

I'll continue with documenting the Review and Stats APIs for the FoodShare application. Let me first analyze the codebase to understand these features. # FoodShare API Documentation - Review APIs

Based on my analysis of the codebase, I'll now document the Review APIs available in the FoodShare application.

6. Review APIs

The FoodShare application includes a comprehensive review system that allows users to provide feedback on donations, rate their experiences, and help build trust within the platform's community. Here are the API endpoints for managing reviews:

6.1 Submit 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 vegetables were very fresh and the pickup process was smooth. Thank you for your generosity!",`

`"reviewerRole": "Beneficiary"`

`}`

```

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

```json

```
{
  "id": "rev101",
  "donationId": "don123",
  "reviewerId": "u12345",
  "reviewerName": "Jane Smith",
  "reviewerRole": "Beneficiary",
  "rating": 5,
  "comment": "The vegetables were very fresh and the pickup process was smooth. Thank you for your generosity!",
  "date": "2025-04-14T15:30:00Z",
  "status": "Active"
}
```

```

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

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

- `201 Created` : Review submitted successfully
- `400 Bad Request` : Invalid input data or rating not provided
- `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\*\*:** Allows a user to submit a review for a donation they were involved with. Donors can review beneficiaries and beneficiaries can review donors after a donation has been completed.

### ### 6.2 Get Reviews for Donation

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

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

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

` ` `

Authorization: Bearer {token}

` ` `

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

- `page` (optional): Page number for pagination (default: 1)

- `limit` (optional): Number of reviews per page (default: 10)

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

` `` `json

{

  "reviews": [

    {

      "id": "rev101",

      "donationId": "don123",

      "reviewerId": "u12345",

      "reviewerName": "Jane Smith",

      "reviewerRole": "Beneficiary",

      "rating": 5,

      "comment": "The vegetables were very fresh and the pickup process was smooth. Thank you for your generosity!",

      "date": "2025-04-14T15:30:00Z",

      "status": "Active"

```
},
{
 "id": "rev102",
 "donationId": "don123",
 "reviewerId": "u67890",
 "reviewerName": "Green Grocers",
 "reviewerRole": "Donor",
 "rating": 4,
 "comment": "The food bank was very professional and picked up the items promptly.",
 "date": "2025-04-14T16:20:00Z",
 "status": "Active"
}
],
"pagination": {
 "total": 2,
 "page": 1,
 "limit": 10,
 "pages": 1
}
}
` ``
```

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

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

- `200 OK` : Successful request
- `401 Unauthorized` : Invalid or missing token
- `404 Not Found` : Donation not found
- `500 Internal Server Error` : Server error

**\*\*Description\*\*:** Retrieves all reviews associated with a specific donation. This includes reviews from both the donor and the recipient.

### ### 6.3 Get User Reviews

**\*\*Endpoint URL\*\*:** `/api/users/{userId}/reviews`

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

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

```

Authorization: Bearer {token}

```

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

- `type` (optional): Type of reviews to retrieve ("given", "received", or "all")
- `page` (optional): Page number for pagination (default: 1)
- `limit` (optional): Number of reviews per page (default: 10)

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

```json

{

"reviews": [

{

"id": "rev101",

"donationId": "don123",

"donationTitle": "Fresh Vegetables Bundle",

"reviewerId": "u12345",

"reviewerName": "Jane Smith",

"reviewerRole": "Beneficiary",

"recipientId": "u67890",

```
"recipientName": "Green Grocers",
"recipientRole": "Donor",
"rating": 5,
"comment": "The vegetables were very fresh and the pickup process was smooth. Thank you
for your generosity!",
"date": "2025-04-14T15:30:00Z",
"status": "Active"
}
],
"pagination": {
"total": 1,
"page": 1,
"limit": 10,
"pages": 1
},
"stats": {
"averageRating": 4.8,
"totalReviews": 15,
"ratingDistribution": {
"5": 10,
"4": 4,
"3": 1,
"2": 0,
"1": 0
}
}
}
` ``
```

****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 reviews
- `404 Not Found` : User not found
- `500 Internal Server Error` : Server error

****Description**:** Retrieves all reviews associated with a specific user, either reviews they've given, reviews they've received, or both. Also includes statistics about the user's review history.

6.4 Update Review

****Endpoint URL**:** `/api/reviews/{reviewId}`

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

****Request Headers**:**

...

Content-Type: application/json

Authorization: Bearer {token}

...

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

```json

```
{
 "rating": 4,
 "comment": "Updated comment: The vegetables were fresh but delivery was delayed."
}
```

...

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

```
` ``json
{
 "id": "rev101",
 "donationId": "don123",
 "reviewerId": "u12345",
 "reviewerName": "Jane Smith",
 "reviewerRole": "Beneficiary",
 "rating": 4,
 "comment": "Updated comment: The vegetables were fresh but delivery was delayed.",
 "date": "2025-04-14T15:30:00Z",
 "updatedAt": "2025-04-14T17:45:00Z",
 "status": "Active"
}
` ``
```

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

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

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

**\*\*Description\*\*:** Allows a user to update a review they previously submitted. Users can only update their own reviews.

### ### 6.5 Delete Review

**\*\*Endpoint URL\*\*:** `/api/reviews/{reviewId}`



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

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

```

Authorization: Bearer {token}

```

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

```json

{

"id": "rev101",

"message": "Review successfully deleted",

"deletedAt": "2025-04-14T17:50:00Z"

}

```

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

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

- `200 OK` : Review deleted successfully
- `401 Unauthorized` : Invalid or missing token
- `403 Forbidden` : User doesn't have permission to delete this review
- `404 Not Found` : Review not found
- `500 Internal Server Error` : Server error

**\*\*Description\*\*:** Allows a user to delete a review they previously submitted. Users can only delete their own reviews, while administrators can delete any review.

### 6.6 Report Review

**\*\*Endpoint URL\*\*:** `/api/reviews/{reviewId}/report`

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

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

```

Content-Type: application/json

Authorization: Bearer {token}

```

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

```json

{

"reason": "Inappropriate content",

"details": "This review contains offensive language."

