**SCHOOL CONFIG**

Great. Let's start with the first endpoint in your school\_config module:

**✅ Endpoint 1: Create and List Academic Years**

**URL:** /school\_config/years/  
**Methods:** GET, POST  
**View Class:** YearListCreateView  
**Permissions:**

* IsschoolAdmin (Full access)
* IsStudentReadOnly, IsTeacherReadOnly, IsSchoolAdminReadOnly (Read-only)

**▶️ GET Request (List All Years)**

* **Description:** Lists all academic years created by the School Admin's school.
* **Filters Available:** status (e.g., ?status=true)
* **Pagination:** Enabled (10 per page by default)

**Example Response:**

[

{

"year\_id": "uuid",

"name": "2024/2025",

"start\_date": "2024-09-01",

"end\_date": "2025-07-31",

"school": "school\_uuid",

"school\_name": "Covenant International School",

"status": true

}

]

**▶️ POST Request (Create a Year)**

* **Automatically sets:** school from the authenticated SchoolAdmin.
* **Validation:** Prevents duplicate school injection from request data.

**Sample Payload:**

{

"name": "2025/2026",

"start\_date": "2025-09-01",

"end\_date": "2026-07-15"

}

**Successful Response:**

{

"year\_id": "new\_uuid",

"name": "2025/2026",

"start\_date": "2025-09-01",

"end\_date": "2026-07-15",

"school": "school\_uuid",

"school\_name": "Covenant International School",

"status": false

}

Ready for the next endpoint (/school\_config/years/<uuid:pk>/) — Year detail, update, and delete?

Perfect. Here’s the breakdown for the next endpoint:

**✅ Endpoint 2: Retrieve, Update, or Delete a Year**

**URL:** /school\_config/years/<uuid:pk>/  
**Methods:** GET, PUT, PATCH, DELETE  
**View Class:** YearDetailView  
**Permissions:** IsschoolAdmin (Full access only)

**▶️ GET Request**

* **Description:** Retrieves the details of a specific academic year.

**Example Response:**

{

"year\_id": "uuid",

"name": "2024/2025",

"start\_date": "2024-09-01",

"end\_date": "2025-07-31",

"school": "school\_uuid",

"school\_name": "Covenant International School",

"status": true

}

**✏️ PUT/PATCH Request**

* **Allows updates to:** name, start\_date, end\_date, status
* **Special Logic:**
  + If status=true is set, all other years for that school will be deactivated (status=false)
  + You **cannot** update year\_id or school

**Sample Payload (PATCH):**

{

"status": true

}

**Effect:** Activates this year and deactivates others for the same school.

**❌ DELETE Request**

* **Validation:** Active years (status=true) **cannot** be deleted.
* **Error Response (if active):**

{

"error": "Cannot delete the active year."

}

Shall we continue with **Endpoint 3: List/Create Terms (/school\_config/terms/)**?

Great. Here's the breakdown for the next endpoint:

**✅ Endpoint 3: List & Create Academic Terms**

**URL:** /school\_config/terms/  
**Methods:** GET, POST  
**View Class:** TermListCreateView  
**Permissions:**

* IsschoolAdmin → Full access
* IsStudentReadOnly, IsTeacherReadOnly, IsSchoolAdminReadOnly → Read-only

**▶️ GET Request**

* **Description:** Returns a list of academic terms for the school.
* **Filterable By:**
  + status: true or false
  + year: UUID of the academic year
* **Searchable By:** name (via DRF SearchFilter)

**Query Example:**  
/school\_config/terms/?status=true&year=123e4567-e89b-12d3-a456-426614174000

**Response:**

[

{

"term\_id": "uuid",

"name": "First Term",

"start\_date": "2024-09-01",

"end\_date": "2024-12-15",

"year": "year\_uuid",

"school": "school\_uuid",

"school\_name": "Covenant International",

"status": true

},

...

]

**✏️ POST Request**

* **Creates a new term** under the authenticated School Admin’s school.
* **Auto-fills:** school field from request user.
* **Required Fields:**
* {
* "name": "First Term",
* "start\_date": "2024-09-01",
* "end\_date": "2024-12-15",
* "year": "year\_uuid",
* "status": true
* }

**Note:** Creation of multiple active terms for the same year is not prevented here (handled in update view).

Proceed with **Endpoint 4: /school\_config/terms/<uuid:pk>/ - Retrieve, Update, Delete Term**?

Great. Here's the breakdown for the next endpoint:

**✅ Endpoint 4: Retrieve, Update, or Delete a Term**

**URL:** /school\_config/terms/<uuid:pk>/  
**Methods:** GET, PUT, PATCH, DELETE  
**View Class:** TermDetailView  
**Permissions:** IsschoolAdmin (Only School Admins have full access)

**▶️ GET Request**

* **Description:** Retrieve details of a specific term by its UUID.

**Example Response:**

{

"term\_id": "uuid",

"name": "First Term",

"start\_date": "2024-09-01",

"end\_date": "2024-12-15",

"year": "year\_uuid",

"school": "school\_uuid",

"school\_name": "AlgorithmX College",

"status": true

}

**✏️ PUT/PATCH Request**

* **Purpose:** Update term details (e.g., status or dates).
* **Special Logic:** If status is set to true, it automatically deactivates all other terms for the same year.

**Example Request Body:**

{

"name": "First Term",

"status": true

}

**🗑️ DELETE Request**

* **Only allowed** if the term is *not* active.
* **If status is true, deletion is blocked.**

**Error on Active Term Deletion:**

{

"error": "Cannot delete the active year."

}

Ready for **Endpoint 5: /school\_config/class\_years/ - List & Create Class Years**?

Here’s the breakdown for the next endpoint:

**✅ Endpoint 5: List & Create Class Years**

**URL:** /school\_config/class\_years/  
**Methods:** GET, POST  
**View Class:** ClassYearListCreateView  
**Permissions:**

* IsschoolAdmin → Full access (list & create)
* IsStudentReadOnly, IsTeacherReadOnly, IsSchoolAdminReadOnly → Read-only

**▶️ GET Request**

* **Purpose:** Returns a list of class year entries for the authenticated school.

