

ACTIVITY 3.1.5

Updates and Deletes

INTRODUCTION

In previous activities of this lesson, you have *created* a trip in the database using the Backendless `save` method and you have *read* a trip from the database using the `find` method. In this activity, you will learn to *update* an existing trip and *delete* a trip from the database. These four basic data storage operations are referred to as “CRUD”.

Materials

- Computer with Android™ Studio
- Android™ tablet and USB cable, or a device emulator
- Free Backendless account per student

RESOURCES



Lesson 3.1 Reference Card for Backendless
Resources available online

Procedure

Part I: Edit a Trip

Your app has an existing user interface to create and read a trip. Once you create the back-end functionality, you will be able to update and delete trips. Once implemented, all of the CRUD operations can be done from the Trip Details screen (`TripFragment`) and the delete can be done from the My Trips screen (`TripListFragment`).

- 1 Open your TripTracker app in Android Studio.

NOTE

If you were unable to complete *Activity 3.1.4 Listing Trips*, import *3.1.4TripTracker_Solution* as directed by your teacher. Recall that if you import the solution, you must update keys values in *strings.xml* Specifically,

- Change `be_app_id` your Backendless App ID key value
- Change `be_android_api_key` to your Android API key value

You can retrieve these from your  **Backendless Console** (**Manage** icon).

- 2 Open `TripListFragment` and observe the `onListItemClick` method.
- 3 Open the `IntentData` interface. Explain what the `onListItemClick` code does and how it relates to this interface; give an example of one piece of intent data.
- 4 The `onListItemClick` method is missing one crucial step—starting the activity. Use the code hints to issue `startActivity` with an intent.
- 5 What part of the intent indicated which activity should be started?
- 6 Test your app to confirm you can edit and save/update a trip.

Part II: Delete a Trip

`TripTracker` provides two ways to delete a trip. The user can:

- Use the trip's context menu on the `TripList` screen.
 - Use the Delete icon in the action bar on the `Trip Details` screen.
- 7 To delete a trip from the `TripList` screen, find the `deleteTrip` method in `TripListFragment`.
 - a. Use the Backendless `remove` method in a new `Thread`. Because the `remove` is called in a new thread, the reference to the trip object must be declared as `final`: If an old thread were to modify an object that is used by a new thread, the new thread would produce inconsistent and unpredictable results.
 - b. Create a copy of the trip parameter using a `final Trip` and use the new trip reference to `remove` the trip.
 - c. After the thread completes, don't return to another activity, rather, refresh the trip list.
 - 8 To delete a trip from the `Trip Details` screen, modify the `deleteTrip` method in `TripFragment`.
 - a. Check that `mTrip` represents an existing trip and not a new trip. (Recall that the user may have selected the Add icon from the `Trip Details` screen, and a trip may not exist yet in Backendless.) You can determine this by checking that `mTrip`'s `getObjectId` method is not `null`.

- b. Use a similar algorithm from the previous step to delete a trip.
 - c. As you did in `updateTrip`, use `getActivity().finish()` to finish the activity and return to the `TripList` screen.
- 9 Call the `deleteTrip` method when the user selects the delete icon (in `onOptionsItemSelected`, the `action_delete` case).

Part III: Log Out of the Trip Details Screen

- 10 If you have not yet tried to log out of your app, log out from the Trip Details screen.
- 11 Log back in and attempt to log out of the Trip Details screen. What is the error that occurs in `logcat`?
- 12 While both fragments use the Backendless logout method, `TripFragment` is missing something. Determine what is missing (`null`) and fix the bug. If necessary, get a hint from your teacher.

In Activity 3.1.6, you will modify both trip screens to accommodate the view of publicly shared trips and personal trips.

CONCLUSION

- 1. Explain which Backendless methods are used to accomplish “CRUD”.