

# Thiết kế API

---

## 1. Quy chuẩn chung

- URL Base: `/api/v1`
- Authentication: Headers: `Authorization: Bearer <access_token>`
- Response Format (JSON):

```
{
  "message": "Mô tả kết quả (Thành công/Lỗi)",
  "data": {}
}
```

- Cụ thể:

```
{
  "message": "string",
  "data": "object | array | null"
}
```

## 2. Chi tiết các API

### Authentication & Authorization (PostgreSQL)

Chức năng: Đăng ký, Đăng nhập, Quản lý Token

#### Đăng ký

- Endpoint: `POST /auth/register`
- Request Body:

```
{
  "email": "user@example.com",
  "password": "securePassword123",
  "fullName": "Nguyen Van A",
  "phoneNum": "0909123456"
}
```

- Response:

```
{
  "message": "Đăng ký thành công",
  "data": {
```

```
{
  "userId": "uuid-gen-123",
  "email": "user@example.com"
}
```

## Đăng nhập

- Endpoint: **POST** /auth/login
- Request Body:

```
{
  "email": "user@example.com",
  "password": "securePassword123",
  "userAgent": "Mozilla/5.0..."
}
```

- Response:

```
{
  "message": "Đăng nhập thành công",
  "data": {
    "accessToken": "eyJ...",
    "refreshToken": "random-string-db-stored",
    "user": { "userId": "...", "role": "CUSTOMER" }
  }
}
```

## Refresh Token (rotation)

- Endpoint: **POST** /auth/refresh-token
- Request Body:

```
{
  "refreshToken": "random-string-db-stored"
}
```

- Response:

```
{
  "message": "Cấp lại token thành công",
  "data": {
    "accessToken": "eyJhbGciOiJIUz...", // Token mới (hạn ngắn: 15p-1h)
    "refreshToken": "new_refresh_token_xyz..." // Token mới (hạn dài: 7-30 ngày)
  }
}
```

```
}  
}
```

## Users & Profile (PostgreSQL)

Chức năng: Quản lý thông tin cá nhân, Sổ địa chỉ

### Lấy thông tin cá nhân

- Endpoint: **GET** /users/profile
- Auth: Required
- Response:

```
{  
  "message": "Lấy thông tin thành công",  
  "data": {  
    "userId": "...", "fullName": "...", "email": "...", "phoneNum": "..."  
  }  
}
```

### Cập nhật thông tin cá nhân

- Endpoint: **PUT** /users/profile
- Request Body:

```
{  
  "fullName": "Nguyen Van B",  
  "phoneNum": "0987654321",  
  "avatar": "url_moi" // Nếu có  
}
```

- Response:

```
{  
  "message": "Cập nhật hồ sơ thành công",  
  "data": { "userId": "...", "fullName": "Nguyen Van B" }  
}
```

### Quản lý địa chỉ giao hàng

- Endpoint: **GET** /users/addresses (Danh sách) | **POST** /users/addresses (Thêm mới)
- Request Body (POST):

```
{
  "addressLine": "123 Ly Thuong Kiet",
  "city": "HCM",
  "district": "District 10",
  "isDefault": true
}
```

- Response:

```
{
  "message": "Thêm địa chỉ thành công",
  "data": { "addressID": 1, "addressLine": "..." }
}
```

### Cập nhật một địa chỉ cụ thể

- Endpoint: **PUT** /users/addresses/:addressID
- Mục đích: Sửa lỗi chính tả hoặc thay đổi chi tiết của một địa chỉ đã lưu.
- Request Body:

```
{
  "addressLine": "So 1, Dai Co Viet",
  "city": "Ha Noi",
  "district": "Hai Ba Trung",
  "isDefault": false
}
```

- Response:

```
{
  "message": "Cập nhật địa chỉ thành công",
  "data": { "addressID": 10, "city": "Ha Noi", ... }
}
```

### Xóa địa chỉ

- Endpoint: **DELETE** /users/addresses/:addressID
- Mục đích: Xóa địa chỉ cũ không dùng nữa.
- Lưu ý: Backend nên kiểm tra xem địa chỉ này có đang gắn với đơn hàng nào đang xử lý (PENDING, PROCESSING) không trước khi cho xóa (hoặc chỉ soft-delete).
- Response:

```
{ "message": "Đã xóa địa chỉ" }
```

### Thiết lập địa chỉ mặc định

- Endpoint: `PATCH /users/addresses/:addressID/set-default`
- Mục đích: Chọn địa chỉ này làm địa chỉ giao hàng ưu tiên.
- Logic: Set `isDefault = true` cho ID này và false cho tất cả các địa chỉ còn lại của user đó.

### Products (MongoDB + Postgres Inventory)

Chức năng: Tìm kiếm nâng cao, Xem chi tiết, Quản lý kho

#### Tìm kiếm & Lọc sản phẩm (MongoDB)

Hỗ trợ tìm kiếm theo tên, category, và attributes động (RAM, Color...)

