Kaushaly Home Learning API Documentation

Overview

This document provides comprehensive information about the Kaushaly Home Learning platform API endpoints, authentication, request/response formats, and usage examples.

Base URL

- **Production**: `https://www.kaushaly.in/api`
- **Development**: `http://localhost:3000/api`

Authentication

All API endpoints (except login and signup) require authentication using HTTP-only cookies for session management and Role-Based Access Control (RBAC).

Headers Required

```
Content-Type: application/json
```

Note: No additional authentication headers are required as authentication is handled via secure HTTP-only cookies.

Session Expiration

- Session cookies expire after 15 days of inactivity

Rate Limiting

- **General endpoints**: 300 requests per hour per IP
- **Authentication endpoints**: 10 requests per minute per IP

- **File upload endpoints**: 20 requests per minute per IP

Error Handling

All errors follow a consistent format:

```
{
  "error": "ERROR_CODE",
  "message": "Human readable error message",
  "code": 400,
  "details": {
    "field": "validation error details"
  }
}
```

Common Error Codes

- `400` - Bad Request (validation errors)
- `401` - Unauthorized (invalid/missing token)
- `403` - Forbidden (insufficient permissions)
- `404` - Not Found
- `429` - Too Many Requests (rate limit exceeded)
- `500` - Internal Server Error

Authentication Endpoints

POST /auth/login

Authenticate user with email and password.

Request Body:

```
{
  "email": "user@example.com",
  "password": "password123"
}
```

Response (200):

```
{
  "message": "Login successful",
  "user": {
    "id": 1,
    "email": "user@example.com",
    "role": "student",
    "firstName": "John",
    "lastName": "Doe"
  }
}
```

Note: Authentication cookies are automatically set in the response headers.

POST /auth/signup

Register a new user account (initial signup).

Request Body:

```
{
  "email": "newuser@example.com",
  "firstName": "Jane",
  "lastName": "Smith",
  "role": "student"
}
```

Response (201):

```
{
  "message": "Registration successful. Please verify your email.",
  "userId": 123
}
```

Note: Email verification token is sent to the provided email address.

POST /auth/register/student

Complete student registration with additional profile information after verification.

Request Body:

```
{
  "email": "newuser@example.com",
  "firstName": "Jane",
  "lastName": "Smith",
  "role": "student",
  "grade": "10th",
  "schoolName": "ABC School",
  "parentName": "Jane Doe",
  "parentPhone": "+91-9876543210",
  "subjectsInterested": ["Mathematics", "Science"],
  "monthlyFee": 5000,
  "paymentStatus": "pending",
  "enrollmentDate": "2024-01-15"
}
```

Response (201):

```
{
  "message": "Student registration successful.",
  "studentId": 123
}
```

POST /auth/register/teacher

Complete teacher registration with qualification details.

Request Body:

```
{
  "email": "newuser@example.com",
  "firstName": "Jane",
  "lastName": "Smith",
  "role": "teacher",
  "qualification": "M.Sc Mathematics",
  "experienceYears": 5,
  "subjectsTaught": ["Mathematics", "Physics"],
  "teachingMode": "both",
  "documents": {
    "marksheet": "file_url_or_id",
    "govtId": "file_url_or_id"
  }
}
```

Response (201):

```
{
  "message": "Teacher registration successful. Pending approval.",
  "teacherId": 456,
  "approvalStatus": "pending"
}
```

POST /auth/logout

Logout user and clear authentication cookies.

Response (200):

```
{
  "message": "Logout successful"
}
```

POST /auth/verify-email

Verify user email with verification token.

Request Body:

```
{
  "token": "verification_token_from_email"
}
```

Response (200):

```
{
  "message": "Email verified successfully"
}
```

User Management

GET /students

Get list of students (Admin only).

