**TÀI LIỆU THIẾT KẾ API**

**HỆ THỐNG QUẢN LÝ BÁN HÀNG (POINT OF SALE - POS)**

Phiên bản: 1.0

Ngày: 22/05/2025

Tác giả: [Cao Đăng Khoa]

Công nghệ Backend: Java (Spring Boot)

Công nghệ Frontend: Vue.js

**1. Giới thiệu**

Tài liệu này mô tả thiết kế các API RESTful cho hệ thống Quản lý Bán hàng (POS). Các API này sẽ được phát triển bằng Java Spring Boot và sẽ là cầu nối cho giao diện Frontend được xây dựng bằng Vue.js để tương tác với cơ sở dữ liệu và logic nghiệp vụ. Mục tiêu là cung cấp các API nhất quán, dễ hiểu, bảo mật và hiệu quả.

Cấu trúc chung của API:

* Base URL: https://api.yourdomain.com/api/v1 (hoặc http://localhost:8080/api/v1 trong môi trường phát triển)
* Định dạng dữ liệu: Tất cả các yêu cầu và phản hồi sẽ sử dụng định dạng JSON.
* Xác thực: Sử dụng JWT (JSON Web Token) cho việc xác thực. Token sẽ được gửi trong header Authorization dưới dạng Bearer <token>.
* Phân quyền: Backend sẽ thực hiện kiểm tra quyền hạn (RBAC) dựa trên vai trò của người dùng trong JWT token cho mỗi yêu cầu API.
* Mã trạng thái HTTP: Sẽ được sử dụng để chỉ thị kết quả của yêu cầu (ví dụ: 200 OK, 201 Created, 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 Internal Server Error).

**2. Cấu trúc phản hồi chung (Common Response Structure)**

Để đảm bảo tính nhất quán, các phản hồi API sẽ tuân theo một cấu trúc chung, đặc biệt là đối với các phản hồi thành công và lỗi.

**2.1. Phản hồi thành công (Success Response)**

{

"status": "success",

"message": "Operation successful",

"data": {

// Dữ liệu trả về (object, array, null)

},

"metadata": {

// Thông tin phân trang, tổng số bản ghi (nếu có)

}

}

* status: Luôn là "success".
* message: Thông báo mô tả kết quả của thao tác.
* data: Payload chính chứa dữ liệu được yêu cầu. Có thể là một đối tượng, một mảng đối tượng, hoặc null cho các thao tác không trả về dữ liệu cụ thể (ví dụ: DELETE).
* metadata: (Tùy chọn) Chứa thông tin bổ sung như phân trang (total, page, limit), số lượng kết quả, v.v.

**2.2. Phản hồi lỗi (Error Response)**

{

"status": "error",

"message": "Error description",

"code": "ERROR\_CODE", // Mã lỗi cụ thể của ứng dụng

"errors": [

// Danh sách lỗi chi tiết (ví dụ: lỗi validation)

{

"field": "fieldName",

"message": "Error message for field"

}

]

}

* status: Luôn là "error".
* message: Mô tả tổng quát về lỗi.
* code: Mã lỗi cụ thể (ví dụ: INVALID\_INPUT, RESOURCE\_NOT\_FOUND, UNAUTHORIZED\_ACCESS).
* errors: (Tùy chọn) Mảng các lỗi chi tiết, thường dùng cho lỗi validation, chỉ rõ trường nào bị lỗi và thông báo lỗi tương ứng.

**3. Các API Endpoint**

Dưới đây là danh sách các API endpoint chính, được nhóm theo từng module chức năng.

