**📄 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": "Rishi Goyal",

"email": "rishi.goyal@gmail.com",

"password": "Rishi@12345"

}

**Response**

**Success (200 Created)**

Json :{

"message": "User registered successfully"

}

**Failure (400 Bad Request)**

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**,

    "message": "User logged in successfully.",

    "data": {

        "token": "",

        "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 )**

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: { "success": **true**,

    "message": "Cancel order list found.",

    "data": {

        "orders": [

            {

                "order\_id": 1,

                "original\_price": "250.00",

                "discounted\_price": "175.00",

                "order\_lat": "17.385044",

                "order\_lng": "78.486671",

                "restaurant\_name": "Veggie Delight",

                "items": [

                    {

                        "item\_name": "Paneer Butter Masala",

                        "category": "veg"

                    },

                    {

                        "item\_name": "Garlic Naan",

                        "category": "veg"

                    }

                ]

            },

], "pagination": {

            "current\_page": 1,

            "total\_pages": 3,

            "total\_orders": 26

        }

    }

}

**Failure (400 Bad Request)**

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 )**

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.