CodeLab Data Persistence

Simone Salvatore La Milia

November 5, 2023

1 Description

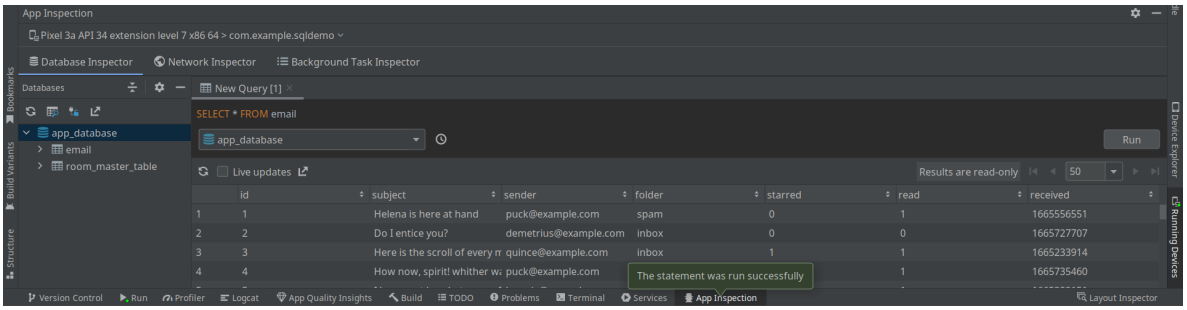
The goal of this CodeLab is learn **how to store data** locally on the device and make your apps work despite network interruptions to deliver a smooth and consistent user experience.
Is divided into 3 parts:

- An introductory part to SQL
- A development part with Room to create a database in an Android app
- A part that will involve using data in a simple way with key-value in an Android App

The link used for theoretical learning is Unidad 6: Persistencia de datos.

2 SQL

The first part explains **how to read and write to a database** using SQL. A test code is provided so you can test the queries downloadable from here. Once you start the App from Android Studio you can go to the **App Inspection** section (**View/Tool Windows/App Inspection**)to get access to the database and be able to enter the queries in the appropriate section.
An example of a query is shown below:



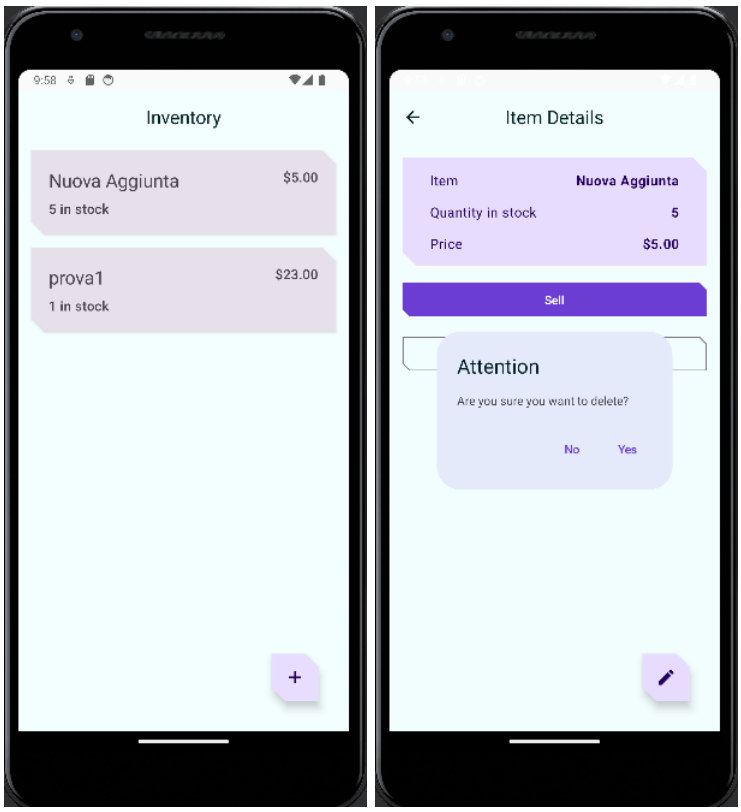
3 Room

Room is an **abstraction layer over SQLite** and provides convenient APIs to setup, configure, and query the database. In this part you learn how to use Room in your Android apps, how to add and modify existing data with Room and finally, the concepts are applied in the implementation of a Bus Schedule App of which only the initial code is provided.

