Ecommerce API Documentation

Base URL

```
http://localhost:8080/api/v1
```

Authentication

- Protected endpoints require JWT token in Authorization header
- Format: `Authorization: Bearer <jwt_token>`
- Get JWT token from login endpoint

Authentication API

Base URL

```
http://localhost:8080/api/v1/auth
```

Endpoints

1. Register User

POST `/public/register`

Creates a new user account.

Request Body

```
{
  "email": "user@example.com",
  "password": "password123",
  "full_name": "John Doe",
  "mobile": "9876543210"
}
```

Request Fields

Field	Type	Required	Validation
email	string	Yes	Must be valid email
password	string	Yes	Minimum 6 characters
full_name	string	Yes	Minimum 2 characters
mobile	string	No	10 digits starting with 6-9

cURL Example

```
curl -X POST http://localhost:8080/api/v1/auth/public/register \
  -H "Content-Type: application/json" \
  -d '{
    "email": "john.doe@example.com",
    "password": "securepass123",
    "full_name": "John Doe",
    "mobile": "9876543210"
  }'
```

Success Response (201)

```
{
  "success": true,
  "message": "Registration successful",
  "data": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "email": "user@example.com",
    "full_name": "John Doe",
    "mobile": "9876543210",
    "status": "Active",
    "role": "user",
    "created_at": "2025-06-12T10:30:00Z"
  }
}
```

Error Responses

- **400 Bad Request**: Email already registered
- **400 Bad Request**: Validation errors (invalid email, short password, etc.)

---

2. Login User

POST `/public/login`

Authenticates a user and returns a JWT token.

Request Body

```
{
  "email": "user@example.com",
  "password": "password123"
}
```

Request Fields

Field	Type	Required	Validation
email	string	Yes	Must be valid email
password	string	Yes	Minimum 6 characters

Success Response (200)

```
{
  "success": true,
  "message": "Login successful",
  "data": {
    "user": {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "email": "user@example.com",
      "full_name": "John Doe",
      "mobile": "9876543210",
      "status": "Active",
      "role": "user",
      "created_at": "2025-06-12T10:30:00Z"
    },
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
  }
}
```

Error Responses

- **401 Unauthorized**: Invalid email or password
- **400 Bad Request**: Validation errors

User Management API

Base URL

```
http://localhost:8080/api/v1/user
```

Endpoints

1. Get Current User Profile

GET `/me/get`

Retrieves the authenticated user's profile information.

Headers

```
 Authorization: Bearer <jwt_token>
Content-Type: application/json
```

Success Response (200)

```
{
  "success": true,
  "message": "User fetched successfully",
  "data": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "email": "user@example.com",
    "full_name": "John Doe",
    "mobile": "9876543210",
    "status": "Active",
    "role": "user",
    "created_at": "2025-06-12T10:30:00Z"
  }
}
```

cURL Example

```
curl -X GET http://localhost:8080/api/v1/user/me/get \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." \
  -H "Content-Type: application/json"
```

Error Responses

- **401 Unauthorized**: User not authenticated
- **403 Forbidden**: Invalid or expired token

---

2. Update User Profile

PUT `/me/update`

Updates the authenticated user's profile information.

Headers

```
 Authorization: Bearer <jwt_token>
Content-Type: application/json
```

Request Body

```
{
  "email": "newemail@example.com",
  "full_name": "John Smith",
  "mobile": "9123456789"
}
```

Request Fields

Field	Type	Required	Validation
email	string	No	Must be valid email format
full_name	string	No	Minimum 2 characters
mobile	string	No	10 digits starting with 6-9

Success Response (200)

```
{
  "success": true,
  "message": "User updated successfully",
  "data": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "email": "newemail@example.com",
    "full_name": "John Smith",
    "mobile": "9123456789",
    "status": "Active",
    "role": "user",
    "created_at": "2025-06-12T10:30:00Z"
  }
}
```

cURL Example

```
curl -X PUT http://localhost:8080/api/v1/user/me/update \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." \
  -H "Content-Type: application/json" \
  -d '{
    "email": "newemail@example.com",
    "full_name": "John Smith",
    "mobile": "9123456789"
  }'
```

Error Responses

- **400 Bad Request**: Validation errors
- **401 Unauthorized**: User not authenticated
- **500 Internal Server Error**: Database error

---

3. Change User Status (Admin Only)

PUT `/admin/status`

Changes a user's status (Active/Inactive). Admin access required.

