RESULT API

Not at all — we proceed **endpoint by endpoint**, reviewing:

1. **Path**
2. **Purpose**
3. **Allowed methods**
4. **Permissions**
5. **Expected request/response structures**
6. **Relevant serializers/views**

Let's begin with the first endpoint:

**🔹 Endpoint 1: Result Visibility List/Create**

POST /result/result-visibility/

GET /result/result-visibility/

**🔸 Purpose:**

* **GET**: List existing result visibility settings for a school.
* **POST**: Create a visibility control if none exists (one per school).

**🔸 Permissions:**

* Only accessible to: IsschoolAdmin

**🔸 Serializer:**

* ResultVisibilityControlSerializer

**🔸 Fields:**

{

"id": "uuid",

"school\_name": "string (read-only)",

"term\_result\_open": true/false,

"annual\_result\_open": true/false,

"updated\_at": "timestamp (read-only)"

}

**🔸 Sample POST Request:**

{

"term\_result\_open": true,

"annual\_result\_open": false

}

**🔸 Responses:**

* ✅ **201 Created**

{

"id": "9e8e7b26-4cb2-4dd2-9e47-cc9c4fae601d",

"school\_name": "Greenfield High",

"term\_result\_open": true,

"annual\_result\_open": false,

"updated\_at": "2025-05-08T10:25:00.123Z"

}

* ⚠️ **400 Bad Request** (If already exists)

{

"detail": "Result visibility control already exists for this school."

}

Ready for the **next endpoint**?

Great! Here's the next endpoint breakdown:

**🔹 Endpoint 2: Result Visibility Detail/Update**

GET /result/result-visibility/<uuid:pk>/

PUT /result/result-visibility/<uuid:pk>/

PATCH /result/result-visibility/<uuid:pk>/

**🔸 Purpose:**

* Retrieve or update the result visibility settings for a specific school instance.

**🔸 Permissions:**

* Only IsschoolAdmin can access this.

**🔸 Serializer:**

* ResultVisibilityControlSerializer

**🔸 Sample GET Response:**

{

"id": "9e8e7b26-4cb2-4dd2-9e47-cc9c4fae601d",

"school\_name": "Greenfield High",

"term\_result\_open": true,

"annual\_result\_open": false,

"updated\_at": "2025-05-08T10:25:00.123Z"

}

**🔸 Sample PATCH Request:**

{

"term\_result\_open": false,

"annual\_result\_open": true

}

**🔸 Sample PATCH Response:**

{

"id": "9e8e7b26-4cb2-4dd2-9e47-cc9c4fae601d",

"school\_name": "Greenfield High",

"term\_result\_open": false,

"annual\_result\_open": true,

"updated\_at": "2025-05-08T11:05:44.412Z"

}

Ready to move to **Assessment Category List/Create**?

Here’s the next endpoint breakdown:

**🔹 Endpoint 3: Assessment Category List & Create**

GET /result/assessment-categories/

POST /result/assessment-categories/

**🔸 Purpose:**

* GET: List all assessment categories (e.g., Tests, Assignments) for the authenticated school.
* POST: Create a new assessment category (with number of times and max score per one).

**🔸 Permissions:**

* Only accessible to authenticated users with the IsschoolAdmin role.

**🔸 Serializer:**

* AssessmentCategorySerializer

**🔸 Sample GET Response:**

[

{

"assessment\_category\_id": "bd7d89e7-1212-46a4-a621-2c4162a3dc21",

"school\_name": "Springfield Academy",

"assessment\_name": "Weekly Test",

"number\_of\_times": 3,

"max\_score\_per\_one": 10,

"created\_at": "2025-05-08T09:15:10.332Z",

"updated\_at": "2025-05-08T09:15:10.332Z"

}

]

**🔸 Sample POST Request:**

{

"assessment\_name": "Assignment",

"number\_of\_times": 2,

"max\_score\_per\_one": 15

}

**🔸 Sample POST Response:**

{

"assessment\_category\_id": "fcb3b90c-b7ae-4531-9429-ffb92f8dd93e",

"school\_name": "Springfield Academy",

"assessment\_name": "Assignment",

"number\_of\_times": 2,

"max\_score\_per\_one": 15,

"created\_at": "2025-05-08T11:47:55.089Z",

"updated\_at": "2025-05-08T11:47:55.089Z"

}

Ready for **Assessment Category Detail (GET/DELETE)**?