**3.1. Xác thực & Quyền hạn (Authentication & Authorization)**

* Đăng nhập người dùng
  + Endpoint: POST /auth/login
  + Mô tả: Xác thực người dùng bằng tên đăng nhập và mật khẩu. Trả về JWT token và thông tin người dùng.
  + Yêu cầu (Request Body):

{

"username": "employee\_username",

"password": "employee\_password"

}

* + Phản hồi thành công (200 OK):

{

"status": "success",

"message": "Login successful",

"data": {

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",

"user": {

"id": 1,

"username": "admin",

"name": "Admin User",

"role": {

"id": 1,

"name": "Admin"

}

}

}

}

* + Phản hồi lỗi (401 Unauthorized): Tên đăng nhập hoặc mật khẩu không đúng.
* Đăng xuất người dùng
  + Endpoint: POST /auth/logout
  + Mô tả: Vô hiệu hóa JWT token hiện tại (thêm vào blacklist nếu cần).
  + Yêu cầu (Headers): Authorization: Bearer <token>
  + Phản hồi thành công (200 OK):

{

"status": "success",

"message": "Logged out successfully"

}

* Lấy thông tin người dùng hiện tại
  + Endpoint: GET /auth/me
  + Mô tả: Trả về thông tin của người dùng đang đăng nhập dựa trên JWT token.
  + Yêu cầu (Headers): Authorization: Bearer <token>
  + Phản hồi thành công (200 OK): (Tương tự user object trong login response)
  + Phản hồi lỗi (401 Unauthorized): Token không hợp lệ hoặc hết hạn.

**3.2. Quản lý Sản phẩm (Product Management)**

* Lấy danh sách sản phẩm
  + Endpoint: GET /products
  + Mô tả: Trả về danh sách các sản phẩm.
  + Tham số truy vấn (Query Params):
    - page: Số trang (mặc định 0)
    - size: Số lượng mục trên mỗi trang (mặc định 10)
    - search: Tìm kiếm theo tên sản phẩm hoặc barcode
    - category\_id: Lọc theo ID danh mục
    - is\_active: Lọc theo trạng thái hoạt động (true/false)
  + Quyền yêu cầu: Cashier, Inventory Manager, Admin
  + Phản hồi thành công (200 OK):

{

"status": "success",

"message": "Products fetched successfully",

"data": [

{ "id": 1, "name": "Cà phê đen", "barcode": "SP001", "price": 25000.00, "quantity": 100, "category\_id": 1, "is\_active": true, ... },

{ "id": 2, "name": "Trà sữa trân châu", "barcode": "SP002", "price": 35000.00, "quantity": 80, "category\_id": 2, "is\_active": true, ... }

],

"metadata": { "total": 200, "page": 0, "size": 10 }

}

* Lấy chi tiết sản phẩm theo ID
  + Endpoint: GET /products/{id}
  + Mô tả: Trả về thông tin chi tiết của một sản phẩm cụ thể.
  + Quyền yêu cầu: Cashier, Inventory Manager, Admin
  + Phản hồi thành công (200 OK): (Object Product)
  + Phản hồi lỗi (404 Not Found): Sản phẩm không tồn tại.
* Thêm sản phẩm mới
  + Endpoint: POST /products
  + Mô tả: Thêm một sản phẩm mới vào hệ thống.
  + Quyền yêu cầu: Inventory Manager, Admin
  + Yêu cầu (Request Body):

{

"barcode": "SP003",

"name": "Nước ép cam",

"price": 40000.00,

"quantity": 50,

"image\_path": "/images/orange\_juice.jpg",

"category\_id": 3

}

* + Phản hồi thành công (201 Created): (Object Product đã tạo)
  + Phản hồi lỗi (400 Bad Request): Dữ liệu không hợp lệ (ví dụ: trùng barcode, thiếu trường).
* Cập nhật thông tin sản phẩm
  + Endpoint: PUT /products/{id}
  + Mô tả: Cập nhật thông tin của một sản phẩm hiện có.
  + Quyền yêu cầu: Inventory Manager, Admin
  + Yêu cầu (Request Body): (Object Product với các trường cần cập nhật, bao gồm version để optimistic locking)

{

"name": "Nước ép cam tươi",

"price": 42000.00,

"quantity": 48,

"version": 0

}

* + Phản hồi thành công (200 OK): (Object Product đã cập nhật)
  + Phản hồi lỗi (400 Bad Request): Dữ liệu không hợp lệ, hoặc version không khớp (Optimistic Locking).
  + Phản hồi lỗi (404 Not Found): Sản phẩm không tồn tại.
* Cập nhật trạng thái hoạt động của sản phẩm
  + Endpoint: PATCH /products/{id}/status
  + Mô tả: Kích hoạt hoặc vô hiệu hóa sản phẩm.
  + Quyền yêu cầu: Inventory Manager, Admin
  + Yêu cầu (Request Body):

