**📄 API Documentation: Food Rescue Feature**

This documentation provides details on the **three APIs** for the **Food Rescue feature**, including authentication, request/response formats, and security considerations.

**Version:** 1.0  
**Base URL:** https://your-api.com/

* **Authentication APIs :** All API requests must include an jwt token for authentication.

**1. User Registration**

**Endpoint:** /register  
**Method:** POST  
**Description:** Registers a new user.

**Request Body (JSON)**

json :{

"name": "John Doe",

"email": "john.doe@example.com",

"password": "password123"

}

**Response**

**Success (201 Created)**

Json :{

"message": "User registered successfully"

}

**Failure (400 Bad Request / 409 Conflict)**

Json : {

"success": false,

"message": "User already exists."

}

**2. User Login**

**Endpoint:** /oauth-token  
**Method:** POST  
**Description:** Authenticates a user and returns access & refresh tokens.

**Request Body (JSON)**

Json: {

"email": "john.doe@example.com",

"password": "password123"

}

**Response**

**Success (200 OK)**

Json: {

"success": true,

"token": "your\_access\_token",

"refresh\_token": "your\_refresh\_token"

}

**Failure (401 Unauthorized)**

Json: {

"success": false,

"message": "Invalid credentials"

}

**Order APIs**

**3. Cancel an Order**

**Endpoint:** /cancel-order  
**Method:** POST  
**Description:** Cancels an order and updates its status.

**Request Headers**

* Authorization: Bearer your\_api\_key

**Request Body (JSON)**

Json: {

"order\_id": 12345,

"discounted\_price": 12.99,

"reason": "Customer changed mind"

}

**Response**

**Success (200 OK)**

Json: {

"success": true,

"message": "Order marked as canceled"

}

**Failure (400 Bad Request / 401 Unauthorized / 404 Not Found)**

Json: {

"success": false,

"message": "Order not found"

}

**4. Get Canceled Orders (Within 3 KM)**

**Endpoint:** /get-canceled-orders  
**Method:** POST  
**Description:** Fetches canceled orders within a 3 KM radius for a user.

**Request Headers**

* Authorization: Bearer your\_api\_key

**Request Body (JSON)**

Json: {

"user\_id": 5678,

"page": 1 }

**Response**

**Success (200 OK)**

Json: {

"orders": [

{

"id": 12345,

"original\_price": 15.99,

"discounted\_price": 12.99,

"order\_lat": 40.7128,

"order\_lng": -74.0060,

"partner\_id": 789,

"partner\_lat": 40.7138,

"partner\_lng": -74.0050,

"items": [

{

"item\_name": "Pizza",

"category": "Vegetarian"

}

]

} ],

"pagination": {

"current\_page": 1,

"total\_pages": 5,

"total\_orders": 50

}

}

**Failure (400 Bad Request / 404 Not Found)**

Json: {

"success": false,

"message": "User not found" }

**5. Claim an Order**

**Endpoint:** /claim-order  
**Method:** POST  
**Description:** Claim an order and updates its status and insert new order details for the user behalf of claimed order.

**Request Headers**

* Authorization: Bearer your\_api\_key

**Request Body (JSON)**

Json: {

"order\_id": 16,

"user\_id": 2

}

**Response**

**Success (200 OK)**

Json: {

"success": true,

"message": "Order claimed successfully"

}

**Failure (400 Bad Request / 401 Unauthorized / 404 Not Found)**

Json: {

"success": false,

"message": "Order not found"

}

**🚀 Conclusion**

This API system is designed for **high performance, security, and scalability** with: ✅ **Optimized queries** for 1M+ orders.  
✅ **Caching (Redis)** for reducing database load.  
✅ **Indexing** for **faster** query execution.  
✅ **Security enhancements** like rate-limiting & API authentication.