**Example Response:**

[

{

"class\_year\_id": "uuid",

"school": "school\_uuid",

"school\_name": "AlgorithmX Academy",

"year": "year\_uuid",

"year\_name": "2024/2025",

"class\_name": "SS1"

},

...

]

**✏️ POST Request**

* **Purpose:** Create a new class year for the current school and year.
* **Auto-filled Fields:**
  + school is inferred from the logged-in School Admin.
  + year must be an active year within the school.

**Example Request Body:**

{

"year": "uuid\_of\_active\_year",

"class\_name": "SS1"

}

**Example Response:**

{

"class\_year\_id": "uuid",

"school": "school\_uuid",

"school\_name": "AlgorithmX Academy",

"year": "uuid",

"year\_name": "2024/2025",

"class\_name": "SS1"

}

Would you like to continue with **Endpoint 6: /school\_config/class\_years/<uuid:pk>/ - ClassYear Detail View**?

Here’s the breakdown for the next endpoint:

**✅ Endpoint 6: Retrieve, Update, or Delete Class Year**

**URL:** /school\_config/class\_years/<uuid:pk>/  
**Methods:** GET, PUT, PATCH, DELETE  
**View Class:** ClassYearDetailView  
**Permissions:**

* IsschoolAdmin → Full access
* Others → No access

**▶️ GET Request**

* **Purpose:** Retrieve a specific class year by its ID.

**Example Response:**

{

"class\_year\_id": "b6e733cc-fd24-4909-aefb-ace6bb9c94b3",

"school": "f83e05ec-ea25-4a61-abb5-998a7414d47a",

"school\_name": "AlgorithmX Academy",

"year": "31bc6a77-e167-4eab-9c49-e1214dd9857f",

"year\_name": "2024/2025",

"class\_name": "JSS3"

}

**✏️ PUT / PATCH Request**

* **Purpose:** Update the class year name or year association.

**Example Request Body:**

{

"year": "new\_year\_uuid",

"class\_name": "SS2"

}

**Note:** School ID and school name are read-only.

**❌ DELETE Request**

* **Purpose:** Delete a class year if it's no longer needed.

**Response:**

{

"message": "Class year deleted successfully."

}

Would you like to proceed to **Endpoint 7: /school\_config/classes/ - List & Create Class Arms**?

Here’s the breakdown for the next endpoint:

**✅ Endpoint 7: List and Create Class Arms**

**URL:** /school\_config/classes/  
**Methods:** GET, POST  
**View Class:** ClassListCreateView  
**Permissions:**

* IsschoolAdmin: Full access
* IsStudentReadOnly, IsTeacherReadOnly, IsSchoolAdminReadOnly: Can view (GET) only

**▶️ GET Request**

* **Purpose:** Retrieve all class arms for the School Admin’s school.

**Example Response:**

[

{

"class\_id": "5eaa17aa-801e-48cd-a2e9-0c2d861d4315",

"arm\_name": "Arm A",

"class\_year": "b6e733cc-fd24-4909-aefb-ace6bb9c94b3",

"class\_year\_name": "JSS3",

"year\_name": "2024/2025",

"school": "f83e05ec-ea25-4a61-abb5-998a7414d47a",

"school\_name": "AlgorithmX Academy"

},

...

]

**➕ POST Request**

* **Purpose:** Create a new class arm under a class year.

**Example Request Body:**

{

"arm\_name": "Arm B",

"class\_year": "b6e733cc-fd24-4909-aefb-ace6bb9c94b3"

}

**Fields Automatically Filled:**

* school: From authenticated School Admin
* class\_year\_name, year\_name, school\_name: Read-only

Ready to move on to **Endpoint 8: /school\_config/classes/<uuid:pk>/ - Retrieve, Update, or Delete Class Arm**?

Here’s the breakdown for the next endpoint:

**✅ Endpoint 8: Retrieve, Update, or Delete Class Arm**

**URL:** /school\_config/classes/<uuid:pk>/  
**Methods:** GET, PUT, PATCH, DELETE  
**View Class:** ClassDetailView  
**Permissions:**

* IsschoolAdmin: Full access

**▶️ GET Request**

**Purpose:** Retrieve detailed information about a specific class arm.

**Example Response:**

{

"class\_id": "5eaa17aa-801e-48cd-a2e9-0c2d861d4315",

"arm\_name": "Arm A",

"class\_year": "b6e733cc-fd24-4909-aefb-ace6bb9c94b3",

"class\_year\_name": "JSS3",

"year\_name": "2024/2025",

"school": "f83e05ec-ea25-4a61-abb5-998a7414d47a",

"school\_name": "AlgorithmX Academy"

}

**📝 PUT/PATCH Request**

**Purpose:** Update the class arm name or class year.

**Example Request Body (PATCH):**

{

"arm\_name": "Arm B"

}

**Example Response:**

{

"class\_id": "5eaa17aa-801e-48cd-a2e9-0c2d861d4315",

"arm\_name": "Arm B",

"class\_year": "b6e733cc-fd24-4909-aefb-ace6bb9c94b3",

"class\_year\_name": "JSS3",

"year\_name": "2024/2025",

"school": "f83e05ec-ea25-4a61-abb5-998a7414d47a",

"school\_name": "AlgorithmX Academy"

}

**❌ DELETE Request**

**Purpose:** Delete the specified class arm (if not referenced by related models).

**Example Success Response:**

{

"message": "Class 'Arm B' has been deleted successfully."

}

**Failure (e.g., if referenced):**

{

"error": "Cannot delete this class as it is referenced by other records."

}

Ready to continue with **Endpoint 9: /school\_config/classrooms/ - List & Create Classrooms**?