- Endpoint: `GET /products`
- Query Params: `?keyword=macbook&category=laptop&price_min=1000&attrs[ram]=16GB&page=1`
- Response:

```
{
  "message": "Tìm thấy 10 sản phẩm",
  "data": [
    {
      "_id": "mongo_object_id",
      "name": "MacBook Pro M3",
      "base_price": 2000,
      "category": "Electronics",
      "images": ["url1.jpg"],
      "attributes": { "ram": "16GB", "storage": "512GB" }
    }
  ]
}
```

#### Chi tiết sản phẩm (Hybrid: Mongo Info + Postgres Stock)

- Endpoint: `GET /products/:id`
- Response:

```
{
  "message": "Lấy chi tiết sản phẩm thành công",
  "data": {
    "_id": "mongo_object_id",
    "name": "MacBook Pro M3",
    "description": "...",
    "attributes": { ... },
    "inventory": [ // Dữ liệu từ Postgres bảng inventories

```

```
{ "sku": "M3-16-512", "stockQuantity": 50 },
{ "sku": "M3-32-1TB", "stockQuantity": 10 }
]
}
}
```

### Admin: Thêm/Sửa sản phẩm (MongoDB)

- Endpoint: **POST** /admin/products
- Auth: Required (Role: ADMIN)
- Request Body:

```
{
  "name": "Ao thun",
  "category": "Clothing",
  "base_price": 200,
  "attributes": { "size": "L", "material": "Cotton" }
}
```

- Response:

```
{
  "message": "Tạo sản phẩm thành công",
  "data": { "_id": "new_mongo_id", ... }
}
```

### Cart & Orders (PostgreSQL - Transaction)

Chức năng: Giỏ hàng, Tạo đơn, Xử lý kho, Lịch sử đơn hàng

#### Quản lý Giỏ hàng

Lưu ý: Có thể lưu ở Client hoặc DB. API này giả định lưu DB hoặc Session

- Endpoint: **POST** /cart/sync (Đồng bộ giỏ hàng local lên server)
- Request Body:

```
{
  "items": [
    { "productId": "mongo_id_1", "sku": "SKU_A", "quantity": 2 }
  ]
}
```

- Response:

```
{
  "message": "Đồng bộ giỏ hàng thành công",
  "data": { "cartTotal": 2, "items": [...] }
}
```

### Tạo đơn hàng (Checkout - ACID Transaction)

Quan trọng: Đây là nơi thực hiện Transaction trừ kho và tạo đơn.

- Endpoint: **POST /orders**
- Auth: Required
- Request Body:

```
{
  "shippingAddressId": 1,
  "paymentMethod": "COD",
  "items": [
    {
      "productId": "mongo_id_1", // Mapping ID Mongo
      "productName": "MacBook Pro", // Snapshot tên
      "sku": "M3-16-512",
      "quantity": 1,
      "unitPrice": 2000
    }
  ]
}
```

- Response:

```
{
  "message": "Đặt hàng thành công",
  "data": {
    "orderId": 1001,
    "totalAmount": 2000,
    "status": "PENDING"
  }
}
```

### Lịch sử đơn hàng

- Endpoint: **GET /orders**
- Response:

```
{
  "message": "Danh sách đơn hàng",

```

```
"data": [  
  { "orderId": 1001, "status": "PENDING", "totalAmount": 2000, "createdAt":  
    "..."}  
]
```

### Chi tiết đơn hàng

- Endpoint: `GET /orders/:orderId`
- Response:

```
{  
  "message": "Chi tiết đơn hàng",  
  "data": {  
    "orderId": 1001,  
    "items": [  
      { "productName": "MacBook Pro", "sku": "M3-16-512", "quantity": 1 }  
    ],  
    "payment": { "status": "PENDING", "method": "COD" }  
  }  
}
```

### Payments (PostgreSQL)

Chức năng: Xử lý thanh toán

#### Thực hiện thanh toán (Mockup)

- Endpoint: `POST /payments/process`
- Request Body:

```
{  
  "orderId": 1001,  
  "paymentMethod": "BANKING"  
}
```

- Response:

```
{  
  "message": "Thanh toán thành công",  
  "data": {  
    "paymentID": 505,  
    "status": "COMPLETED",  
    "transactionDate": "..."  
  }  
}
```



## Reviews (MongoDB)

Chức năng: Đánh giá sản phẩm

### Lấy danh sách Review

- Endpoint: **GET** `/products/:productId/reviews`
- Response:

```
{
  "message": "Danh sách đánh giá",
  "data": [
    {
      "_id": "review_id_1",
      "user_name": "Nguyen Van A", // Snapshot từ lúc review
      "rating": 5,
      "comment": "Máy chạy nhanh",
      "images": ["url_img.jpg"]
    }
  ]
}
```

### Gửi đánh giá mới

Logic: Check backend xem user đã mua hàng chưa (optional) rồi lưu vào Mongo

- Endpoint: **POST** `/products/:productId/reviews`
- Auth: Required
- Request Body:

```
{
  "rating": 5,
  "comment": "Sản phẩm tốt",
  "images": []
}
```

- Response:

```
{
  "message": "Gửi đánh giá thành công",
  "data": { "_id": "new_review_id" }
}
```

## System Logging (MongoDB)

Chức năng: Log hành vi người dùng

### Ghi Log (User Tracking)

- Endpoint: **POST /logs**
- Request Body:

```
{
  "action_type": "VIEW_PRODUCT", // hoặc SEARCH, ADD_TO_CART
  "target_id": "mongo_product_id",
  "metadata": {
    "search_keyword": "Laptop gaming",
    "device": "Mobile"
  }
}
```

- Response

```
{
  "message": "Đã ghi log",
  "data": null
}
```

### Statistics (Postgres Aggregation)

Chức năng: Thống kê doanh thu

- Endpoint: **GET /admin/stats/revenue**
- Auth: Required (Role: ADMIN)
- Query Params: **?startDate=2026-01-01&endDate=2026-02-01&type=day** (theo ngày/tháng)
- Response:

```
{
  "message": "Thống kê doanh thu",
  "data": [
    { "date": "2026-01-01", "totalRevenue": 50000000, "orderCount": 20 },
    { "date": "2026-01-02", "totalRevenue": 30000000, "orderCount": 15 }
  ]
}
```