Query Parameters:

- `page` (integer, default: 1) - Page number
- `limit` (integer, default: 15) - Items per page
- `search` (string) - Search by name or email
- `city` (string) - Filter by city
- `paymentStatus` (string) - Filter by payment status

Response (200):

```
{
  "students": [
    {
      "id": 1,
      "email": "student@example.com",
      "firstName": "John",
      "lastName": "Doe",
      "grade": "10th",
      "schoolName": "ABC School",
      "parentName": "Jane Doe",
      "parentPhone": "+91-9876543210",
      "subjectsInterested": ["Mathematics", "Science"],
      "monthlyFee": 5000,
      "paymentStatus": "paid",
      "enrollmentDate": "2024-01-15"
    }
  ],
  "total": 150,
  "page": 1,
  "totalPages": 15
}
```


GET /teachers

Get list of teachers.

Query Parameters:

- `page` (integer) - Page number
- `limit` (integer) - Items per page
- `subject` (string) - Filter by subject taught
- `city` (string) - Filter by city
- `approvalStatus` (string) - Filter by approval status

Response (200):

```
{
  "teachers": [
    {
      "id": 1,
      "email": "teacher@example.com",
      "firstName": "Alice",
      "lastName": "Johnson",
      "qualification": "M.Sc Mathematics",
      "experienceYears": 5,
      "subjectsTaught": ["Mathematics", "Physics"],
      "teachingMode": "both",
      "hourlyRate": 500,
      "approvalStatus": "approved",
      "rating": 4.8,
      "totalReviews": 25,
      "currentStudents": 15,
      "maxStudents": 20
    }
  ],
  "total": 45,
  "page": 1,
  "totalPages": 5
}
```


Assignment Management

POST /assignments

Create a new assignment (Teacher only).

Request Body:

```
{
  "studentId": 123,
  "title": "Algebra Practice Problems",
  "description": "Complete exercises 1-20 from chapter 5",
  "subject": "Mathematics",
  "dueDate": "2024-02-15",
  "assignmentType": "homework",
  "maxMarks": 100,
  "instructions": "Show all working steps clearly"
}
```

Response (201):

```
{
  "id": 456,
  "teacherId": 789,
  "studentId": 123,
  "title": "Algebra Practice Problems",
  "description": "Complete exercises 1-20 from chapter 5",
  "subject": "Mathematics",
  "dueDate": "2024-02-15",
  "assignmentType": "homework",
  "maxMarks": 100,
  "status": "assigned",
  "createdAt": "2024-02-01T10:00:00Z"
}
```

POST /assignments/{id}/submit

Submit an assignment (Student only).

Request (multipart/form-data):

```
submissionText: "Here is my solution to the problems..."
file: [uploaded file]
```

Response (200):

```
{
  "message": "Assignment submitted successfully",
  "submissionId": 789,
  "submittedAt": "2024-02-14T15:30:00Z"
}
```


Attendance Management

POST /attendance

Mark attendance.

Request Body:


```
{
  "studentId": 123,
  "date": "2024-02-01",
  "status": "present",
  "notes": "Participated actively in class",
  "sessionDuration": 60,
  "subject": "Mathematics"
}
```


Response (201):


```
{
  "id": 456,
  "studentId": 123,
  "teacherId": 789,
  "date": "2024-02-01",
  "status": "present",
  "notes": "Participated actively in class",
  "sessionDuration": 60,
  "subject": "Mathematics",
  "markedAt": "2024-02-01T10:00:00Z"
}
```

GET /attendance

Get attendance records.

Query Parameters:

- `studentId` (integer) - Filter by student
- `teacherId` (integer) - Filter by teacher
- `startDate` (date) - Start date range
- `endDate` (date) - End date range
- `subject` (string) - Filter by subject

Payment Management

POST /payments

Process a payment.