{

"is\_active": false

}

* + Phản hồi thành công (200 OK): (Object Product đã cập nhật trạng thái)
* Xóa sản phẩm
  + Endpoint: DELETE /products/{id}
  + Mô tả: Xóa một sản phẩm khỏi hệ thống.
  + Quyền yêu cầu: Admin
  + Phản hồi thành công (204 No Content): (Không có nội dung trả về)
  + Phản hồi lỗi (404 Not Found): Sản phẩm không tồn tại.
  + Phản hồi lỗi (409 Conflict): Không thể xóa vì có ràng buộc với các hóa đơn hoặc lịch sử kho.

**3.3. Quản lý Danh mục (Category Management)**

* Lấy danh sách danh mục
  + Endpoint: GET /categories
  + Mô tả: Trả về danh sách các danh mục sản phẩm.
  + Quyền yêu cầu: Cashier, Inventory Manager, Admin
  + Phản hồi thành công (200 OK):

{

"status": "success",

"data": [

{ "id": 1, "name": "Đồ uống", "description": "Các loại đồ uống", ... },

{ "id": 2, "name": "Đồ ăn vặt", "description": "Các món ăn nhẹ", ... }

]

}

* Thêm danh mục mới
  + Endpoint: POST /categories
  + Mô tả: Thêm một danh mục sản phẩm mới.
  + Quyền yêu cầu: Admin
  + Yêu cầu (Request Body):

{

"name": "Tráng miệng",

"description": "Các món tráng miệng ngọt"

}

* + Phản hồi thành công (201 Created): (Object Category đã tạo)
* Cập nhật danh mục
  + Endpoint: PUT /categories/{id}
  + Mô tả: Cập nhật thông tin của một danh mục.
  + Quyền yêu cầu: Admin
  + Yêu cầu (Request Body): (Object Category với các trường cần cập nhật, bao gồm version)
  + Phản hồi thành công (200 OK): (Object Category đã cập nhật)
* Xóa danh mục
  + Endpoint: DELETE /categories/{id}
  + Mô tả: Xóa một danh mục.
  + Quyền yêu cầu: Admin
  + Phản hồi thành công (204 No Content):
  + Phản hồi lỗi (409 Conflict): Không thể xóa nếu có sản phẩm thuộc danh mục này.

**3.4. Quản lý Bán hàng (Sales Management - Invoices & Invoice Items)**

* Tạo hóa đơn mới
  + Endpoint: POST /invoices
  + Mô tả: Khởi tạo một hóa đơn mới.
  + Quyền yêu cầu: Cashier, Admin
  + Yêu cầu (Request Body):

{

"customer\_id": 1, // Tùy chọn, có thể null

"table\_id": 1 // Tùy chọn, có thể null

}

* + Phản hồi thành công (201 Created):

{

"status": "success",

"message": "Invoice created successfully",

"data": {

"id": 101,

"invoice\_number": "INV-20250522-0001",

"status": "pending",

"employee\_id": 5, // ID của nhân viên tạo (lấy từ JWT)

"sub\_total": 0, "discount\_amount": 0, "tax\_amount": 0, "total\_amount": 0,

"created\_at": "...", "updated\_at": "...", "version": 0

}

}

* Lấy chi tiết hóa đơn
  + Endpoint: GET /invoices/{id}
  + Mô tả: Trả về thông tin chi tiết của một hóa đơn, bao gồm các invoice\_items và payments.
  + Quyền yêu cầu: Cashier, Admin
  + Phản hồi thành công (200 OK):

