

# Online Restaurant Ordering System API - User Manual

## Table of Contents

1. [Overview](#)
2. [Getting Started](#)
3. [Authentication](#)
4. [API Usage Examples](#)
5. [Common Use Cases](#)
6. [Error Handling](#)
7. [Support](#)

## Overview

The Online Restaurant Ordering System API is a comprehensive solution that enables restaurants to manage their digital ordering operations. This API provides functionality for menu management, order processing, customer management, inventory tracking, and payment processing.

## Key Features

- **Menu Management:** Browse available menu items with filtering by category, price, and dietary preferences
- **Order Placement:** Place orders as registered customers or guests
- **Order Tracking:** Track order status using unique tracking numbers
- **Customer Management:** Register and manage customer accounts
- **Inventory Tracking:** Monitor ingredient availability
- **Reviews & Ratings:** Submit and view customer reviews
- **Promotional Codes:** Apply discounts and special offers

## Target Audience

- Restaurant staff and managers
- Mobile app developers
- Web application developers
- Third-party integration partners

## Getting Started

## Base URL

All API requests should be made to:

```
http://localhost:8000
```

## Prerequisites

- Internet connection
- HTTP client (Postman, curl, or your preferred tool)
- Basic understanding of REST APIs

## Making Your First Request

Test the API connection by browsing the menu:

### Request:

```
http  
  
GET /menu_items/
```

### Response:

```
json  
  
[  
  {  
    "id": 1,  
    "name": "Classic Burger",  
    "description": "Juicy beef patty with lettuce, tomato, and cheese",  
    "price": 12.99,  
    "calories": 650,  
    "food_category": "regular",  
    "is_available": true,  
    "average_rating": 4.2,  
    "review_count": 15  
  }  
]
```

## Authentication

The current version of the API does not require authentication for basic operations. However, some administrative functions may require proper authorization in future versions.

## API Usage Examples

### 1. Browse Menu Items

**Get all available menu items:**

http

```
GET /menu_items/?available_only=true
```

**Search menu items by category:**

http

```
GET /menu_items/search?category=vegetarian&sort_by=price_asc
```

**Search menu items by name:**

http

```
GET /menu_items/search?search_term=burger&max_price=15.00
```

### 2. Place an Order as Guest

**Step 1: Prepare your order data**

json

```
{  
  "guest_name": "John Smith",  
  "guest_phone": "5551234567",  
  "guest_email": "john@email.com",  
  "order_type": "takeout"  
}
```

**Step 2: Prepare order items**

json

```
[
  {
    "menu_item_id": 1,
    "quantity": 2
  },
  {
    "menu_item_id": 5,
    "quantity": 1
  }
]
```

### Step 3: Place the order

```
http
POST /customer_actions/orders/guest
Content-Type: application/json

{
  "guest_info": {
    "guest_name": "John Smith",
    "guest_phone": "5551234567",
    "guest_email": "john@email.com",
    "order_type": "takeout"
  },
  "order_items": [
    {"menu_item_id": 1, "quantity": 2},
    {"menu_item_id": 5, "quantity": 1}
  ]
}
```

### Response:

```
json

{
  "order_id": 123,
  "tracking_number": "ORD4F2A8B1C",
  "message": "Order placed successfully! Save your tracking number.",
  "estimated_completion": "30-45 minutes",
  "total_amount": 28.47
}
```

### 3. Track Your Order

Using your tracking number:

http

GET /customer\_actions/orders/track/ORD4F2A8B1C

Response:

json

```
{
  "id": 123,
  "tracking_number": "ORD4F2A8B1C",
  "status": "in_progress",
  "order_date": "2024-08-07T14:30:00Z",
  "order_type": "takeout",
  "total_amount": 28.47,
  "estimated_completion": "2024-08-07T15:15:00Z"
}
```

### 4. Submit a Review

Rate and review a menu item:

http

POST /reviews/

Content-Type: application/json

```
{
  "menu_item_id": 1,
  "customer_name": "John Smith",
  "rating": 5,
  "review_text": "Absolutely delicious! Best burger in town."
}
```

### 5. Register as a Customer

Create a customer account:

http

POST /customers/

Content-Type: application/json

```
{  
  "customer_name": "Jane Doe",  
  "customer_email": "jane@email.com",  
  "customer_phone": "5559876543",  
  "customer_address": "123 Main St, City, State 12345"  
}
```

## Common Use Cases

### For Customers

#### 1. Finding Vegetarian Options

http

GET /customer\_actions/menu/search?category=vegetarian&sort\_by=price\_asc

#### 2. Finding Budget-Friendly Meals

http

GET /customer\_actions/menu/search?max\_price=10.00&sort\_by=price\_asc

#### 3. Checking Order Status

http

GET /customer\_actions/orders/track/{your\_tracking\_number}

### For Restaurant Staff

#### 1. View All Orders for Today

http

GET /staff\_actions/orders/date-range?start\_date=2024-08-07&end\_date=2024-08-07

#### 2. Check Low Stock Items

http

```
GET /staff_actions/inventory/low-stock?threshold=10
```

### 3. Calculate Daily Revenue

```
http
```

```
GET /staff_actions/revenue/daily?target_date=2024-08-07
```

## Error Handling

The API uses standard HTTP status codes:

### Success Codes

- `200 OK` - Request successful
- `201 Created` - Resource created successfully
- `204 No Content` - Request successful, no content to return

### Client Error Codes

- `400 Bad Request` - Invalid request data
- `404 Not Found` - Resource not found
- `409 Conflict` - Resource already exists
- `422 Unprocessable Entity` - Invalid data format

### Server Error Codes

- `500 Internal Server Error` - Server-side error

### Example Error Response

```
json
{
  "detail": "Menu item with id 999 not found"
}
```

## Data Formats

### Order Status Types

- `pending` - Order received, not yet confirmed

- `confirmed` - Order confirmed and being prepared
- `in_progress` - Order is being prepared
- `awaiting_pickup` - Order ready for pickup
- `out_for_delivery` - Order is being delivered
- `completed` - Order completed
- `cancelled` - Order cancelled

## Order Types

- `dine_in` - Customer dining in restaurant
- `takeout` - Customer picking up order
- `delivery` - Order will be delivered

## Food Categories

- `vegetarian` - Contains no meat
- `vegan` - Contains no animal products
- `gluten_free` - Contains no gluten
- `regular` - Standard menu item
- `keto` - Ketogenic diet friendly
- `low_carb` - Low carbohydrate content

## Payment Types

- `cash` - Cash payment
- `credit_card` - Credit card payment
- `debit_card` - Debit card payment

## Rate Limits

Currently, there are no rate limits imposed on API requests. However, please use the API responsibly to ensure optimal performance for all users.

## Support

For technical support or questions about using the API:

1. **Review this documentation** first for common solutions
2. **Check error messages** - they often contain helpful information



3. **Test with sample data** to isolate issues
4. **Contact support** with specific error messages and request details

## Helpful Tips

- Always validate your JSON data before sending requests
- Use appropriate HTTP methods (GET for reading, POST for creating, etc.)
- Check that menu items are available before adding them to orders
- Save tracking numbers immediately after placing orders
- Use the search functionality to find specific menu items quickly

---

*This manual covers the essential features of the Online Restaurant Ordering System API. For advanced technical details and development information, please refer to the Technical Documentation.*