  
 **realm**.beginTransaction();  
 **noteToEdit**.removeFromRealm();  
 **realm**.commitTransaction();  
}

1. Locate **onOptionsItemSelected** method and modify the existing code in order to use both methods that you created in the previous step.

@Override  
**public boolean** onOptionsItemSelected(MenuItem item) {  
 **if** (item.getItemId() == R.id.***SaveNote***) {  
 String title = ((EditText) findViewById(R.id.***title\_edit***)).getText().toString();  
 String content = ((EditText) findViewById(R.id.***note\_info\_edit***)).getText().toString();  
 *//updating the note* updateNote(title, content);  
 } **else if** (item.getItemId() == R.id.***DeleteNote***) {  
 deleteNote();  
 }

*//Return to notesList activity*  
 finish();   
 **return true**;  
}

1. The last step is to **override** **onDestroy** method in order to close the realm connection.

@Override  
**protected void** onDestroy() {  
 **super**.onDestroy();  
 **realm**.close();  
}

1. Completed code for EditNote:

**package** androidcourse.notes;  
  
**import** android.content.Intent;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.Menu;  
**import** android.view.MenuInflater;  
**import** android.view.MenuItem;  
**import** android.widget.EditText;  
  
**import** java.util.Date;  
  
**import** androidcourse.notes.Models.Note;  
**import** io.realm.Realm;  
  
**public class** EditNote **extends** AppCompatActivity {  
  
 **private** Note **noteToEdit**;  
 **private** Realm **realm**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_edit\_note***);  
  
 **realm** = Realm.*getDefaultInstance*();  
  
 *//obtaining the note from notesList activity* Intent intent = getIntent();  
 **if** (intent.hasExtra(**"note to edit"**)) {  
 **int** noteId = intent.getIntExtra(**"note to edit"**, 0);  
 **noteToEdit** = findNote(noteId);  
 }  
 *//updating the UI with the note info* EditText title = (EditText) findViewById(R.id.***title\_edit***);  
 title.setText(**noteToEdit**.getmTitle());  
 EditText content = (EditText) findViewById(R.id.***note\_info\_edit***);  
 content.setText(**noteToEdit**.getmContent());  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.***edit\_note\_menu***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **if** (item.getItemId() == R.id.***SaveNote***) {  
 String title = ((EditText) findViewById(R.id.***title\_edit***)).getText().toString();  
 String content = ((EditText) findViewById(R.id.***note\_info\_edit***)).getText().toString();  
 *//updating the note* updateNote(title, content);  
 } **else if** (item.getItemId() == R.id.***DeleteNote***) {  
 deleteNote();  
 }  
 *//Return to notesList activity* finish();  
 **return true**;  
 }  
  
 **private** Note findNote(**int** id) {  
 **return realm**.where(Note.**class**).equalTo(**"mId"**, id).findFirst();  
 }  
  
 **private void** updateNote(String title, String content) {  
 **realm**.beginTransaction();  
 **noteToEdit**.setmTitle(title);  
 **noteToEdit**.setmContent(content);  
 **noteToEdit**.setmLastModified(**new** Date());  
 **realm**.commitTransaction();  
 }  
  
 **private void** deleteNote() {  
 **realm**.beginTransaction();  
 **noteToEdit**.removeFromRealm();  
 **realm**.commitTransaction();  
 }  
  
 @Override  
 **protected void** onDestroy() {  
 **super**.onDestroy();  
 **realm**.close();  
 }  
}

1. Test your application in the emulator/device and make sure all the functions are working properly.