

Let me continue with documenting the Donation APIs for the FoodShare application. I'll first explore the codebase to understand the donation-related functionality. Based on my analysis of the codebase, here's the documentation for the Donation APIs in the FoodShare application:

FoodShare API Documentation - Donation APIs

2. Donation APIs

2.1 Create Donation

****Endpoint URL**:** `/api/donations/create`

****HTTP Method**:** `POST`

****Request Headers**:**

``

Content-Type: application/json

Authorization: Bearer {token}

``

****Request Body Schema**:**

```json

{

"title": "Fresh Vegetables Bundle",

"foodType": "Vegetables",

"quantity": 5,

"unit": "kg",

"category": "Produce",

"expiresAt": "2025-04-18T15:30:00Z",

"location": "123 Main St, Anytown",

"description": "Assorted fresh vegetables including carrots, broccoli, and bell peppers"

}

...

**\*\*Response Body Schema\*\*:**

```json

```
{
  "id": "don123",
  "title": "Fresh Vegetables Bundle",
  "foodType": "Vegetables",
  "quantity": 5,
  "unit": "kg",
  "category": "Produce",
  "expiresAt": "2025-04-18T15:30:00Z",
  "location": "123 Main St, Anytown",
  "description": "Assorted fresh vegetables including carrots, broccoli, and bell peppers",
  "donorId": "u12345",
  "donorName": "Green Grocers",
  "status": "Pending Approval",
  "createdAt": "2025-04-14T10:20:30Z"
}
```

...

****Authentication Requirements**:** JWT Token Required (Donor role)

****Status Codes and Messages**:**

- `201 Created` : Successfully created donation
- `400 Bad Request` : Invalid input data
- `401 Unauthorized` : Invalid or missing token
- `403 Forbidden` : User doesn't have appropriate role
- `500 Internal Server Error` : Server error

****Description**:** Creates a new food donation listing with details provided by a donor. The donation initially has a status of "Pending Approval" until reviewed by an admin.

2.2 Get All Donations

****Endpoint URL**:** `/api/donations/list`

****HTTP Method**:** `GET`

****Request Headers**:**

```

Authorization: Bearer {token}

```

****Request Parameters**:**

- `category` (optional): Filter by food category
- `status` (optional): Filter by donation status
- `fromDate` (optional): Filter donations created after date
- `toDate` (optional): Filter donations created before date
- `page` (optional): Page number for pagination
- `limit` (optional): Number of results per page

****Response Body Schema**:**

```json

```
{
 "donations": [
 {
 "id": "don123",
 "title": "Fresh Vegetables Bundle",
 "foodType": "Vegetables",
 "quantity": 5,
```

```
"unit": "kg",
"category": "Produce",
"expiresAt": "2025-04-18T15:30:00Z",
"location": "123 Main St, Anytown",
"description": "Assorted fresh vegetables including carrots, broccoli, and bell peppers",
"donorId": "u12345",
"donorName": "Green Grocers",
"status": "Pending Approval",
"createdAt": "2025-04-14T10:20:30Z"
}
],
"total": 1,
"page": 1,
"totalPages": 1
}
...

```

**\*\*Authentication Requirements\*\*:** JWT Token Required

**\*\*Status Codes and Messages\*\*:**

- `200 OK` : Successful request
- `401 Unauthorized` : Invalid or missing token
- `500 Internal Server Error` : Server error

**\*\*Description\*\*:** Retrieves a list of all donation listings with optional filtering and pagination.

### ### 2.3 Get Donation Details

**\*\*Endpoint URL\*\*:** `/api/donations/{id}`

**\*\*HTTP Method\*\*:** `GET`

**\*\*Request Headers\*\*:**

```

Authorization: Bearer {token}

```

**\*\*Response Body Schema\*\*:**

```json

{

"id": "don123",

"title": "Fresh Vegetables Bundle",

"foodType": "Vegetables",

"quantity": 5,

"unit": "kg",

"category": "Produce",

"expiresAt": "2025-04-18T15:30:00Z",

"location": "123 Main St, Anytown",

"description": "Assorted fresh vegetables including carrots, broccoli, and bell peppers",

"donor": {

"id": "u12345",

"name": "Green Grocers",

"phone": "(555) 123-4567",

"address": "123 Main St, Anytown"

},

"recipient": {

"id": "u67890",

"name": "Community Food Bank",

"phone": "(555) 987-6543",

"address": "456 Oak Ave, Townsville"

},

"ngo": {

```

    "id": "ngo789",
    "name": "Food Rescue Initiative",
    "phone": "(555) 456-7890"
  },
  "status": "In Transit",
  "createdAt": "2025-04-14T10:20:30Z",
  "timestamps": {
    "created": "2025-04-14T10:20:30Z",
    "accepted": "2025-04-14T12:30:00Z",
    "readyForPickup": "2025-04-14T15:00:00Z",
    "inTransit": "2025-04-14T16:30:00Z"
  },
  "logistics": {
    "pickupAddress": "123 Main St, Cityville",
    "dropoffAddress": "456 Oak Ave, Townsville",
    "deliveryType": "NGO Logistics",
    "eta": "2025-04-14T18:00:00Z"
  }
}
...