Headers

```
 Authorization: Bearer <admin_jwt_token>
Content-Type: application/json
```

Request Body

```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "status": "Inactive"
}
```

Request Fields

Field	Type	Required	Validation
user_id	UUID	Yes	Valid UUID format
status	string	No	Must be "Active" or "Inactive"

Success Response (200)

```
{
  "success": true,
  "message": "User status changed successfully",
  "data": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "email": "user@example.com",
    "full_name": "John Doe",
    "mobile": "9876543210",
    "status": "Inactive",
    "role": "user",
    "created_at": "2025-06-12T10:30:00Z"
  }
}
```

cURL Example

```
curl -X PUT http://localhost:8080/api/v1/user/admin/status \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." \
  -H "Content-Type: application/json" \
  -d '{
    "user_id": "550e8400-e29b-41d4-a716-446655440000",
    "status": "Inactive"
  }'
```

Error Responses

- **400 Bad Request**: Validation errors (invalid status)
- **401 Unauthorized**: User not authenticated
- **403 Forbidden**: Admin access required
- **404 Not Found**: User not found

Product API Documentation

Base URL

```
http://localhost:8080/api/v1/products
```

Public Endpoints (No Authentication Required)

1. Get All Products

GET `/public`

Retrieves all available products for public viewing.

Success Response (200)

```
{
  "success": true,
  "message": "Products retrieved successfully",
  "data": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "name": "iPhone 15 Pro",
      "description": "Latest iPhone with advanced features and premium build quality",
      "images": [
        "https://example.com/image1.jpg",
        "https://example.com/image2.jpg"
      ],
      "brand": "Apple",
      "category": "Smartphones",
      "price": 999.99,
      "is_available": true,
      "count_in_stock": 50,
      "created_at": "2025-06-12T10:30:00Z"
    }
  ]
}
```

Example Usage

```
curl -X GET http://localhost:8080/api/v1/products/public
```

---

2. Get Product by ID

GET `/public/{id}`

Retrieves a specific product by its ID.

Path Parameters

Parameter	Type	Required	Description
id	UUID	Yes	Product ID

Success Response (200)

```
{
  "success": true,
  "message": "Product retrieved successfully",
  "data": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "iPhone 15 Pro",
    "description": "Latest iPhone with advanced features and premium build quality",
    "images": [
      "https://example.com/image1.jpg",
      "https://example.com/image2.jpg"
    ],
    "brand": "Apple",
    "category": "Smartphones",
    "price": 999.99,
    "is_available": true,
    "count_in_stock": 50,
    "created_at": "2025-06-12T10:30:00Z"
  }
}
```

Error Responses

- **404 Not Found**: Product not found

Example Usage

```
curl -X GET http://localhost:8080/api/v1/products/public/550e8400-e29b-41d4-a716-446655440000
```

Admin Endpoints (Authentication Required)

> **Note**: All admin endpoints require JWT authentication and admin role.
>
> **Headers Required:**
>
> ```
>  Authorization: Bearer <your_jwt_token>
> Content-Type: application/json
> ```

3. Create Product

POST `/admin/create`

Creates a new product (Admin only).

Request Body

```
{
  "name": "iPhone 15 Pro",
  "description": "Latest iPhone with advanced features and premium build quality",
  "images": [
    "https://example.com/image1.jpg",
    "https://example.com/image2.jpg"
  ],
  "brand": "Apple",
  "category": "Smartphones",
  "price": 999.99,
  "count_in_stock": 50
}
```

Request Fields

Field	Type	Required	Validation
name	string	Yes	Minimum 2 characters
description	string	Yes	Minimum 10 characters
images	array	Yes	At least 1 image URL
brand	string	Yes	Minimum 2 characters
category	string	Yes	Minimum 2 characters
price	number	Yes	Must be greater than 0
count_in_stock	integer	Yes	Cannot be negative

Success Response (201)

```
{
  "success": true,
  "message": "Product created successfully",
  "data": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "iPhone 15 Pro",
    "description": "Latest iPhone with advanced features and premium build quality",
    "images": [
      "https://example.com/image1.jpg",
      "https://example.com/image2.jpg"
    ],
    "brand": "Apple",
    "category": "Smartphones",
    "price": 999.99,
    "is_available": true,
    "count_in_stock": 50,
    "created_at": "2025-06-12T10:30:00Z"
  }
}
```

