Let me continue with documenting the Notification APIs for the FoodShare application. I'll analyze the codebase first to understand how the notification system works.# FoodShare API Documentation - Notification APIs

After analyzing the notification service implementation in the codebase, I'll now document the Notification APIs available in the FoodShare application.

4. Notification APIs

4.1 Send In-App Notification

Endpoint URL: `/api/notifications`

HTTP Method: `POST`

Request Headers:
```
Content-Type: application/json
Authorization: Bearer {token}
```

Request Body Schema:
```json
{
  "userId": "u12345",
  "title": "Donation Matched!",
  "message": "Your donation has been matched with Food Rescue Initiative.",
  "type": "success",
  "linkTo": "/donations/don123",
  "metadata": {
    "donationId": "don123",
    "recipientId": "u67890"
  }
}
```

Response Body Schema:

```json
{
  "id": "not789",
  "userId": "u12345",
  "title": "Donation Matched!",
  "message": "Your donation has been matched with Food Rescue Initiative.",
  "type": "success",
  "isRead": false,
  "createdAt": "2025-04-14T15:30:00Z",
  "linkTo": "/donations/don123",
  "metadata": {
    "donationId": "don123",
    "recipientId": "u67890"
  }
}
```

Authentication Requirements: JWT Token Required

Status Codes and Messages:

- `201 Created`: Notification created successfully
- `400 Bad Request`: Invalid input data
- `401 Unauthorized`: Invalid or missing token
- `403 Forbidden`: Insufficient permissions
- `500 Internal Server Error`: Server error

Description: Creates a new in-app notification for a specific user. The notification can include a link and additional metadata for context.

4.2 Get User Notifications

Endpoint URL: `/api/notifications/user/{userId}`

HTTP Method: `GET`

Request Headers:
```
Authorization: Bearer {token}
```

Request Parameters:
- `limit` (optional): Maximum number of notifications to return (default: 20)
- `offset` (optional): Number of notifications to skip (default: 0)
- `unreadOnly` (optional): Only return unread notifications if true

Response Body Schema:
```json
[
 {
   "id": "not789",
   "userId": "u12345",
   "title": "Donation Matched!",
   "message": "Your donation has been matched with Food Rescue Initiative.",
   "type": "success",
   "isRead": false,
   "createdAt": "2025-04-14T15:30:00Z",
   "linkTo": "/donations/don123",
    "metadata": {
      "donationId": "don123"
    }
  },
  {
    "id": "not788",
    "userId": "u12345",
    "title": "Donation Expiring Soon!",
    "message": "Your donation 'Fresh Vegetables Bundle' will expire in less than an hour.",
    "type": "warning",
    "isRead": true,
    "createdAt": "2025-04-14T14:00:00Z",
    "linkTo": "/donations/don122",
    "metadata": {
      "donationId": "don122"
    }
  }
]
```

Authentication Requirements: JWT Token Required

Status Codes and Messages:

- `200 OK`: Successful request
- `401 Unauthorized`: Invalid or missing token
- `403 Forbidden`: User doesn't have permission to view these notifications
- `404 Not Found`: User not found
- `500 Internal Server Error`: Server error

Description: Retrieves a list of notifications for a specific user with optional filtering for unread notifications and pagination support.

4.3 Mark Notification as Read

Endpoint URL: `/api/notifications/{notificationId}/read`

HTTP Method: `PATCH`

Request Headers:
```
Authorization: Bearer {token}
```

Response Body Schema:
```json
{
  "id": "not789",
  "isRead": true,
  "updatedAt": "2025-04-14T15:35:00Z"
}
```

Authentication Requirements: JWT Token Required

Status Codes and Messages:
- `200 OK`: Notification marked as read successfully
- `401 Unauthorized`: Invalid or missing token
- `403 Forbidden`: User doesn't have permission to update this notification
- `404 Not Found`: Notification not found
- `500 Internal Server Error`: Server error

Description: Marks a specific notification as read. This endpoint should be called when a user clicks on or views a notification.

4.4 Mark All Notifications as Read

Endpoint URL: `/api/notifications/user/{userId}/read-all`

HTTP Method: `POST`

Request Headers:
```
Authorization: Bearer {token}
```

Response Body Schema:
```json
{
 "success": true,
 "count": 5,
 "message": "All notifications marked as read"
}
```

Authentication Requirements: JWT Token Required

Status Codes and Messages:
- `200 OK`: Notifications marked as read successfully
- `401 Unauthorized`: Invalid or missing token
- `403 Forbidden`: User doesn't have permission to update these notifications
- `404 Not Found`: User not found
- `500 Internal Server Error`: Server error

Description: Marks all unread notifications for a specific user as read. This endpoint is typically called when a user clicks "Mark all as read" in the notification panel.

