SLMS API – End‑to‑End Guide & Reference

This document explains how to use the School Learning Management System (SLMS) APIs end‑to‑end, including authentication, sessions, class and subject setup, user creation, fees, attendance, TC requests, and timetables. It also clarifies which endpoints require admin, teacher, student, or non‑teaching staff JWTs.

**Tip:** You can explore and test these APIs interactively through the built-in **Swagger UI**.  
Just run the application and open <http://localhost:8080/swagger-ui/index.html> in your browser to access the complete API documentation and try out each endpoint directly.

# Quick Start Workflow (as requested)

1) The file slms.txt (attached) contains the initial admin credentials. Use them to log in and obtain an Admin JWT.

ADMIN\_USERNAME=admin@company.com  
ADMIN\_PASSWORD=temporaryStrongPassword!123

2) With Admin JWT: Create a 12‑month Session. Only one session should be active at a time; if you create a new one, first deactivate the previous active session.

3) With Admin JWT: Create Classes for that Session, including fee amounts per class.

4) With Admin JWT: Create Students under those Classes (and in the active Session).

5) With Admin or Non‑Teaching Staff JWT: Record Fees for students, and view fee catalogs.

6) With Admin JWT: Manage Subjects and assign Teachers to Subjects for specific Classes.

7) With Teacher JWT: Use /me endpoints to view personal data and timetable. With Student JWT: use /me endpoints to view the student’s own data and timetable.

8) With Admin JWT: Create the class Timetable using Timetable endpoints.

(Skip score/exam features as requested.)

# Authentication & JWTs

Obtain JWTs by logging in as Staff (admins, teachers, non‑teaching staff) or as Students. Include the JWT in the Authorization header as a Bearer token for protected endpoints:

Authorization: Bearer <JWT\_TOKEN>

## Auth Endpoints

### Register Staff

Method: POST

Path: /api/auth/register/staff

Required Roles / JWT: Admin JWT (to create staff accounts)

Purpose: Create a new staff user (roles can include ROLE\_ADMIN, ROLE\_TEACHER, non‑teaching).

Sample Request Body:

{  
 "name": "Teacher and Admin",  
 "email": "TeacherAndAdmin11111@example.com",  
 "qualification": "MSc",  
 "salaryGrade": "A1",  
 "contactNumber": "+1234567890",  
 "roles": ["ROLE\_TEACHER", "ROLE\_ADMIN"],  
 "password": "securePassword123",  
 "designation": "Senior Teacher",  
 "joiningDate": "2025-08-21"  
}

### Login Staff

Method: POST

Path: /api/auth/login

Required Roles / JWT: No JWT required

Purpose: Authenticate a staff user (admin, teacher, non‑teaching). Returns JWT.

Sample Request Body:

{  
 "email": "admin@company.com",  
 "password": "temporaryStrongPassword!123"  
}

### Register Student

Method: POST

Path: /api/auth/register/student

Required Roles / JWT: Admin JWT

Purpose: Create a new student and place them into a Class and Session.

Sample Request Body:

{  
 "panNumber": "ABC135",  
 "name": "Rahul Kumar",  
 "photo": "rahul\_kumar.jpg",  
 "password": "securePassword123",  
 "feeStatus": "PAID",  
 "feeCatalogStatus": "UP\_TO\_DATE",  
 "parentName": "Rajesh Kumar",  
 "mobileNumber": "9876543210",  
 "dateOfBirth": "2008-05-15",  
 "gender": "MALE",  
 "address": "123 Main Street, New Delhi, India",  
 "emergencyContact": "9876543211",  
 "bloodGroup": "B+",  
 "admissionDate": "2020-04-01",  
 "previousSchool": "Delhi Public School",  
 "classId": 1,  
 "sessionId": 1  
}

### Login Student

Method: POST

Path: /api/auth/student/login

Required Roles / JWT: No JWT required

Purpose: Authenticate a student by PAN and password. Returns JWT.

Sample Request Body:

{  
 "panNumber": "ABC133",  
 "password": "securePassword123"  
}

# Session Management

### Create Session

Method: POST

Path: /api/sessions

Required Roles / JWT: Admin JWT

Purpose: Create an academic session (target 12 months). Only one active session should exist. Deactivate any current active session before activating a new one.

Sample Request Body:

{  
 "name": "2026-27",  
 "startDate": "2026-04-01",  
 "endDate": "2027-03-31"  
}

Notes: Use 'Deactivate Session' to ensure only one active session.

### Get All Sessions

Method: GET

Path: /api/sessions

Required Roles / JWT: Admin JWT

Purpose: List all sessions.

### Get Session by ID

Method: GET

Path: /api/sessions/{id}

Required Roles / JWT: Admin JWT

Purpose: Fetch a session by its ID.