}

```

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

```json

{

"id": "rep123",

"reviewId": "rev101",

"reporterId": "u12345",

"reason": "Inappropriate content",

"details": "This review contains offensive language.",

"status": "Pending",

"submittedAt": "2025-04-14T17:55:00Z"

}

```

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

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

- `201 Created` : Report submitted successfully
- `400 Bad Request` : Invalid input data
- `401 Unauthorized` : Invalid or missing token
- `404 Not Found` : Review not found
- `500 Internal Server Error` : Server error

**\*\*Description\*\*:** Allows a user to report a review that they believe violates the platform's guidelines. An administrator will review the report and take appropriate action.

### 6.7 Moderate Review (Admin)

**\*\*Endpoint URL\*\*:** `/api/admin/reviews/{reviewId}/moderate`

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

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

```

Content-Type: application/json

Authorization: Bearer {token}

```

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

```json

```
{  
  "action": "approve",  
  "notes": "Review meets community guidelines.",  
  "notifyUser": true  
}
```

```

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

```json

```
{
  "id": "rev101",
  "status": "Approved",
  "moderatorId": "admin123",
  "moderationNotes": "Review meets community guidelines.",
  "moderatedAt": "2025-04-14T18:00:00Z"
}
```

```

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

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

- `200 OK` : Review moderated successfully
- `400 Bad Request` : Invalid input data
- `401 Unauthorized` : Invalid or missing token
- `403 Forbidden` : User doesn't have admin role
- `404 Not Found` : Review not found
- `500 Internal Server Error` : Server error

**\*\*Description\*\*:** Allows administrators to moderate reported reviews. The possible actions are "approve", "flag", or "remove".

### ### 6.8 Get Review Reports (Admin)

**\*\*Endpoint URL\*\*:** `/api/admin/reviews/reports`

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

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

```

Authorization: Bearer {token}

```

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

- `status` (optional): Filter by report status ("pending", "resolved", "all")
- `page` (optional): Page number for pagination (default: 1)
- `limit` (optional): Number of reports per page (default: 20)

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

```json

```
{
  "reports": [
    {
      "id": "rep123",
      "reviewId": "rev101",
      "reviewContent": {
        "rating": 2,
        "comment": "This service was terrible...",
        "reviewerName": "John Doe"
      },
      "reporterId": "u12345",
      "reporterName": "Jane Smith",
      "reason": "Inappropriate content",
      "details": "This review contains offensive language.",
      "status": "Pending",
      "submittedAt": "2025-04-14T17:55:00Z"
    }
  ],
}
```

```
"pagination": {  
  "total": 1,  
  "page": 1,  
  "limit": 20,  
  "pages": 1  
}  
}  
` ``
```

****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 reported reviews for administrator moderation. Includes details about both the report and the reported review.

FoodShare API Documentation - Statistics APIs

Now, let's document the Statistics APIs for the FoodShare application:

7. Statistics APIs

7.1 Get User Statistics

****Endpoint URL****: `/api/users/{userId}/stats`

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

****Request Headers**:**

```

Authorization: Bearer {token}

```

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

```json

{

"userId": "u12345",

"role": "Donor",

"donations": {

"total": 42,

"active": 3,

"completed": 37,

"cancelled": 2

},

"claimed": {

"total": 0,

"active": 0,

"completed": 0

},

"impact": {

"foodSaved": 320,

"mealsProvided": 960,

"co2Reduced": 640

},

"points": 2350,

"rating": {

"average": 4.8,

"count": 37

```
},
"memberSince": "2024-11-15T10:30:00Z",
"lastActive": "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 comprehensive statistics about a user's activity on the platform, including donations, claims, impact metrics, points, and ratings.

### ### 7.2 Get Donation Statistics

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

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

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

` ` `

Authorization: Bearer {token}

` ` `

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

` `` json



```

{
 "id": "don123",
 "title": "Fresh Vegetables Bundle",
 "createdAt": "2025-04-10T10:30:00Z",
 "completedAt": "2025-04-12T15:45:00Z",
 "timeToMatch": 5.25,
 "timeToComplete": 53.25,
 "viewCount": 42,
 "requestCount": 3,
 "environmentalImpact": {
 "co2Saved": 15,
 "waterSaved": 2500,
 "landUseSaved": 10
 },
 "socialImpact": {
 "mealsProvided": 30,
 "beneficiariesServed": 1
 },
 "reviews": {
 "donorRating": 5,
 "recipientRating": 4
 }
}
` ``

```

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

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

- `200 OK` : Successful request
- `401 Unauthorized` : Invalid or missing token
- `404 Not Found` : Donation not found

- `500 Internal Server Error` : Server error

**\*\*Description\*\***: Retri