kenzi.ai

Transforming Business with Technology

Kenzi Tracker

App Documentation

1. Introduction

Kenzi Tracker is a mobile application designed to track attendance for events or classes by scanning QR codes. It enables users to log in using a username and password, retrieve a list of events or classes, and then scan QR codes during the event for attendance tracking. The app interacts with two external Web APIs (webhooks) to facilitate data retrieval and QR code submission.

---

2. User Flow

1. **Login:**
    - The user logs in with a username and password.
    - Upon successful login, the user is prompted to provide two Web APIs (webhooks).
        1. **GET API:** Retrieves a list of events or classes based on the customer's configuration.
        2. **POST API:** Sends the scanned QR code text along with the user's username.
2. **Data Retrieval:**
    - The app will send a GET request to the first API, passing the username in the headers.
    - A list of events/classes is displayed to the user, each with an ID, title, and optional notes.
3. **Scanning:**
    - The user selects an event/class from the list.
    - The user is taken to a screen displaying the event details and a button to scan a QR code.

   - After scanning, the app sends a POST request to the second API, including the scanned QR code and username in the headers.
4. **Local Storage:**
   - The app stores scanned QR codes locally to ensure no data is lost in the event of connectivity issues.

---

3. API Specifications

3.1. GET API - Retrieve Events or Classes

- **Method:** GET
- **Headers:**
  - Username: [User's username] (sent in the header).

Response (JSON Array format):

```
[{
 "id": [number, required],
 "title": [string, required],
 "notes": [string, must be sent, but can be an empty string]
}]
```

- **Behavior:**
  - The API will return a list of events or classes based on the username provided.
  - Each entry in the list will contain an id, title, and notes.

3.2. POST API - Send Scanned QR Code

- **Method:** `POST`
- **Headers:** `Content-Type`: `application/json`
- **Body:** The POST request will include the following JSON object:

```
{

          "username": "logged in username",
          "id": "event or class id",
          "scanResult": "scan result",
          "action": "check-in, check-out, break-in, or break-out"

}
```

- **Explanation:**
  - `username`: The username of the logged-in user.
  - `id`: The ID of the event or class that the user is attending.
  - `qr`: The content of the QR code that was scanned during the event or class.
- **Response (if applicable):**
  - The API may return a status indicating whether the scanned data was successfully recorded.

---

4. App Workflow

1. **Login Screen:**
   - The user inputs their username and password.
   - Upon successful login, the user is asked to input two Web API URLs.
2. **Data Fetch Screen:**
   - The app sends the username as a header to the provided GET API.
   - The API returns a list of events or classes.
   - The user selects an item from the list.
3. **Details Screen:**
   - The selected item's details (ID, title, notes) are shown.
   - A **SCAN** button is available to trigger the QR code scanning functionality.
4. **QR Code Scan:**
   - Upon scanning, the QR code data is sent to the POST API along with the username.
   - The app locally stores the scanned data for future reference.

---

5. Local Data Management

- **Scanned IDs:**
  - The app stores the IDs of scanned QR codes locally in the device's storage.
  - This ensures data persistence in case of network issues.

---

6. Error Handling

- **GET API Errors:**
  - If the GET API request fails, the app will notify the user and allow them to retry.
- **POST API Errors:**
  - If the POST API fails, the scanned data is saved locally and reattempted when the connection is restored.

---

7. Security Considerations

- All API requests should be sent over HTTPS to ensure secure communication between the app and the server.
- User authentication via username and password should be managed securely, following best practices for password storage and handling.

8. Contact Information

If you need any help, please contact **connect@kenzi.ai**.

Watch this video to see how it works

https://tracker.kenzi.ai/how_it_works.mp4

This Application is developed and maintained by kenzi.ai

www.kenzi.ai