Thêm sản phẩm vào hóa đơn

{

"status": "success",

"message": "Invoice fetched successfully",

"data": {

"id": 101,

"invoice\_number": "INV-20250522-0001",

// ... các trường của hóa đơn

"customer": { "id": 1, "name": "Nguyễn Văn A" }, // Nếu có customer\_id

"employee": { "id": 5, "name": "Nhân viên B" }, // Nếu có employee\_id

"table": { "id": 1, "name": "Bàn 1" }, // Nếu có table\_id

"discount": { "id": 1, "code": "SALE10", ... }, // Nếu có discount\_id

"items": [

{ "id": 1, "product\_id": 1, "product\_name": "Cà phê đen", "quantity": 1, "unit\_price": 25000.00, "item\_total": 25000.00 },

{ "id": 2, "product\_id": 2, "product\_name": "Trà sữa", "quantity": 1, "unit\_price": 35000.00, "item\_total": 35000.00 }

],

"payments": [

{ "id": 1, "method": "cash", "amount": 60000.00 }

]

}

}

* + Endpoint: POST /invoices/{invoiceId}/items
  + Mô tả: Thêm một sản phẩm vào hóa đơn hiện có. Backend sẽ tự động cập nhật tổng tiền hóa đơn.
  + Quyền yêu cầu: Cashier, Admin
  + Yêu cầu (Request Body):

{

"product\_id": 1,

"quantity": 2

}

* + Phản hồi thành công (200 OK): (Object InvoiceItem đã thêm, kèm theo thông tin invoice được cập nhật)
  + Phản hồi lỗi (400 Bad Request): Số lượng không đủ, hoặc dữ liệu không hợp lệ.
  + Phản hồi lỗi (404 Not Found): Hóa đơn hoặc sản phẩm không tồn tại.
* Cập nhật số lượng sản phẩm trong hóa đơn
  + Endpoint: PUT /invoices/{invoiceId}/items/{itemId}
  + Mô tả: Cập nhật số lượng của một sản phẩm đã có trong hóa đơn. Backend sẽ tự động cập nhật tổng tiền.
  + Quyền yêu cầu: Cashier, Admin
  + Yêu cầu (Request Body):

JSON

{

"quantity": 3,

"version": 0 // của invoice\_item

}

* + Phản hồi thành công (200 OK): (Object InvoiceItem đã cập nhật, kèm theo thông tin invoice được cập nhật)
* Xóa sản phẩm khỏi hóa đơn
  + Endpoint: DELETE /invoices/{invoiceId}/items/{itemId}
  + Mô tả: Xóa một sản phẩm khỏi hóa đơn. Backend sẽ tự động cập nhật tổng tiền.
  + Quyền yêu cầu: Cashier, Admin
  + Phản hồi thành công (204 No Content):
* Áp dụng/Xóa khuyến mãi cho hóa đơn
  + Endpoint: PATCH /invoices/{id}/discount
  + Mô tả: Áp dụng một mã khuyến mãi vào hóa đơn hoặc xóa khuyến mãi đã áp dụng.
  + Quyền yêu cầu: Cashier, Admin
  + Yêu cầu (Request Body - Áp dụng):

JSON

{

"discount\_code": "SALE10"

}

* + Yêu cầu (Request Body - Xóa):

JSON

{

"discount\_code": null

}

* + Phản hồi thành công (200 OK): (Object Invoice đã cập nhật)
  + Phản hồi lỗi (400 Bad Request): Mã khuyến mãi không hợp lệ, hết hạn, hoặc không đủ điều kiện.
* Thanh toán hóa đơn
  + Endpoint: POST /invoices/{id}/pay
  + Mô tả: Xử lý thanh toán cho hóa đơn và cập nhật trạng thái hóa đơn thành 'completed'.
  + Quyền yêu cầu: Cashier, Admin
  + Yêu cầu (Request Body):

JSON

{

"method": "cash",

"amount\_paid": 60000.00 // Số tiền khách hàng trả

}

* + Phản hồi thành công (200 OK):