4.5 Get Notification Statistics

Endpoint URL: `/api/notifications/user/{userId}/stats`

HTTP Method: `GET`

Request Headers:
```
Authorization: Bearer {token}
```

Response Body Schema:
```json
{
  "total": 15,
  "unread": 3,
  "categories": {
    "success": 5,
    "warning": 3,
    "info": 7,
    "error": 0
  },
  "recentActivity": "2025-04-14T15:30:00Z"
}
```

Authentication Requirements: JWT Token Required

Status Codes and Messages:

- `200 OK`: Successful request

- `401 Unauthorized`: Invalid or missing token

- `403 Forbidden`: User doesn't have permission to view these statistics

- `404 Not Found`: User not found

- `500 Internal Server Error`: Server error

Description: Retrieves notification statistics for a specific user, including the total count, unread count, and breakdown by notification type.

4.6 Subscribe to Real-Time Notifications

Endpoint URL: `/api/notifications/subscribe`

HTTP Method: `GET` (WebSocket connection)

Request Headers:

```
Authorization: Bearer {token}
```

Response: WebSocket connection established

Authentication Requirements: JWT Token Required

Status Codes and Messages:

- `101 Switching Protocols`: WebSocket connection established successfully

- `401 Unauthorized`: Invalid or missing token

- `403 Forbidden`: User doesn't have permission to subscribe

- `500 Internal Server Error`: Server error

Description: Establishes a WebSocket connection to receive real-time notifications. New notifications will be pushed to the client as they are created, without requiring polling.

4.7 Send Email Notification

Endpoint URL: `/api/notifications/email`

HTTP Method: `POST`

Request Headers:
```
Content-Type: application/json
Authorization: Bearer {token}
```

Request Body Schema:
```json
{
  "toEmail": "recipient@example.com",
  "subject": "Your FoodShare donation has been matched",
  "htmlBody": "<div><h1>Donation Matched!</h1><p>Your donation has been matched with Food Rescue Initiative.</p></div>",
  "templateParams": {
    "donorName": "John Smith",
    "recipientName": "Food Rescue Initiative"
  }
}
```

Response Body Schema:
```json
```

```
{
  "success": true,
  "messageId": "email-1713171000000",
  "status": 200
}
```

Authentication Requirements: JWT Token Required (Admin or System role)

Status Codes and Messages:

- `200 OK`: Email sent successfully
- `400 Bad Request`: Invalid input data
- `401 Unauthorized`: Invalid or missing token
- `403 Forbidden`: User doesn't have permission to send emails
- `500 Internal Server Error`: Server error

Description: Sends an email notification to a specified recipient. This endpoint is typically used by the system for sending transactional emails like donation matches, expiration alerts, etc.

4.8 Send System Notification

Endpoint URL: `/api/notifications/system`

HTTP Method: `POST`

Request Headers:

```
Content-Type: application/json
Authorization: Bearer {token}
```

Request Body Schema:

```json
{
  "title": "Scheduled Maintenance",
  "message": "FoodShare will be undergoing scheduled maintenance on April 20th from 2-4 AM EST.",
  "type": "info",
  "roles": ["admin", "donor", "ngo", "beneficiary"]
}
```

Response Body Schema:

```json
{
  "success": true,
  "notificationId": "sys123",
  "recipientCount": 248,
  "createdAt": "2025-04-14T15:30:00Z"
}
```

Authentication Requirements: JWT Token Required (Admin role only)

Status Codes and Messages:

- `201 Created`: System notification sent successfully
- `400 Bad Request`: Invalid input data
- `401 Unauthorized`: Invalid or missing token
- `403 Forbidden`: User doesn't have admin role
- `500 Internal Server Error`: Server error

Description: Creates and sends a system-wide notification to all users or to users with specific roles. This is used for announcements, alerts, and other system-wide communications.

