

## Assignment 7

In this assignment, you will continue working on your code for Database Assignment by adding server communication to your application.

### The REST API to Use:

In this course, we use Back4App service based on the Parse platform as the RESTful server, which provides general REST APIs without needing any extra coding or settings on the server side.

If you've not done this before, first sign up as a new user in [Back4App website](#). In your Back4App dashboard, you can create a new app. Then you can see your Application ID and REST API key in the App Settings > Security & Keys section in the left sidebar. These strings should be used respectively in X-Parse- Application-Id and X-Parse-REST-API-Key headers of all requests sent to Back4App.

In Back4App dashboard, you can always check the data saved in your app by opening the Core > Database Browser section in the left sidebar. If you want to test sending requests to Back4App outside Android code, you can use REST request builders like [Postman](#).

The Back4App REST API host address is: <https://parseapi.back4app.com>. You should first watch the Android networking videos to learn how to consume the Back4App REST API. You can also check its [documentation](#) for more details.

### User Sign-Up:

Your application already has the ability to sign up new users. In this assignment, you should change the sign-up process so that the new user is saved to the server instead of the local database.

In order to save the new user to the server, a POST request should be sent to /users route of the Back4App server. The server assumes that you send the user data including at least email and password fields in the body of the request.

### User Sign-In:

When the user opens the app, the sign-in page should be opened only if no previous user is already signed in. Otherwise, if there is any user that has recently signed in to

the app, the home page of the app should be displayed. This means that when a user signs in, he/she should remain logged-in until he/she logs out of the app using the exit button in the home page. For this purpose, you need to keep the login state of different users in the User table in the local database.

In the sign-in page, after the user enters his/her username and password, a GET request should be sent to /login route in the server, with username and password added to the URL as query strings. If the sign-in request is successful, the server returns the full data of the logged-in user, which includes at least the server-generated ID of the user in objectId field, and the user session token in the sessionToken field.

Now the user data should be saved to the local database.

Because the ID fields of the main entities of your app are now generated in server, you should change your previous database schema to store the ID fields as string values provided by server.

### **Saving Changes to Server:**

Getting, adding, updating and removing reports should now be done in communication with the server. After saving to the server successfully, the related item should be saved to or removed from the local database.

For each of these operations, a request with appropriate methods should be sent to /classes/report endpoint. The appropriate GET, POST, PUT or DELETE methods should be used for each of these requests. Also, all these requests should have the session token of the current user in the X-Parse-Session-Token header.

### **Notes about Implementation:**

You should respect common implementation notes mentioned in your previous assignments in this assignment too. In addition:

1. Before working on this assignment, make sure the main functionalities of your app from Database Assignment work as expected, because in this assignment you should change and complete your code for the previous assignment.
2. Because in this assignment you may change the previous schema of your local database, you may need to uninstall your app for the previous assignment from your device, or at least clear its data, in order to prevent conflicts with the previous database.

The ideal way for handling changes in database schema is using [Room migrations](#), but you're not required to use migration in this assignment.

3. For network requests, you should use the [Retrofit](#) networking library. For JSON processing, you should use the Gson library.

### **Notes about Submission:**

1. You should submit a zipped file which includes both your root project folder and the final APK file of your project. You can find the generated APK file of your project in the path YOURPROJECT/app/build/outputs/apk/ on disk.

2. In order to reduce the size of the project for submission, you can remove the folder build in the path /app/build of your project folder. Make sure to get out the APK file before removing this folder.