```

****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 access to this donation
- `404 Not Found` : Donation not found
- `500 Internal Server Error` : Server error

****Description**:** Retrieves detailed information about a specific donation, including donor, recipient, and logistics information.

2.4 Update Donation Status

****Endpoint URL**:** `/api/donations/{id}/status`

****HTTP Method**:** `PUT`

****Request Headers**:**

...

Content-Type: application/json

Authorization: Bearer {token}

...

****Request Body Schema**:**

```json

{

"status": "Ready for Pickup"

}

```

****Response Body Schema**:**

```json

{

"id": "don123",

"status": "Ready for Pickup",

"updatedAt": "2025-04-14T14:30:00Z",

"timestamps": {

"readyForPickup": "2025-04-14T14:30:00Z"

},

```
"message": "Donation status updated successfully"
}
...
```

**\*\*Authentication Requirements\*\*:** JWT Token Required (Donor, Beneficiary, NGO or Admin role)

**\*\*Status Codes and Messages\*\*:**

- `200 OK` : Status updated successfully
- `400 Bad Request` : Invalid status value
- `401 Unauthorized` : Invalid or missing token
- `403 Forbidden` : User doesn't have permission to update this donation
- `404 Not Found` : Donation not found
- `500 Internal Server Error` : Server error

**\*\*Description\*\*:** Updates the status of a donation. Different roles can set different statuses based on the workflow: Donors can mark as "Ready for Pickup", NGOs can mark as "In Transit", Beneficiaries can mark as "Delivered", and Admins can set any status.

### ### 2.5 Get Donor Donations

**\*\*Endpoint URL\*\*:** `/api/donations/donor/{userId}`

**\*\*HTTP Method\*\*:** `GET`

**\*\*Request Headers\*\*:**

...

Authorization: Bearer {token}

...

**\*\*Request Parameters\*\*:**

- `status` (optional): Filter by donation status
- `page` (optional): Page number for pagination



- `limit` (optional): Number of results per page

**\*\*Response Body Schema\*\*:**

```
```\n{\n  "donations": [\n    {\n      "id": "don123",\n      "title": "Fresh Vegetables Bundle",\n      "foodType": "Vegetables",\n      "quantity": 5,\n      "unit": "kg",\n      "category": "Produce",\n      "expiresAt": "2025-04-18T15:30:00Z",\n      "status": "Pending Approval",\n      "createdAt": "2025-04-14T10:20:30Z"\n    }\n  ],\n  "total": 1,\n  "page": 1,\n  "totalPages": 1\n}\n```\n
```

****Authentication Requirements**:** JWT Token Required (Donor or Admin role)

****Status Codes and Messages**:**

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

****Description**:** Retrieves all donations created by a specific donor with optional filtering and pagination.

2.6 Claim Donation

****Endpoint URL**:** `/api/donations/claim`

****HTTP Method**:** `POST`

****Request Headers**:**

```

Content-Type: application/json

Authorization: Bearer {token}

```

****Request Body Schema**:**

```json

```
{
 "donationId": "don123",
 "beneficiaryId": "u67890",
 "claimNotes": "Will pick up on Thursday afternoon"
}
```

```

****Response Body Schema**:**

```json

```
{
 "id": "don123",
 "status": "Accepted",
 "recipientId": "u67890",

```

```
"recipientName": "Community Food Bank",
"claimedAt": "2025-04-14T11:45:00Z",
"message": "Donation claimed successfully"
}
` ` `
```

**\*\*Authentication Requirements\*\*:** JWT Token Required (Beneficiary or NGO role)

**\*\*Status Codes and Messages\*\*:**

- `200 OK` : Donation claimed successfully
- `400 Bad Request` : Invalid request or donation not available
- `401 Unauthorized` : Invalid or missing token
- `403 Forbidden` : User doesn't have appropriate role
- `404 Not Found` : Donation not found
- `409 Conflict` : Donation already claimed
- `500 Internal Server Error` : Server error

**\*\*Description\*\*:** Claims an available donation on behalf of a beneficiary or NGO, updating the status to "Accepted".

### ### 2.7 Get Claimed Donations

**\*\*Endpoint URL\*\*:** `/api/donations/claimed/{beneficiaryId}`

**\*\*HTTP Method\*\*:** `GET`

**\*\*Request Headers\*\*:**

` ` `

Authorization: Bearer {token}

` ` `

**\*\*Request Parameters\*\*:**

- `status` (optional): Filter by donation status
- `page` (optional): Page number for pagination
- `limit` (optional): Number of results per page

**\*\*Response Body Schema\*\*:**