Here's the next endpoint:

**🔹 Endpoint 4: Assessment Category Detail (Retrieve & Delete)**

GET /result/assessment-categories/<uuid:assessment\_category\_id>/

DELETE /result/assessment-categories/<uuid:assessment\_category\_id>/

**🔸 Purpose:**

* GET: Retrieve the details of a single assessment category.
* DELETE: Delete the selected assessment category.

**🔸 Permissions:**

* Requires authentication and IsschoolAdmin role.

**🔸 Serializer Used:**

* AssessmentCategorySerializer

**✅ Sample GET Response:**

{

"assessment\_category\_id": "bd7d89e7-1212-46a4-a621-2c4162a3dc21",

"school\_name": "Springfield Academy",

"assessment\_name": "Assignment",

"number\_of\_times": 2,

"max\_score\_per\_one": 15,

"created\_at": "2025-05-08T09:15:10.332Z",

"updated\_at": "2025-05-08T09:15:10.332Z"

}

**🗑️ Sample DELETE Response:**

{

"detail": "Assessment category deleted successfully."

}

*(Django REST Framework usually returns a 204 No Content on successful deletion)*

Ready for **Result Configuration (List & Create)**?

Here is the next endpoint:

**🔹 Endpoint 5: Result Configuration List & Create**

GET /result/result-configurations/

POST /result/result-configurations/

**🔸 Purpose:**

* GET: Retrieve the current result configuration for the school.
* POST: Create a new result configuration (allowed only if one doesn't already exist).

**🔸 Permissions:**

* Only accessible to users with the IsschoolAdmin permission.

**🔸 Serializer Used:**

* ResultConfigurationSerializer

**🔸 Validation Rule:**

* The sum of total\_ca\_score and total\_exam\_score must equal **100**.

**🧾 Sample POST Body:**

{

"total\_ca\_score": 40,

"total\_exam\_score": 60

}

**✅ Sample Response (POST / GET):**

{

"id": "f9ab4127-49c0-4a8e-b6d0-b1a19ad7fa0c",

"school\_name": "Springfield Academy",

"total\_ca\_score": 40,

"total\_exam\_score": 60

}

**⚠️ Error Response:**

{

"non\_field\_errors": ["Total CA and Exam scores must sum up to 100."]

}

Ready for the next one: **Result Configuration Detail (Retrieve, Update, Delete)**?

Here is the next endpoint:

**🔹 Endpoint 6: Result Configuration Detail (Retrieve, Update, Delete)**

GET /result/result-configurations/<uuid:id>/

PUT /result/result-configurations/<uuid:id>/

PATCH /result/result-configurations/<uuid:id>/

DELETE /result/result-configurations/<uuid:id>/

**🔸 Purpose:**

* Retrieve, update, or delete a specific result configuration by its ID.

**🔸 Permissions:**

* Only accessible to IsschoolAdmin.

**🔸 Validation Rule on Update:**

* The updated total\_ca\_score and total\_exam\_score **must sum to 100**.

**🧾 Sample PUT Body:**

{

"total\_ca\_score": 45,

"total\_exam\_score": 55

}

**✅ Sample Response (GET or PUT):**

{

"id": "f9ab4127-49c0-4a8e-b6d0-b1a19ad7fa0c",

"school\_name": "Springfield Academy",

"total\_ca\_score": 45,

"total\_exam\_score": 55

}

**⚠️ Error Example:**

{

"non\_field\_errors": ["Total CA and Exam scores must sum up to 100."]

}

Shall we move on to **Annual Weight Config List & Create**?

Here is the next endpoint:

**🔹 Endpoint 7: Annual Weight Config List & Create**

GET /result/annual-weight-configs/

POST /result/annual-weight-configs/

**🔸 Purpose:**

* GET: List all configured annual result weightings for the authenticated School Admin.
* POST: Create a new weight configuration for a specific class\_year and department.

**🔸 Permissions:**

* Only accessible to IsschoolAdmin.

**🔸 Validation Rule (POST):**

* The sum of first\_term\_weight, second\_term\_weight, and third\_term\_weight **must be exactly 1.0**.
* Each school/class\_year/department combination must be unique.