Here’s the breakdown for the next endpoint:

**✅ Endpoint 9: List & Create Classrooms**

**URL:** /school\_config/classrooms/  
**Methods:** GET, POST  
**View Class:** ClassroomListCreateView  
**Permissions:**

* IsschoolAdmin: Full access
* IsStudentReadOnly, IsTeacherReadOnly, IsSchoolAdminReadOnly: Read-only access

**▶️ GET Request**

**Purpose:** Retrieve a paginated list of all classrooms in the School Admin’s school.

**Example Response:**

{

"count": 2,

"next": null,

"previous": null,

"results": [

{

"classroom\_id": "81484c5e-9923-4f0e-a57e-3f443e27d3c5",

"block\_name": "A Block",

"room\_number": "101",

"capacity": 30,

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

},

{

"classroom\_id": "b83d5020-f204-4e56-a11c-1bc028ec2485",

"block\_name": "B Block",

"room\_number": "201",

"capacity": 35,

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

}

]

}

**📝 POST Request**

**Purpose:** Create a new classroom. The school field is auto-filled from the authenticated School Admin.

**Example Request Body:**

{

"block\_name": "C Block",

"room\_number": "301",

"capacity": 40

}

**Example Response:**

{

"classroom\_id": "f2dfd882-e7aa-4ac3-8f6f-e9467a94e146",

"block\_name": "C Block",

"room\_number": "301",

"capacity": 40,

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

}

Next up is **Endpoint 10: /school\_config/classrooms/<uuid:pk>/ – Retrieve, Update, or Delete Classroom**. Ready?

Here’s the breakdown for:

**✅ Endpoint 10: Retrieve, Update, or Delete Classroom**

**URL:** /school\_config/classrooms/<uuid:pk>/  
**Methods:** GET, PUT, PATCH, DELETE  
**View Class:** ClassroomDetailView  
**Permissions:** IsschoolAdmin (Full access)

**▶️ GET Request**

**Purpose:** Retrieve details of a specific classroom by UUID.

**Example Response:**

{

"classroom\_id": "f2dfd882-e7aa-4ac3-8f6f-e9467a94e146",

"block\_name": "C Block",

"room\_number": "301",

"capacity": 40,

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

}

**🛠️ PUT/PATCH Request**

**Purpose:** Update fields of a specific classroom.

* PUT updates all fields.
* PATCH updates selected fields.

**Example PATCH Request:**

{

"capacity": 45

}

**Example Response:**

{

"classroom\_id": "f2dfd882-e7aa-4ac3-8f6f-e9467a94e146",

"block\_name": "C Block",

"room\_number": "301",

"capacity": 45,

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

}

**❌ DELETE Request**

**Purpose:** Delete a specific classroom.

**Example Response:**

{

"message": "Classroom deleted successfully."

}

Next is **Endpoint 11: /school\_config/departments/ – List and Create Departments**. Proceed?

Here’s the breakdown for:

**✅ Endpoint 11: List and Create Departments**

**URL:** /school\_config/departments/  
**Methods:** GET, POST  
**View Class:** DepartmentListCreateView  
**Permissions:**

* GET: IsschoolAdmin, IsStudentReadOnly, IsTeacherReadOnly, IsSchoolAdminReadOnly
* POST: IsschoolAdmin

**▶️ GET Request**

**Purpose:** List all departments in the authenticated School Admin’s school.

**Example Response:**

[

{

"department\_id": "3a9a3f7c-c1d2-4b56-92a3-923cd4578124",

"name": "Science",

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

},

{

"department\_id": "2b56a3df-9c23-4e56-8820-a1e3aa724c9d",

"name": "Commercial",

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

}

]

**🛠️ POST Request**

**Purpose:** Create a new department in the authenticated School Admin’s school.

**Request Body:**

{

"name": "Art"

}

**Example Response:**

{

"department\_id": "7e0d45f2-4210-4e23-91c7-9f18a6cddb2f",

"name": "Art",

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

}

Next is **Endpoint 12: /school\_config/departments/<uuid:pk>/ – Retrieve, Update, Delete a Department**. Ready?

Here’s the breakdown for:

**✅ Endpoint 12: Retrieve, Update, Delete a Department**

**URL:** /school\_config/departments/<uuid:pk>/  
**Methods:** GET, PUT, PATCH, DELETE  
**View Class:** DepartmentDetailView  
**Permissions:** IsschoolAdmin

**▶️ GET Request**

**Purpose:** Retrieve a specific department by its UUID.

**Example Response:**

{

"department\_id": "7e0d45f2-4210-4e23-91c7-9f18a6cddb2f",

"name": "Art",

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

}

**🛠️ PUT / PATCH Request**

**Purpose:** Update a department’s name.

**Request Body:**

{

"name": "Arts and Humanities"

}

**Example Response:**

{

"department\_id": "7e0d45f2-4210-4e23-91c7-9f18a6cddb2f",

"name": "Arts and Humanities",

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

}

**❌ DELETE Request**

**Purpose:** Permanently delete a department by its UUID.

**Example Response:**

{

"message": "Department deleted successfully."

}

Next is **Endpoint 13: /school\_config/subjects/ – List and Create Subjects**. Shall we continue?

Here’s the breakdown for:

**✅ Endpoint 13: List & Create Subjects**

**URL:** /school\_config/subjects/  
**Methods:** GET, POST  
**View Class:** SubjectListCreateView  
**Permissions:** IsschoolAdmin | IsStudentReadOnly | IsTeacherReadOnly

**▶️ GET Request**

**Purpose:** List all subjects in the authenticated School Admin’s school.  
**Supports Filtering:**

* By department\_id via query params → /school\_config/subjects/?department\_id=<uuid>

**Example Response:**

[

{

"subject\_id": "845aab4a-888f-4293-8031-b20d99f274f1",

"name": "Mathematics",

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

},

{

"subject\_id": "a18e8b1f-7685-4122-a4f9-f7a75e8a564a",

"name": "Biology",

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

}

]

**🛠️ POST Request**

**Purpose:** Create a new subject for the school.

**Request Body:**

{

"name": "Physics"

}

**Example Response:**

{

"subject\_id": "5f2343b8-9b90-4e7a-85f3-7e6be94e64f9",

"name": "Physics",

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

}

Next is **Endpoint 14: /school\_config/subjects/<uuid:pk>/ – Retrieve, Update, Delete Subject**. Shall I proceed?