```
```\n{\n  "donations": [\n    {\n      "id": "don123",\n      "title": "Fresh Vegetables Bundle",\n      "foodType": "Vegetables",\n      "quantity": 5,\n      "unit": "kg",\n      "category": "Produce",\n      "expiresAt": "2025-04-18T15:30:00Z",\n      "donorName": "Green Grocers",\n      "status": "Accepted",\n      "claimedAt": "2025-04-14T11:45:00Z"\n    }\n  ],\n  "total": 1,\n  "page": 1,\n  "totalPages": 1\n}\n```\n
```

****Authentication Requirements**:** JWT Token Required (Beneficiary, NGO or Admin role)

****Status Codes and Messages**:**

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

****Description****: Retrieves all donations claimed by a specific beneficiary or NGO with optional filtering and pagination.

2.8 Browse Available Donations

****Endpoint URL****: `/api/donations/available`

****HTTP Method****: `GET`

****Request Headers****:

...

Authorization: Bearer {token}

...

****Request Parameters****:

- `category` (optional): Filter by food category
- `distance` (optional): Maximum distance in km
- `latitude` (optional): Latitude for location-based search
- `longitude` (optional): Longitude for location-based search
- `expiryRange` (optional): Filter by expiration timeframe
- `page` (optional): Page number for pagination
- `limit` (optional): Number of results per page

****Response Body Schema****:

```json

{

```
"donations": [
 {
 "id": "don123",
 "title": "Fresh Vegetables Bundle",
 "type": "Vegetables",
 "quantity": 5,
 "unit": "kg",
 "expiresAt": "2025-04-18T15:30:00Z",
 "status": "Available",
 "donorName": "Green Grocers",
 "distance": "1.2 km",
 "location": "123 Main St, Anytown"
 },
],
 "total": 1,
 "page": 1,
 "totalPages": 1
}
```

...

**\*\*Authentication Requirements\*\*:** JWT Token Required (Beneficiary or NGO role)

**\*\*Status Codes and Messages\*\*:**

- `200 OK` : Successful request
- `401 Unauthorized` : Invalid or missing token
- `500 Internal Server Error` : Server error

**\*\*Description\*\*:** Retrieves a list of available donations that can be claimed, with optional filtering by category, distance, and expiration timeframe.

### ### 2.9 Request Donation

**\*\*Endpoint URL\*\*:** `/api/donations/request`

**\*\*HTTP Method\*\*:** `POST`

**\*\*Request Headers\*\*:**

```

Content-Type: application/json

Authorization: Bearer {token}

```

**\*\*Request Body Schema\*\*:**

```json

```
{  
  "donationId": "don123",  
  "beneficiaryId": "u67890",  
  "requestNotes": "Need for community meal program"  
}
```

```

**\*\*Response Body Schema\*\*:**

```json

```
{  
  "id": "req456",  
  "donationId": "don123",  
  "beneficiaryId": "u67890",  
  "status": "Pending",  
  "requestedAt": "2025-04-14T13:20:00Z",  
  "message": "Donation request submitted successfully"  
}
```

```

**\*\*Authentication Requirements\*\***: JWT Token Required (Beneficiary or NGO role)

**\*\*Status Codes and Messages\*\***:

- `201 Created` : Request submitted successfully
- `400 Bad Request` : Invalid request
- `401 Unauthorized` : Invalid or missing token
- `403 Forbidden` : User doesn't have appropriate role
- `404 Not Found` : Donation not found
- `409 Conflict` : Donation already requested or unavailable
- `500 Internal Server Error` : Server error

**\*\*Description\*\***: Submits a request for a specific donation, which must be approved by the donor before it's claimed.

### ### 2.10 Submit Donation Review

**\*\*Endpoint URL\*\***: `/api/donations/{donationId}/review`

**\*\*HTTP Method\*\***: `POST`

**\*\*Request Headers\*\***:

`,`

Content-Type: application/json

Authorization: Bearer {token}

`,`

**\*\*Request Body Schema\*\***:

`` `json

{

  "rating": 5,



```
"comment": "The food was fresh and the pickup process was smooth. Thank you!"
}
` ` `
```

**\*\*Response Body Schema\*\*:**