4.9 Get System Notifications

Endpoint URL: `/api/notifications/system`

HTTP Method: `GET`

Request Headers:
```
Authorization: Bearer {token}
```

Request Parameters:
- `limit` (optional): Maximum number of notifications to return (default: 20)
- `offset` (optional): Number of notifications to skip (default: 0)
- `type` (optional): Filter by notification type (info, warning, error)

Response Body Schema:
```json
[
 {
  "id": "sys123",
  "title": "Scheduled Maintenance",
  "message": "FoodShare will be undergoing scheduled maintenance on April 20th from 2-4 AM EST.",
  "type": "info",
  "createdAt": "2025-04-14T15:30:00Z",
  "targetRoles": ["admin", "donor", "ngo", "beneficiary"],
  "createdBy": "admin",
  "priority": "medium"
 },
 {
   "id": "sys122",
   "title": "Multiple Food Donations Approaching Expiration",
   "message": "5 donations are expiring in the next 3 hours.",
   "type": "warning",
   "createdAt": "2025-04-14T14:00:00Z",
   "targetRoles": ["admin"],
   "priority": "high"
 }
]
```

Authentication Requirements: JWT Token Required (Admin role only)

Status Codes and Messages:

- `200 OK`: Successful request
- `401 Unauthorized`: Invalid or missing token
- `403 Forbidden`: User doesn't have admin role
- `500 Internal Server Error`: Server error

Description: Retrieves a list of system notifications for admin users to monitor system alerts and announcements.

4.10 Notify Donation Match

Endpoint URL: `/api/notifications/donation-match`

HTTP Method: `POST`

Request Headers:

```
```

```
Content-Type: application/json
Authorization: Bearer {token}
```

Request Body Schema:

```json
{
  "donorId": "u12345",
  "recipientId": "u67890",
  "donationDetails": {
    "id": "don123",
    "title": "Fresh Vegetables Bundle",
    "quantity": "10 kg",
    "expiresAt": "2025-04-18T15:30:00Z",
    "location": "123 Main St, Anytown"
  }
}
```

Response Body Schema:

```json
{
  "success": true,
  "notifications": [
    {
      "type": "in-app",
      "userId": "u12345",
      "notificationId": "not789",
      "success": true
    },
    {
      "type": "in-app",
      "userId": "u67890",
      "notificationId": "not790",
      "success": true
    },
    {
      "type": "email",
      "recipient": "donor@example.com",
      "success": true,
      "messageId": "email-1713171000000"
    }
  ]
}
```

Authentication Requirements: JWT Token Required (Admin or System role)

Status Codes and Messages:

- `200 OK`: Notifications sent successfully
- `400 Bad Request`: Invalid input data
- `401 Unauthorized`: Invalid or missing token
- `403 Forbidden`: User doesn't have permission
- `404 Not Found`: Donor or recipient not found
- `500 Internal Server Error`: Server error

Description: Sends notifications to both donor and recipient when a donation is matched. This includes both in-app notifications and email notifications if email addresses are available.

4.11 Notify Donation Expiration

Endpoint URL: `/api/notifications/donation-expiration`

HTTP Method: `POST`

Request Headers:
```
Content-Type: application/json
Authorization: Bearer {token}
```

Request Body Schema:
```json
{
  "donationId": "don123",
  "expirationDate": "2025-04-18T15:30:00Z"
}
```

Response Body Schema:
```json
{
  "success": true,
  "notified": true,
  "hoursLeft": 2,
  "notifications": [
    {
      "type": "in-app",
      "userId": "u12345",
      "notificationId": "not791",
      "success": true
    },
    {
      "type": "in-app",
      "userId": "admin",
      "notificationId": "not792",
      "success": true
    },
    {
      "type": "email",
      "recipient": "donor@example.com",
      "success": true,
      "messageId": "email-1713171000001"
    }
  ]
}
```

Authentication Requirements: JWT Token Required (Admin or System role)

Status Codes and Messages:

- `200 OK`: Process completed successfully (may not have sent notifications if not within threshold)
- `400 Bad Request`: Invalid input data
- `401 Unauthorized`: Invalid or missing token
- `403 Forbidden`: User doesn't have permission
- `404 Not Found`: Donation not found
- `500 Internal Server Error`: Server error

Description: Checks if a donation is nearing expiration and sends notifications to relevant parties if it is within the notification threshold (e.g., 3 hours before expiration). Notifications are sent to the donor and to admins.

This completes the documentation for the Notification APIs in the FoodShare application, which allow users to receive real-time updates about their donations, matches, and other important events.