Error Responses

- **400 Bad Request**: Validation errors
- **401 Unauthorized**: Invalid or missing token
- **403 Forbidden**: Not admin user

Example Usage

```
curl -X POST http://localhost:8080/api/v1/products/admin/create \
  -H "Authorization: Bearer <your_jwt_token>" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "iPhone 15 Pro",
    "description": "Latest iPhone with advanced features",
    "images": ["https://example.com/image1.jpg"],
    "brand": "Apple",
    "category": "Smartphones",
    "price": 999.99,
    "count_in_stock": 50
  }'
```

4. Get All Products (Admin)

GET `/admin/all`

Retrieves all products including unavailable ones (Admin only).

Success Response (200)

```
{
  "success": true,
  "message": "Products retrieved successfully",
  "data": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "name": "iPhone 15 Pro",
      "description": "Latest iPhone with advanced features",
      "images": ["https://example.com/image1.jpg"],
      "brand": "Apple",
      "category": "Smartphones",
      "price": 999.99,
      "is_available": false,
      "count_in_stock": 0,
      "created_at": "2025-06-12T10:30:00Z"
    }
  ]
}
```

Example Usage

```
curl -X GET http://localhost:8080/api/v1/products/admin/all \
  -H "Authorization: Bearer <your_jwt_token>"
```

5. Update Product

PUT `/admin/{id}`

Updates an existing product (Admin only).

Path Parameters

Parameter	Type	Required	Description
id	UUID	Yes	Product ID

Request Body

```
{
  "name": "iPhone 15 Pro Max",
  "description": "Updated description for the latest iPhone",
  "images": [
    "https://example.com/new-image1.jpg",
    "https://example.com/new-image2.jpg"
  ],
  "brand": "Apple",
  "category": "Smartphones",
  "price": 1099.99,
  "count_in_stock": 30
}
```

Success Response (200)

```
{
  "success": true,
  "message": "Product updated successfully",
  "data": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "iPhone 15 Pro Max",
    "description": "Updated description for the latest iPhone",
    "images": [
      "https://example.com/new-image1.jpg",
      "https://example.com/new-image2.jpg"
    ],
    "brand": "Apple",
    "category": "Smartphones",
    "price": 1099.99,
    "is_available": true,
    "count_in_stock": 30,
    "created_at": "2025-06-12T10:30:00Z"
  }
}
```

Error Responses

- **400 Bad Request**: Validation errors
- **404 Not Found**: Product not found
- **401 Unauthorized**: Invalid or missing token
- **403 Forbidden**: Not admin user

Example Usage

```
curl -X PUT http://localhost:8080/api/v1/products/admin/550e8400-e29b-41d4-a716-446655440000 \
  -H "Authorization: Bearer <your_jwt_token>" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "iPhone 15 Pro Max",
    "description": "Updated description",
    "images": ["https://example.com/new-image.jpg"],
    "brand": "Apple",
    "category": "Smartphones",
    "price": 1099.99,
    "count_in_stock": 30
  }'
```

---

6. Delete Product

DELETE `/admin/{id}`

Deletes a product (Admin only).

Path Parameters

Parameter	Type	Required	Description
id	UUID	Yes	Product ID

Success Response (200)

```
{
  "success": true,
  "message": "Product deleted successfully"
}
```

Error Responses

- **404 Not Found**: Product not found
- **401 Unauthorized**: Invalid or missing token
- **403 Forbidden**: Not admin user

Example Usage

```
curl -X DELETE http://localhost:8080/api/v1/products/admin/550e8400-e29b-41d4-a716-446655440000 \
  -H "Authorization: Bearer <your_jwt_token>"
```

---

7. Update Product Status

PUT `/admin/{id}/status`

Updates product availability status (Admin only).

Path Parameters

Parameter	Type	Required	Description
id	UUID	Yes	Product ID

Request Body

```
{
  "is_available": false
}
```

Request Fields

Field	Type	Required	Description
is_available	boolean	Yes	Product availability status

Success Response (200)

```
{
  "success": true,
  "message": "Product status updated successfully",
  "data": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "is_available": false
  }
}
```

Example Usage

```
curl -X PUT http://localhost:8080/api/v1/products/admin/550e8400-e29b-41d4-a716-446655440000/status \
  -H "Authorization: Bearer <your_jwt_token>" \
  -H "Content-Type: application/json" \
  -d '{"is_available": false}'
```

8. Update Product Stock

PUT `/admin/{id}/stock`

Updates product stock count (Admin only).

