

Server Library Datastore

1 Fields

The elements that we would store in the database, for each session, would be:

- `api_key`: unique identifier for the user given a session
- `device_name`: name of the device for this session
- `app_name`: name of the application for this session
- `os_type`: type of logs being stored. Android is currently supported, and will provide plans for iOS
- `start_time`: start time of session
- `log_entries`: An array of JSON objects of the logs including time, text, tag, and `logType`, `processId` and `threadId` of the log
- `ended`: a boolean for if the session has ended or not

In addition to the collection of sessions, the database will also have a collection each for the device alias and the app alias. These collections will both have the following values:

- `api_key`: unique identifier for the user for the selected apps
- `device_name`: actual name of the device used, sent from the mobile app

The device collection will have the additional value of `device_alias`, which will be the alias given by the user and the value that is displayed on the web page.

The `app_name` collection will have the additional value of the `app_name` which will hold the value given by the mobile app. The `app_alias` supplied by the user and displayed on the web page will also be in this collection.

2 Storing in Database

2.1 Session Collection

The `log_entries` and `os_type` will be provided by the web backend after parsing the `rawLogData`. The `device_name` and `app_name` will be updated based on the WebSocket information sent to the web app backend from the mobile API. The `start_time` and `ended` will be populated based on the time when `startSession` and `endSession` is sent, respectively. The `api_key` will be retrieved from the user management interface.

2.2 Unique Devices and Unique Apps Collections

When a mobile API starts streaming at first, the `api_key`, the `device_name`, and `app_name` will be compared with the `devices` and `app_name` collections. If it's a new value, then the `api_key/device_name` is added with the `device_name` serving as the alias. The `api_key`, `device_name`, and `app_name` will also be inserted into the `app_name` collection with the raw `app_name` acting as the `app_alias`. If the values are already in the collection(s), nothing is done.

3 Retrieving from Database

3.1 Session Collection

If historical logs are requested then as soon as a user is chosen, a device name selected, an app name selected, and a start time selected, then the log entries are retrieved from the database. If streaming a current session, then the parsing library will send the `log_entries` directly to the web interface as HTML and then simultaneously store in the database as JSON objects.

3.2 Unique Devices and Unique Apps Collections

This information is collected when the AJAX request for devices and apps is requested. The `api_key` has to be determined and then collects all relevant device aliases. When a device is selected, then the `api_key` and `device_name` is used to collect all relevant app aliases.