```
` ` `json
{
 "id": "rev789",
 "donationId": "don123",
 "reviewerId": "u67890",
 "reviewerName": "Community Food Bank",
 "reviewerRole": "Beneficiary",
 "rating": 5,
 "comment": "The food was fresh and the pickup process was smooth. Thank you!",
 "createdAt": "2025-04-14T18:45:00Z"
}
` ` `
```

**\*\*Authentication Requirements\*\*:** JWT Token Required (Donor, Beneficiary, or NGO role)

**\*\*Status Codes and Messages\*\*:**

- ` 201 Created ` : Review submitted successfully
- ` 400 Bad Request ` : Invalid review data
- ` 401 Unauthorized ` : Invalid or missing token
- ` 403 Forbidden ` : User not authorized to review this donation
- ` 404 Not Found ` : Donation not found
- ` 409 Conflict ` : User has already reviewed this donation
- ` 500 Internal Server Error ` : Server error

**\*\*Description\*\*:** Submits a review for a completed donation. The user must be a participant (donor, recipient, or NGO) in the donation process.

### ### 2.11 Delete Donation

**\*\*Endpoint URL\*\*:** `/api/donations/{id}`

**\*\*HTTP Method\*\*:** `DELETE`

**\*\*Request Headers\*\*:**

```

Authorization: Bearer {token}

```

**\*\*Response Body Schema\*\*:**

```json

```
{  
  "message": "Donation deleted successfully",  
  "id": "don123"  
}
```

```

**\*\*Authentication Requirements\*\*:** JWT Token Required (Donor or Admin role)

**\*\*Status Codes and Messages\*\*:**

- `200 OK` : Donation deleted successfully
- `401 Unauthorized` : Invalid or missing token
- `403 Forbidden` : User not authorized to delete this donation
- `404 Not Found` : Donation not found
- `500 Internal Server Error` : Server error

**\*\*Description\*\*:** Deletes a donation listing. Donors can only delete their own donations that are in "Pending" or "Available" status, while admins can delete any donation.

### ### 2.12 Search Donations

**\*\*Endpoint URL\*\*:** `/api/donations/search`

**\*\*HTTP Method\*\*:** `GET`

**\*\*Request Headers\*\*:**

` ` `

Authorization: Bearer {token}

` ` `

**\*\*Request Parameters\*\*:**

- `query` (optional): Text search query
- `category` (optional): Filter by food category
- `status` (optional): Filter by donation status
- `fromDate` (optional): Filter by date range start
- `toDate` (optional): Filter by date range end
- `page` (optional): Page number for pagination
- `limit` (optional): Number of results per page

**\*\*Response Body Schema\*\*:**

` `` `json

{

  "donations": [

    {

      "id": "don123",

      "title": "Fresh Vegetables Bundle",

      "type": "Vegetables",

      "quantity": 5,

      "unit": "kg",

```
"expiresAt": "2025-04-18T15:30:00Z",
"status": "Available",
"donorName": "Green Grocers",
"createdAt": "2025-04-14T10:20:30Z"
}
],
"total": 1,
"page": 1,
"totalPages": 1
}
` ` `
```

**\*\*Authentication Requirements\*\*:** JWT Token Required

**\*\*Status Codes and Messages\*\*:**

- `200 OK` : Successful request
- `400 Bad Request` : Invalid search parameters
- `401 Unauthorized` : Invalid or missing token
- `500 Internal Server Error` : Server error

**\*\*Description\*\*:** Searches for donations based on various criteria, including text search and filters.

### ### 2.13 Get Donation Statistics

**\*\*Endpoint URL\*\*:** `/api/donations/stats/{userId}`

**\*\*HTTP Method\*\*:** `GET`

**\*\*Request Headers\*\*:**

` ` `

Authorization: Bearer {token}

```

****Response Body Schema**:**

```json

```
{
 "totalDonations": 15,
 "activeDonations": 3,
 "completedDonations": 12,
 "impactMetrics": {
 "totalQuantity": 87,
 "peopleFed": 348,
 "foodSaved": "87 kg",
 "co2Saved": "174 kg"
 },
 "topCategories": [
 {"name": "Vegetables", "count": 7},
 {"name": "Bakery", "count": 5},
 {"name": "Canned Goods", "count": 3}
],
 "monthlyStats": [
 {"month": "January", "count": 2},
 {"month": "February", "count": 3},
 {"month": "March", "count": 5},
 {"month": "April", "count": 5}
]
}
```

```

****Authentication Requirements**:** JWT Token Required

****Status Codes and Messages**:**

- `200 OK` : Successful request
- `401 Unauthorized` : Invalid or missing token
- `403 Forbidden` : User not authorized to view these statistics
- `404 Not Found` : User not found
- `500 Internal Server Error` : Server error

****Description**:** Retrieves statistical information about a user's donation activity, including counts, impact metrics, and trends over time.