**🧾 Sample POST Body:**

{

"class\_year": "3884bb10-d48a-4dc5-a542-2edaa13cc20f",

"department": "f13b92b5-8b0a-41c5-92a2-b7ad09a3d144",

"first\_term\_weight": 0.2,

"second\_term\_weight": 0.3,

"third\_term\_weight": 0.5

}

**✅ Sample Response:**

{

"id": "3c5a3ff7-6e7a-4030-9f13-56f49a066f95",

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

"class\_year": "3884bb10-d48a-4dc5-a542-2edaa13cc20f",

"class\_year\_name": "JSS 3",

"department": "f13b92b5-8b0a-41c5-92a2-b7ad09a3d144",

"department\_name": "Junior",

"first\_term\_weight": 0.2,

"second\_term\_weight": 0.3,

"third\_term\_weight": 0.5

}

**⚠️ Error Example:**

{

"non\_field\_errors": ["Sum of term weights must be equal to 1.0"]

}

Ready for **Annual Weight Config Detail**?

Here is the next endpoint:

**🔹 Endpoint 8: Annual Weight Config Detail (Retrieve, Update, Delete)**

GET /result/annual-weight-configs/<uuid:id>/

PUT /result/annual-weight-configs/<uuid:id>/

PATCH /result/annual-weight-configs/<uuid:id>/

DELETE /result/annual-weight-configs/<uuid:id>/

**🔸 Purpose:**

* Retrieve, update, or delete a specific annual result weight configuration.

**🔸 Permissions:**

* Only accessible to IsschoolAdmin.

**🔸 Validation Rule (PUT/PATCH):**

* The sum of weights (first\_term\_weight, second\_term\_weight, third\_term\_weight) must be exactly 1.0.

**🧾 Sample PATCH Request:**

{

"second\_term\_weight": 0.25,

"third\_term\_weight": 0.55

}

**✅ Sample Response:**

{

"id": "3c5a3ff7-6e7a-4030-9f13-56f49a066f95",

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

"class\_year": "3884bb10-d48a-4dc5-a542-2edaa13cc20f",

"class\_year\_name": "JSS 3",

"department": "f13b92b5-8b0a-41c5-92a2-b7ad09a3d144",

"department\_name": "Junior",

"first\_term\_weight": 0.2,

"second\_term\_weight": 0.25,

"third\_term\_weight": 0.55

}

Shall we proceed to **Grading System List & Create**?

Here is the next endpoint:

**🔹 Endpoint 9: Grading System List and Create**

GET /result/grading-systems/

POST /result/grading-systems/

**🔸 Purpose:**

* GET: Retrieve all grading ranges for the authenticated school.
* POST: Create a new grading rule (e.g., A = 70–100).

**🔸 Permissions:**

* Only IsschoolAdmin can access.

**🧾 Sample POST Request:**

{

"min\_score": 70,

"max\_score": 100,

"grade": "A",

"remarks": "Excellent performance"

}

**✅ Sample Response:**

{

"id": "2c4d1f7b-0df2-4b95-8e65-dc7a8d19d719",

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

"min\_score": 70,

"max\_score": 100,

"grade": "A",

"remarks": "Excellent performance"

}

Would you like to continue to the **Grading System Detail (Retrieve/Update/Delete)** next?

Here is the next endpoint:

**🔹 Endpoint 10: Grading System Detail (Retrieve, Update, Delete)**

GET /result/grading-systems/<uuid:id>/

PUT /result/grading-systems/<uuid:id>/

PATCH /result/grading-systems/<uuid:id>/

DELETE /result/grading-systems/<uuid:id>/

**🔸 Purpose:**

Retrieve, update, or delete a specific grading range by its id.

**🔸 Permissions:**

* Only accessible by IsschoolAdmin.

**🧾 Sample GET Response:**

{

"id": "2c4d1f7b-0df2-4b95-8e65-dc7a8d19d719",

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

"min\_score": 70,

"max\_score": 100,

"grade": "A",

"remarks": "Excellent performance"

}

**🧾 Sample PATCH Request:**

{

"remarks": "Outstanding performance"

}

**🧾 Sample DELETE Response:**

{

"detail": "Grading system deleted successfully."

}

Shall we proceed to the **Score Per Assessment (Instance-based scores)** endpoint?

