API Documentation

Complete API reference for the E-commerce Backend.

Base URL

```
http://localhost:5000/api/v1
```

Response Format

All responses follow this structure:

Success Response:

```
json

{
  "success": true,
  "message": "Optional message",
  "data": { ... }
}
```

Error Response:

```
json

{
  "success": false,
  "error": "Error message"
}
```

Paginated Response:

```
json
```

```
{
  "success": true,
  "data": [...],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 100,
    "pages": 5
  }
}
```

Authentication

Register User

Create a new user account.

Endpoint: `POST /auth/register`

Request Body:

json

```
{
  "firstName": "John",
  "lastName": "Doe",
  "email": "john@example.com",
  "password": "Password@123",
  "phone": "9876543210"
}
```

Validation Rules:

- firstName: Required, max 50 chars
- lastName: Required, max 50 chars
- email: Required, valid email format
- password: Required, min 8 chars, must contain uppercase, lowercase, number, special char
- phone: Optional, 10-15 digits

Response: `201 Created`

json

```
{
  "success": true,
  "message": "User registered successfully",
  "data": {
    "user": {
      "_id": "507f1f77bcf86cd799439011",
      "firstName": "John",
      "lastName": "Doe",
      "email": "john@example.com",
      "role": "user"
    },
    "tokens": {
      "accessToken": "eyJhbGci...",
      "refreshToken": "eyJhbGci..."
    }
  }
}
```

Login

Authenticate and receive tokens.

Endpoint: `POST /auth/login`

Request Body:

json

```
{
  "email": "john@example.com",
  "password": "Password@123"
}
```

Response: `200 OK`

json

```
{
  "success": true,
  "message": "Login successful",
  "data": {
    "user": {
      "_id": "507f1f77bcf86cd799439011",
      "firstName": "John",
      "lastName": "Doe",
      "email": "john@example.com",
      "role": "user"
    },
    "tokens": {
      "accessToken": "eyJhbGci...",
      "refreshToken": "eyJhbGci..."
    }
  }
}
```

Refresh Token

Get new access token using refresh token.

Endpoint: `POST /auth/refresh-token`

Request Body:

```json
{
  "refreshToken": "eyJhbGci..."
}
```

Response: `200 OK`

```json
```

```
{
  "success": true,
  "data": {
    "tokens": {
      "accessToken": "eyJhbGci...",
      "refreshToken": "eyJhbGci..."
    }
  }
}
```

Logout

Invalidate refresh token.

Endpoint: `POST /auth/logout` **Auth Required:** Yes

Response: `200 OK`

```json
{
  "success": true,
  "message": "Logout successful"
}
```

Change Password

Change user password.

Endpoint: `POST /auth/change-password` **Auth Required:** Yes

Request Body:

```json
{
  "currentPassword": "OldPassword@123",
  "newPassword": "NewPassword@123"
}
```

Response: `200 OK`

Products

List Products

Get paginated list of products with filters.

Endpoint: `GET /products`

Query Parameters:

- `page` (number): Page number (default: 1)
- `limit` (number): Items per page (default: 20, max: 100)
- `sort` (string): Sort field (default: createdAt)
- `order` (string): asc or desc (default: desc)
- `category` (string): Category ID
- `subcategory` (string): Subcategory ID
- `minPrice` (number): Minimum price
- `maxPrice` (number): Maximum price
- `minRating` (number): Minimum rating (1-5)
- `featured` (boolean): Filter featured products
- `isNew` (boolean): Filter new products
- `inStock` (boolean): Filter in-stock products
- `search` (string): Search query
- `tags` (string): Comma-separated tags

Example: `/products?category=507f...&minPrice=1000&maxPrice=5000&page=1&limit=20`

Response: `200 OK`

```
json
```

```
{
  "success": true,
  "data": [
    {
      "_id": "507f1f77bcf86cd799439011",
      "name": "Wireless Headphones",
      "slug": "wireless-headphones",
      "description": "Premium noise-cancelling headphones",
      "price": 2999,
      "originalPrice": 4999,
      "category": {
        "_id": "507f...",
        "name": "Electronics",
        "slug": "electronics"
      },
      "images": ["https://..."],
      "thumbnail": "https://...",
      "sku": "ELEC-WH-001",
      "stock": 50,
      "rating": 4.5,
      "reviewCount": 128,
      "inStock": true,
      "featured": true,
      "isNew": true,
      "tags": ["headphones", "wireless"],
      "createdAt": "2024-01-01T00:00:00.000Z"
    }
  ],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 100,
    "pages": 5
  }
}
```

Get Product by ID

Get single product details.

Endpoint: `GET /products/:id`

Response: `200 OK`

```json
{
  "success": true,
  "data": {
    "_id": "507f1f77bcf86cd799439011",
    "name": "Wireless Headphones",
    ...
  }
}
```

Get Product by Slug

Get single product by slug.

Endpoint: `GET /products/slug/:slug`

Get Featured Products

Get featured products.