Here’s the breakdown for:

**✅ Endpoint 14: Retrieve, Update, or Delete a Subject**

**URL:** /school\_config/subjects/<uuid:pk>/  
**Methods:** GET, PUT, PATCH, DELETE  
**View Class:** SubjectDetailView  
**Permissions:** IsschoolAdmin

**▶️ GET Request**

**Purpose:** Retrieve a single subject's details.  
**Example Response:**

{

"subject\_id": "5f2343b8-9b90-4e7a-85f3-7e6be94e64f9",

"name": "Physics",

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

}

**🛠️ PUT or PATCH Request**

**Purpose:** Update the subject’s name.

**Request Body:**

{

"name": "Basic Physics"

}

**Response:**

{

"subject\_id": "5f2343b8-9b90-4e7a-85f3-7e6be94e64f9",

"name": "Basic Physics",

"school": "ccab9b11-9645-4c92-b7a9-d94459a7b184",

"school\_name": "AlgorithmX Academy"

}

**❌ DELETE Request**

**Purpose:** Delete the specified subject.  
**Response:**

{

"message": "Subject 'Basic Physics' has been deleted successfully."

}

Ready to continue with **Endpoint 15: Bulk Create Class Teachers (/assignments/class\_teachers/bulk\_create/)**?

**ASSIGNMENT**

Here’s the breakdown for:

**✅ Endpoint 15: Bulk Create Class Teachers**

**URL:** /assignments/class\_teachers/bulk\_create/  
**Method:** POST  
**View Class:** BulkClassTeacherCreateView  
**Permissions:** IsschoolAdmin

**▶️ POST Request**

**Purpose:** Assign multiple teachers to multiple classes in a single request.

**Request Body:**

{

"assignments": [

{

"teacher\_id": "6e8b153e-243a-49cb-a1f2-d2c3ac82d4d2",

"class\_id": "abf1cb7e-5b49-4ab5-8ab6-c70a6870f312"

},

{

"teacher\_id": "7d45f3c0-f2e6-4989-bef2-88ad4e68f0aa",

"class\_id": "bfc0cacc-2c6e-4d8c-a1d2-4f5453c29f0d"

}

]

}

**✅ Success Response:**

{

"message": "Teachers assigned successfully.",

"data": [

{

"class\_teacher\_id": "3d2f2e66-711e-45db-b5f0-b7d64fd295e3",

"teacher\_id": "6e8b153e-243a-49cb-a1f2-d2c3ac82d4d2",

"teacher\_name": "Ayo Dare",

"class\_id": "abf1cb7e-5b49-4ab5-8ab6-c70a6870f312",

"class\_name": "JSS1 A",

"school\_id": "acb993ef-d5b9-46db-8b7c-18c7bdfd0e3b",

"school\_name": "AlgorithmX Academy"

},

{

"class\_teacher\_id": "4e6fa621-9d9a-4b8c-8e9e-16e55e21670b",

"teacher\_id": "7d45f3c0-f2e6-4989-bef2-88ad4e68f0aa",

"teacher\_name": "Olu Bisi",

"class\_id": "bfc0cacc-2c6e-4d8c-a1d2-4f5453c29f0d",

"class\_name": "JSS1 B",

"school\_id": "acb993ef-d5b9-46db-8b7c-18c7bdfd0e3b",

"school\_name": "AlgorithmX Academy"

}

]

}

**❌ Error Response (Some Failures):**

{

"message": "Teachers assigned successfully.",

"data": [

{

"class\_teacher\_id": "...",

...

}

],

"errors": [

{

"teacher\_id": "xxxx",

"class\_id": "yyyy",

"error": "Already assigned."

},

{

"teacher\_id": "zzzz",

"error": "Teacher not found in this school."

}

]

}

Ready for **Endpoint 16: List All Class-Teacher Assignments (/assignments/class\_teachers/)**?

Here’s the breakdown for:

**✅ Endpoint 16: List All Class-Teacher Assignments**

**URL:** /assignments/class\_teachers/  
**Method:** GET  
**View Class:** ClassTeacherListView  
**Permissions:** IsschoolAdmin  
**Pagination:** Standard pagination (page, page\_size)  
**Searchable Fields:** class\_assigned.class\_name

**▶️ Query Parameters (Optional)**

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| search | string | Search by class name |
| page | integer | Page number |
| page\_size | integer | Number of items per page |

**✅ Sample Response:**

{

"count": 2,

"next": null,

"previous": null,

"results": [

{

"class\_teacher\_id": "b3c0f229-2a3e-46e0-8998-173d319fd5b1",

"class\_assigned": "18b4c314-2099-48a0-b674-51135d45b25b",

"teacher": "d8d15d70-8402-4e2e-8724-1980bd1664c2",

"school": "a57f032d-bb1f-4f2a-a6bc-77b6a950f70d",

"class\_assigned\_name": "JSS1 A",

"teacher\_name": "Olu Bisi",

"teacher\_lastname": "Dare",

"school\_name": "AlgorithmX Academy"

},

{

"class\_teacher\_id": "e3f44ddf-4c74-4388-9875-5c2557e12d78",

"class\_assigned": "90ef71f0-b9e4-4e20-b1b8-3ef2b7a2eafb",

"teacher": "f80d1e04-fb71-4a10-a39a-512c9dbfdf91",

"school": "a57f032d-bb1f-4f2a-a6bc-77b6a950f70d",

"class\_assigned\_name": "JSS1 B",

"teacher\_name": "Adaobi",

"teacher\_lastname": "Okeke",

"school\_name": "AlgorithmX Academy"

}

]

}

Would you like to continue with **Endpoint 17: Update a Single Class-Teacher Assignment (/assignments/class\_teachers/<uuid:pk>/)**?

Here’s the breakdown for:

**✅ Endpoint 17: Update a Single Class-Teacher Assignment**

**URL:** /assignments/class\_teachers/<uuid:pk>/  
**Method:** PUT or PATCH  
**View Class:** ClassTeacherUpdateView  
**Permissions:** IsschoolAdmin  
**Description:** Updates a teacher’s assignment to a class within the same school.