JSON

{

"status": "success",

"message": "Invoice paid successfully",

"data": {

"invoice\_id": 101,

"total\_amount": 55000.00,

"amount\_paid": 60000.00,

"change\_due": 5000.00,

"payment\_details": { "id": 1, "method": "cash", "amount": 60000.00 }

}

}

* + Phản hồi lỗi (400 Bad Request): Số tiền thanh toán không đủ, hoặc hóa đơn đã được thanh toán.
* Hủy hóa đơn
  + Endpoint: PATCH /invoices/{id}/cancel
  + Mô tả: Hủy một hóa đơn.
  + Quyền yêu cầu: Admin
  + Phản hồi thành công (200 OK): (Object Invoice với status 'canceled')
* Lấy danh sách hóa đơn (tìm kiếm, lọc, phân trang)
  + Endpoint: GET /invoices
  + Mô tả: Lấy danh sách hóa đơn với các tùy chọn lọc.
  + Tham số truy vấn: page, size, status, start\_date, end\_date, employee\_id, customer\_id, invoice\_number.
  + Quyền yêu cầu: Cashier, Admin
  + Phản hồi thành công (200 OK): (List of Invoices)

**3.5. Quản lý Khách hàng (Customer Management)**

* Lấy danh sách khách hàng
  + Endpoint: GET /customers
  + Mô tả: Trả về danh sách khách hàng.
  + Tham số truy vấn: page, size, search (tên, số điện thoại, email).
  + Quyền yêu cầu: Cashier, Admin
* Lấy chi tiết khách hàng theo ID
  + Endpoint: GET /customers/{id}
  + Mô tả: Trả về thông tin chi tiết của một khách hàng.
  + Quyền yêu cầu: Cashier, Admin
* Thêm khách hàng mới
  + Endpoint: POST /customers
  + Mô tả: Thêm một khách hàng mới.
  + Quyền yêu cầu: Cashier, Admin
  + Yêu cầu (Request Body):

JSON

{

"name": "Nguyễn Thị B",

"phone": "0987654321",

"email": "b.nguyen@example.com",

"address": "123 Đường ABC"

}

* + Phản hồi thành công (201 Created): (Object Customer đã tạo)
* Cập nhật thông tin khách hàng
  + Endpoint: PUT /customers/{id}
  + Mô tả: Cập nhật thông tin của một khách hàng.
  + Quyền yêu cầu: Cashier, Admin
  + Yêu cầu (Request Body): (Object Customer với các trường cần cập nhật, bao gồm version)
  + Phản hồi thành công (200 OK): (Object Customer đã cập nhật)
* Lấy lịch sử mua hàng của khách hàng
  + Endpoint: GET /customers/{id}/invoices
  + Mô tả: Trả về danh sách các hóa đơn của một khách hàng cụ thể.
  + Quyền yêu cầu: Cashier, Admin
  + Tham số truy vấn: page, size, status.
  + Phản hồi thành công (200 OK): (List of Invoices)

**3.6. Quản lý Kho (Inventory Management)**

* Lấy lịch sử nhập kho
  + Endpoint: GET /stock-in
  + Mô tả: Trả về danh sách các giao dịch nhập kho.
  + Tham số truy vấn: page, size, product\_id, employee\_id, start\_date, end\_date.
  + Quyền yêu cầu: Inventory Manager, Admin
* Thêm giao dịch nhập kho
  + Endpoint: POST /stock-in
  + Mô tả: Ghi nhận một giao dịch nhập sản phẩm vào kho.
  + Quyền yêu cầu: Inventory Manager, Admin
  + Yêu cầu (Request Body):

JSON

{

"product\_id": 1,

"quantity": 50,

"note": "Nhập hàng từ nhà cung cấp XYZ"

}

* + Phản hồi thành công (201 Created): (Object StockIn đã tạo, kèm theo thông tin product được cập nhật quantity)
* Lấy lịch sử xuất kho
  + Endpoint: GET /stock-out
  + Mô tả: Trả về danh sách các giao dịch xuất kho.
  + Tham số truy vấn: page, size, product\_id, employee\_id, start\_date, end\_date.
  + Quyền yêu cầu: Inventory Manager, Admin
* Thêm giao dịch xuất kho
  + Endpoint: POST /stock-out
  + Mô tả: Ghi nhận một giao dịch xuất sản phẩm khỏi kho.
  + Quyền yêu cầu: Inventory Manager, Admin
  + Yêu cầu (Request Body):

JSON

{

"product\_id": 1,

"quantity": 5,

"note": "Xuất cho chi nhánh A"

}

* + Phản hồi thành công (201 Created): (Object StockOut đã tạo, kèm theo thông tin product được cập nhật quantity)
  + Phản hồi lỗi (400 Bad Request): Số lượng xuất lớn hơn tồn kho hiện có.

