

Testing Report

General Approach

We tested our system the way it is meant to be used, by building applications. The two applications we built were a Notes app and a Chat app. Both of these applications used our client-side SDK, server and Admin UI to manage the application.

The Notes app tested our system's capability of building any general CRUD application, along with OAuth integration and file handling. In the Notes app user has two ways to sign up, using our auth or using Google's OAuth. Each user has their own avatar which they can update. This is possible through a way our system allows any file to be linked to a record and then can be fetched with the record. The notes can be created and deleted. All notes are fetched in descending order, newest first, through the SDK and server.

The chat app tests the real-time database feature of our system. There are two collections chats and messages. Each message is filtered based on which chat it belongs to and sent to those users which are part of that chat. Files can be sent as messages as well. The user can be a part of different chats and receive real-time messages and notifications.

Both applications require no additional third-party library. The demo video of these applications is uploaded to the repository.

Server Tests

```
✓ Collection CRUD > Get all collections [13.40ms]
Collection: Collection created successfully
✓ Collection CRUD > Create collection [9.96ms]
✓ Collection CRUD > Delete collection - force [10.58ms]
✓ Record CRUD > Create record [1.88ms]
✓ Record CRUD > Read record [1.34ms]
✓ Record CRUD > Read all records [0.57ms]
✓ Record CRUD > Update record [1.07ms]
✓ Record CRUD > Delete record [9.51ms]
✓ Authentication - Admin > Check if admin exists before creation [0.91ms]
✓ Authentication - Admin > Create Admin User [194.41ms]
✓ Authentication - Admin > Check if admin exists [0.71ms]
✓ Authentication - Admin > Get all Admin Users [0.40ms]
✓ Authentication - Admin > Login Admin User [214.19ms]
✓ Authentication - Admin > Delete Admin User [4.91ms]
✓ Authentication - Admin > Check if admin exists after deletion [0.51ms]
✓ Rule based Access Control > Fetch Rules [0.89ms]
✓ Rule based Access Control > Update Rules [1.66ms]
✓ Rule based Access Control > Fetch Rules after update [0.56ms]

18 pass
0 fail
48 expect() calls
Ran 18 tests across 1 files. [717.00ms]
```