Here is the next endpoint:

**🔹 Endpoint 11: Assessment Scores (Instance) - List and Create**

GET /result/assessment-scores/

POST /result/assessment-scores/

**🔸 Purpose:**

* GET: List all assessment instance scores (by teachers, students, school admins).
* POST: Create a new instance-based assessment score (only by assigned subject teachers).

**🔸 Permissions:**

* GET: Accessible by IsschoolAdmin, ISteacher, or ISstudent.
* POST: Only ISteacher (teacher must be assigned to the subject and class).

**🧾 Sample GET Response:**

[

{

"scoreperassessment\_id": "f8aa481e-c7a1-11ee-bb38-acde48001122",

"registration": "d72a1cce-c7a1-11ee-bb38-acde48001122",

"category": "Midterm Test",

"assessment\_name": "Midterm Test",

"instance\_number": 1,

"score": 8.0,

"student\_name": "Opeyemi Dare",

"created\_at": "2025-05-08T11:20:33.170Z",

"updated\_at": "2025-05-08T11:20:33.170Z"

}

]

**🧾 Sample POST Request:**

{

"registration": "d72a1cce-c7a1-11ee-bb38-acde48001122",

"category": "18a38ee1-c7a1-11ee-bb38-acde48001122",

"instance\_number": 1,

"score": 7.5

}

**🔄 Logic:**

* After saving a score instance, the system auto-updates the **total score** in ScoreObtainedPerAssessment for that category.

Shall we continue to the **Retrieve/Update/Delete for assessment scores** endpoint?

Here is the next endpoint:

**🔹 Endpoint 12: Assessment Score (Instance) - Retrieve, Update, Delete**

GET /result/assessment-scores/<uuid:scoreperassessment\_id>/

PUT /result/assessment-scores/<uuid:scoreperassessment\_id>/

PATCH /result/assessment-scores/<uuid:scoreperassessment\_id>/

DELETE /result/assessment-scores/<uuid:scoreperassessment\_id>/

**🔸 Purpose:**

* Retrieve a specific assessment instance score.
* Update or delete it if you're the assigned teacher.

**🔸 Permissions:**

* GET: ISteacher, ISstudent, or IsschoolAdmin.
* PUT/PATCH/DELETE: Only ISteacher.

**🧾 Sample GET Response:**

{

"scoreperassessment\_id": "f8aa481e-c7a1-11ee-bb38-acde48001122",

"registration": "d72a1cce-c7a1-11ee-bb38-acde48001122",

"category": "18a38ee1-c7a1-11ee-bb38-acde48001122",

"assessment\_name": "Midterm Test",

"instance\_number": 1,

"score": 7.5,

"student\_name": "Opeyemi Dare",

"created\_at": "2025-05-08T11:20:33.170Z",

"updated\_at": "2025-05-08T11:20:33.170Z"

}

**🧾 Sample PUT Request:**

{

"registration": "d72a1cce-c7a1-11ee-bb38-acde48001122",

"category": "18a38ee1-c7a1-11ee-bb38-acde48001122",

"instance\_number": 1,

"score": 9.0

}

**🔄 Logic:**

* Updates also trigger recalculation of the corresponding ScoreObtainedPerAssessment total.
* Deletion also triggers the total recalculation for the remaining instances.

Ready to continue to **total scores per assessment (view-only)?**

Here is the next endpoint:

**🔹 Endpoint 13: Total Scores per Assessment Category**

GET /result/assessment-score-totals/

**🔸 Purpose:**

View the total scores obtained by students per assessment category (e.g., "Test", "Quiz").

**🔸 Permissions:**

Accessible to:

* ISstudent (own scores)
* ISteacher (students in assigned classes/subjects)
* IsschoolAdmin (all students)

**🧾 Sample Response:**

[

{

"scoreperassessment\_id": "7c2f2c72-c7a3-11ee-bb38-acde48001122",

"registration": "d72a1cce-c7a1-11ee-bb38-acde48001122",

"category": "Midterm",

"assessment\_name": "Midterm",

"total\_score": 24.5,

"student\_name": "Opeyemi Dare",

"created\_at": "2025-05-08T11:55:18.170Z",

"updated\_at": "2025-05-08T11:55:18.170Z"

},

{

"scoreperassessment\_id": "9a0127ee-c7a3-11ee-bb38-acde48001122",

"registration": "d72a1cce-c7a1-11ee-bb38-acde48001122",

"category": "Quiz",

"assessment\_name": "Quiz",

"total\_score": 18.0,

"student\_name": "Opeyemi Dare",

"created\_at": "2025-05-08T12:01:03.210Z",

"updated\_at": "2025-05-08T12:01:03.210Z"

}

]

**🔄 Logic:**

* These totals are updated automatically when instance scores are added, updated, or deleted.