**3.7. Quản lý Nhân viên & Vai trò (Employee & Role Management)**

* Lấy danh sách nhân viên
  + Endpoint: GET /employees
  + Mô tả: Trả về danh sách nhân viên.
  + Tham số truy vấn: page, size, search (tên, username), role\_id, is\_active.
  + Quyền yêu cầu: Admin
* Lấy chi tiết nhân viên theo ID
  + Endpoint: GET /employees/{id}
  + Mô tả: Trả về thông tin chi tiết của một nhân viên.
  + Quyền yêu cầu: Admin (hoặc Cashier cho me endpoint)
* Thêm nhân viên mới
  + Endpoint: POST /employees
  + Mô tả: Thêm một nhân viên mới vào hệ thống.
  + Quyền yêu cầu: Admin
  + Yêu cầu (Request Body):

JSON

{

"name": "Nhân viên C",

"username": "nvien.c",

"password": "secure\_password",

"role\_id": 2, // ID của vai trò Cashier

"phone": "0123456789",

"email": "nv.c@example.com",

"is\_active": true

}

* + Phản hồi thành công (201 Created): (Object Employee đã tạo, password\_hash không được trả về)
* Cập nhật thông tin nhân viên
  + Endpoint: PUT /employees/{id}
  + Mô tả: Cập nhật thông tin của một nhân viên.
  + Quyền yêu cầu: Admin
  + Yêu cầu (Request Body): (Object Employee với các trường cần cập nhật, bao gồm version)
  + Phản hồi thành công (200 OK): (Object Employee đã cập nhật)
* Thay đổi trạng thái hoạt động của nhân viên
  + Endpoint: PATCH /employees/{id}/status
  + Mô tả: Kích hoạt hoặc vô hiệu hóa tài khoản nhân viên.
  + Quyền yêu cầu: Admin
  + Yêu cầu (Request Body): {"is\_active": false}
  + Phản hồi thành công (200 OK):
* Đặt lại mật khẩu nhân viên
  + Endpoint: PUT /employees/{id}/password-reset
  + Mô tả: Admin đặt lại mật khẩu cho nhân viên.
  + Quyền yêu cầu: Admin
  + Yêu cầu (Request Body): {"new\_password": "new\_secure\_password"}
  + Phản hồi thành công (200 OK):
* Lấy danh sách vai trò
  + Endpoint: GET /roles
  + Mô tả: Trả về danh sách các vai trò có sẵn.
  + Quyền yêu cầu: Admin
* Thêm vai trò mới
  + Endpoint: POST /roles
  + Mô tả: Thêm một vai trò mới.
  + Quyền yêu cầu: Admin
  + Yêu cầu (Request Body): {"name": "Supervisor"}
  + Phản hồi thành công (201 Created):
* Cập nhật vai trò
  + Endpoint: PUT /roles/{id}
  + Mô tả: Cập nhật tên vai trò.
  + Quyền yêu cầu: Admin
  + Yêu cầu (Request Body): {"name": "Manager", "version": 0}
  + Phản hồi thành công (200 OK):
* Xóa vai trò
  + Endpoint: DELETE /roles/{id}
  + Mô tả: Xóa một vai trò.
  + Quyền yêu cầu: Admin
  + Phản hồi thành công (204 No Content):
  + Phản hồi lỗi (409 Conflict): Không thể xóa nếu có nhân viên đang sử dụng vai trò này.

