Great — let’s begin with the **first endpoint** in your notification module:

**🔹 Endpoint: List & Create Notifications**

**URL:** POST & GET /notifications/  
**View:** NotificationListCreateView

**🔸 GET /notifications/**

* **Permission:** IsschoolAdmin
* **Function:** Lists all notifications created in the logged-in School Admin’s school.
* **Serializer:** NotificationSerializer
* **Response (Sample):**

[

{

"id": "uuid-of-notification",

"title": "Exam Notice",

"content": "The exam starts on Monday.",

"recipient\_user": null,

"recipient\_group": "Teacher",

"notification\_type": "Information",

"school": "uuid-of-school",

"created\_at": "2025-05-08T10:00:00Z",

"is\_read": false

},

...

]

**🔸 POST /notifications/**

* **Permission:** IsschoolAdmin
* **Serializer:** NotificationCreateSerializer
* **Fields Required:**
  + title (str)
  + content (str)
  + recipient\_user (optional UUID)
  + recipient\_group (e.g., "Teacher", "Student", "Everyone")
  + notification\_type (e.g., "Alert", "Information", "Reminder")
* **Function:** Creates a new notification for the selected user or group.
* **Auto-added:** School is assigned from the authenticated School Admin.
* **Sample Request Body:**

{

"title": "Midterm Exam",

"content": "Midterm exams begin next week.",

"recipient\_group": "Student",

"notification\_type": "Reminder"

}

* **Sample Response:**

{

"title": "Midterm Exam",

"content": "Midterm exams begin next week.",

"recipient\_user": null,

"recipient\_group": "Student",

"notification\_type": "Reminder",

"school": "uuid-of-school",

"created\_at": "2025-05-08T11:20:00Z",

"is\_read": false

}

✅ Ready for the next endpoint?

Great — here’s the breakdown for the **second endpoint**:

**🔹 Endpoint: Retrieve, Update, or Delete a Notification**

**URL:** GET, PUT, PATCH, DELETE /notifications/<uuid:pk>/  
**View:** NotificationDetailView

**🔸 GET /notifications/<uuid:pk>/**

* **Permissions:** IsschoolAdmin, ISteacher, ISstudent
* **Function:** Retrieves details of a single notification if it belongs to the user’s school.
* **Serializer:** NotificationSerializer
* **Sample Response:**

{

"id": "uuid-of-notification",

"title": "Exam Notice",

"content": "The exam starts on Monday.",

"recipient\_user": null,

"recipient\_group": "Teacher",

"notification\_type": "Information",

"school": "uuid-of-school",

"created\_at": "2025-05-08T10:00:00Z",

"is\_read": false

}

**🔸 PUT or PATCH /notifications/<uuid:pk>/**

* **Permissions:** Same as above
* **Serializer Used:** NotificationUpdateSerializer
* **Fields Allowed to Update:**
  + title
  + content
  + is\_read
* **Sample Request Body:**

{

"title": "Updated Exam Notice",

"content": "Exam now starts on Tuesday.",

"is\_read": true

}

* **Sample Response:**

{

"title": "Updated Exam Notice",

"content": "Exam now starts on Tuesday.",

"is\_read": true

}

**🔸 DELETE /notifications/<uuid:pk>/**

* **Function:** Deletes the specified notification if it belongs to the user's school.
* **Response:**

{

"detail": "Notification deleted successfully."

}

✅ Ready for the next one?

Perfect — here's the breakdown for the **third endpoint**:

**🔹 Endpoint: Get 5 Most Recent Notifications**

**URL:** GET /notifications/recent/  
**View:** RecentNotificationsView

**🔸 Method: GET**

* **Permissions:** IsschoolAdmin
* **Function:** Returns the **5 most recent notifications** for the currently authenticated school admin’s school.
* **Serializer Used:** NotificationSerializer
* **Queryset Logic:**

Notification.objects.filter(school=self.request.user.schooladmin.school).order\_by('-created\_at')[:5]

