Online Test Platform - REST API Documentation

Base URL: `/api/v1`

Table of Contents

Overview

This API enables schools to manage classes, teachers, students, and online tests. Teachers create tests for their classes; all students in the class can access and take the test. Student test responses are accessible to the student and the class teacher.

Entities

School

```
{
  "id": "sch001",
  "name": "Springfield High School"
}
```

Class

```
{
  "id": "cls101",
  "school_id": "sch001",
  "name": "Grade 10 - A",
  "teacher_ids": ["t001", "t002"],
  "student_ids": ["s101", "s102", "s103"]
}
```

Teacher

```
{
  "id": "t001",
  "name": "Ms. Smith",
  "school_id": "sch001",
  "class_ids": ["cls101"]
}
```

Student

```
{
  "id": "s101",
  "name": "Bart Simpson",
  "school_id": "sch001",
  "class_id": "cls101"
}
```

Question

```
{
  "id": "q123",
  "subject": "Mathematics",
  "topic": "Algebra",
  "question_text": "What is 2 + 2?",
  "type": "mcq",
  "choices": ["2", "3", "4", "5"]
}
```

Test

```
{
  "id": "t101",
  "name": "Algebra Basics Test",
  "question_ids": ["q123", "q124"],
  "due_date": "2025-07-20T23:59:59Z",
  "instructions": "Answer all questions.",
  "class_id": "cls101",
  "teacher_id": "t001"
}
```

Student Test Response

```
{
  "student_id": "s101",
  "test_id": "t101",
  "answers": [
    {
      "question_id": "q123",
      "answer": "4",
    },
   {
      "question_id": "q124",
      "answer": "3",
    }

  ],
}
```

Relationships

- **A School** has multiple Classes.
- **A Class** belongs to a School, has multiple Students and Teachers.
- **A Teacher** can teach multiple Classes, but is always part of a School.
- **A Student** belongs to one Class and one School.
- **A Test** is created by a Teacher for a Class; all Students in that Class can access the Test.
- **A Student Test Response** can be viewed by the Student and the Teacher(s) of the Class.

Security, Authentication, and Authorization

Authentication

- All endpoints require a valid JWT (JSON Web Token).
- Include the JWT in the `Authorization` header for every request.

Authorization

- **Students:** Can only access their own data and tests assigned to their class.
- **Teachers:** Can manage tests for their classes and view responses from their students.
- **Admins:** Full access to all resources.

Access Rules

- **Tests** are only visible to students in the class for which the test was created.
- **Student Responses** are only visible to the student and the teacher(s) of the class.

Security Best Practices

- All API traffic must use HTTPS.

- JWT tokens should have reasonable expiry.
- Input validation and logging are enforced.

API Endpoints Summary

Use Case	**HTTP Verb**	**Endpoint**
List all schools	GET	/schools
List classes in a school	GET	/schools/{school_id}/classes
List teachers in a class	GET	/classes/{class_id}/teachers
List students in a class	GET	/classes/{class_id}/students
List questions (filter by subject/topic)	GET	/questions?subject={subject}&topic={topic}
Create a test for a class	POST	/classes/{class_id}/tests
List all tests for a class	GET	/classes/{class_id}/tests
View questions in a test	GET	/tests/{test_id}/questions
Delete a test	DELETE	/tests/{test_id}
View student responses for a test (teacher/admin)	GET	/tests/{test_id}/students/{student_id}/responses
View my school and class (student)	GET	/students/me/profile
List tests assigned to my class (student)	GET	/students/me/tests
Get all questions for a test (student)	GET	/students/me/tests/{test_id}/questions
Submit answer to a question (student)	POST	/students/me/tests/{test_id}/questions/{question_id}/answer
Get a specific question in a test (student)	GET	/students/me/tests/{test_id}/questions/{question_id}

Use Case	HTTP Verb	Endpoint
View my response to a test (student)	GET	/students/me/tests/{test_id}/responses

- All endpoints require authentication and are subject to role-based access control.
- Endpoints with `/me/` are automatically scoped to the authenticated user.
- Teacher/admin endpoints for student responses are restricted to teachers of the class or admins.

Teacher Endpoints

1. Get List of Schools

GET `/schools`

Response:

```
[
  {
    "id": "sch001",
    "name": "Springfield High School"
  }
]
```

2. Get Classes for a School

GET `/schools/{school_id}/classes`

Response:

```
[
  {
    "id": "cls101",
    "name": "Grade 10 - A",
    "teacher_ids": ["t001"],
    "student_ids": ["s101", "s102"]
  }
]
```

3. Get List of Questions

GET `/questions?subject={subject}&topic={topic}`

Response: *As previously described*

4. Create a Test for a Class

POST `/classes/{class_id}/tests`

Request Body:

```
{
  "name": "Algebra Basics Test",
  "question_ids": ["q123", "q124"],
  "due_date": "2025-07-20T23:59:59Z",
  "instructions": "Answer all questions."
}
```

Response:

```
{
  "id": "t101",
  "status": "created"
}
```

5. View All Tests for a Class

GET `/classes/{class_id}/tests`

Response:

```
[
  {
    "id": "t101",
    "name": "Algebra Basics Test",
    "due_date": "2025-07-20T23:59:59Z",
    "instructions": "Answer all questions.",
    "teacher_id": "t001"
  }
]
```

6. View Questions in a Test

GET `/tests/{test_id}/questions`

Response: *As previously described*

7. Delete a Test

DELETE `/tests/{test_id}`

Response: *As previously described*

8. View Student Responses for a Test (Teacher Only)

GET `/tests/{test_id}/students/{student_id}/responses`

Response: *As previously described*

Student Endpoints

All endpoints are scoped to the authenticated student.