Path Parameters

Parameter	Type	Required	Description
id	UUID	Yes	Product ID

Request Body

```
{
  "count_in_stock": 25
}
```

Request Fields

Field	Type	Required	Validation
count_in_stock	integer	Yes	Cannot be negative

Success Response (200)

```
{
  "success": true,
  "message": "Product stock updated successfully",
  "data": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "count_in_stock": 25
  }
}
```

Example Usage

```
curl -X PUT http://localhost:8080/api/v1/products/admin/550e8400-e29b-41d4-a716-446655440000/stock \
  -H "Authorization: Bearer <your_jwt_token>" \
  -H "Content-Type: application/json" \
  -d '{"count_in_stock": 25}'
```

Order Management API Documentation

Base URL

```
http://localhost:8080/api/v1/orders
```

Authentication

All order endpoints require JWT authentication. Include the token in the Authorization header:

```
Authorization: Bearer <your_jwt_token>
```

Endpoints

1. Create Order

POST `/me/create`

Creates a new order from the user's current cart items.

Request Body

```
{
  "address_id": "550e8400-e29b-41d4-a716-446655440000"
}
```

Request Fields

Field	Type	Required	Description
address_id	UUID	Yes	ID of the delivery address

curl Example

```
curl -X POST http://localhost:8080/api/v1/orders/me/create \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." \
  -H "Content-Type: application/json" \
  -d '{
    "address_id": "550e8400-e29b-41d4-a716-446655440000"
  }'
```

Success Response (201)

```
{
  "success": true,
  "message": "Order created successfully",
  "data": {
    "order_id": "ORD-2025-001234",
    "total_amount": 299.99
  }
}
```

Error Responses

- **400 Bad Request**: Invalid address ID or empty cart
- **401 Unauthorized**: Missing or invalid JWT token
- **404 Not Found**: Address not found

2. Get User Orders

GET `/me/all`

Retrieves all orders for the authenticated user.

curl Example

```
curl -X GET http://localhost:8080/api/v1/orders/me/all \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

Success Response (200)

```
{
  "success": true,
  "message": "Orders retrieved successfully",
  "data": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "order_id": "ORD-2025-001234",
      "total_amount": 299.99,
      "order_status": "Processing",
      "payment_status": "Paid",
      "created_at": "2025-06-12T10:30:00Z",
      "items": [
        {
          "product_id": "660e8400-e29b-41d4-a716-446655440001",
          "product_name": "Wireless Headphones",
          "quantity": 2,
          "price_at_order_time": 149.99
        }
      ]
    }
  ]
}
```

Error Responses

- **401 Unauthorized**: Missing or invalid JWT token
- **404 Not Found**: No orders found

---

3. Update Order Status (Admin Only)

PUT `/admin/status/{id}`

Updates the status of an order. Requires admin privileges.

URL Parameters

Parameter	Type	Description
id	UUID	Order ID to update

Request Body

```
{
  "order_status": "Shipped",
  "payment_status": "Paid",
  "user_id": "550e8400-e29b-41d4-a716-446655440000"
}
```

Request Fields

Field	Type	Required	Validation
order_status	string	Yes	Must be one of: "Pending", "Processing", "Shipped", "Delivered", "Cancelled"
payment_status	string	No	Must be one of: "Pending", "Paid", "Failed", "Refunded"

Field	Type	Required	Validation
user_id	UUID	Yes	ID of the user who owns the order

curl Example

```
curl -X PUT http://localhost:8080/api/v1/orders/admin/status/550e8400-e29b-41d4-a716-446655440000 \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." \
  -H "Content-Type: application/json" \
  -d '{
    "order_status": "Shipped",
    "payment_status": "Paid",
    "user_id": "550e8400-e29b-41d4-a716-446655440000"
  }'
```

Success Response (200)