**▶️ Request Body (PATCH)**

{

"class\_assigned": "18b4c314-2099-48a0-b674-51135d45b25b",

"teacher": "d8d15d70-8402-4e2e-8724-1980bd1664c2"

}

Note: class\_assigned and teacher must both belong to the same school as the authenticated School Admin.

**✅ Sample Response**

{

"class\_teacher\_id": "e3f44ddf-4c74-4388-9875-5c2557e12d78",

"class\_assigned": "18b4c314-2099-48a0-b674-51135d45b25b",

"teacher": "d8d15d70-8402-4e2e-8724-1980bd1664c2",

"school": "a57f032d-bb1f-4f2a-a6bc-77b6a950f70d",

"class\_assigned\_name": "JSS1 A",

"teacher\_name": "Olu Bisi",

"teacher\_lastname": "Dare",

"school\_name": "AlgorithmX Academy"

}

Would you like to proceed with **Endpoint 18: Delete Multiple Class-Teacher Assignments**?

Here’s the breakdown for:

**✅ Endpoint 18: Delete Multiple Class-Teacher Assignments**

**URL:** /assignments/class\_teachers/delete-multiple/  
**Method:** DELETE  
**View Class:** DeleteMultipleClassTeachersView  
**Permissions:** IsschoolAdmin  
**Description:** Allows a School Admin to delete multiple class-teacher assignments by providing their IDs.

**▶️ Request Body**

{

"assignment\_ids": [

"f94b35a7-74d6-4433-8d67-930a78dc0381",

"b7097c79-3ef3-464e-a13f-bdcb4e62b1da"

]

}

**✅ Sample Response**

{

"message": "2 teacher-class assignment(s) deleted successfully."

}

**⚠️ Possible Errors**

* If assignment\_ids is not a list:

{

"error": "A list of assignment IDs is required."

}

Ready for **Endpoint 19: SubjectClassListCreateView**?

Here is the detailed API specification for the next endpoint:

**✅ Endpoint: POST /assignments/class\_departments/**

**View:** ClassDepartmentListCreateView  
**Permissions:** IsAuthenticated (must be a school admin)

**🔹 Purpose:**

To **create a new mapping** between a class and a department in the school.

**🔸 Request Body (application/json):**

{

"classes": "class-uuid",

"department": "department-uuid"

}

🔍 classes is the UUID of the class arm.  
🔍 department is the UUID of the department.

**✅ Success Response (201 Created):**

{

"subject\_class\_id": "uuid-generated",

"school": "school-uuid",

"classes": "class-uuid",

"department": "department-uuid",

"class\_name": "SS 1 A",

"school\_name": "AlgorithmX High School",

"department\_name": "Science"

}

**❌ Error Responses:**

**🔹 Missing Required Fields**

{

"classes": ["This field is required."],

"department": ["This field is required."]

}

**🔹 Department Not Belonging to the School**

{

"detail": "Department must belong to the same school."

}

Would you like me to continue with the next endpoint?

Here is the detailed API documentation for the next endpoint:

**✅ Endpoint:**

GET /assignments/class\_departments/<uuid:subject\_class\_id>/  
PUT /assignments/class\_departments/<uuid:subject\_class\_id>/  
DELETE /assignments/class\_departments/<uuid:subject\_class\_id>/

**View:** ClassDepartmentDetailView  
**Permissions:** IsAuthenticated (must be a School Admin)

**🔹 Purpose:**

To **retrieve**, **update**, or **delete** a single Class–Department mapping.

**📥 GET: Retrieve a Class–Department Mapping**

**🔸 Response (200 OK):**

{

"subject\_class\_id": "15f65e52-45d9-4d68-a234-3de4b209b63c",

"school": "a671a6dc-8e54-48c3-bd5e-e497322b1ad2",

"classes": "17d6ea8d-5cde-4be0-a42e-10b2cebe2023",

"department": "5e9123a1-b64e-4413-aea0-2be8d0aa56dd",

"class\_name": "SS 2 B",

"school\_name": "Smart Future School",

"department\_name": "Commercial"

}

**✏️ PUT: Update a Class–Department Mapping**

**🔸 Request Body (application/json):**

{

"classes": "17d6ea8d-5cde-4be0-a42e-10b2cebe2023",

"department": "4e9131fa-d927-4c72-bfb1-b3a84521f313"

}

**✅ Success Response (200 OK):**

{

"subject\_class\_id": "15f65e52-45d9-4d68-a234-3de4b209b63c",

"school": "a671a6dc-8e54-48c3-bd5e-e497322b1ad2",

"classes": "17d6ea8d-5cde-4be0-a42e-10b2cebe2023",

"department": "4e9131fa-d927-4c72-bfb1-b3a84521f313",

"class\_name": "SS 2 B",

"school\_name": "Smart Future School",

"department\_name": "Science"

}

**❌ Validation Error:**

{

"detail": "Department must belong to the same school."

}

**🗑️ DELETE: Delete a Class–Department Mapping**

**✅ Success Response (204 No Content):**

// No content returned

Ready for the next one?  
GET | POST /assignments/subject\_department/ (SubjectClassListCreateView)

Here is the next endpoint documentation:

**✅ Endpoint:**

GET /assignments/subject\_department/  
POST /assignments/subject\_department/

**View:** SubjectClassListCreateView  
**Permissions:** IsschoolAdmin

**🔹 Purpose:**

Manage the mapping between subjects and departments in a school.

**📥 GET: List Subject–Department Mappings**

**🔸 Query Parameters (optional):**

* subject=<subject\_id> — Filter by a specific subject UUID
* department=<department\_id> — Filter by a specific department UUID
* search=<text> — Search by subject or department name

**✅ Response (200 OK):**

[

{

"subject\_class\_id": "2341d72c-5a9b-445f-983f-7c8ea3b193eb",

"subject": "64f23b3c-0d4c-4cfb-bf7c-8267590de930",

"school": "ecf33921-3d5f-42e1-8b5a-dcd187d1e0d7",

"department": "b681e65b-e8e4-4db5-b929-cb865172045e",

"subject\_name": "Mathematics",

"school\_name": "SmartFuture School",

"department\_name": "Science"

},

...

