**EzSkool API Documentation**

**Overview**

This document outlines the API endpoints used in the EzSkool Flutter application. The application allows school staff to manage student attendance, view student information, and access reports.

**Base URL**

All API endpoints are prefixed with {{ezskool}}, which represents the base URL of the API server.

**Authentication**

Most API endpoints require authentication via Bearer token. The token is obtained during login and should be included in the Authorization header for subsequent requests:

Authorization: Bearer <token>

**API Endpoints**

**Public Endpoints (No Authentication Required)**

**1. Institution Registration**

* **Endpoint**: /inquiry/create
* **Method**: POST
* **Description**: Registers a new institution in the system
* **Request Parameters**:
* { "inst\_name": "MyInstitute", "inst\_cp": "TestPerson", "inst\_no": "9999999999", "inst\_email": "testmail@gmail.com", "inst\_type": "1", "inst\_add": "Ahmedabad", "inst\_city": "Ahmedabad", "pincode": "380009"}
* **Response**:
* { "success": true, "data": { "inst\_name": "MyInstitute", "inst\_add": "Ahmedabad", "inst\_cp": "TestPerson", "inst\_no": "9999999999", "inst\_email": "testmail@gmail.com", "inst\_city": "Ahmedabad", "pincode": "380009", "inst\_type": "1", "created\_by": null, "updated\_at": "2025-03-08 04:33:07", "created\_at": "2025-03-08 04:33:07", "id": 18 }, "message": "Inquiry submitted successfully."}

**2. Staff Login**

* **Endpoint**: /login
* **Method**: POST
* **Description**: Authenticates a staff member and provides an access token
* **Request Parameters**:
* { "client\_id": "client\_id\_value", "user\_name": "username", "password": "password"}
* **Response**:
* { "success": true, "data": { "user": { "id": 2, "inst\_id": 2, "name": "User One", "email": "user1@example.com", "user\_name": "user1", "mobile": null, "email\_verified\_at": null, "status": 1, "deleted\_at": null, "created\_at": "2024-12-30T13:10:05.000000Z", "updated\_at": "2024-12-30T13:10:05.000000Z" }, "treasure\_url": "https://ezs.raktdaan.org/public/treasure/2", "token": "330|PTEXCDjEpwZq32sZlpjk3Mb3f2bW40v4Pdmp9eUF53e4f778" }, "message": "Login successful."}

**Authenticated Endpoints (Bearer Token Required)**

**3. Staff Logout**

* **Endpoint**: /logout
* **Method**: POST
* **Description**: Logs out the current user and invalidates their token
* **Headers**: Authorization Bearer Token
* **Response**:
* { "success": true, "data": null, "message": "Logged out successfully."}

**4. Initial Data Sync**

* **Endpoint**: /get\_login\_sync
* **Method**: POST
* **Description**: Fetches initial data after login for local storage (classes, leave reasons, etc.)
* **Headers**: Authorization Bearer Token
* **Request Parameters**:
* {
* "sync\_req": "lr,div" // For leave reasons and divisions
* }
* **Response**:
* {
* "success": true,
* "data": {
* "lr": "Sick, Family Occasion, Others",
* "div": "A,B,C"
* },
* "message": ""
* }
* **For Classes**:
  + **Request Parameters**:
  + { "sync\_req": "class"}
  + **Response**:
  + { "success": true, "data": { "class": [ {"id": 1, "std\_id": 11, "division": "A", "class\_name": "10"}, {"id": 2, "std\_id": 1, "division": "A", "class\_name": "1-A"}, {"id": 3, "std\_id": 1, "division": "B", "class\_name": "1-B"} // Additional classes... ] }, "message": ""}

**5. Get Class Students for Attendance**

* **Endpoint**: /get\_class\_students\_att
* **Method**: GET
* **Description**: Retrieves student list for a specific class with attendance status
* **Headers**: Authorization Bearer Token
* **Query Parameters**:
  + class\_id: ID of the class
  + att\_date: Date for attendance (YYYY-MM-DD)
* **Response**:
* { "class\_att\_id": 0, "class\_id": 2, "class\_name": "1-A", "students": [ {"student\_id": 105, "name": "Krina Patel", "roll\_no": 1, "gender": 2}, {"student\_id": 106, "name": "Kamlash patel", "roll\_no": 2, "gender": 1}, // Additional students... ]}

*Note*: class\_att\_id: 0 means attendance not marked yet, non-zero means already marked