**3.8. Quản lý Bàn (Table Management - If Applicable)**

* Lấy danh sách bàn
  + Endpoint: GET /tables
  + Mô tả: Trả về danh sách các bàn.
  + Tham số truy vấn: status (available, occupied, ...).
  + Quyền yêu cầu: Cashier, Admin
* Lấy chi tiết bàn theo ID
  + Endpoint: GET /tables/{id}
  + Mô tả: Trả về thông tin chi tiết của một bàn.
  + Quyền yêu cầu: Cashier, Admin
* Cập nhật trạng thái bàn
  + Endpoint: PATCH /tables/{id}/status
  + Mô tả: Cập nhật trạng thái của một bàn (ví dụ: 'cleaning', 'available').
  + Quyền yêu cầu: Cashier, Admin
  + Yêu cầu (Request Body): {"status": "cleaning"}
  + Phản hồi thành công (200 OK): (Object Table đã cập nhật)

**3.9. Quản lý Khuyến mãi (Discount Management)**

* Lấy danh sách khuyến mãi
  + Endpoint: GET /discounts
  + Mô tả: Trả về danh sách các chương trình khuyến mãi.
  + Tham số truy vấn: page, size, is\_active, code\_search.
  + Quyền yêu cầu: Cashier, Admin
* Lấy chi tiết khuyến mãi theo ID
  + Endpoint: GET /discounts/{id}
  + Mô tả: Trả về thông tin chi tiết của một khuyến mãi.
  + Quyền yêu cầu: Cashier, Admin
* Thêm khuyến mãi mới
  + Endpoint: POST /discounts
  + Mô tả: Thêm một chương trình khuyến mãi mới.
  + Quyền yêu cầu: Admin
  + Yêu cầu (Request Body):

JSON

{

"code": "SALE20",

"description": "Giảm 20% cho đơn hàng trên 500k",

"discount\_type\_id": 1, // ID của 'percent'

"value": 20.00,

"start\_date": "2025-05-22T00:00:00Z",

"end\_date": "2025-06-30T23:59:59Z",

"minimum\_order\_amount": 500000.00,

"maximum\_discount\_amount": 100000.00,

"is\_active": true

}

* + Phản hồi thành công (201 Created): (Object Discount đã tạo)
* Cập nhật thông tin khuyến mãi
  + Endpoint: PUT /discounts/{id}
  + Mô tả: Cập nhật thông tin của một khuyến mãi.
  + Quyền yêu cầu: Admin
  + Yêu cầu (Request Body): (Object Discount với các trường cần cập nhật, bao gồm version)
  + Phản hồi thành công (200 OK): (Object Discount đã cập nhật)
* Thay đổi trạng thái hoạt động của khuyến mãi
  + Endpoint: PATCH /discounts/{id}/status
  + Mô tả: Kích hoạt hoặc vô hiệu hóa khuyến mãi.
  + Quyền yêu cầu: Admin
  + Yêu cầu (Request Body): {"is\_active": false}
  + Phản hồi thành công (200 OK):

**3.10. Báo cáo & Thống kê (Reporting & Analytics)**

* Báo cáo Doanh thu
  + Endpoint: GET /reports/sales-summary
  + Mô tả: Trả về tổng quan doanh thu theo khoảng thời gian.
  + Tham số truy vấn: start\_date, end\_date, group\_by (day, month, year, payment\_method).
  + Quyền yêu cầu: Admin
  + Phản hồi thành công (200 OK):

JSON

{

"status": "success",

"data": {

"total\_revenue": 150000000.00,

"total\_discount\_applied": 15000000.00,

"total\_tax\_collected": 1500000.00,

"net\_revenue": 136500000.00,

"details\_by\_period": [

{ "date": "2025-05-20", "revenue": 5000000.00 },

{ "date": "2025-05-21", "revenue": 5500000.00 }

],

"details\_by\_payment\_method": [

{"method": "cash", "revenue": 80000000.00},

{"method": "card", "revenue": 60000000.00}

]

}

}

* Báo cáo sản phẩm bán chạy
  + Endpoint: GET /reports/top-selling-products
  + Mô tả: Trả về danh sách các sản phẩm bán chạy nhất.
  + Tham số truy vấn: start\_date, end\_date, limit (số lượng sản phẩm top), order\_by (quantity, revenue).
  + Quyền yêu cầu: Admin
  + Phản hồi thành công (200 OK):