]

**✏️ POST: Create a Subject–Department Mapping**

**🔸 Request Body:**

{

"subject": "64f23b3c-0d4c-4cfb-bf7c-8267590de930",

"department": "b681e65b-e8e4-4db5-b929-cb865172045e"

}

**✅ Success Response (201 Created):**

{

"subject\_class\_id": "e3adbc33-8f4f-4fd1-9ad6-10cce2e1ebfb",

"subject": "64f23b3c-0d4c-4cfb-bf7c-8267590de930",

"school": "ecf33921-3d5f-42e1-8b5a-dcd187d1e0d7",

"department": "b681e65b-e8e4-4db5-b929-cb865172045e",

"subject\_name": "Mathematics",

"school\_name": "SmartFuture School",

"department\_name": "Science"

}

**❌ Validation Errors:**

* If subject or department is not from the current school:

{

"detail": "Subject and department must belong to the same school."

}

Ready for the next one?  
GET | PUT | DELETE /assignments/subject\_department/<uuid:subject\_class\_id>/ (SubjectClassDetailView)

Here is the next endpoint documentation:

**✅ Endpoint:**

GET /assignments/subject\_department/<uuid:subject\_class\_id>/  
PUT /assignments/subject\_department/<uuid:subject\_class\_id>/  
DELETE /assignments/subject\_department/<uuid:subject\_class\_id>/

**View:** SubjectClassDetailView  
**Permissions:** IsschoolAdmin  
**Lookup Field:** subject\_class\_id

**📥 GET: Retrieve Subject–Department Mapping**

**✅ Success Response (200 OK):**

{

"subject\_class\_id": "e3adbc33-8f4f-4fd1-9ad6-10cce2e1ebfb",

"subject": "64f23b3c-0d4c-4cfb-bf7c-8267590de930",

"school": "ecf33921-3d5f-42e1-8b5a-dcd187d1e0d7",

"department": "b681e65b-e8e4-4db5-b929-cb865172045e",

"subject\_name": "Mathematics",

"school\_name": "SmartFuture School",

"department\_name": "Science"

}

**✏️ PUT: Update Subject–Department Mapping**

**🔸 Request Body:**

{

"subject": "64f23b3c-0d4c-4cfb-bf7c-8267590de930",

"department": "98c48dd1-1c95-4b39-9df1-cc1f65fbe234"

}

**✅ Success Response (200 OK):**

{

"subject\_class\_id": "e3adbc33-8f4f-4fd1-9ad6-10cce2e1ebfb",

"subject": "64f23b3c-0d4c-4cfb-bf7c-8267590de930",

"school": "ecf33921-3d5f-42e1-8b5a-dcd187d1e0d7",

"department": "98c48dd1-1c95-4b39-9df1-cc1f65fbe234",

"subject\_name": "Mathematics",

"school\_name": "SmartFuture School",

"department\_name": "Commercial"

}

**❌ Validation Error Example:**

{

"detail": "Subject and department must belong to the same school."

}

**🗑️ DELETE: Remove Subject–Department Mapping**

**✅ Success Response (204 No Content):**

{}

Ready for the next one?  
GET | POST /assignments/teacher\_assignments/ (TeacherAssignmentListCreateView)?

Here is the next endpoint documentation:

**✅ Endpoint:**

GET /assignments/teacher\_assignments/  
POST /assignments/teacher\_assignments/

**View:** TeacherAssignmentListCreateView  
**Permissions:** IsschoolAdmin  
**Searchable Fields:**

* teacher.first\_name, teacher.last\_name
* subject.subject.name, class\_assigned.arm\_name, subject.department.name

**📥 GET: List Teacher Assignments**

**✅ Success Response (200 OK):**

[

{

"teacher\_subject\_id": "b5e1e775-c71f-4083-88b3-b489e3f78d30",

"teacher": "ea4f7ec3-4f14-4456-9188-648d2d1f64dd",

"subject": "192d0422-382e-406a-804e-9f139b1c7b0c",

"class\_assigned": "4c95c407-1871-4386-8b12-9dc991d16b9f",

"school": "15d2ab83-02a9-42f2-8a2e-8cc52a985e51",

"teacher\_name": "Grace",

"teacher\_lastname": "Johnson",

"subject\_name": "Physics",

"class\_name": "SS2 B",

"school\_name": "Summit Academy",

"department\_name": "Science"

}

]

**✏️ POST: Create Teacher Assignment**

**🔸 Request Body:**

{

"teacher": "ea4f7ec3-4f14-4456-9188-648d2d1f64dd",

"subject": "192d0422-382e-406a-804e-9f139b1c7b0c",

"class\_assigned": "4c95c407-1871-4386-8b12-9dc991d16b9f"

}

**✅ Success Response (201 Created):**

{

"teacher\_subject\_id": "af918bd9-9d26-47d0-bc3a-6c22ef75595e",

"teacher": "ea4f7ec3-4f14-4456-9188-648d2d1f64dd",

"subject": "192d0422-382e-406a-804e-9f139b1c7b0c",

"class\_assigned": "4c95c407-1871-4386-8b12-9dc991d16b9f",

"school": "15d2ab83-02a9-42f2-8a2e-8cc52a985e51",

"teacher\_name": "Grace",

"teacher\_lastname": "Johnson",

"subject\_name": "Physics",

"class\_name": "SS2 B",

"school\_name": "Summit Academy",

"department\_name": "Science"

}

**❌ Validation Error Example:**

{

"non\_field\_errors": ["Teacher, subject, and class must belong to the same school."]

}

Next:  
GET | PUT | DELETE /assignments/teacher\_assignments/<uuid:pk>/ (TeacherAssignmentDetailView)?