**6. Mark Class Attendance**

* **Endpoint**: /markclass\_att
* **Method**: POST
* **Description**: Marks or updates attendance for a class
* **Headers**: Authorization Bearer Token
* **Request Parameters for New Attendance**:
* {
* "att\_date": "2025-03-08",
* "ay\_id": "2024-25",
* "shift\_id": "1",
* "class\_id": "2",
* "absent\_roll\_no": "1,3,5",
* "action": "1",
* "ta": "3",
* "tp": "11"
* }
* **Request Parameters for Updating Attendance**:
* {
* "att\_date": "2025-03-08",
* "ay\_id": "2024-25",
* "shift\_id": "1",
* "class\_id": "2",
* "absent\_roll\_no": "1,3,5",
* "action": "2",
* "class\_att\_id": "26",
* "ta": "3",
* "tp": "11"
* }

*Note*: action: 1 for new attendance, 2 for updating existing attendance

* **Response**:
* {
* "success": true,
* "message": "Attendance marked successfully.",
* "attendance\_id": 26
* }

**7. Get Students List by Class**

* **Endpoint**: /get\_class\_student\_list
* **Method**: GET
* **Description**: Retrieves students list for a specific class (without attendance info)
* **Headers**: Authorization Bearer Token
* **Query Parameters**:
  + class\_id: ID of the class
* **Response**:
* { "success": true, "data": [ {"id": 136, "gender": 1, "roll\_no": 1, "name": "Steve Rogers"}, {"id": 137, "gender": 1, "roll\_no": 2, "name": "Tony Stark"}, // Additional students... ], "message": "Records retrieved successfully."}

**8. Get Student Details**

* **Endpoint**: /get\_student\_details
* **Method**: POST
* **Description**: Retrieves detailed information about a specific student
* **Headers**: Authorization Bearer Token
* **Request Parameters**:
* { "student\_id": "104"}
* **Response**:
* { "success": true, "data": { "id": 104, "apaar\_id": "1234", "pen\_no": "123456", "uid\_no": "12345", "gr\_no": "2345", "adhar\_no": "123456789098", "gender": 2, "bg": "O+", "dob": "2003-12-12", "doj": "0225-12-12", "dol": null, "hw": null, "current\_std\_id": 1, "current\_class\_id": 2, "current\_roll\_no": 46, "class\_name": "1-A", "std": "1", "category\_id": 5, "caste": null, "religion": null, "sib\_ids": "78,79", "pg\_link": "{\"0\": {\"id\": 2, \"rel\": \"Unknown\", \"name\": \"Mohita Patel\", \"email\": \"mp@gmail.com\"}}", "roll\_no": 46, "name": "Akshita Mathur", "sp\_remark": null, "ppsp": null, "siblings": [ {"student\_id": 78, "name": "Aryan Patel Makwana", "gender": 1, "class\_name": "5-A", "siblings\_roll\_no": 9}, {"student\_id": 79, "name": "Mahir Makwana", "gender": 1, "class\_name": "4-A", "siblings\_roll\_no": 25} ] }, "message": "Student details retrieved successfully."}

**9. Get Parents Contact Information**

* **Endpoint**: /get\_parents\_contact
* **Method**: GET
* **Description**: Retrieves contact information for a student's parents
* **Headers**: Authorization Bearer Token
* **Query Parameters**:
  + student\_id: ID of the student
* **Response**:
* { "student\_id": "128", "parents": [ {"parent\_id": 156, "parent\_name": "DineshBhai", "contact\_no": "9316709018", "relation": 1}, {"parent\_id": 158, "parent\_name": "Nipa", "contact\_no": "9316102032", "relation": 2} ]}

**10. Get Student Birthdays**

* **Endpoint**: /get\_student\_bday
* **Method**: POST
* **Description**: Retrieves students with birthdays within a specified date range
* **Headers**: Authorization Bearer Token
* **Request Parameters**:
* { "from\_date": "2025-02-01", "to\_date": "2025-02-28"}
* **Response**:
* { "success": true, "data": [ { "date": "2025-02-08", "classes": [ { "class\_id": 2, "students": [ {"student\_id": 103, "name": "Lacki Parmar", "gender": 1, "class": "1-A", "dob": "2025-02-08"}, {"student\_id": 128, "name": "Jagdish Parmar", "gender": 1, "class": "1-A", "dob": "2025-02-08"}, {"student\_id": 129, "name": "Bharat Makwana", "gender": 1, "class": "1-A", "dob": "2025-02-08"} ] } ] }, // Additional dates with birthdays... ], "message": "Birthdays retrieved successfully."}