**🔸 Sample Response:**

[

{

"id": "31b4e4fe-2e15-4eb1-a7de-f084b8c67e2d",

"title": "Midterm Exams",

"content": "Midterm exams start on May 10th.",

"recipient\_user": null,

"recipient\_group": "Everyone",

"notification\_type": "Announcement",

"school": "88a42df2-987c-4e0e-9a2e-88d1c35623ba",

"created\_at": "2025-05-08T11:45:23Z",

"is\_read": false

},

{

"id": "5cbb6bb4-4209-413b-9857-76493ac2ebcb",

"title": "Staff Meeting",

"content": "Meeting at 10AM Friday.",

"recipient\_user": null,

"recipient\_group": "Teacher",

"notification\_type": "Reminder",

"school": "88a42df2-987c-4e0e-9a2e-88d1c35623ba",

"created\_at": "2025-05-07T17:21:49Z",

"is\_read": false

},

...

]

✅ Shall we continue to the next one?

Great! Here's the breakdown for the **fourth endpoint**:

**🔹 Endpoint: List Notifications for Teachers and Everyone**

**URL:** GET /notifications/teacher-and-everyone/  
**View:** TeacherAndEveryoneNotificationView

**🔸 Method: GET**

* **Permissions:** IsschoolAdmin, ISteacher
* **Function:** Returns all notifications for the authenticated user's school where the recipient\_group is either "Teacher" or "Everyone".
* **Serializer Used:** NotificationSerializer
* **Queryset Logic:**

Notification.objects.filter(

school=self.request.user.schooladmin.school,

recipient\_group\_\_in=['Teacher', 'Everyone']

).order\_by('-created\_at')

**🔸 Sample Response:**

[

{

"id": "92fe9d79-f0c2-4f3d-8c9f-14ebd02bd780",

"title": "Weekly Report Deadline",

"content": "All teachers must submit reports by Friday.",

"recipient\_user": null,

"recipient\_group": "Teacher",

"notification\_type": "Reminder",

"school": "88a42df2-987c-4e0e-9a2e-88d1c35623ba",

"created\_at": "2025-05-07T08:30:00Z",

"is\_read": false

},

{

"id": "b93cbf72-2f3d-42d0-90d2-13a9be4c4024",

"title": "Holiday Notice",

"content": "School will be closed on Monday.",

"recipient\_user": null,

"recipient\_group": "Everyone",

"notification\_type": "Announcement",

"school": "88a42df2-987c-4e0e-9a2e-88d1c35623ba",

"created\_at": "2025-05-06T14:10:22Z",

"is\_read": false

}

]

✅ Ready for the final endpoint?

Here’s the final breakdown for the **last endpoint**:

**🔹 Endpoint: List Notifications for Students and Everyone**

**URL:** GET /notifications/student-and-everyone/  
**View:** StudentAndEveryoneNotificationView

**🔸 Method: GET**

* **Permissions:** IsschoolAdmin, ISstudent
* **Functionality:**  
  Returns all notifications for the authenticated user's school where the recipient\_group is either "Student" or "Everyone".
* **Serializer Used:** NotificationSerializer
* **Query Logic:**

Notification.objects.filter(

school=self.request.user.schooladmin.school,

recipient\_group\_\_in=['Student', 'Everyone']

).order\_by('-created\_at')

**🔸 Sample Response:**

[

{

"id": "349d4045-7d2b-437b-9622-bf81f832ad8c",

"title": "Midterm Exams Schedule",

"content": "Midterm exams start next Monday. Check the notice board.",

"recipient\_user": null,

"recipient\_group": "Student",

"notification\_type": "Alert",

"school": "88a42df2-987c-4e0e-9a2e-88d1c35623ba",

"created\_at": "2025-05-06T10:05:00Z",

"is\_read": false

},

{

"id": "d69e6fc2-f3fd-4d90-8158-fd964b6b2e8f",

"title": "Public Holiday Notice",

"content": "All students and staff are to observe a public holiday on Wednesday.",

"recipient\_user": null,

"recipient\_group": "Everyone",

"notification\_type": "Announcement",

"school": "88a42df2-987c-4e0e-9a2e-88d1c35623ba",

"created\_at": "2025-05-05T12:45:30Z",

"is\_read": false

}

]

✅ All notification endpoints have now been fully documented and walked through.

Would you like to proceed to the next module or generate documentation (e.g., Swagger/OpenAPI or markdown)?