JSON

{

"status": "success",

"data": [

{"product\_id": 1, "product\_name": "Cà phê đen", "total\_quantity\_sold": 500, "total\_revenue": 12500000.00},

{"product\_id": 2, "product\_name": "Trà sữa", "total\_quantity\_sold": 450, "total\_revenue": 15750000.00}

]

}

* Báo cáo tồn kho
  + Endpoint: GET /reports/inventory-summary
  + Mô tả: Trả về tổng quan về tồn kho hiện tại.
  + Tham số truy vấn: category\_id, min\_quantity, max\_quantity, is\_active.
  + Quyền yêu cầu: Inventory Manager, Admin
  + Phản hồi thành công (200 OK):

JSON

{

"status": "success",

"data": {

"total\_products": 100,

"total\_stock\_value": 50000000.00,

"low\_stock\_products": [

{"product\_id": 5, "product\_name": "Sữa chua", "current\_quantity": 5, "min\_threshold": 10}

],

"all\_products\_inventory": [

{"product\_id": 1, "product\_name": "Cà phê đen", "current\_quantity": 100},

// ...

]

}

}

**3.11. Cấu hình Hệ thống (System Configuration)**

* Lấy tất cả cài đặt hệ thống
  + Endpoint: GET /settings
  + Mô tả: Trả về tất cả các cặp key-value của cài đặt hệ thống.
  + Quyền yêu cầu: Admin
  + Phản hồi thành công (200 OK):

JSON

{

"status": "success",

"data": {

"store\_name": "Your Awesome Cafe",

"tax\_rate": "0.10",

"currency\_symbol": "VND",

"invoice\_prefix": "INV-"

}

}

* Cập nhật cài đặt hệ thống
  + Endpoint: PUT /settings
  + Mô tả: Cập nhật một hoặc nhiều cài đặt hệ thống.
  + Quyền yêu cầu: Admin
  + Yêu cầu (Request Body):

JSON

{

"store\_name": "New Awesome Cafe",

"tax\_rate": "0.08"

}

* + Phản hồi thành công (200 OK): (Object chứa các cài đặt đã cập nhật)

**4. Các cân nhắc quan trọng**

* Xử lý ngày/giờ: Tất cả các trường ngày/giờ sẽ được trao đổi dưới dạng chuỗi ISO 8601 (ví dụ: 2025-05-22T10:30:00Z). Backend sẽ lưu trữ dưới dạng UTC và Frontend sẽ chuyển đổi sang múi giờ địa phương để hiển thị.
* Phân trang (Pagination): Hầu hết các API trả về danh sách đều hỗ trợ phân trang thông qua các tham số page, size và trả về metadata chứa total, page, size.
* Lọc & Tìm kiếm: Cung cấp các tham số truy vấn linh hoạt để lọc và tìm kiếm dữ liệu.
* Xử lý lỗi chi tiết: Đảm bảo các phản hồi lỗi cung cấp đủ thông tin cho Frontend để hiển thị thông báo phù hợp cho người dùng hoặc để debug.
* Validation: Backend sẽ thực hiện validation mạnh mẽ cho tất cả dữ liệu đầu vào.
* Optimistic Locking: Đối với các thao tác cập nhật (PUT), hãy đảm bảo gửi trường version trong request body. Backend sẽ kiểm tra và trả về lỗi nếu version không khớp để ngăn ngừa xung đột.
* Rate Limiting (Tùy chọn): Để bảo vệ API khỏi các cuộc tấn công DDoS hoặc lạm dụng, có thể xem xét áp dụng rate limiting cho một số endpoint nhất định.
* API Documentation (Swagger/OpenAPI): Khi triển khai thực tế, nên sử dụng các công cụ như Swagger/OpenAPI để tự động tạo tài liệu API tương tác từ mã nguồn Spring Boot, giúp quá trình phát triển Frontend dễ dàng hơn rất nhiều.