MXEduLearners API Documentation

Table of Contents

1. [Introduction](introduction)

2. [Authentication](authentication)

3. [Base URL](base-url)

4. [Endpoints](endpoints)

- [User Management](user-management)

- [Course Management](course-management)

- [Tutor Management](tutor-management)

- [Admin Management](admin-management)

- [Quiz Management](quiz-management)

- [Payment Management](payment-management)

5. [Error Handling](error-handling)

## Introduction

This document provides detailed information about the MXEduLearners API, which offers functionality for user management, course creation and consumption, tutor-student interactions, and administrative controls.

## Authentication

- JWT (JSON Web Token): Used for user authentication and authorization.

- Google Authentication: Supported for user sign-in.

## Base URL

All API requests should be made to the following base URL:

<http://mxedulearners/api/v1>

**## Endpoints**

### User Management

#### User Registration

| Method | Endpoint | Description |

|--------|----------|-------------|

| POST | /auth/register | Registers a new user |

Request Body:

```json

{

"username": "string",

"email": "string",

"password": "string",

"role": "string" // "student" or "tutor"

}

Response:

{

"userId": "string",

"username": "string",

"email": "string",

"role": "string"

}

#### **User Login**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /auth/login | Authenticates a user |

Request Body:

{

"email": "string",

"password": "string"

}

Response:

{

"token": "string",

"userId": "string",

"role": "string"

}

#### **Google Sign-In**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /auth/google-signin | Authenticates a user using Google |

Request Body:

{

"googleToken": "string"

}

Response:

{

"token": "string",

"userId": "string",

"role": "string"

}

#### **Get User Profile**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /users/profile | Retrieves the authenticated user’s profile |

Headers:

1. Authorization: Bearer {token}

Response:

{

"userId": "string",

"username": "string",

"email": "string",

"role": "string",

"profilePicture": "string",

"educationalQualification": "string", // for tutors

"experience": "string" // for tutors

}

#### **Update User Profile**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| PUT | /users/profile | Updates the authenticated user’s profile |

Headers:

1. Authorization: Bearer {token}

Request Body:

{

"username": "string",

"email": "string",

"password": "string",

"profilePicture": "string",

"educationalQualification": "string", // for tutors

"experience": "string" // for tutors

}

Response:

{

"userId": "string",

"username": "string",

"email": "string",

"role": "string",

"profilePicture": "string",

"educationalQualification": "string", // for tutors

"experience": "string" // for tutors

}

### **Course Management**

#### **Create Course (Tutor only)**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /courses | Creates a new course |

Headers:

1. Authorization: Bearer {token}

Request Body:

{

"title": "string",

"description": "string",

"price": "number"

}

Response:

{

"courseId": "string",

"title": "string",

"description": "string",

"price": "number",

"tutorId": "string"

}

#### **Get Course Details**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /courses/{courseId} | Retrieves details of a specific course |

Response:

{

"courseId": "string",

"title": "string",

"description": "string",

"price": "number",

"tutorId": "string",

"lessons": [

{

"lessonId": "string",

"title": "string",

"videoUrl": "string"

}

],

"quizzes": [

{

"quizId": "string",

"title": "string"

}

]

}

#### **Add Lesson to Course (Tutor only)**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /courses/{courseId}/lessons | Adds a new lesson to a course |

Headers:

1. Authorization: Bearer {token}

Request Body:

{

"title": "string",

"videoUrl": "string",

"pdfNotes": "string" // URL to PDF file

}

Response:

{

"lessonId": "string",

"title": "string",

"videoUrl": "string",

"pdfNotes": "string"

}

#### **Enroll in Course (Student only)**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /courses/{courseId}/enroll | Enrolls a student in a course |

Headers:

1. Authorization: Bearer {token}

Response:

{

"enrollmentId": "string",

"courseId": "string",

"userId": "string",

"enrollmentDate": "string"

}

### **Tutor Management**

#### **Get Tutor’s Courses**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /tutors/courses | Retrieves all courses created by the authenticated tutor |

Headers:

1. Authorization: Bearer {token}

Response:

[

{

"courseId": "string",

"title": "string",

"description": "string",

"price": "number",

"enrollmentCount": "number"

}

]

#### **Get Tutor’s Ledger**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /tutors/ledger | Retrieves the tutor’s income ledger |

Headers:

1. Authorization: Bearer {token}

Response:

{

"totalIncome": "number",

"transactions": [

{

"transactionId": "string",

"courseId": "string",

"amount": "number",

"date": "string"

}

]

}

### **Admin Management**

#### **Get All Users (Admin only)**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /admin/users | Retrieves all users |

Headers:

1. Authorization: Bearer {token}

Response:

[

{

"userId": "string",

"username": "string",

"email": "string",

"role": "string"

}

]

#### **Block/Unblock User (Admin only)**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| PUT | /admin/users/{userId}/toggle-block | Blocks or unblocks a user |

Headers:

1. Authorization: Bearer {token}

Response:

1. {
2. "userId": "string",
3. "username": "string",
4. "isBlocked": "boolean"
5. }

#### **Get Admin Ledger (Admin only)**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /admin/ledger | Retrieves the admin’s profit ledger |

Headers:

1. Authorization: Bearer {token}

Response:

1. {
2. "totalProfit": "number",
3. "transactions": [
4. {
5. "transactionId": "string",
6. "courseId": "string",
7. "tutorId": "string",
8. "amount": "number",
9. "date": "string"
10. }
11. ]
12. }

### **Quiz Management**

#### **Create Quiz (Tutor only)**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /courses/{courseId}/quizzes | Creates a new quiz for a course |

Headers:

1. Authorization: Bearer {token}

Request Body:

1. {
2. "title": "string",
3. "timeLimit": "number",
4. "questions": [
5. {
6. "question": "string",
7. "options": ["string", "string", "string", "string"],
8. "correctAnswer": "number",
9. "score": "number"
10. }
11. ]
12. }

Response:

1. {
2. "quizId": "string",
3. "title": "string",
4. "timeLimit": "number",
5. "questionCount": "number"
6. }

#### **Submit Quiz (Student only)**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /quizzes/{quizId}/submit | Submits a completed quiz |

Headers:

1. Authorization: Bearer {token}

Request Body:

1. {
2. "answers": [
3. {
4. "questionId": "string",
5. "selectedOption": "number"
6. }
7. ]
8. }

Response:

1. {
2. "quizId": "string",
3. "score": "number",
4. "totalScore": "number",
5. "passed": "boolean"
6. }

### **Payment Management**

#### **Process Payment**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /payments/process | Processes a payment for course enrollment |

Headers:

1. Authorization: Bearer {token}

Request Body:

1. {
2. "courseId": "string",
3. "amount": "number",
4. "paymentMethod": "string"
5. }

Response:

1. {
2. "transactionId": "string",
3. "status": "string",
4. "amount": "number"
5. }

## **Error Handling**

### **Error Response Format**

All errors follow this format:

1. {
2. "error": {
3. "code": "integer",
4. "message": "string"
5. }
6. }

### **Common Error Codes**

| **Code** | **Description** |
| --- | --- |
| 400 | Bad Request |
| 401 | Unauthorized |
| 403 | Forbidden |
| 404 | Not Found |
| 500 | Internal Server Error |

```