

Toodle API Contract

Tutor Management System — REST API Reference

Base URL: http://localhost:3000/api/v1	Version: 1.0.0	Auth: Bearer JWT (Auth0 RS256)	Format: JSON
--	--------------------------	---------------------------------------	------------------------

CONTENTS

- [1. Authentication & Conventions](#)
- [2. Error Responses](#)
- [3. Auth Endpoints \(/auth\)](#)
- [4. User Endpoints \(/users\)](#)
- [5. Course Endpoints \(/courses\)](#)
- [6. Tutor Endpoints \(/tutors\)](#)
- [7. Allocation Endpoints \(/allocations\)](#)
- [8. Health Check](#)

1. Authentication & Conventions

Every endpoint (except `GET /health`) requires a valid Auth0 JWT in the **Authorization** header.

AUTHORIZATION HEADER

Authorization: Bearer <access_token>

ROLES

ROLE	DESCRIPTION	ACCESS LEVEL
ORGANISER	Course organiser / staff member	Full access — manage courses, tutors, allocations, users
TUTOR	Teaching assistant	Read courses, manage own profile & availability
STUDENT	Enrolled learner	Read-only access to courses

STANDARD RESPONSE ENVELOPE

```
{
  "success": true,
  "data": { ... }           // 200 / 201
}

{
  "success": false,
  "error": "Message",       // 4xx / 5xx
  "details": [ ... ]       // optional – validation errors (422)
}
```

COMMON CONVENTIONS

CONVENTION	DETAIL
IDs	UUID v4 strings
Timestamps	ISO 8601 — <code>createdAt</code> , <code>updatedAt</code>
Times	<code>HH:MM</code> 24-hour format (e.g. <code>"09:00"</code>)

CONVENTION	DETAIL
No-content deletes	HTTP 204 — empty body
Content-Type	<code>application/json</code>
Timeout	10 000 ms (client-side)

2. Error Responses

STATUS	TYPE	DESCRIPTION
400	BadRequestError	Malformed request or missing required data
401	UnauthorisedError	Missing or invalid JWT token
403	ForbiddenError	Authenticated but insufficient role permissions
404	NotFoundError	Resource does not exist
409	ConflictError	Duplicate resource (e.g. same tutor allocated twice to same course)
422	ValidationError	Zod schema validation failure — includes <code>details[]</code> array
500	AppError	Unexpected server error

VALIDATION ERROR EXAMPLE (422)

```
{
  "success": false,
  "error": "Validation failed",
  "details": [
    { "field": "mark", "message": "Number must be less than or equal to 100" }
  ]
}
```

3. Auth Endpoints

Base path: `/api/v1/auth` — All require JWT authentication.

POST `/api/v1/auth/callback`

Sync Auth0 user to database after login

JWT Required

DESCRIPTION

Called immediately after Auth0 login. Upserts the user record from JWT claims into the local database (auth0Id, email, name, role).

REQUEST BODY

None — user data is extracted from the validated JWT payload.

SUCCESS RESPONSE — 200

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "auth0Id": "auth0|...",
    "email": "user@example.com",
    "name": "Alice Smith",
    "role": "TUTOR",
    "studentNumber": null,
    "maxHoursPerWeek": 10,
    "createdAt": "2026-08-22T10:00:00.000Z",
    "updatedAt": "2026-08-22T10:00:00.000Z"
  }
}
```

GET `/api/v1/auth/me`

Get current authenticated user's profile

JWT Required

SUCCESS RESPONSE — 200

```
{
  "success": true,
  "data": { /* User object – same shape as POST /auth/callback */ }
}
```

ERRORS

STATUS	CONDITION
404	User not yet synced — call POST /auth/callback first

DELETE `/api/v1/auth/me`

Delete current user's account

JWT Required

SUCCESS RESPONSE

HTTP 204 — No content

4. User Endpoints

Base path: `/api/v1/users`

GET `/api/v1/users`

List all users

JWT Required

ORGANISER only

QUERY PARAMETERS

PARAMETER	TYPE	REQUIRED	DESCRIPTION
<code>role</code>	string	optional	Filter by role: <code>ORGANISER</code> <code>TUTOR</code> <code>STUDENT</code>

SUCCESS RESPONSE — 200

```
{ "success": true, "data": [ /* User[] */ ] }
```

GET `/api/v1/users/:id`

Get user by ID

JWT Required

PATH PARAMETERS

PARAMETER	TYPE	DESCRIPTION
<code>id</code>	UUID	User ID

SUCCESS RESPONSE — 200

