System Design & API Contract: E-Commerce Platform v1.1

1. Architectural Overview & Global Conventions

Our platform is a **microservice architecture**. Each service is an independent application that communicates via APIs. The system is supported by the following core infrastructure:

- **API Gateway:** The single entry point for all client requests. It routes traffic to the appropriate service.
- **Eureka Service Registry:** Allows services to discover each other dynamically by name.
- **Config Server:** Provides centralized configuration for all services.

The following conventions apply to **ALL** services without exception:

- **Base URL:** All public APIs are accessed through the Gateway at /api/v1.
- **Service Discovery:** Each service registers with Eureka using its spring.application.name in uppercase (e.g., PRODUCT-SERVICE).
- **Data Format:** All request and response bodies **must** be application/json.
- **Authentication:** Protected endpoints require a JWT in the Authorization: Bearer <token> header, which is validated by the API Gateway.
- **ID Format:** All public-facing resource IDs (productId, orderId, etc.) **must** be **UUIDs**.
- **Timestamps:** All timestamps (createdAt, updatedAt) **must** be in **ISO 8601** format and in **UTC**.
- **Standard Error Response:**

```
{
  "error": {
    "type": "Validation Error",
    "message": "One or more fields are invalid.",
    "details": [
      { "field": "email", "issue": "Must be a valid email address." }
    ]
  }
}
```

2. Account Service

- **Responsibility:** Manages user identity, authentication, and profile information.
- **Service Discovery ID:** ACCOUNT-SERVICE
- **Gateway Route Prefix:** /api/v1/accounts

Data Models

- **User:**
  {
  "id": "uuid",
  "firstName": "string",
  "lastName": "string",
  "email": "string",
  "defaultShippingAddress": "object (Address Model)",
  "createdAt": "string (ISO 8601)"
  }

- **Address:**
  {
  "street": "string",
  "city": "string",
  "postalCode": "string",
  "country": "string"
  }

API Endpoints

- **POST /api/v1/accounts/register**
  - **Description:** Creates a new user account.
  - **Auth:** Public.
  - **Request Body:** { "firstName": "string", "lastName": "string", "email": "string", "password": "string" }
  - **Success Response (201 Created):** Returns the new User object.
  - **Error Response (409 Conflict):** If an account with the email already exists.
- **POST /api/v1/accounts/login**
  - **Description:** Authenticates a user and returns a JWT.
  - **Auth:** Public.
  - **Request Body:** { "email": "string", "password": "string" }
  - **Success Response (200 OK):** { "accessToken": "string", "expiresIn": 3600, "user": { ... } }
  - **Error Response (401 Unauthorized):** If credentials are invalid.
- **GET /api/v1/accounts/me**
  - **Description:** Retrieves the profile of the currently authenticated user.
  - **Auth:** User Required.
  - **Success Response (200 OK):** Returns the User object.

- **PUT /api/v1/accounts/me**
  - **Description:** Updates the profile of the currently authenticated user.
  - **Auth:** User Required.
  - **Request Body:** { "firstName": "string", "lastName": "string", "defaultShippingAddress": { ... } }
  - **Success Response (200 OK):** Returns the updated User object.

3. Product Service

- **Responsibility:** Manages the product catalog, categories, pricing, and inventory.
- **Service Discovery ID:** PRODUCT-SERVICE
- **Gateway Route Prefixes:** /api/v1/products, /api/v1/categories

Data Models

- **Product:**
  ```
  {
    "id": "uuid",
    "name": "string",
    "description": "string",
    "price": "number",
    "sku": "string",
    "stockQuantity": "integer",
    "category": { "id": "uuid", "name": "string" },
    "imageUrl": "string (url)",
    "isActive": "boolean"
  }
  ```

- **Category:** { "id": "uuid", "name": "string", "description": "string" }

API Endpoints

- **GET /api/v1/products**: (Public) Retrieves a paginated list of active products.
- **GET /api/v1/products/{productId}**: (Public) Retrieves a single product by its ID.
- **POST /api/v1/products**: (Admin) Creates a new product.
- **PUT /api/v1/products/{productId}**: (Admin) Updates an existing product.
- **DELETE /api/v1/products/{productId}**: (Admin) Soft-deletes a product by setting isActive to false.
- **GET /api/v1/categories**: (Public) Retrieves a list of all product categories.

4. Order Service

- **Responsibility:** Manages the order lifecycle, orchestrating calls to other services.
- **Service Discovery ID:** ORDER-SERVICE

- **Gateway Route Prefixes:** /api/v1/orders, /api/v1/accounts/{userId}/orders

Data Models

- **Order:**
  {
  "id": "uuid",
  "userId": "uuid",
  "items": [/* Array of OrderItem */],
  "totalAmount": "number",
  "status": "string (PENDING, AWAITING_PAYMENT, PROCESSING, SHIPPED, DELIVERED, CANCELLED)",
  "shippingAddress": "object (Address Model)",
  "createdAt": "string (ISO 8601)",
  "updatedAt": "string (ISO 8601)"
  }

- **OrderItem:** { "productId": "uuid", "name": "string", "quantity": "integer", "priceAtPurchase": "number" }

API Endpoints

- **POST /api/v1/orders**: (User) Creates a new order. Must verify product stock and price internally.
- **GET /api/v1/accounts/{userId}/orders**: (User) Retrieves the order history for a specific user.
- **GET /api/v1/orders/{orderId}**: (User/Admin) Retrieves the details of a single order.
- **POST /api/v1/orders/{orderId}/cancel**: (User) Cancels an order if it has not been shipped.

5. Payment Service

- **Responsibility:** Securely handles payment processing via a third-party provider.
- **Service Discovery ID:** PAYMENT-SERVICE
- **Gateway Route Prefix:** /api/v1/payments

Data Models

- **Payment:**
  {
  "id": "uuid",
  "orderId": "uuid",
  "amount": "number",

```
  "currency": "string (e.g., USD, LKR)",
  "status": "string (PENDING, SUCCEEDED, FAILED)",
  "provider": "string (e.g., stripe)",
  "providerTransactionId": "string",
  "createdAt": "string (ISO 8601)"
}
```

API Endpoints

- **POST /api/v1/payments/initiate**: (User) Initiates a payment for an order, returning a clientSecret for the frontend.
- **POST /api/v1/payments/webhook**: (Public/Webhook) Endpoint for the payment provider to send asynchronous status updates. Must be secured with signature verification. On success, it must notify the Order Service to update the order status.
- **GET /api/v1/orders/{orderId}/payment**: (User) Retrieves the payment status for a specific order.