Request Body:

```
{
  "studentId": 123,
  "amount": 5000,
  "paymentType": "monthly_fee",
  "paymentMethod": "upi",
  "notes": "February 2024 fee payment"
}
```

Response (201):

```
{
  "id": 789,
  "studentId": 123,
  "amount": 5000,
  "paymentType": "monthly_fee",
  "paymentMethod": "upi",
  "paymentStatus": "completed",
  "transactionId": "TXN123456789",
  "paymentDate": "2024-02-01",
  "dueDate": "2024-02-01"
}
```


Admin Analytics

GET /admin/analytics

Get comprehensive platform analytics (Admin only).

Response (200):

```
{
  "totalUsers": 500,
  "totalStudents": 350,
  "totalTeachers": 45,
  "approvedTeachers": 40,
  "pendingTeachers": 5,
  "totalRevenue": 1750000,
  "monthlyGrowth": 15.5,
  "yearlyGrowth": 180.2,
  "monthlyData": [
    {
      "month": "Jan",
      "students": 320,
      "teachers": 38,
      "revenue": 1600000,
      "expenses": 400000
    }
  ],
  "subjectDistribution": [
    {
      "subject": "Mathematics",
      "students": 180,
      "teachers": 15
    }
  ],
  "locationStats": [
    {
      "city": "Mumbai",
      "students": 120,
      "teachers": 18
    }
  ]
}
```


Notification System

POST /notifications

Send a notification.

Request Body:

```
{
  "userId": 123,
  "title": "Assignment Due Reminder",
  "message": "Your Mathematics assignment is due tomorrow",
  "type": "assignment",
  "priority": "medium",
  "deliveryMethod": "email",
  "scheduledAt": "2024-02-14T09:00:00Z"
}
```

GET /notifications

Get user notifications.

Query Parameters:

- `isRead` (boolean) - Filter by read status
- `type` (string) - Filter by notification type
- `priority` (string) - Filter by priority level

File Upload

POST /upload

Upload files (assignments, documents, profile images).

Request (multipart/form-data):

```
file: [uploaded file]
type: "assignment" | "profile" | "document"
```

Response (200):

```
{
  "url": "https://storage.kaushaly.com/files/abc123.pdf",
  "filename": "assignment.pdf",
  "size": 1024000,
  "type": "application/pdf"
}
```

Webhook Events

The platform sends webhook events for important actions:

Payment Completed

```
{
  "event": "payment.completed",
  "data": {
    "paymentId": 789,
    "studentId": 123,
    "amount": 5000,
    "paymentDate": "2024-02-01T10:00:00Z"
  },
  "timestamp": "2024-02-01T10:00:00Z"
}
```

Teacher Approved

```
{
  "event": "teacher.approved",
  "data": {
    "teacherId": 456,
    "approvedBy": 1,
    "approvedAt": "2024-02-01T10:00:00Z"
  },
  "timestamp": "2024-02-01T10:00:00Z"
}
```

SDK Examples

JavaScript/Node.js

```
const KaushalyAPI = require('@kaushaly/api-client');

const client = new KaushalyAPI({
  baseURL: 'https://api.kaushaly.com/v1',
  apiKey: 'your_api_key'
});

// Get students
const students = await client.students.list({
  page: 1,
  limit: 10,
  search: 'john'
});

// Create assignment
const assignment = await client.assignments.create({
  studentId: 123,
  title: 'Math Homework',
  subject: 'Mathematics',
  dueDate: '2024-02-15'
});
```

Python

```python
from kaushaly_api import KaushalyClient

client = KaushalyClient(
    base_url='https://api.kaushaly.com/v1',
    api_key='your_api_key'
)

# Get teachers
teachers = client.teachers.list(
    subject='Mathematics',
    city='Mumbai'
)

# Mark attendance
attendance = client.attendance.create(
    student_id=123,
    date='2024-02-01',
    status='present'
)
```

Testing

Test Environment

- **Base URL**: `https://api-test.kaushaly.com/v1`
- **Test API Key**: Contact support for test credentials

Sample Test Data

The test environment includes sample data:

- 50 test students
- 10 test teachers
- Sample assignments and attendance records
- Mock payment transactions

For more information or support, contact: support@kaushaly.com