Ready for the next one on **exam scores**?

Here is the next endpoint:

**🔹 Endpoint 14: Exam Scores**

GET /result/exam-scores/

POST /result/exam-scores/

**🔸 Purpose:**

* **GET**: View all exam scores for the active term.
* **POST**: Allow teachers to submit exam scores for students in their assigned classes/subjects.

**🔸 Permissions:**

* **GET**: IsschoolAdmin, ISteacher, ISstudent
* **POST**: ISteacher only

**🔸 POST Payload Example:**

{

"registration": "d72a1cce-c7a1-11ee-bb38-acde48001122",

"score": 65.0

}

**🧾 Sample GET Response:**

[

{

"examscore\_id": "e19b3b42-c7a3-11ee-bb38-acde48001122",

"registration": "d72a1cce-c7a1-11ee-bb38-acde48001122",

"subject\_name": "Mathematics",

"score": 65.0,

"student\_name": "Opeyemi Dare",

"created\_at": "2025-05-08T12:05:44.579Z",

"updated\_at": "2025-05-08T12:05:44.579Z"

}

]

**🔄 Logic:**

* A teacher can only submit scores for:
  + The **active term**
  + **Subjects** and **class arms** they are assigned to
* Multiple submissions overwrite previous values.

Shall we continue with **updating and retrieving individual exam scores** next?

Here is the next endpoint:

**🔹 Endpoint 15: Retrieve, Update, or Delete an Exam Score**

GET /result/exam-scores/<uuid:examscore\_id>/

PUT /result/exam-scores/<uuid:examscore\_id>/

PATCH /result/exam-scores/<uuid:examscore\_id>/

DELETE /result/exam-scores/<uuid:examscore\_id>/

**🔸 Purpose:**

* View, update, or delete a specific exam score for a student.

**🔸 Permissions:**

* **GET**: ISteacher, ISstudent, IsschoolAdmin
* **PUT/PATCH/DELETE**: ISteacher only

**🔸 PUT/PATCH Payload Example:**

{

"score": 78.5

}

**🧾 Sample GET Response:**

{

"examscore\_id": "e19b3b42-c7a3-11ee-bb38-acde48001122",

"registration": "d72a1cce-c7a1-11ee-bb38-acde48001122",

"subject\_name": "Mathematics",

"score": 78.5,

"student\_name": "Opeyemi Dare",

"created\_at": "2025-05-08T12:05:44.579Z",

"updated\_at": "2025-05-08T12:30:20.129Z"

}

**🔄 Logic:**

* Only the assigned **subject teacher** can update or delete exam scores.
* Students can **view** their scores.
* Admins can **view** all scores for the active term.

Continue to the **continuous assessment** endpoint?

Here is the next endpoint:

**🔹 Endpoint 16: List Continuous Assessment Records**

GET /result/continuous-assessments/

**🔸 Purpose:**

Retrieve the list of Continuous Assessment (CA) totals for students in the current term.

**🔸 Permissions:**

* Accessible by:
  + ISteacher
  + ISstudent
  + IsschoolAdmin

**🧾 Sample Response:**

[

{

"continuous\_assessment\_id": "c21d5ae0-c7a5-11ee-bb38-acde48001122",

"registration": "8cd121b4-c7a2-11ee-bb38-acde48001122",

"subject\_name": "Physics",

"ca\_total": 24.6,

"student\_name": "Adebayo Joy",

"created\_at": "2025-05-08T13:44:32.541Z",

"updated\_at": "2025-05-08T13:44:32.541Z"

},

...

]

**🔄 Logic:**

* Automatically computes missing CA totals for the active term using compute\_continuous\_assessment.
* Teachers only see students in their assigned subjects and classes.
* Students see only their own CA totals.
* Admins see all CA totals for the active term.

Ready to continue with the **GET individual CA detail** endpoint?

Here is the next endpoint:

**🔹 Endpoint 16: List Continuous Assessment Records**

GET /result/continuous-assessments/

**🔸 Purpose:**

Retrieve the list of Continuous Assessment (CA) totals for students in the current term.