```
{
  "success": true,
  "message": "Order status updated successfully",
  "data": {
    "order_id": "ORD-2025-001234",
    "new_status": "Shipped"
  }
}
```

Error Responses

- **400 Bad Request**: Invalid status values or validation errors
- **401 Unauthorized**: Missing or invalid JWT token
- **403 Forbidden**: Insufficient privileges (not admin)
- **404 Not Found**: Order not found

Order Status Values

Order Status

- **Pending**: Order placed but not yet processed
- **Processing**: Order is being prepared
- **Shipped**: Order has been dispatched
- **Delivered**: Order has been delivered to customer
- **Cancelled**: Order has been cancelled

Payment Status

- **Pending**: Payment not yet processed
- **Paid**: Payment completed successfully
- **Failed**: Payment failed or declined
- **Refunded**: Payment has been refunded

Cart API Documentation

Base URL

```
http://localhost:8080/api/v1/cart
```

Authentication

All cart endpoints require JWT authentication. Include the token in the Authorization header:

```
Authorization: Bearer <your_jwt_token>
```

Endpoints

1. Add Product to Cart

POST `/me/add`

Adds a product to the user's cart or updates quantity if product already exists.

Request Body

```
{
  "product_id": "550e8400-e29b-41d4-a716-446655440000",
  "quantity": 2
}
```

Request Fields

Field	Type	Required	Default	Validation
product_id	UUID	Yes	-	Must be valid UUID
quantity	integer	No	1	Must be between 1 and 100

curl Example

```
curl -X POST http://localhost:8080/api/v1/cart/me/add \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." \
  -H "Content-Type: application/json" \
  -d '{
    "product_id": "550e8400-e29b-41d4-a716-446655440000",
    "quantity": 2
  }'
```

Success Response (200)

```
{
  "success": true,
  "message": "Product added to cart successfully",
  "data": {
    "id": "550e8400-e29b-41d4-a716-446655440001",
    "user_id": "550e8400-e29b-41d4-a716-446655440002",
    "product_id": "550e8400-e29b-41d4-a716-446655440000",
    "quantity": 2,
    "created_at": "2025-06-12T10:30:00Z"
  }
}
```

Error Responses

- **400 Bad Request**: Invalid product_id or quantity validation failed
- **401 Unauthorized**: Invalid or missing JWT token
- **404 Not Found**: Product not found
- **409 Conflict**: Not enough stock available

---

2. Get Cart Items

GET `/me`

Retrieves all items in the user's cart with product details.

curl Example

```
curl -X GET http://localhost:8080/api/v1/cart/me \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

Success Response (200)

```
{
  "success": true,
  "message": "Cart retrieved successfully",
  "data": {
    "cart_items": [
      {
        "id": "550e8400-e29b-41d4-a716-446655440001",
        "user_id": "550e8400-e29b-41d4-a716-446655440002",
        "product_id": "550e8400-e29b-41d4-a716-446655440000",
        "quantity": 2,
        "created_at": "2025-06-12T10:30:00Z",
        "product": {
          "id": "550e8400-e29b-41d4-a716-446655440000",
          "name": "iPhone 15 Pro",
          "price": 999.99,
          "images": [
            "https://example.com/image1.jpg",
            "https://example.com/image2.jpg"
          ],
          "count_in_stock": 50,
          "is_available": true
        }
      }
    ],
    "total_items": 2,
    "total_amount": 1999.98
  }
}
```

Error Responses

- **401 Unauthorized**: Invalid or missing JWT token
- **404 Not Found**: Cart is empty

---

3. Remove Item from Cart

DELETE `/me/{cartItemId}`

Removes a specific item from the user's cart.

Path Parameters

Parameter	Type	Required	Description
cartItemId	UUID	Yes	ID of the cart item to remove

curl Example

```
curl -X DELETE http://localhost:8080/api/v1/cart/me/550e8400-e29b-41d4-a716-446655440001 \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

Success Response (200)

```
{
  "success": true,
  "message": "Item removed from cart successfully"
}
```

Error Responses

- **400 Bad Request**: Invalid cartItemId format
- **401 Unauthorized**: Invalid or missing JWT token
- **404 Not Found**: Cart item not found or doesn't belong to user

---

4. Clear Cart

DELETE `/me`

Removes all items from the user's cart.

curl Example

```
curl -X DELETE http://localhost:8080/api/v1/cart/me \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

Success Response (200)

```
{
  "success": true,
  "message": "Cart cleared successfully"
}
```