Here is the documentation for the next endpoint:

**✅ Endpoint:**

GET /assignments/teacher\_assignments/<uuid:pk>/  
PUT /assignments/teacher\_assignments/<uuid:pk>/  
DELETE /assignments/teacher\_assignments/<uuid:pk>/

**View:** TeacherAssignmentDetailView  
**Permissions:** IsschoolAdmin

**📥 GET: Retrieve Teacher Assignment**

**✅ Success Response (200 OK):**

{

"teacher\_subject\_id": "af918bd9-9d26-47d0-bc3a-6c22ef75595e",

"teacher": "ea4f7ec3-4f14-4456-9188-648d2d1f64dd",

"subject": "192d0422-382e-406a-804e-9f139b1c7b0c",

"class\_assigned": "4c95c407-1871-4386-8b12-9dc991d16b9f",

"school": "15d2ab83-02a9-42f2-8a2e-8cc52a985e51",

"teacher\_name": "Grace",

"teacher\_lastname": "Johnson",

"subject\_name": "Physics",

"class\_name": "SS2 B",

"school\_name": "Summit Academy",

"department\_name": "Science"

}

**✏️ PUT: Update Teacher Assignment**

**🔸 Request Body:**

{

"teacher": "ea4f7ec3-4f14-4456-9188-648d2d1f64dd",

"subject": "192d0422-382e-406a-804e-9f139b1c7b0c",

"class\_assigned": "4c95c407-1871-4386-8b12-9dc991d16b9f"

}

**✅ Success Response (200 OK):**

{

"teacher\_subject\_id": "af918bd9-9d26-47d0-bc3a-6c22ef75595e",

"teacher": "ea4f7ec3-4f14-4456-9188-648d2d1f64dd",

"subject": "192d0422-382e-406a-804e-9f139b1c7b0c",

"class\_assigned": "4c95c407-1871-4386-8b12-9dc991d16b9f",

"school": "15d2ab83-02a9-42f2-8a2e-8cc52a985e51",

"teacher\_name": "Grace",

"teacher\_lastname": "Johnson",

"subject\_name": "Physics",

"class\_name": "SS2 B",

"school\_name": "Summit Academy",

"department\_name": "Science"

}

**❌ Validation Error Example:**

{

"non\_field\_errors": [

"Teacher, subject, and class must belong to the same school."

]

}

**🗑️ DELETE: Delete Teacher Assignment**

**✅ Success Response (204 No Content):**

{}

Continue with:  
GET | POST /assignments/student-classes/ (StudentClassListView)?

Here is the documentation for the next endpoint:

**✅ Endpoint:**

GET /assignments/student-classes/

**View:** StudentClassListView  
**Permissions:**

* IsschoolAdmin → Can view all student-class assignments in their school.
* IsClassTeacher → Can view students in assigned classes.
* ISstudent → Can view only their own class.

**📥 GET: List Student-Class Assignments**

**✅ Success Response (200 OK):**

[

{

"student\_class\_id": "2a3a7937-4e94-41e8-9f63-457b27f2c70b",

"student": "d89d9643-123e-48b9-a7e3-9f761ec4a925",

"class\_year": "SS2",

"class\_arm": "Science",

"student\_firstname": "Samuel",

"student\_surname": "Okoye",

"student\_name": "Okoye Samuel",

"class\_name": "SS2 B"

},

{

"student\_class\_id": "cf183f4b-e882-4f83-a72f-f2931f7c8c23",

"student": "a111de5f-12a7-4e43-b1b7-08a5b3dd4ba7",

"class\_year": "JSS3",

"class\_arm": "Art",

"student\_firstname": "Amina",

"student\_surname": "Yusuf",

"student\_name": "Yusuf Amina",

"class\_name": "JSS3 A"

}

]

**🔐 Access Behavior by Role:**

| **Role** | **Access to Records** |
| --- | --- |
| School Admin | All students in their school |
| Class Teacher | Students in their assigned classes |
| Student | Only their own record/class info |

Proceed to:  
PUT /assignments/student-classes/<uuid:student\_class\_id>/update/ (StudentClassUpdateView)?

Here is the documentation for the next endpoint:

**✅ Endpoint:**

PUT /assignments/student-classes/<uuid:student\_class\_id>/update/

**View:** StudentClassUpdateView  
**Permissions:**

* IsschoolAdmin, IsClassTeacher, or ISstudent

**📤 PUT: Update a Student-Class Assignment**

**🔐 Access Rules:**

* **School Admin**: Can update any student-class mapping within their school.
* **Class Teacher**: Can update students only in classes assigned to them.
* **Student**: May be allowed to update their own record depending on backend validation logic (if enabled).

**🧾 Request Body Example:**

{

"student": "e51c62a4-0dc6-49cd-9aa9-bd2b5c9bc273",

"class\_year": "2025/2026",

"class\_arm": "d7fdc48e-0485-4f8b-92d0-72e21a5d34d0"

}

**✅ Success Response (200 OK):**

{

"student\_class\_id": "4ab1a555-5cb1-4416-8782-b3cf8e25a491",

"student": "e51c62a4-0dc6-49cd-9aa9-bd2b5c9bc273",

"class\_year": "2025/2026",

"class\_arm": "Science",

"student\_firstname": "David",

"student\_surname": "Adebayo",

"student\_name": "Adebayo David",

"class\_name": "SS2 B"

}

**❌ Error Responses:**

* 403 Forbidden: User role is not allowed to perform this update.
* 404 Not Found: StudentClass record does not exist for the given ID.

Proceed to:  
GET or POST /assignments/student-subject-registrations/ (StudentSubjectRegistrationListCreateView)?

Here is the documentation for the next endpoint:

**✅ Endpoint:**

GET /assignments/student-subject-registrations/  
POST /assignments/student-subject-registrations/

**View:** StudentSubjectRegistrationListCreateView  
**Permissions:**

* IsschoolAdmin, ISstudent, or IsClassTeacher

**📥 POST: Register a Student for a Subject**

**🔐 Access Control:**

* **School Admin**: Can register any student.
* **Class Teacher**: Can register students only in classes assigned to them.
* **Student**: Can register only themselves.