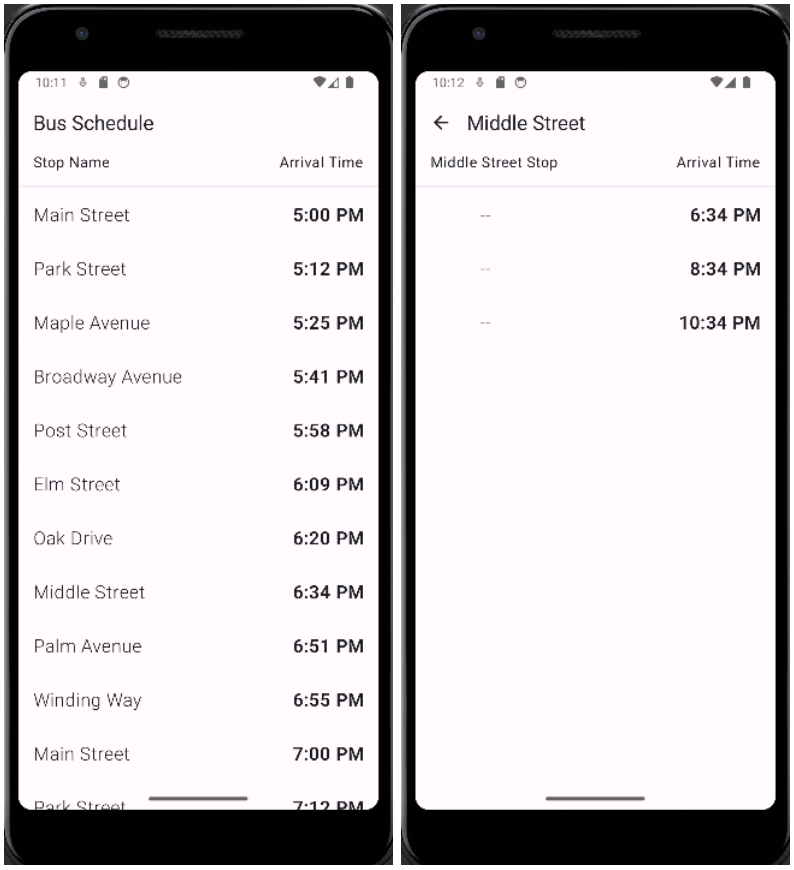
The Codes for this CodeLab can be seen in my GitHub in the following links:

- Inventory
- Bus Schedule App

Below are two screenshots of the application "**Inventory**" with data entered into the database and viewable on the screen:



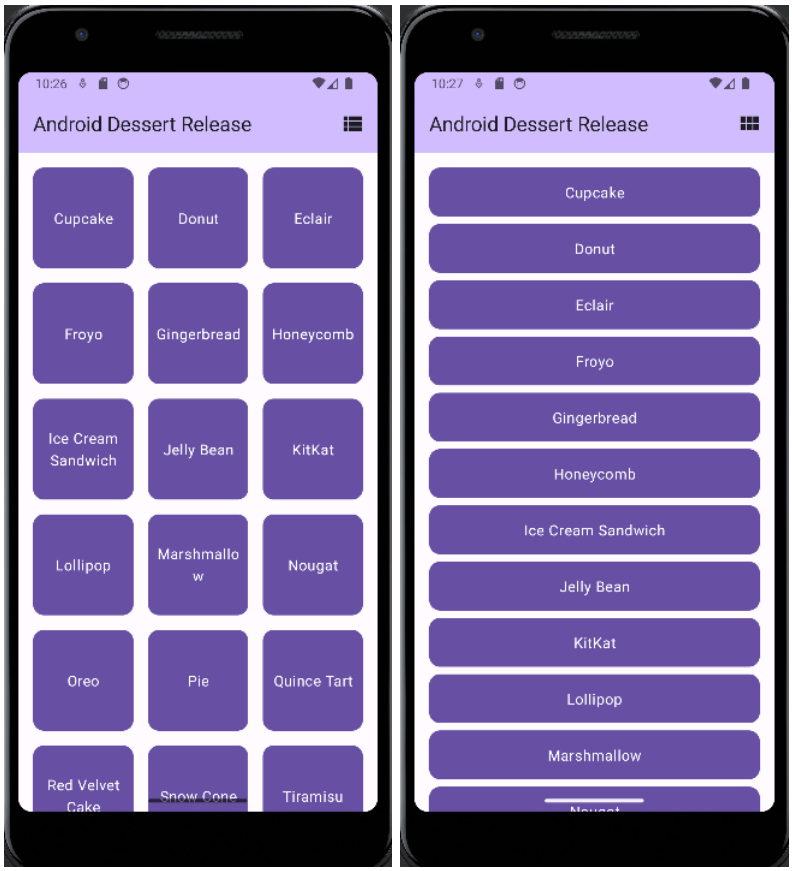
The screens of the "**Bus Schedule App**" part are shown below:



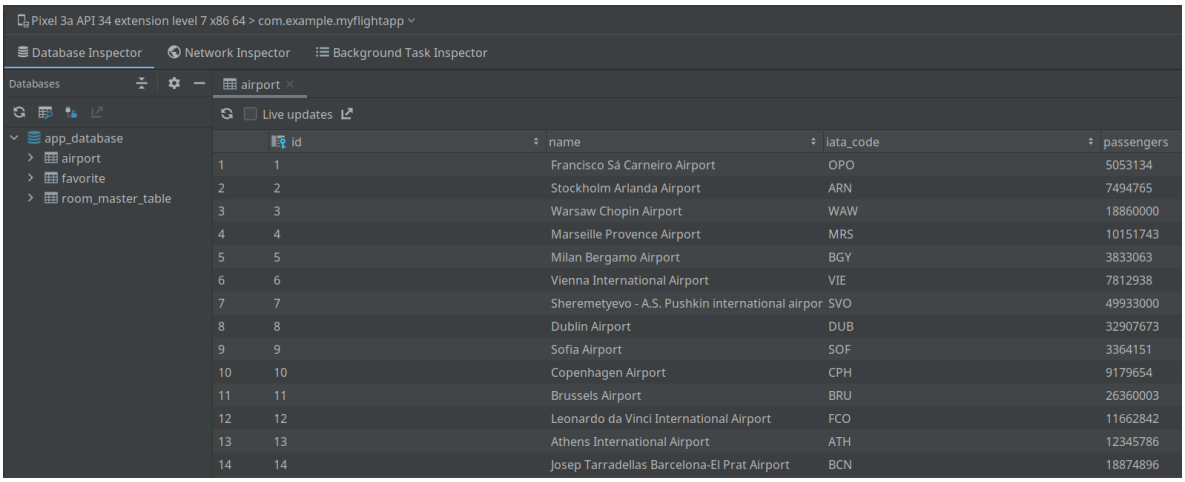
4 Keys with Datastore

In this part you learn how to store user preferences on the device with **DataStore**.

- The first activity involves making a **menu of desserts** that can be displayed in two different modes via a button. The app code can be found on my GitHub and downloaded from here.
  Preferences DataStore is a great way to store user-controlled settings, and in this CodeLab, we can learn how to do.
  Below are the two visible screenshots of the app:

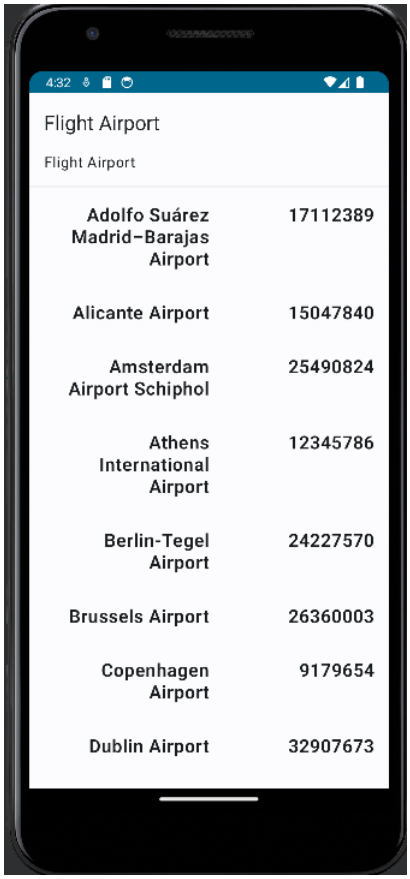


- For the second activity was created an app that lets users select a **flight departure**.
  The app code can be found on my GitHub and downloaded from here.
  Below is the **App Inspection**:



	id	name	iata_code	passengers
1	1	Francisco Sá Carneiro Airport	OPO	5053134
2	2	Stockholm Arlanda Airport	ARN	7494765
3	3	Warsaw Chopin Airport	WAW	18860000
4	4	Marseille Provence Airport	MRS	10151743
5	5	Milan Bergamo Airport	BGY	3833063
6	6	Vienna International Airport	VIE	7812938
7	7	Sheremetyevo - A.S. Pushkin international airpor	SVO	49933000
8	8	Dublin Airport	DUB	32907673
9	9	Sofia Airport	SOF	3364151
10	10	Copenhagen Airport	CPH	9179654
11	11	Brussels Airport	BRU	26360003
12	12	Leonardo da Vinci International Airport	FCO	11662842
13	13	Athens International Airport	ATH	12345786
14	14	Josep Tarradellas Barcelona-El Prat Airport	BCN	18874896

And the Screen of app:



5 Problems

Also for this CodeLab there were no major problems, except **in part 8** of the Room section where a brace and a comma were missing.
Below is the error in the example code and the correction in my code:

```
import kotlinx.coroutines.launch

ItemDetailsBody(
    itemUiState = uiState.value,
    onSellItem = { viewModel.reduceQuantityByOne() },
    onDelete = {
        coroutineScope.launch {
            viewModel.deleteItem()
    }
    modifier = modifier.padding(innerPadding)
)
```

```
ItemDetailsBody(
    itemUiState = uiState.value,
    onSellItem = { viewModel.reduceQuantityByOne() },
    onDelete = {
        coroutineScope.launch { this: CoroutineScope
            viewModel.deleteItem()
            navigateBack()
        }
    },
    modifier = modifier.padding(innerPadding)
)
```

An additional problem is the absence in the "implements" section of beginning CodeLab (for the last part of the CodeLab on room) of the following property:

```
buildscript { this: ScriptHandlerScope
    extra.apply { this: ExtraPropertiesExtension
        set("nav_version", "2.5.3")
        set("room_version", "2.5.1")
    }
}
```