**🔸 Permissions:**

* Accessible by:
  + ISteacher
  + ISstudent
  + IsschoolAdmin

**🧾 Sample Response:**

[

{

"continuous\_assessment\_id": "c21d5ae0-c7a5-11ee-bb38-acde48001122",

"registration": "8cd121b4-c7a2-11ee-bb38-acde48001122",

"subject\_name": "Physics",

"ca\_total": 24.6,

"student\_name": "Adebayo Joy",

"created\_at": "2025-05-08T13:44:32.541Z",

"updated\_at": "2025-05-08T13:44:32.541Z"

},

...

]

**🔄 Logic:**

* Automatically computes missing CA totals for the active term using compute\_continuous\_assessment.
* Teachers only see students in their assigned subjects and classes.
* Students see only their own CA totals.
* Admins see all CA totals for the active term.

Ready to continue with the **GET individual CA detail** endpoint?

Here is the next endpoint:

**🔹 Endpoint 17: Retrieve a Specific Continuous Assessment Record**

GET /result/continuous-assessments/<uuid:continuous\_assessment\_id>/

**🔸 Purpose:**

Retrieve detailed information about a specific student's Continuous Assessment record.

**🔸 Permissions:**

* Accessible by:
  + ISteacher
  + ISstudent
  + IsschoolAdmin

**🧾 Sample Response:**

{

"continuous\_assessment\_id": "5dce0524-c7a6-11ee-bb38-acde48001122",

"registration": "2bb5584c-c7a2-11ee-bb38-acde48001122",

"subject\_name": "Mathematics",

"ca\_total": 27.5,

"student\_name": "Okoro Blessing",

"created\_at": "2025-05-08T14:01:21.132Z",

"updated\_at": "2025-05-08T14:01:21.132Z"

}

**🔄 Logic:**

* Uses the continuous\_assessment\_id to retrieve the record.
* Access is filtered by the requester's role:
  + Teachers: students in assigned subjects/classes.
  + Students: own record.
  + Admins: any student under the school.

Continue with the **result list view** endpoint?

Here is the next endpoint:

**🔹 Endpoint 20: List All Annual Results**

GET /result/annual-results/

**🔸 Purpose:**

Returns all computed **annual results** for the current active academic year for:

* A student (own results)
* A class teacher (students in their class)
* A subject teacher (students in their assigned subject & class)
* A school admin (all students in the school)

**🔸 Permissions:**

Accessible by:

* ISstudent
* ISteacher
* IsClassTeacher
* IsschoolAdmin

**🧾 Sample Response:**

[

{

"annual\_result\_id": "fc86b6a6-c7ab-11ee-bb38-acde48001122",

"registration": "c70adf66-c7ab-11ee-bb38-acde48001122",

"student": {

"student\_id": "d5095c40-c7a9-11ee-bb38-acde48001122",

"first\_name": "Peter",

"last\_name": "Akinwale",

"gender": "male",

"admission\_number": 10523,

"status": "Active"

},

"subject": {

"subject\_id": "b42cd85e-c7a9-11ee-bb38-acde48001122",

"name": "Mathematics"

},

"first\_term\_score": 75.0,

"second\_term\_score": 88.0,

"third\_term\_score": 85.0,

"third\_term\_detail": {

"ca\_total": 24.0,

"exam\_score": 65.0,

"assessment\_scores": [

{

"assessment\_name": "Weekly Test",

"obtained\_score": 12.0,

"max\_score": 20.0

},

{

"assessment\_name": "Project",

"obtained\_score": 12.0,

"max\_score": 20.0

}

]

},

"annual\_average": 82.6,

"grade": "A",

"remarks": "Excellent",

"created\_at": "2025-05-08T15:39:14.000Z",

"updated\_at": "2025-05-08T15:39:14.000Z"

}

]

Continue to view **Retrieve a Single Annual Result** endpoint?

Here is the next endpoint:

**🔹 Endpoint 21: Retrieve a Single Annual Result**

GET /result/annual-results/<uuid:annual\_result\_id>/

**🔸 Purpose:**

Fetches the **details** of one annual result record by its ID.

