

## useAppNotificationsData

### Overview

The `useAppNotificationsData` hook provides a comprehensive solution for managing application notifications. It consolidates functionality for fetching, marking as read, and managing unread notification counts, offering an optimized way to handle notification-related tasks in a unified interface. Unlike hooks like `useFetchAppNotifications` or `useCountUnreadNotifications`, this hook centralizes multiple responsibilities, ensuring smoother notification workflows with added features such as message templating and pagination.

### Usage Example

```
import { useAppNotificationsData } from "@replyke/react-js";

function NotificationsPanel() {
  const {
    appNotifications,
    unreadAppNotificationsCount,
    loading,
    hasMore,
    loadMore,
    markNotificationAsRead,
    resetAppNotifications,
  } = useAppNotificationsData({ limit: 10 });

  return (
    <div>
      <h3>Notifications</h3>
      <p>Unread: {unreadAppNotificationsCount}</p>

      <ul>
        {appNotifications.map((notification) => (
          <li key={notification.id}>
            <span>{notification.message}</span>
            {!notification.isRead && (
              <button onClick={() => markNotificationAsRead(notification.id)}>
                Mark as Read
              </button>
            )}
          </li>
        ))}
      </ul>
    </div>
  );
}
```

```

    {loading && <p>Loading...</p>}
    {hasMore && !loading && <button onClick={loadMore}>Load More</button>}

    <button onClick={resetAppNotifications}>Refresh Notifications</button>
  </div>
);
}

```

## Parameters & Returns

### Parameters

Parameter	Type	Required	Description
<code>limit</code>	<code>number</code>	No	The number of notifications to fetch per page. Default is <code>10</code> .
<code>notificationTemplates</code>	<code>Partial&lt;NotificationTemplates&gt;</code>	No	Templates to format or add custom messages to notifications.

### Returns

Return Value	Type	Description
<code>appNotifications</code>	<code>UnifiedAppNotification[]</code>	A list of notifications fetched.
<code>unreadAppNotificationsCount</code>	<code>number</code>	The count of unread notifications.
<code>loading</code>	<code>boolean</code>	Indicates whether notifications are being loaded.

Return Value	Type	Description
hasMore	boolean	Indicates whether there are more notifications to fetch.
loadMore	() => void	Fetches the next page of notifications.
markNotificationAsRead	(notificationId: string) => Promise<void>	Marks a notification as read and updates its state locally.
resetAppNotifications	() => Promise<void>	Resets the notifications list and fetches the first page of data.

## Key Features

### 1. Unified Notifications Management

Combines fetching, counting, and managing notifications into one hook.

### 2. Unread Notifications Counter

Tracks unread notifications and provides functionality to decrement the count.

### 3. Message Templating

Supports custom notification messages using the `notificationTemplates` parameter.

### 4. Pagination

Fetches notifications in pages with options to load more data incrementally.

### 5. Marking as Read

Allows marking individual notifications as read while updating their state locally.

© Replyke 2025