Error Responses

- **401 Unauthorized**: Invalid or missing JWT token
- **404 Not Found**: Cart is already empty

Data Models

CartProduct

```
{
  "id": "UUID",           // Unique cart item identifier
  "user_id": "UUID",      // User who owns this cart item
  "product_id": "UUID",   // Product being added to cart
  "quantity": "integer",  // Number of items (1-100)
  "created_at": "DateTime" // When item was added to cart
}
```

Product

```
{
  "id": "UUID",              // Unique product identifier
  "name": "string",          // Product name
  "price": "float",          // Product price
  "images": ["string"],      // Array of image URLs
  "count_in_stock": "integer", // Available stock count
  "is_available": "boolean"   // Whether product is available for purchase
}
```

Address Management API Documentation

Base URL

```
http://localhost:8080/api/v1/addresses/me
```

Authentication

All address endpoints require JWT authentication. Include the token in the Authorization header:

```
Authorization: Bearer <your_jwt_token>
```

Endpoints

1. Create Address

POST `/`

Creates a new address for the authenticated user.

Request Body

```
{
  "address_line1": "123 Main Street, Apartment 4B",
  "city": "Mumbai",
  "state": "Maharashtra",
  "pincode": "400001",
  "country": "India",
  "mobile": "9876543210",
  "selected": true
}
```

Request Fields

Field	Type	Required	Validation
address_line1	string	Yes	Minimum 5 characters
city	string	Yes	Minimum 2 characters
state	string	Yes	Minimum 2 characters
pincode	string	Yes	5-6 digits
country	string	No	Minimum 2 characters (defaults to "India")
mobile	string	Yes	10 digits starting with 6-9
selected	boolean	No	Mark as selected address

curl Example

```
curl -X POST http://localhost:8080/api/v1/addresses/me/ \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." \
  -H "Content-Type: application/json" \
  -d '{
    "address_line1": "123 Main Street, Apartment 4B",
    "city": "Mumbai",
    "state": "Maharashtra",
    "pincode": "400001",
    "country": "India",
    "mobile": "9876543210",
    "selected": true
  }'
```

Success Response (201)

```
{
  "success": true,
  "message": "Address created successfully",
  "data": {
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "address_line1": "123 Main Street, Apartment 4B",
    "city": "Mumbai",
    "state": "Maharashtra",
    "pincode": "400001",
    "country": "India",
    "mobile": "9876543210",
    "selected": true,
    "user_id": "550e8400-e29b-41d4-a716-446655440000",
    "created_at": "2025-06-12T10:30:00Z"
  }
}
```

Error Responses

- **400 Bad Request**: Validation failed for required fields or invalid format
- **401 Unauthorized**: Missing or invalid JWT token

---

2. Get All User Addresses

GET `/all`

Retrieves all addresses for the authenticated user.

curl Example

```
curl -X GET http://localhost:8080/api/v1/addresses/me/all \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

Success Response (200)

```
{
  "success": true,
  "message": "Addresses retrieved successfully",
  "data": [
    {
      "id": "123e4567-e89b-12d3-a456-426614174000",
      "address_line1": "123 Main Street, Apartment 4B",
      "city": "Mumbai",
      "state": "Maharashtra",
      "pincode": "400001",
      "country": "India",
      "mobile": "9876543210",
      "selected": true,
      "user_id": "550e8400-e29b-41d4-a716-446655440000",
      "created_at": "2025-06-12T10:30:00Z"
    }
  ]
}
```

Error Responses

- **401 Unauthorized**: Missing or invalid JWT token
- **404 Not Found**: No addresses found for user

Common Error Response Format

```
{
  "success": false,
  "message": "Error description",
  "error": "Detailed error information"
}
```

3. Get Single Address

GET `/{id}` Retrieves a specific address by ID for the authenticated user.

URL Parameters

Parameter	Type	Description
id	UUID	Address ID

Success Response (200)

```
{
  "success": true,
  "message": "Address retrieved successfully",
  "data": {
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "address_line1": "123 Main Street, Apartment 4B",
    "city": "Mumbai",
    "state": "Maharashtra",
    "pincode": "400001",
    "country": "India",
    "mobile": "9876543210",
    "selected": true,
    "user_id": "550e8400-e29b-41d4-a716-446655440000",
    "created_at": "2025-06-12T10:30:00Z"
  }
}
```

Error Responses

- **404 Not Found**: Address not found or doesn't belong to user

cURL Example