**🔸 Permissions:**

Accessible by:

* ISstudent
* ISteacher
* IsClassTeacher
* IsschoolAdmin

**📥 Path Parameter:**

* annual\_result\_id (UUID): The ID of the annual result to retrieve.

**🧾 Sample Response:**

{

"annual\_result\_id": "fc86b6a6-c7ab-11ee-bb38-acde48001122",

"registration": "c70adf66-c7ab-11ee-bb38-acde48001122",

"student": {

"student\_id": "d5095c40-c7a9-11ee-bb38-acde48001122",

"first\_name": "Peter",

"last\_name": "Akinwale",

"gender": "male",

"admission\_number": 10523,

"status": "Active"

},

"subject": {

"subject\_id": "b42cd85e-c7a9-11ee-bb38-acde48001122",

"name": "Mathematics"

},

"first\_term\_score": 75.0,

"second\_term\_score": 88.0,

"third\_term\_score": 85.0,

"third\_term\_detail": {

"ca\_total": 24.0,

"exam\_score": 65.0,

"assessment\_scores": [

{

"assessment\_name": "Weekly Test",

"obtained\_score": 12.0,

"max\_score": 20.0

},

{

"assessment\_name": "Project",

"obtained\_score": 12.0,

"max\_score": 20.0

}

]

},

"annual\_average": 82.6,

"grade": "A",

"remarks": "Excellent",

"created\_at": "2025-05-08T15:39:14.000Z",

"updated\_at": "2025-05-08T15:39:14.000Z"

}

Continue to **Class Teacher Comment Endpoints**?

Here is the next endpoint:

**🔹 Endpoint 22: List/Create Class Teacher Comments**

GET /result/classteacher-comments/

POST /result/classteacher-comments/

**🔸 Purpose:**

* GET: View **all class teacher comments** made by the authenticated class teacher for their assigned class and active term.
* POST: **Create a new comment** for a student based on skill ratings.

**🔸 Permissions:**

* Requires authentication.
* Only accessible by users with IsClassTeacher role.

**📥 Fields Required for POST:**

| **Field** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| student | UUID | ✅ | ID of the student receiving the comment. |
| skills | Dict | ✅ | A dictionary of skills and their ratings (e.g., {"neatness": "Excellent"}) |

⚠️ Skills are used to generate an **AI-powered comment** based on a FLAN-T5 model.

**✅ Example Request Body (POST):**

{

"student": "8dbfa080-1c1f-4b9e-b7d2-baa490b49e93",

"skills": {

"neatness": "Excellent",

"punctuality": "Very Good",

"leadership": "Good"

}

}

**🧾 Sample Response (POST):**

{

"classteacher\_comment\_id": "b5fe0c40-9a20-4c89-8b0f-1be1f5f1eeb0",

"student": "8dbfa080-1c1f-4b9e-b7d2-baa490b49e93",

"school": "620e1a5d-5275-44b5-9ed5-7c3dfef80c7d",

"term": "2024/2025 - Second Term",

"comment": "He shows exceptional commitment in neatness, demonstrates strong ability in punctuality, and performs well in leadership.",

"created\_at": "2025-05-08T16:40:11.000Z"

}

Ready for **Endpoint 23: Retrieve/Update/Delete a Class Teacher Comment**?

Here is the next endpoint:

**🔹 Endpoint 23: Retrieve, Update, or Delete a Class Teacher Comment**

GET /result/classteacher-comments/<uuid:classteacher\_comment\_id>/

PUT /result/classteacher-comments/<uuid:classteacher\_comment\_id>/

PATCH /result/classteacher-comments/<uuid:classteacher\_comment\_id>/

DELETE /result/classteacher-comments/<uuid:classteacher\_comment\_id>/

**🔸 Purpose:**

* GET: Retrieve a specific comment made by the class teacher.
* PUT/PATCH: Update the content of the comment or skills.
* DELETE: Delete a comment created by the class teacher.

**🔸 Permissions:**

* Requires authentication.
* Only accessible by users with the IsClassTeacher role.

**📥 Fields for Update (PUT/PATCH):**

| **Field** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| skills | Dict | ❌ | New dictionary of skills (to regenerate comment). |
| comment | String | ❌ | Optionally override AI comment with manual input. |

If skills are provided, a new AI-generated comment will be produced.

