

# CentraMart: A Single-Vendor E-Commerce Management System API Documentation

---

## Overview

This document describes the RESTful APIs for the CentraMart single-vendor e-commerce management system. The APIs support product catalog management, cart and checkout workflows, order lifecycle enforcement, payment and wallet handling, coupon validation, and administrative reporting. The backend follows REST principles using Node.js, Express.js, and MongoDB.

All APIs return JSON responses and enforce authentication and authorization based on user roles.

## 1. Authentication & Authorization APIs

### User Signup:-

POST /auth/signup

Auth: No

Request Body :-

```
{ "name": "string", "email": "string", "phone": "string",  
"password": "string" }
```

Success (201)

```
{ "status": "success", "message": "User registered  
successfully. OTP sent." }
```

Errors

- 400 Invalid input
- 409 User already exists

## OTP Verification

POST /auth/verify-otp

Auth: No

Request Body

```
{ "email": "string", "otp": "string" }
```

Success

```
{ "status": "success", "verified": true }
```

Errors

- 400 Invalid or expired OTP

## User Login

POST /auth/login

Auth: No

Request Body

```
{ "email": "string", "password": "string" }
```

Success

```
{ "status": "success", "token": "jwt_token", "user": {  
    "id": "string", "name": "string", "role": "customer" } }
```

Errors

- 401 Invalid credentials
- 403 OTP not verified

## 2. User Management APIs

### Get User Profile

GET /users/profile

Auth: Customer

Success

```
{"status": "success", "user": { "name": "string", "email": "string", "phone": "string", "addresses": []}}
```

Errors

- 401 Unauthorized

## **Update User Profile**

PUT /users/profile

Auth: Customer

Request Body

```
{ "name": "string", "phone": "string"}
```

Success

```
{ "status": "success", "message": "Profile updated successfully"}
```

Errors

- 400 Invalid input
- 401 Unauthorized

## **3. Product Catalog APIs**

### **Get Product List**

GET /products

Auth: No

Query Params (optional)

- category
- brand
- search
- sort

Success

```
{ "status": "success", "products": []}
```

Errors

- 400 Invalid query parameters
- 500 Internal server error

## Get Product Details

GET /products/:id

Auth: No

Success

```
{ "status": "success", "product": {}}
```

Errors

- 404 Product not found

## Create Product

POST /admin/products

Auth: Admin

Request Body

```
{ "name": "string", "description": "string", "price": "number", "categoryId": "string", "brandId": "string", "stock": "number"}
```

Success

```
{"status": "success", "productId": "string"}
```

Errors

- 403 Forbidden
- 400 Invalid input

## 4. Cart Management APIs

### Add Item to Cart

POST /cart/items

Auth: Customer

Request Body

```
{ "productId": "string", "quantity": "number"}
```

Success

```
{ "status": "success", "cart": {}}
```

Errors

- 404 Product not found
- 400 Insufficient stock

### Get Cart

GET /cart

Auth: Customer

Request Body

N/A

Success

```
{ "status": "success", "cart": {} }
```

Errors

- 401 Unauthorized

## 5. Checkout & Order APIs

### Create Order

POST /orders

Auth: Customer

Request Body

```
{ "addressId": "string", "paymentMethod": "ONLINE | COD | WALLET" }
```

Success

```
{"status": "success", "orderId": "string"}
```

Errors

- 400 Invalid checkout data
- 401 Unauthorized

### Get Order Details

GET /orders/:id

Auth: Customer / Admin

Request Body

N/A

Success

```
{ "status": "success", "order": {} }
```

Errors

- 404 Order not found
- 403 Forbidden

## 6. Payment & Wallet APIs

### Initiate Payment

POST /payments/initiate

Auth: Customer

Request Body

```
{ "orderId": "string" }
```

Success

```
{ "status": "success", "paymentReference": "string" }
```

Errors

- 400 Invalid request
- 401 Unauthorized

### Verify Payment

POST /payments/verify

Auth: Customer

Request Body

```
{ "paymentReference": "string" }
```

Success

```
{ "status": "success", "paymentStatus": "Paid" }
```

Errors

- 400 Verification failed

## **Get Wallet**

GET /wallet

Auth: Customer

Request Body

N/A

Success

```
{ "status": "success", "balance": "number" }
```

Errors

- 401 Unauthorized

## **Get Wallet Transactions**

GET /wallet/transactions

Auth: Customer

Request Body

N/A

Success

```
{ "status": "success", "transactions": [ ] }
```

Errors

401 Unauthorized

# **7. Administrative Order APIs**

## **Get All Orders**

GET /admin/orders

Auth: Admin

Request Body

N/A

Success

```
{ "status": "success", "orders": []}
```

Errors

- 403 Forbidden

## Update Order Status

PUT /admin/orders/:id/status

Auth: Admin

Request Body

```
{ "status": "Processing | Completed | Cancelled"}
```

Success

```
{ "status": "success", "message": "Order status updated"}
```

Errors

- 400 Invalid status transition
- 403 Forbidden

## 8. Coupon APIs

### Create Coupon

POST /admin/coupons

Auth: Admin

Request Body: { "code": "string", "discountType": "PERCENTAGE | FLAT", "discountValue": "number", "expiryDate": "date", "usageLimit": "number"}

Success

```
{ "status": "success", "couponId": "string"}
```

Errors

- 400 Invalid input

## Apply Coupon

POST /cart/coupon

Auth: Customer

Request Body

```
{ "couponCode": "string"}
```

Success

```
{"status": "success", "discountApplied": "number",  
"updatedTotal": "number"}
```

Errors

- 400 Invalid or expired coupon

## 9. Review & Ratings APIs

### Add Product Review

POST /products/:id/reviews

Auth: Customer

Request Body

```
{ "rating": "number", "comment": "string"}
```

Success

```
{ "status": "success", "message": "Review submitted  
successfully"}
```

## Errors

- 400 Invalid rating

# 10. Sales & Reporting APIs

## Generate Sales Report

GET /admin/reports/sales

Auth: Admin

Query Params

- fromDate
- toDate
- reportType (overall | brand | category | payment)

Success

```
{ "status": "success", "report": []}
```

## Errors

- 400 Invalid report parameters
- 403 Forbidden

# 11. Admin Roles & Permissions APIs

## Get Roles

GET /admin/roles

Auth: Admin

## Update Role Permissions

PUT /admin/roles/:id

Auth: Admin

Request Body

N/A

Success

```
{ "status": "success", "message": "Role permissions updated" }
```

Errors

400 Invalid permissions data

403 Forbidden