Endpoint: `GET /products/featured?limit=10`

Get New Products

Get new arrivals.

Endpoint: `GET /products/new?limit=10`

Create Product (Admin)

Create a new product.

Endpoint: `POST /products` **Auth Required:** Yes (Admin)

Request Body:

```json
```

```
{
  "name": "Product Name",
  "description": "Product description",
  "price": 2999,
  "originalPrice": 4999,
  "category": "507f1f77bcf86cd799439011",
  "images": ["https://...", "https://..."],
  "sku": "PROD-001",
  "stock": 100,
  "featured": false,
  "isNew": true,
  "tags": ["tag1", "tag2"]
}
```

Response: `201 Created`

Update Product (Admin)

Update product details.

Endpoint: `PUT /products/:id` **Auth Required:** Yes (Admin)

Delete Product (Admin)

Soft delete a product.

Endpoint: `DELETE /products/:id` **Auth Required:** Yes (Admin)

Cart

Get Cart

Get user's cart.

Endpoint: `GET /cart` **Auth Required:** Yes

Response: `200 OK`

```
json
```

```
{
  "success": true,
  "data": {
    "_id": "507f1f77bcf86cd799439011",
    "user": "507f...",
    "items": [
      {
        "product": {
          "_id": "507f...",
          "name": "Product Name",
          "price": 2999,
          "images": ["https://..."],
          "inStock": true,
          "stock": 50
        },
        "quantity": 2,
        "price": 2999
      }
    ],
    "totalItems": 2,
    "subtotal": 5998
  }
}
```

Add to Cart

Add item to cart.

Endpoint: `POST /cart/items` **Auth Required:** Yes

Request Body:

```json
{
  "productId": "507f1f77bcf86cd799439011",
  "quantity": 2
}
```

Response: `200 OK`

Update Cart Item

Update item quantity.

Endpoint: `PUT /cart/items/:productId` **Auth Required:** Yes

Request Body:

```json
{
  "quantity": 3
}
```

Remove from Cart

Remove item from cart.

Endpoint: `DELETE /cart/items/:productId` **Auth Required:** Yes

Clear Cart

Remove all items from cart.

Endpoint: `DELETE /cart` **Auth Required:** Yes

Orders

Create Order

Create order from cart.

Endpoint: `POST /orders` **Auth Required:** Yes

Request Body:

```json
```

```
{
  "shippingAddress": {
    "fullName": "John Doe",
    "phone": "9876543210",
    "addressLine1": "123 Main Street",
    "addressLine2": "Apt 4B",
    "city": "Mumbai",
    "state": "Maharashtra",
    "postalCode": "400001",
    "country": "India"
  },
  "paymentMethod": "card",
  "couponCode": "SAVE20",
  "notes": "Please deliver before 6 PM"
}
```

Response: 201 Created

json

```json
{
  "success": true,
  "message": "Order created successfully",
  "data": {
    "_id": "507f1f77bcf86cd799439011",
    "orderNumber": "ORD17066...",
    "user": "507f...",
    "items": [...],
    "shippingAddress": {...},
    "subtotal": 5998,
    "discount": 0,
    "shippingFee": 50,
    "tax": 1079.64,
    "total": 7127.64,
    "status": "pending",
    "paymentMethod": "card",
    "paymentStatus": "pending",
    "createdAt": "2024-01-01T00:00:00.000Z"
  }
}
```

Get User Orders

Get current user's order history.

Endpoint: `GET /orders/my-orders?page=1&limit=20` **Auth Required:** Yes

Get Order Details

Get single order details.

Endpoint: `GET /orders/:id` **Auth Required:** Yes

Cancel Order

Cancel an order.

Endpoint: `POST /orders/:id/cancel` **Auth Required:** Yes

Request Body:

```json
{
  "reason": "Changed my mind"
}
```

Get All Orders (Admin)

Get all orders.

Endpoint: `GET /orders/admin/all?page=1&limit=20` **Auth Required:** Yes (Admin)

Update Order Status (Admin)

Update order status.

Endpoint: `PATCH /orders/:id/status` **Auth Required:** Yes (Admin)

Request Body:

```json
{
  "status": "shipped"
}
```

Valid statuses: pending, paid, processing, shipped, delivered, cancelled, refunded

Get Order Stats (Admin)

Get order statistics.

Endpoint: `GET /orders/admin/stats` **Auth Required:** Yes (Admin)

Error Codes

Code	Description
400	Bad Request - Invalid input
401	Unauthorized - Invalid or missing token
403	Forbidden - Insufficient permissions
404	Not Found - Resource not found
409	Conflict - Duplicate entry
500	Internal Server Error

Rate Limiting

- **Window:** 15 minutes
- **Max Requests:** 100 per IP
- **Response:** 429 Too Many Requests

Notes

- All timestamps are in ISO 8601 format (UTC)
- All prices are in INR (paise for calculations)
- File uploads not yet implemented
- Payment integration ready but requires gateway setup