**🧾 POST Request Example:**

{

"student\_class": "4f7e9219-9f45-4386-918b-e287a310cae9",

"subject\_class": "b0f28d64-d46b-4f88-a519-e749d6d84788"

}

**✅ POST Success Response (201 Created):**

{

"registration\_id": "693cd1e5-1a29-4e63-bbb2-7f8a1880d924",

"student\_class": "4f7e9219-9f45-4386-918b-e287a310cae9",

"subject\_class": "b0f28d64-d46b-4f88-a519-e749d6d84788",

"term": "First Term",

"school": "72a2c9c8-d02d-4eaf-a931-7c3a08769f00",

"status": "Pending",

"created\_at": "2025-05-08T14:15:12.925Z",

"updated\_at": "2025-05-08T14:15:12.925Z",

"student\_firstname": "Blessing",

"student\_surname": "Ojo",

"student\_admission\_number": "2023/0032",

"student\_name": "Ojo Blessing",

"class\_year": "2025/2026",

"class\_arm": "Science",

"department": "Science",

"subject\_name": "Biology",

"term\_name": "First Term",

"school\_name": "Victory Academy"

}

**📤 GET: List Registered Subjects**

* Returns subject registration records based on the current user's role:
  + School Admin → all students
  + Class Teacher → only students in assigned classes
  + Student → only their own registrations

**✅ GET Response Example:**

[

{

"registration\_id": "693cd1e5-1a29-4e63-bbb2-7f8a1880d924",

"student\_firstname": "Blessing",

"student\_surname": "Ojo",

"student\_admission\_number": "2023/0032",

"student\_name": "Ojo Blessing",

"class\_year": "2025/2026",

"class\_arm": "Science",

"department": "Science",

"subject\_name": "Biology",

"term\_name": "First Term",

"school\_name": "Victory Academy",

"status": "Pending",

"created\_at": "2025-05-08T14:15:12.925Z",

"updated\_at": "2025-05-08T14:15:12.925Z"

}

]

Ready for GET, PUT, DELETE /assignments/student-subject-registrations/<uuid:registration\_id>/?

Here is the documentation for the next endpoint:

**✅ Endpoint:**

GET /assignments/student-subject-registrations/<uuid:registration\_id>/  
PUT /assignments/student-subject-registrations/<uuid:registration\_id>/  
DELETE /assignments/student-subject-registrations/<uuid:registration\_id>/

**View:** StudentSubjectRegistrationDetailView  
**Permissions:**

* **GET**: School Admin, Class Teacher, and Student
* **PUT/DELETE**: Only School Admin and Class Teacher
* **Student**: Can only **view**, not update or delete

**📤 GET: View a Registered Subject**

**🔐 Role-Based View Access:**

* **School Admin**: View any record
* **Class Teacher**: View students in their assigned classes
* **Student**: View only their own registration

**✅ GET Response Example:**

{

"registration\_id": "693cd1e5-1a29-4e63-bbb2-7f8a1880d924",

"student\_class": "4f7e9219-9f45-4386-918b-e287a310cae9",

"subject\_class": "b0f28d64-d46b-4f88-a519-e749d6d84788",

"term": "First Term",

"school": "72a2c9c8-d02d-4eaf-a931-7c3a08769f00",

"status": "Pending",

"created\_at": "2025-05-08T14:15:12.925Z",

"updated\_at": "2025-05-08T14:15:12.925Z",

"student\_firstname": "Blessing",

"student\_surname": "Ojo",

"student\_admission\_number": "2023/0032",

"student\_name": "Ojo Blessing",

"class\_year": "2025/2026",

"class\_arm": "Science",

"department": "Science",

"subject\_name": "Biology",

"term\_name": "First Term",

"school\_name": "Victory Academy"

}

**✏️ PUT: Update a Registration (Status/Subject)**

**🧾 Request Example:**

{

"status": "Approved"

}

**✅ PUT Success Response:**

{

"registration\_id": "693cd1e5-1a29-4e63-bbb2-7f8a1880d924",

"status": "Approved",

...

}

**❌ DELETE: Remove a Registration**

**✅ DELETE Response:**

{

"message": "Registration deleted successfully."

}

Ready for the next endpoint:  
GET/PUT /assignments/registration-control/ (SubjectRegistrationControlView)?

Here is the documentation for the next endpoint:

**✅ Endpoint:**

GET /assignments/registration-control/  
PUT /assignments/registration-control/

**View:** SubjectRegistrationControlView  
**Permissions:** Only **School Admins** can access

**📤 GET: View Subject Registration Control**

This fetches the current registration status for the school (whether students can register for subjects).

**✅ Sample GET Response:**

{

"school": "f274cc35-5a7f-45e3-92ec-9e51e9d1dc57",

"school\_name": "Gracefield High School",

"is\_open": true,

"start\_date": "2025-05-01",

"end\_date": "2025-05-30"

}

**✏️ PUT: Update Registration Window**

Use this to open/close subject registration or modify its window (start and end date).

**🧾 Sample PUT Request:**

{

"is\_open": true,

"start\_date": "2025-05-10",

"end\_date": "2025-05-31"

}

**✅ Sample PUT Response:**

{

"school": "f274cc35-5a7f-45e3-92ec-9e51e9d1dc57",

"is\_open": true,

"start\_date": "2025-05-10",

"end\_date": "2025-05-31"

}

Ready for the next endpoint:  
PUT /assignments/subject-registration-status/<uuid:registration\_id>/ (UpdateSubjectRegistrationStatusView)?

Here is the documentation for the next and final endpoint in this batch:

**✅ Endpoint:**

PUT /assignments/subject-registration-status/<uuid:registration\_id>/  
**View:** UpdateSubjectRegistrationStatusView

**🔐 Permissions:**

* ✅ **School Admins**
* ✅ **Class Teachers** (only for students in their assigned classes)

**🧾 Purpose:**

Update the status of a student's subject registration entry (e.g., Pending → Approved/Rejected).

**🧾 Sample Request:**

{

"status": "Approved"

}

Valid options for status:

* "Pending"
* "Approved"
* "Rejected"

**✅ Sample Success Response:**

{

"status": "Approved"

}

**❌ Sample Error (Invalid Status):**

{

"status": [

"Status must be Pending, Approved, or Rejected."

]

}

**❌ Sample Permission Denied (Class Teacher trying to update outside assigned class):**

{

"detail": "You do not have permission to update this student's subject registration."

}

✅ That completes the school\_config module endpoints!  
Would you like to proceed to the next module or review a summary of what we’ve documented so far?