**✅ Sample PATCH Request:**

{

"skills": {

"neatness": "Good",

"empathy": "Very Good"

}

}

**🧾 Sample Response:**

{

"classteacher\_comment\_id": "b5fe0c40-9a20-4c89-8b0f-1be1f5f1eeb0",

"student": "8dbfa080-1c1f-4b9e-b7d2-baa490b49e93",

"school": "620e1a5d-5275-44b5-9ed5-7c3dfef80c7d",

"term": "2024/2025 - Second Term",

"comment": "He demonstrates strong ability in empathy and performs well in neatness.",

"created\_at": "2025-05-08T16:45:11.000Z"

}

Let me know when you're ready for **Endpoint 24: Full Student Result View**.

Here is the next endpoint:

**🔹 Endpoint 24: Full Student Result View (All Terms or Annual)**

GET /result/full-student-result/<uuid:student\_id>/?year\_id=<uuid>&term\_id=<uuid:optional>

**🔸 Purpose:**

Retrieve a student's complete result for:

* A **specific term** (if term\_id is provided),
* OR the **entire academic year** (if term\_id is omitted).

**🔸 Permissions:**

* Must be authenticated.
* Access allowed for:
  + The student themselves (ISstudent)
  + The student’s class teacher (IsClassTeacher)
  + School admin (IsschoolAdmin)

**📥 Required Query Parameters:**

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| year\_id | UUID | ✅ | Academic year of the result |
| term\_id | UUID | ❌ | If provided, returns term result. |

**🔐 Result Visibility:**

* Term result is shown **only** if term\_result\_open is True.
* Annual result is shown **only** if annual\_result\_open is True.

**✅ Sample Term Result Response:**

{

"school": {

"school\_id": "123...",

"school\_name": "AlgorithmX College",

...

},

"student": {

"student\_id": "456...",

"first\_name": "John",

"last\_name": "Doe",

...

},

"year": "2024/2025",

"term": "First Term",

"term\_results": [

{

"subject": {

"subject\_id": "789...",

"name": "Mathematics"

},

"class\_year": "JSS1",

"class\_arm": "A",

"ca\_total": 24.0,

"exam\_score": 56.0,

"total\_score": 80.0,

"grade": "A",

"remarks": "Excellent",

"assessments": [

{

"assessment\_name": "Homework",

"obtained\_score": 10.0,

"max\_score": 10.0

},

...

]

}

]

}

Let me know when you're ready for **Endpoint 25: Broadsheet View and Excel Export**.

Here is the next endpoint:

**🔹 Endpoint 25: Broadsheet View and Excel Export**

GET /result/broadsheet/?year\_id=<uuid>&term\_id=<uuid:optional>&class\_year\_id=<uuid:optional>&class\_arm\_id=<uuid:optional>&download=excel:optional

**🔸 Purpose:**

Return either:

* A **JSON broadsheet** view of term or annual results
* Or an **Excel file** if download=excel is included

**🔸 Permissions:**

* Authenticated users with:
  + IsschoolAdmin OR
  + IsClassTeacher

**📥 Required Query Parameters:**

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| year\_id | UUID | ✅ | Academic year of the broadsheet |
| term\_id | UUID | ❌ | Include for **term** broadsheet; omit for **annual** |
| class\_year\_id | UUID | ❌ | Optionally filter for a specific class year |
| class\_arm\_id | UUID | ❌ | Optionally filter for a specific class arm |
| download | str | ❌ | If set to excel, returns a downloadable Excel file |

**📤 Response (JSON):**

{

"broadsheet\_type": "term",

"year": "2024/2025",

"term": "First Term",

"class\_year": "JSS1",

"class\_arm": "A",

"department": "Junior",

"pass\_mark": 50.0,

"subjects": ["Mathematics", "English"],

"students": [

{

"student\_id": "abc123",

"name": "Jane Doe",

"scores": {

"Mathematics": { "score": 85.0, "grade": "A" },

"English": { "score": 70.0, "grade": "B" }

},

"average\_score": 77.5,

"position\_in\_class\_year": "1st",

"position\_in\_class\_arm": "1st",

"passed\_subjects": 2,

"failed\_subjects": 0

}

]

}

**📥 Excel Export:**

If download=excel is in the query, the response is a .xlsx file titled:

broadsheet\_JSS1\_A\_2024\_2025\_First\_Term.xlsx

Let me know when you're ready to continue to **AI-Based Class Teacher Comment Integration**.