**11. Get Class Attendance Report**

* **Endpoint**: /get\_shift\_att
* **Method**: POST
* **Description**: Retrieves attendance report for all classes for a specific shift and date
* **Headers**: Authorization Bearer Token
* **Request Parameters**:
* { "shift\_id": "1", "att\_date": "2025-03-03"}
* **Response**:
* [ {"class\_id": 1, "class\_name": "10", "att": null}, {"class\_id": 2, "class\_name": "1-A", "att": "10,3,13", "lut": "03-03-2025 02:40 PM"}, {"class\_id": 3, "class\_name": "1-B", "att": "3,2,5", "lut": "03-03-2025 11:04 AM"}, // Additional classes...]

*Note*: att format is "present,absent,total"; null means attendance not marked; lut is Last Updated Time

**12. Get Whole School Attendance Report**

* **Endpoint**: /get\_shift\_wise\_school\_att
* **Method**: POST
* **Description**: Retrieves school-wide attendance summary for a date range
* **Headers**: Authorization Bearer Token
* **Request Parameters**:
* { "frm\_dt": "2025-03-01", "to\_dt": "2025-03-07", "shift\_id": "1"}
* **Response**:
* [ { "date": "2025-03-01", "att": [ {"shift\_id": 1, "shift": "Morning Shift", "att": "0,0,0"}, {"shift\_id": 2, "shift": "Afternoon Shift", "att": "0,0,0"} ] }, { "date": "2025-03-03", "att": [ {"shift\_id": 1, "shift": "Morning Shift", "att": "25,8,33"}, {"shift\_id": 2, "shift": "Afternoon Shift", "att": "0,0,0"} ] }, // Additional dates...]

*Note*: att format is "present,absent,total"

**13. Get Class Attendance Summary**

* **Endpoint**: /get\_class\_att\_summary
* **Method**: POST
* **Description**: Retrieves attendance summary for a specific class over a date range
* **Headers**: Authorization Bearer Token
* **Request Parameters**:
* { "class\_id": "2", "frm\_dt": "2025-03-01", "to\_dt": "2025-03-06"}
* **Response**:
* { "class\_id": 2, "class\_name": "1-A", "attendance": [ {"dt": "2025-03-01", "att": null}, {"dt": "2025-03-02", "att": null}, {"dt": "2025-03-03", "att": "10

**13. Get Class Attendance Summary (continued)**

* **Response**:
* { "class\_id": 2, "class\_name": "1-A", "attendance": [ {"dt": "2025-03-01", "att": null}, {"dt": "2025-03-02", "att": null}, {"dt": "2025-03-03", "att": "10,3,13"}, {"dt": "2025-03-04", "att": null}, {"dt": "2025-03-05", "att": "13,0,13"}, {"dt": "2025-03-06", "att": "13,1,14"} ]}

*Note*: att format is "present,absent,total"; null means attendance not marked for that date

**Error Handling**

All API endpoints follow a consistent error response format:

{

"success": false,

"message": "Error message describing what went wrong"

}

Common error scenarios:

* Authentication failure (401): Token missing or invalid
* Authorization failure (403): User doesn't have permission
* Validation errors (422): Invalid input data
* Server errors (500): Unexpected server issues

**Data Types and Formats**

**Date Formats**

* All dates should be sent in the format: YYYY-MM-DD (e.g., "2025-03-08")

**Gender Codes**

* 1: Male
* 2: Female

**Attendance Status Format**

* In responses, attendance data is often formatted as "present,absent,total"
  + Example: "10,3,13" means 10 present students, 3 absent students, total 13 students

**Action Codes (for Attendance)**

* 1: Mark new attendance
* 2: Update existing attendance

**Local Storage Implementation**

The application stores several data types in local databases for offline access:

1. **Classes**: Class information including ID, name, and division
2. **Students**: Student information including personal details and class assignment
3. **Leave Reasons**: Standard reasons for student absences
4. **Attendance Data**: Temporary storage during the attendance marking process

**API Flow Summary**

1. User downloads and opens the app
2. User registers institution (if new) or logs in
3. Upon successful login, app fetches and stores:
   * Classes data
   * Leave reasons
   * Divisions
4. For marking attendance:
   * User selects class
   * App fetches students and current attendance status
   * User marks absences
   * App submits attendance data
5. For student information:
   * User selects class from list
   * App fetches students in that class
   * User can select individual student to view details
   * App can fetch parent contact information
6. For reports:
   * User can view attendance reports by class, date, or for the whole school
   * Birthday reports can be viewed by date range