LifePlus Healthcare Platform - API V2 Complete Reference

Version: 2.0 **Last Updated:** December 26, 2025 **Status:** DRAFT

CREATED BY: FARUK HANNAN

Table of Contents

API Design Standards

This section defines the core design principles and standards that govern the LifePlus API v2.0.

Core Principles

1. REST-First Design

The API strictly follows REST (Representational State Transfer) architectural principles:

- **Resources as Nouns:** All URIs represent resources using nouns (e.g., `/users`, `/orders`)
- **Stateless:** Each request contains all information needed to process it
- **Cacheable:** Responses explicitly indicate cacheability
- **Uniform Interface:** Consistent patterns across all endpoints

2. OpenAPI 3.0+ Compliance

All API definitions conform to OpenAPI Specification 3.0 or higher:

- Machine-readable API specifications
- Automated SDK generation
- Interactive documentation (Swagger UI)
- Contract-first development

3. JSON:API Specification

Response format follows JSON:API v1.0 specification:

- Consistent response structure
- Built-in relationship handling
- Standardized error format
- Sparse fieldsets and includes support

4. Security by Default

- All endpoints require authentication (no public endpoints without explicit design)
- HTTPS/TLS 1.2+ required for all communications
- Bearer token authentication for users
- API key authentication for partners
- Rate limiting on all endpoints

RESTful Architecture

Resource-Oriented Design

Every endpoint represents a **resource** (noun), not an action (verb):

✅ **Correct:**

```
GET    /api/v2/users
POST   /api/v2/orders
PATCH  /api/v2/users/123
DELETE /api/v2/cart/items/456
```

❌ **Incorrect:**

```
POST /api/v2/getUsers
POST /api/v2/createOrder
POST /api/v2/updateUser
POST /api/v2/deleteCartItem
```

Resource Hierarchy

Use nested resources for clear relationships:

```
/api/v2/users/123/addresses          # User's addresses
/api/v2/orders/456/items             # Order items
/api/v2/doctors/789/slots            # Doctor's appointment slots
/api/v2/package-orders/012/usage     # Package usage
```

URL Design

URL Structure Rules

1. **Use Nouns Only (No Verbs)**

   ```
   ✅ /api/v2/products
   ❌ /api/v2/getProducts
   ❌ /api/v2/list-products
   ```

2. **Use Plural Nouns for Collections**

   ```
   ✅ /api/v2/orders
   ✅ /api/v2/doctors
   ❌ /api/v2/order
   ❌ /api/v2/doctor
   ```

3. **Use Kebab-Case for Multi-Word Resources**

   ```
   ✅ /api/v2/home-sample-orders
   ✅ /api/v2/ambulance-orders
   ❌ /api/v2/homeSampleOrders
   ❌ /api/v2/home_sample_orders
   ```

4. **Actions as Query Parameters** When an action is required, use `?action=` parameter:

   ```
   ✅ POST /api/v2/orders/123?action=cancel
   ✅ POST /api/v2/phone-verifications?action=resend
   ❌ POST /api/v2/orders/123/cancel
   ❌ POST /api/v2/resend-otp
   ```

5. **Filters as Query Parameters**

   ```
   ✅ GET /api/v2/orders?filter[status]=pending
   ✅ GET /api/v2/products?filter[category]=medicine
   ❌ GET /api/v2/orders/pending
   ❌ GET /api/v2/products/medicine
   ```

HTTP Methods

Use standard HTTP methods with their intended semantics:

Method	Purpose	Idempotent	Safe	Example
GET	Retrieve resource(s)	✅ Yes	✅ Yes	`GET /api/v2/users/123`
POST	Create new resource	❌ No	❌ No	`POST /api/v2/orders`
PATCH	Partial update	❌ No	❌ No	`PATCH /api/v2/users/123`
PUT	Full replacement	✅ Yes	❌ No	`PUT /api/v2/addresses/456`
DELETE	Remove resource	✅ Yes	❌ No	`DELETE /api/v2/cart/items/789`

Method Usage Guidelines

GET - Retrieve Resources

- List collection: `GET /api/v2/products`
- Get single item: `GET /api/v2/products/123`
- Must not modify data
- Should be cacheable