### Update Session

Method: PUT

Path: /api/sessions/{id}

Required Roles / JWT: Admin JWT

Purpose: Update a session’s metadata or active status.

Sample Request Body:

{  
 "name": "2024-25 Revised",  
 "startDate": "2024-04-05",  
 "endDate": "2025-03-30",  
 "active": false  
}

### Deactivate Current Active Session

Method: PUT

Path: /api/sessions/deactivate

Required Roles / JWT: Admin JWT

Purpose: Deactivate the current active session so you can activate/create another.

# Class Management

### Add New Class With SessionId

Method: POST

Path: /api/classes

Required Roles / JWT: Admin JWT

Purpose: Create a class under a Session, including the standard fee amount for that class.

Sample Request Body:

{  
 "className": "8V",  
 "sessionId": 1,  
 "feesAmount": 7000  
}

### All Classes with students in active session

Method: GET

Path: /api/classes

Required Roles / JWT: Admin JWT

Purpose: List classes (and students) in the active session.

### Get Class By Id And SessionId

Method: GET

Path: /api/classes/{classId}/session/{sessionId}

Required Roles / JWT: Admin JWT

Purpose: Fetch a class by ID within a session.

### Update Class By ClassId

Method: PATCH

Path: /api/classes/{classId}

Required Roles / JWT: Admin JWT

Purpose: Rename class and adjust fee amount.

Sample Request Body:

{  
 "className": "9C",  
 "sessionId": 1,  
 "feesAmount": 10000  
}

### Delete Class By ClassId and SessionId

Method: DELETE

Path: /api/classes/{classId}/session/{sessionId}

Required Roles / JWT: Admin JWT

Purpose: Delete a class within a specific session.

# Student APIs

### Get All Students

Method: GET

Path: /api/students

Required Roles / JWT: Admin JWT

Purpose: List all students.

### Get Student By PAN

Method: GET

Path: /api/students/{pan}

Required Roles / JWT: Admin or Teacher JWT

Purpose: Fetch a specific student by PAN.

### Get Current Student (/me)

Method: GET

Path: /api/students/me

Required Roles / JWT: Student JWT

Purpose: Get the authenticated student’s profile.

### Get Active Students

Method: GET

Path: /api/students/active

Required Roles / JWT: Admin or Teacher JWT

Purpose: List active students.

### Get Active Students by classId

Method: GET

Path: /api/students/class/{classId}

Required Roles / JWT: Admin or Teacher JWT

Purpose: List active students within a class.

### Mark Student Graduate Or Inactive

Method: PUT

Path: /api/students/status

Required Roles / JWT: Admin JWT

Purpose: Bulk update status (e.g., GRADUATED or INACTIVE) for PAN list.

Sample Request Body:

{  
 "panNumbers": ["ABC129", "ABC130"],  
 "status": "GRADUATED"  
}

### Update Student Details

Method: PUT

Path: /api/students/{pan}

Required Roles / JWT: Admin JWT (student self‑service may be restricted)

Purpose: Update a student’s personal and academic details.

Sample Request Body:

{  
 "name": "John Doe",  
 "photo": "base64EncodedPhotoStringOrUrl",  
 "classRollNumber": 25,  
 "parentName": "Jane Doe",  
 "mobileNumber": "+1234567890",  
 "dateOfBirth": "2005-08-20",  
 "gender": "MALE",  
 "address": "123 Main Street, City, Country",  
 "emergencyContact": "+0987654321",  
 "bloodGroup": "O+",  
 "previousSchool": "Sunshine High School"  
}

### Get Student's Fee Catalog (self)

Method: GET

Path: /api/fees/catalogs/me

Required Roles / JWT: Student JWT

Purpose: View the authenticated student’s fee catalog.

### Get Student's Present Today By Class

Method: GET

Path: /api/students/present-today/{classId}

Required Roles / JWT: Admin or Teacher JWT

Purpose: List students present today for a class.

# Teacher APIs

### Get Teacher By ID

Method: GET

Path: /api/teachers/{id}

Required Roles / JWT: Admin JWT

Purpose: Fetch a teacher by ID.

### Get ALL Teachers

Method: GET

Path: /api/teachers

Required Roles / JWT: Admin JWT

Purpose: List all teachers.

### Update/Deactivate Teacher

Method: PUT

Path: /api/teachers/{id}

Required Roles / JWT: Admin JWT

Purpose: Update teacher details; set status ACTIVE/INACTIVE.

Sample Request Body:

{  
 "name": "John Doe",  
 "email": "john.doe@example.com",  
 "qualification": "M.Sc. Mathematics",  
 "status": "INACTIVE"  
}

### Get Active Teachers

Method: GET

Path: /api/teachers/active

Required Roles / JWT: Admin JWT

Purpose: List active teachers.

### Get Current Teacher (/me)

Method: GET

Path: /api/teachers/me

Required Roles / JWT: Teacher JWT

Purpose: Get the authenticated teacher’s profile.