```
{ "success": true, "data": { /* User object */ } }
```

ERRORS

STATUS	CONDITION
404	User not found

PATCH `/api/v1/users/:id`

Update user profile

JWT Required

REQUEST BODY (ALL FIELDS OPTIONAL)

FIELD	TYPE	CONSTRAINTS
<code>name</code>	string	—
<code>studentNumber</code>	string	—
<code>maxHoursPerWeek</code>	integer	—

SUCCESS RESPONSE — 200

```
{ "success": true, "data": { /* Updated User */ } }
```

5. Course Endpoints

Base path: `/api/v1/courses`

GET `/api/v1/courses`

List all courses

JWT Required

QUERY PARAMETERS

PARAMETER	TYPE	REQUIRED	DESCRIPTION
<code>year</code>	integer	optional	Filter by year
<code>semester</code>	integer	optional	Filter by semester (1 or 2)

SUCCESS RESPONSE — 200

```
{ "success": true, "data": [ /* Course[] */ ] }
```

GET `/api/v1/courses/:id`

Get course detail with sessions and allocations

JWT Required

SUCCESS RESPONSE — 200

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "code": "COMS3011A",
    "name": "Operating Systems",
    "description": "...",
    "year": 2026,
    "semester": 1,
    "requiredTutors": 3,
    "minMarkRequired": 65,
    "sessions": [ /* CourseSession[] */ ],
    "allocations": [ /* Allocation[] */ ],
    "createdAt": "...",
    "updatedAt": "..."
  }
}
```

ERRORS

STATUS	CONDITION
404	Course not found

POST /api/v1/courses

Create a new course

JWT Required

ORGANISER only

Zod Validated

REQUEST BODY

FIELD	TYPE	REQUIRED	CONSTRAINTS
code	string	required	1–20 chars (e.g. "COMS3011A"), unique
name	string	required	1–255 chars
description	string	optional	–
year	integer	required	2020–2030
semester	integer	required	1 or 2
requiredTutors	integer	optional	≥ 0, default 1
minMarkRequired	integer	optional	0–100, default 50

SUCCESS RESPONSE — 201

```
{ "success": true, "data": { /* Created Course */ } }
```

ERRORS

STATUS	CONDITION
409	Course code already exists
422	Validation failure

PATCH /api/v1/courses/:id

Update a course

JWT Required

ORGANISER only

Zod Validated

REQUEST BODY — ALL FIELDS OPTIONAL (SAME FIELDS AS POST)**SUCCESS RESPONSE — 200**

```
{ "success": true, "data": { /* Updated Course */ } }
```

DELETE /api/v1/courses/:id

Delete a course (cascades sessions and allocations)

JWT Required

ORGANISER only

SUCCESS RESPONSE

HTTP 204 — No content

GET /api/v1/courses/:id/sessions

List sessions for a course

JWT Required

SUCCESS RESPONSE — 200

```
{
  "success": true,
  "data": [{
    "id": "uuid",
    "dayOfWeek": "MONDAY",
    "startTime": "09:00",
    "endTime": "11:00",
    "venue": "WSS1",
    "sessionType": "TUTORIAL",
    "courseId": "uuid"
  }]
}
```

POST /api/v1/courses/:id/sessions

Add a session to a course

JWT Required

ORGANISER only

REQUEST BODY

FIELD	TYPE	REQUIRED	CONSTRAINTS
dayOfWeek	string	required	MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY SUNDAY
startTime	string	required	HH:MM format
endTime	string	required	HH:MM format
venue	string	optional	—
sessionType	string	required	LECTURE TUTORIAL LAB

SUCCESS RESPONSE — 201

```
{ "success": true, "data": { /* CourseSession */ } }
```

6. Tutor Endpoints

Base path: `/api/v1/tutors`

GET `/api/v1/tutors`

List all tutors with marks and availability

JWT Required

ORGANISER only

QUERY PARAMETERS

PARAMETER	TYPE	REQUIRED	DESCRIPTION
<code>courseId</code>	UUID	optional	Filter tutors eligible for a specific course

SUCCESS RESPONSE — 200

```
{
  "success": true,
  "data": [{
    "id": "uuid",
    "name": "Alice Smith",
    "email": "alice@students.wits.ac.za",
    "role": "TUTOR",
    "studentNumber": "2001234",
    "maxHoursPerWeek": 10,
    "tutorMarks": [ { "courseId": "...", "mark": 78 } ],
    "availability": [ { "dayOfWeek": "MONDAY", "startTime": "09:00", "endTime": "13:00" } ]
  }]
}
```