```
curl -X GET https://api.example.com/addresses/123e4567-e89b-12d3-a456-426614174000 \
  -H "Authorization: Bearer YOUR_TOKEN_HERE"
```

4. Update Address

PUT `/{id}` Updates an existing address for the authenticated user.

URL Parameters

Parameter	Type	Description
id	UUID	Address ID

Request Body

```
{
  "address_line1": "456 Updated Street",
  "city": "Delhi",
  "pincode": "110001"
}
```

Note: All fields are optional in update requests

Success Response (200)

```
{
  "success": true,
  "message": "Address updated successfully",
  "data": {
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "address_line1": "456 Updated Street",
    "city": "Delhi",
    "state": "Maharashtra",
    "pincode": "110001",
    "country": "India",
    "mobile": "9876543210",
    "selected": true,
    "user_id": "550e8400-e29b-41d4-a716-446655440000",
    "created_at": "2025-06-12T10:30:00Z"
  }
}
```

cURL Example

```
curl -X PUT https://api.example.com/addresses/123e4567-e89b-12d3-a456-426614174000 \
  -H "Authorization: Bearer YOUR_TOKEN_HERE" \
  -H "Content-Type: application/json" \
  -d '{
    "address_line1": "456 Updated Street",
    "city": "Delhi",
    "pincode": "110001"
  }'
```

---

5. Delete Address

DELETE `/{id}` Deletes an address for the authenticated user.

URL Parameters

Parameter	Type	Description
id	UUID	Address ID

Success Response (200)

```
{
  "success": true,
  "message": "Address deleted successfully"
}
```

Error Responses

- **404 Not Found**: Address not found or doesn't belong to user

cURL Example

```
curl -X DELETE https://api.example.com/addresses/123e4567-e89b-12d3-a456-426614174000 \
  -H "Authorization: Bearer YOUR_TOKEN_HERE"
```

6. Set Selected Address

PUT `/select/{address_id}` Sets an address as the selected/default address for the user.

URL Parameters

Parameter	Type	Description
address_id	UUID	Address ID to set as selected

Success Response (200)

```
{
  "success": true,
  "message": "Address selected successfully",
  "data": {
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "address_line1": "123 Main Street, Apartment 4B",
    "city": "Mumbai",
    "state": "Maharashtra",
    "pincode": "400001",
    "country": "India",
    "mobile": "9876543210",
    "selected": true,
    "user_id": "550e8400-e29b-41d4-a716-446655440000",
    "created_at": "2025-06-12T10:30:00Z"
  }
}
```

cURL Example

```
curl -X PUT https://api.example.com/addresses/select/123e4567-e89b-12d3-a456-426614174000 \
  -H "Authorization: Bearer YOUR_TOKEN_HERE"
```

7. Get Selected Address

GET `/selected` Retrieves the currently selected/default address for the authenticated user.

Success Response (200)

```
{
  "success": true,
  "message": "Selected address retrieved successfully",
  "data": {
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "address_line1": "123 Main Street, Apartment 4B",
    "city": "Mumbai",
    "state": "Maharashtra",
    "pincode": "400001",
    "country": "India",
    "mobile": "9876543210",
    "selected": true,
    "user_id": "550e8400-e29b-41d4-a716-446655440000",
    "created_at": "2025-06-12T10:30:00Z"
  }
}
```

Error Responses

- **404 Not Found**: No selected address found

cURL Example

```
curl -X GET https://api.example.com/addresses/selected \
  -H "Authorization: Bearer YOUR_TOKEN_HERE"
```

---

8. Search Addresses

GET `/search` Search addresses by city, state, or pincode for the authenticated user.

Query Parameters

Parameter	Type	Description
q	String	Search query (city, state, or pincode)
limit	Integer	Number of results to return (default: 10, max: 50)

Success Response (200)

```
{
  "success": true,
  "message": "Search completed successfully",
  "data": [
    {
      "id": "123e4567-e89b-12d3-a456-426614174000",
      "address_line1": "123 Main Street, Apartment 4B",
      "city": "Mumbai",
      "state": "Maharashtra",
      "pincode": "400001",
      "country": "India",
      "mobile": "9876543210",
      "selected": true,
      "user_id": "550e8400-e29b-41d4-a716-446655440000",
      "created_at": "2025-06-12T10:30:00Z"
    }
  ],
  "meta": {
    "total": 1,
    "query": "Mumbai"
  }
}
```

cURL Example

