

Notifications Dropdown - Frontend Implementation Notes

Overview

This document captures what was implemented on the frontend for the notification dropdown and what is still needed from the backend team to fully support the feature.

What We Implemented (Frontend)

1) API Integration

- Base path: `/api/v1` is already handled by `ApiClient`.
- Primary endpoints used:
 - `GET /dashboard/notifications`
 - `PUT /dashboard/notifications/{notificationId}/read`
 - `PUT /dashboard/notifications/read-all`
 - `DELETE /dashboard/notifications/{notificationId}` (attempted)
- Fallback delete endpoint:
 - `DELETE /notifications/{notificationId}` (legacy fallback if dashboard delete is not supported)

2) Response Mapping

The backend response is normalized to match existing UI types:

- `isRead` → `read`
- `timestamp` → `createdAt`
- `summary.unread` → `unreadCount`
- `pagination.pages` used to compute `hasMore`

3) Type Normalization

Backend type values are mapped to existing UI types:

Backend Type	UI Type
collaboration	collaboration
payment	payment_received
match	match
system	system
social	new_follower
engagement	achievement
reminder	system
marketplace	marketplace_order
analytics	system

4) userId Handling

- The backend response does not include `userId`. It is now optional in the UI type.
- If metadata contains `requesterId` or `senderId`, it is used as `userId`.

5) Dropdown UX

- The notifications list is scrollable.
- A dark scrollbar is applied for both light and dark themes.

What We Need From Backend

⚠ REQUIRED - Delete Endpoint Not Implemented

Issue: The delete notification functionality is currently broken because the backend does not implement DELETE for notifications.

Current Behavior:

- Frontend attempts: `DELETE /api/v1/dashboard/notifications/{notificationId}`
- Backend returns: 404 or 405 (Method Not Allowed)

- User sees error: “Failed to delete notification”

Required Implementation:

- **Endpoint:** DELETE /api/v1/dashboard/notifications/{notificationId}
- **Request:** DELETE request with notification ID in path
- **Authentication:** Bearer token required (already handled by apiClient)
- **Expected Response:**

```
{  
  "success": true,  
  "message": "Notification deleted"  
}
```

- **Error Responses:**

- 404 : Notification not found
- 401 : Unauthorized
- 500 : Server error

Priority: High - Delete button is visible in UI but non-functional, causing user confusion.

Other Requirements

1. Confirm notification type list and any missing types.
 - If new types exist, provide mapping guidance or add to frontend types.

Nice-to-Have

1. Return count for PUT /dashboard/notifications/read-all .
 - Current backend response has no count so UI cannot display accurate “X notifications updated”.
2. Provide userId directly in the notifications response.
 - This will improve grouping and attribution logic.
3. Add preferences endpoints (if planned):
 - GET /dashboard/notifications/preferences
 - PUT /dashboard/notifications/preferences

Current Behavior Summary

- The dropdown uses the correct Dashboard API (/dashboard/notifications)
- Mark as read and mark all as read work correctly
- The scrollbar styling is dark in both light and dark modes
- **Delete functionality is broken** - Backend does not implement DELETE endpoint
 - Frontend calls: DELETE /api/v1/dashboard/notifications/{notificationId}
 - Backend returns: 404/405 error
 - User sees error toast: "Failed to delete notification"