1. View My School and Class

GET `/students/me/profile`

Response:

```
{
  "student_id": "s101",
  "name": "Bart Simpson",
  "school": {
    "id": "sch001",
    "name": "Springfield High School"
  },
  "class": {
    "id": "cls101",
    "name": "Grade 10 - A"
  }
}
```

2. View List of My Class Tests

GET `/students/me/tests`

Query Parameters:

- `status` (optional: "pending", "completed")
- `due_date_before` (optional)
- `due_date_after` (optional)

Response: *As previously described, filtered to tests for the student's class*

3. Get All Questions for a Test

GET `/students/me/tests/{test_id}/questions`

Response: *As previously described*

4. Submit Answer to a Question

POST `/students/me/tests/{test_id}/questions/{question_id}/answer`

Request Body: *As previously described*

5. Get Specific Question in a Test

GET `/students/me/tests/{test_id}/questions/{question_id}`

Response: *As previously described*

6. View My Response to a Test

GET `/students/me/tests/{test_id}/responses`

Response: *As previously described*

Admin/Teacher: View Student Responses

GET `/tests/{test_id}/students/{student_id}/responses`

Restricted to teachers of the class and admins.

Response: *As previously described*

Error Handling

Standard Error Response:

```
{

  "error": "ErrorType",
  "message": "Detailed error message."
}
```

- `401 Unauthorized` for missing/invalid tokens.
- `403 Forbidden` for insufficient permissions.
- `404 Not Found` for missing resources.
- `400 Bad Request` for invalid input.

Example Workflow

Teacher creates a test for a class:

1. GET `/schools/{school_id}/classes`
2. GET `/questions?subject=Mathematics&topic=Algebra`
3. POST `/classes/{class_id}/tests` with selected questions.

Student takes a test:

1. GET `/students/me/tests` (see available tests for their class)
2. GET `/students/me/tests/{test_id}/questions`
3. POST `/students/me/tests/{test_id}/questions/{question_id}/answer` for each question.

Teacher reviews student responses:

1. GET `/tests/{test_id}/students/{student_id}/responses`

Student reviews their own response:

1. GET `/students/me/tests/{test_id}/responses`

Additional Notes

- **Pagination:** Use `?page=1&limit=20` for list endpoints as needed.
- **Timestamps:** All dates/times are in ISO 8601 format.
- **Extensibility:** Add endpoints for analytics, time limits, grading, etc., as needed.

- **OpenAPI/Swagger:** This structure can be directly translated to an OpenAPI 3.0 spec.

Open Items

List of open questions for discussion

Authentication & Authorization

- How will roles (student, teacher, admin) be managed and enforced at the API level for each endpoint?
- How will you handle multi-teacher classes or co-teaching scenarios?
- How will you ensure a teacher can only access responses for students in their own classes?

Entities & Relationships

- Can a student or teacher belong to multiple classes or schools? If yes, how is this modeled and managed?
- What happens if a student changes classes or schools? How is test access and response history managed?
- How are class rosters (student/teacher lists) managed and updated?

Test Lifecycle & Access

- Can tests be edited or updated after creation? If so, what are the rules and restrictions (especially if students have already started or completed the test)?
- How do you handle test versioning or re-use of questions across tests?
- What is the process for archiving or deleting old tests and responses?

Student Experience

- Are students allowed to retake tests? If so, how are multiple attempts tracked and presented?
- How is the test status (pending, completed, overdue) determined and updated?

Teacher Experience

- How can teachers assign tests to multiple classes or groups at once, if needed?
- Can teachers see aggregate analytics (e.g., average scores) for a test or class?

Data Privacy & Security

- How is sensitive student data (responses, scores) protected at rest and in transit?
- What audit logging is in place for access to student responses?
- How are API rate limits and abuse prevention handled?

API Usability

- What pagination, filtering, and sorting options are supported for large lists (e.g., classes, tests, responses)?
- Are partial updates (PATCH) supported for entities like tests or users?
- How are error responses and validation errors structured and documented for clients?

Notifications & Events

- Will there be notifications (email, in-app) for students when a new test is assigned or due?
- Are there webhook or event-driven APIs for integration with other school systems?

Internationalization & Accessibility

- How will the API handle multiple languages for questions, instructions, and responses?
- Are there accessibility considerations for students with special needs?

Testing & Documentation

- Will there be a sandbox or mock server for client development and testing?

End of Document