```
curl -X GET "https://api.example.com/addresses/search?q=Mumbai&limit=10" \
  -H "Authorization: Bearer YOUR_TOKEN_HERE"
```

---

9. Validate Address

POST `/validate` Validates an address without saving it to the database.

Request Body

```
{
  "address_line1": "123 Main Street, Apartment 4B",
  "city": "Mumbai",
  "state": "Maharashtra",
  "pincode": "400001",
  "country": "India",
  "mobile": "9876543210"
}
```

Success Response (200)

```
{
  "success": true,
  "message": "Address is valid",
  "data": {
    "valid": true,
    "formatted_address": "123 Main Street, Apartment 4B, Mumbai, Maharashtra 400001, India",
    "suggestions": []
  }
}
```

Error Response (400)

```
{
  "success": false,
  "message": "Address validation failed",
  "data": {
    "valid": false,
    "errors": [
      {
        "field": "pincode",
        "message": "Invalid pincode format"
      }
    ],
    "suggestions": [
      {
        "field": "pincode",
        "value": "400001"
      }
    ]
  }
}
```

cURL Example

```
curl -X POST https://api.example.com/addresses/validate \
  -H "Authorization: Bearer YOUR_TOKEN_HERE" \
  -H "Content-Type: application/json" \
  -d '{
    "address_line1": "123 Main Street, Apartment 4B",
    "city": "Mumbai",
    "state": "Maharashtra",
    "pincode": "400001",
    "country": "India",
    "mobile": "9876543210"
  }'
```

---

10. Bulk Create Addresses

POST `/bulk` Creates multiple addresses in a single request for the authenticated user.

Request Body

```
{
  "addresses": [
    {
      "address_line1": "123 Main Street, Apartment 4B",
      "city": "Mumbai",
      "state": "Maharashtra",
      "pincode": "400001",
      "country": "India",
      "mobile": "9876543210"
    },
    {
      "address_line1": "456 Another Street",
      "city": "Delhi",
      "state": "Delhi",
      "pincode": "110001",
      "country": "India",
      "mobile": "9876543211"
    }
  ]
}
```

Success Response (201)

```
{
  "success": true,
  "message": "Addresses created successfully",
  "data": {
    "created": [
      {
        "id": "123e4567-e89b-12d3-a456-426614174000",
        "address_line1": "123 Main Street, Apartment 4B",
        "city": "Mumbai",
        "state": "Maharashtra",
        "pincode": "400001",
        "country": "India",
        "mobile": "9876543210",
        "selected": false,
        "user_id": "550e8400-e29b-41d4-a716-446655440000",
        "created_at": "2025-06-12T10:30:00Z"
      }
    ],
    "failed": [
      {
        "index": 1,
        "errors": [
          {
            "field": "pincode",
            "message": "Invalid pincode format"
          }
        ]
      }
    ]
  }
}
```

cURL Example

```
curl -X POST https://api.example.com/addresses/bulk \
  -H "Authorization: Bearer YOUR_TOKEN_HERE" \
  -H "Content-Type: application/json" \
  -d '{
    "addresses": [
      {
        "address_line1": "123 Main Street, Apartment 4B",
        "city": "Mumbai",
        "state": "Maharashtra",
        "pincode": "400001",
        "country": "India",
        "mobile": "9876543210"
      },
      {
        "address_line1": "456 Another Street",
        "city": "Delhi",
        "state": "Delhi",
        "pincode": "110001",
        "country": "India",
        "mobile": "9876543211"
      }
    ]
  }'
```

Common Error Response Format

```
{
  "success": false,
  "message": "Error description",
  "error": "Detailed error information"
}
```

HTTP Status Codes

- **200**: Success
- **201**: Created successfully
- **400**: Bad Request (validation errors)
- **401**: Unauthorized (invalid/missing token)
- **403**: Forbidden (insufficient permissions)
- **404**: Not Found
- **422**: Unprocessable Entity (validation failed)
- **429**: Too Many Requests (rate limit exceeded)
- **500**: Internal Server Error