# Non‑Teaching Staff (Fee Staff) APIs

### Get Active Staff

Method: GET

Path: /api/feestaff/active

Required Roles / JWT: Admin JWT

Purpose: List active non‑teaching fee staff.

### Get All Fee Staff

Method: GET

Path: /api/feestaff

Required Roles / JWT: Admin JWT

Purpose: List all fee staff.

### Get Fee Staff By ID

Method: GET

Path: /api/feestaff/{id}

Required Roles / JWT: Admin JWT

Purpose: Fetch a fee staff member by ID.

### Deactivate Fee Staff

Method: PUT

Path: /api/feestaff/{id}

Required Roles / JWT: Admin JWT

Purpose: Deactivate a fee staff member.

# Generic User Management

### Change Password of Any User

Method: PUT

Path: /api/users/{userId}/change-password

Required Roles / JWT: Admin JWT

Purpose: Reset a user’s password (admin‑initiated).

Sample Request Body:

{  
 "password": "this is temporary password"  
}

### Update User By UserId

Method: PUT

Path: /api/users/{userId}/update

Required Roles / JWT: Admin JWT (or self if permitted by backend rules)

Purpose: Update generic user fields.

Sample Request Body:

{  
 "name": "teacher updated not admin update",  
 "qualification": "Bachelor's Degree",  
 "salaryGrade": "Grade A",  
 "designation": "Software Engineer",  
 "contactNumber": "+1234567890"  
}

### Delete User

Method: DELETE

Path: /api/users/{userId}

Required Roles / JWT: Admin JWT

Purpose: Delete/deactivate a user.

# Subjects APIs

### Create & Assign Subject to Teacher

Method: POST

Path: /api/subjects

Required Roles / JWT: Admin JWT

Purpose: Create a subject for a class and assign a teacher.

Sample Request Body:

{  
 "classId": 4,  
 "subjectName": "Science",  
 "teacherId": 3  
}

### Create & Assign Multiple Subjects

Method: POST

Path: /api/subjects/multiple

Required Roles / JWT: Admin JWT

Purpose: Bulk create/assign subjects for a class.

Sample Request Body:

{  
 "classId": 3,  
 "subjects": [  
 {"subjectName": "Physics", "teacherId": 5},  
 {"subjectName": "Chemistry", "teacherId": 4},  
 {"subjectName": "Maths", "teacherId": 3}  
 ]  
}

### All Subjects

Method: GET

Path: /api/subjects

Required Roles / JWT: Admin or Teacher JWT

Purpose: List all subjects.

### Get Subject by ID

Method: GET

Path: /api/subjects/{id}

Required Roles / JWT: Admin or Teacher JWT

Purpose: Fetch a subject by ID.

### Get Subjects by ClassId

Method: GET

Path: /api/subjects/class/{classId}

Required Roles / JWT: Admin or Teacher JWT

Purpose: List subjects for a class.

### Update Subject by Id

Method: PUT

Path: /api/subjects/{id}

Required Roles / JWT: Admin JWT

Purpose: Update subject’s class/teacher assignment and name.

Sample Request Body:

{  
 "classId": 1,  
 "subjectName": "Changed Subject",  
 "teacherId": 2  
}

### Delete Subject by ClassId

Method: DELETE

Path: /api/subjects/subject/{subjectId}/class/{classId}

Required Roles / JWT: Admin JWT

Purpose: Delete a subject for a class.

# Fees APIs

### Pay Fees

Method: PUT

Path: /api/fees/pay

Required Roles / JWT: Admin or Non‑Teaching Staff JWT

Purpose: Record a fee payment for a student.

Sample Request Body:

{  
 "studentPanNumber": "ABC133",  
 "amount": 7000,  
 "month": "AUGUST",  
 "receiptNumber": "RCPT20250820XYZ"  
}

### Fee Catalog (Admin view)

Method: GET

Path: /api/fees/catalogs

Required Roles / JWT: Admin or Non‑Teaching Staff JWT

Purpose: Fetch fee catalogs (optionally by student).

### Fee Catalog by Student PAN

Method: GET

Path: /api/fees/catalogs/{pan}

Required Roles / JWT: Admin or Non‑Teaching Staff JWT

Purpose: Get fee catalog for a specific student by PAN.

### Update Overdue Fees

Method: PUT

Path: /api/fees/update-overdue

Required Roles / JWT: Admin JWT

Purpose: Trigger overdue fees recalculation/update.

# Attendance APIs

### Mark Attendance (Today)

Method: POST

Path: /api/attendance

Required Roles / JWT: Teacher or Admin JWT

Purpose: Mark attendance for today for a class.