GET `/api/v1/tutors/:id`

Get tutor detail

JWT Required

SUCCESS RESPONSE — 200

```
{ "success": true, "data": { /* Tutor with marks, availability, allocations */ } }
```

ERRORS

STATUS	CONDITION
404	Tutor not found

POST **/api/v1/tutors/:id/marks**

Add or update a tutor's mark for a course

JWT Required

ORGANISER only

Zod Validated

REQUEST BODY

FIELD	TYPE	REQUIRED	CONSTRAINTS
courseId	UUID	required	Valid course UUID
mark	integer	required	0–100

SUCCESS RESPONSE — 200 (UPSERT)

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "userId": "uuid",
    "courseId": "uuid",
    "mark": 78,
    "createdAt": "...",
    "updatedAt": "..."
  }
}
```

PUT **/api/v1/tutors/:id/availability**

Replace tutor's availability slots (full replace)

JWT Required

Zod Validated

REQUEST BODY

FIELD	TYPE	REQUIRED	CONSTRAINTS
slots	array	required	Array of availability slot objects (can be empty)
slots[].dayOfWeek	string	required	MONDAY–SUNDAY
slots[].startTime	string	required	HH:MM format
slots[].endTime	string	required	HH:MM format
slots[].isRecurring	boolean	optional	Default: true

SUCCESS RESPONSE — 200

```
{ "success": true, "data": [ /* Availability[] */ ] }
```

7. Allocation Endpoints

Base path: `/api/v1/allocations` — All require ORGANISER role.

GET

/api/v1/allocations

List all allocations

JWT Required

ORGANISER only

QUERY PARAMETERS

PARAMETER	TYPE	REQUIRED	DESCRIPTION
courseId	UUID	optional	Filter by course
userId	UUID	optional	Filter by tutor
status	string	optional	ACTIVE PENDING REMOVED

SUCCESS RESPONSE — 200

```
{ "success": true, "data": [ /* Allocation[] */ ] }
```

GET

/api/v1/allocations/validate

Dry-run allocation constraint check (no DB write)

JWT Required

ORGANISER only

QUERY PARAMETERS

PARAMETER	TYPE	REQUIRED	DESCRIPTION
userId	UUID	required	Tutor to validate
courseId	UUID	required	Course to assign
hoursPerWeek	integer	required	Proposed hours

SUCCESS RESPONSE — 200

```
{
  "success": true,
  "data": {
    "valid": false,
    "constraints": [
      { "type": "MARK_BELOW_THRESHOLD", "message": "Tutor mark (45) is below required
minimum (65)" },
      { "type": "TIMETABLE_CLASH",      "message": "Tutor is unavailable on MONDAY 09:00-
11:00" },
      { "type": "HOURS_EXCEEDED",      "message": "Would exceed 10 h/week cap (current: 8
+ proposed: 4)" }
    ]
  }
}
```

POST /api/v1/allocations

Assign a tutor to a course

JWT Required

ORGANISER only

Zod Validated

REQUEST BODY

FIELD	TYPE	REQUIRED	CONSTRAINTS
userId	UUID	required	Tutor's user ID
courseId	UUID	required	Course ID
hoursPerWeek	integer	required	1–40
reason	string	optional	Override reason (if bypassing a constraint)

SUCCESS RESPONSE — 201

```
{
  "success": true,
  "data": {
    "id": "uuid",
    "userId": "uuid",
    "courseId": "uuid",
    "hoursPerWeek": 6,
    "status": "PENDING",
    "reason": null,
    "assignedById": "uuid",
    "createdAt": "...",
    "updatedAt": "..."
  }
}
```

ERRORS

STATUS	CONDITION
409	Tutor already allocated to this course
422	Validation failure

PATCH /api/v1/allocations/:id

Update an allocation

JWT Required

ORGANISER only

Zod Validated

REQUEST BODY (ALL FIELDS OPTIONAL)

FIELD	TYPE	CONSTRAINTS
hoursPerWeek	integer	1–40
status	string	ACTIVE PENDING REMOVED
reason	string	–

SUCCESS RESPONSE — 200

```
{ "success": true, "data": { /* Updated Allocation */ } }
```

DELETE /api/v1/allocations/:id

Remove an allocation

JWT Required

ORGANISER only

SUCCESS RESPONSE

HTTP 204 — No content

8. Health Check

GET

/health

Server liveness check — no auth required

SUCCESS RESPONSE — 200

```
{
  "status": "ok",
  "timestamp": "2026-08-23T12:00:00.000Z"
}
```