Sample Request Body:

{  
 "classId": 1,  
 "studentAttendances": [  
 {"panNumber": "ABC142", "present": true},  
 {"panNumber": "ABC141", "present": false},  
 {"panNumber": "ABC140", "present": true}  
 ]  
}

### Update Attendance For Specific Date

Method: PUT

Path: /api/attendance/{YYYY-MM-DD}

Required Roles / JWT: Teacher or Admin JWT

Purpose: Update attendance for a given date.

Sample Request Body:

{  
 "sessionId": 1,  
 "studentAttendances": [  
 {"panNumber": "ABC141", "present": true},  
 {"panNumber": "ABC142", "present": false}  
 ]  
}

### Get Attendance By Pan and SessionId

Method: GET

Path: /api/attendance/student/{pan}/session/{sessionId}?month={MONTH}

Required Roles / JWT: Admin or Teacher JWT

Purpose: Fetch a student’s attendance for a month.

### Get Attendance of Current Student (/me)

Method: GET

Path: /api/attendance/me?month={MONTH}

Required Roles / JWT: Student JWT

Purpose: Authenticated student retrieves their attendance for a month.

### Get Attendance By classId and sessionId

Method: GET

Path: /api/attendance/class/{classId}/session/{sessionId}?month={MONTH}

Required Roles / JWT: Admin or Teacher JWT

Purpose: Fetch class attendance for a given month.

# Timetable APIs

### Create Timetable

Method: POST

Path: /api/timetables

Required Roles / JWT: Admin JWT

Purpose: Create a timetable item (class, subject, day, start/end time).

Sample Request Body:

{  
 "classId": 1,  
 "subjectId": 2,  
 "day": "TUESDAY",  
 "startTime": "10:00",  
 "endTime": "11:00"  
}

### Get Timetable by Teacher ID

Method: GET

Path: /api/timetables/teacher/{teacherId}

Required Roles / JWT: Admin or Teacher JWT

Purpose: Fetch timetable entries for a teacher.

### Get Timetable by Class and Day

Method: GET

Path: /api/timetables/class/{classId}/day/{DAY}

Required Roles / JWT: Admin or Teacher JWT

Purpose: Fetch timetable entries for a class/day.

### Update Timetable

Method: PUT

Path: /api/timetables/{id}

Required Roles / JWT: Admin JWT

Purpose: Update a timetable entry.

Sample Request Body:

{  
 "classId": 1,  
 "subjectId": 3,  
 "day": "TUESDAY",  
 "startTime": "15:00",  
 "endTime": "18:00"  
}

### Get Timetable of Current Student (/student/me)

Method: GET

Path: /api/timetables/student/me

Required Roles / JWT: Student JWT

Purpose: Authenticated student retrieves their timetable.

### Get Timetable of Current Teacher (/teacher/me)

Method: GET

Path: /api/timetables/teacher/me

Required Roles / JWT: Teacher JWT

Purpose: Authenticated teacher retrieves their timetable.

# Transfer Certificate (TC) Requests

### Raise TC Request (Student)

Method: POST

Path: /api/tc/request

Required Roles / JWT: Student JWT

Purpose: Student raises a TC request with a reason.

Sample Request Body:

{  
 "reason": "Leaving school due to relocation"  
}

### Approve/Reject TC Request (Admin)

Method: PUT

Path: /api/tc/process/{tcRequestId}

Required Roles / JWT: Admin JWT

Purpose: Admin processes a TC request with a decision and reply.

Sample Request Body:

{  
 "adminReply": "All documents verified, approved.",  
 "decision": "APPROVED"  
}

### Get TC Requests by Student PAN (Admin)

Method: GET

Path: /api/tc/requests/{pan}

Required Roles / JWT: Admin JWT

Purpose: Admin lists TC requests for a student.

### Get All TC Requests of Current Student (/me)

Method: GET

Path: /api/tc/requests/me

Required Roles / JWT: Student JWT

Purpose: Student lists their own TC requests.

### List All TC Requests (filter by status)

Method: GET

Path: /api/tc/requests?status={APPROVED|PENDING|REJECTED}

Required Roles / JWT: Admin JWT

Purpose: Admin lists TC requests optionally filtered by status.

# Role Access Matrix (summary)

• Admin JWT: sessions, classes, students (create/update/status), teachers (list/update), fee staff, subjects (create/update/delete), timetables (create/update), attendance (view/update), fees (pay/update/lookup), users (update/delete/password), TC processing/listing.

• Teacher JWT: /me endpoints for teachers, attendance marking/updating (if permitted), view students in classes, timetables (own), attendance/class views.

• Student JWT: /me endpoints for students, attendance/me, fees/catalogs/me, timetables/student/me, TC request/create and list.

• Non‑Teaching Staff JWT: fees/pay, fees/catalogs, and other fee‑related lookups where permitted.

# Appendix: Endpoint counts by group

(unnamed): 66 endpoints