- Should support filtering, sorting, pagination

POST - Create Resources

- Create new: `POST /api/v2/orders`
- Trigger action: `POST /api/v2/orders/123?action=cancel`
- Returns `201 Created` with `Location` header
- Non-idempotent (multiple calls create multiple resources)

PATCH - Partial Update

- Update specific fields: `PATCH /api/v2/users/123`
- Send only fields to update
- Returns `200 OK` with updated resource
- More efficient than PUT

DELETE - Remove Resources

- Delete resource: `DELETE /api/v2/addresses/456`
- Returns `204 No Content` (no body)
- Idempotent (deleting twice has same effect)

Request/Response Format

Request Format

Content-Type: `application/json` (or `multipart/form-data` for file uploads)

Standard Request Body (JSON:API):

```
{
  "data": {
    "type": "orders",
    "attributes": {
      "cart_id": "cart_123",
      "address_id": "addr_456",
      "payment_method": "cash_on_delivery"
    }
  }
}
```

Simple Request (Accepted):

```
{
  "cart_id": "cart_123",
  "address_id": "addr_456",
  "payment_method": "cash_on_delivery"
}
```

Response Format

All responses follow JSON:API specification:

Single Resource:

```
{
  "data": {
    "type": "users",
    "id": "user_123",
    "attributes": {
      "name": "John Doe",
      "email": "john@example.com",
      "phone": "+8801234567890"
    }
  },
  "meta": {
    "timestamp": "2025-12-26T10:00:00Z"
  }
}
```

Collection:

```
{
  "data": [
    {
      "type": "products",
      "id": "prod_1",
      "attributes": {
        "name": "Paracetamol 500mg",
        "price": 2.50
      }
    }
  ],
  "meta": {
    "total": 150,
    "page": {
      "number": 1,
      "size": 20
    }
  },
  "links": {
    "self": "/api/v2/products?page[number]=1",
    "next": "/api/v2/products?page[number]=2",
    "last": "/api/v2/products?page[number]=8"
  }
}
```

Status Codes

Code	Name	Usage
200	OK	Successful GET, PATCH, PUT
201	Created	Successful POST (resource created)
204	No Content	Successful DELETE
400	Bad Request	Invalid request format
401	Unauthorized	Authentication required/failed
403	Forbidden	Insufficient permissions
404	Not Found	Resource doesn't exist
409	Conflict	Resource conflict (duplicate)
422	Unprocessable Entity	Validation error
429	Too Many Requests	Rate limit exceeded
500	Internal Server Error	Server error
503	Service Unavailable	Temporary downtime

Data Models & Naming

Entity Models (Nouns Only)

All domain entities use **singular nouns**:

✅ **Correct Names:**

- `User` (not `UserModel`, `UserEntity`)
- `Order` (not `CreateOrder`, `OrderData`)
- `Product` (not `ProductInfo`, `ProductDetails`)

- `Appointment` (not `AppointmentBooking`)

❌ **Avoid:**

- Verbs: `CreateUser`, `UpdateOrder`, `DeleteProduct`
- Suffixes: `UserData`, `OrderMeta`, `ProductInner`
- Prefixes: `GetUser`, `ListOrders`

DTO/Payload Models

Operation payloads use **descriptive nouns**:

✅ **Correct:**

- `OrderCreation` (not `CreateOrderRequest`)
- `ProfileUpdate` (not `UpdateProfileRequest`)
- `CartItemAddition` (not `AddToCartPayload`)

Field Naming

Use `snake_case` for JSON field names:

```
{
  "first_name": "John",
  "last_name": "Doe",
  "date_of_birth": "1990-01-15",
  "phone_number": "+8801234567890",
  "is_verified": true
}
```

ID Format

- Use descriptive prefixes: `user_123`, `order_456`, `prod_789`
- Or use UUIDs: `550e8400-e29b-41d4-a716-446655440000`
- Never expose auto-increment integers directly

Error Handling Standards

Error Response Format

```
{
  "errors": [
    {
      "id": "error_unique_id",
      "status": "400",
      "code": "VALIDATION_ERROR",
      "title": "Validation Failed",
      "detail": "The phone number format is invalid",
      "source": {
        "pointer": "/data/attributes/phone"
      },
      "meta": {
        "timestamp": "2025-12-26T10:00:00Z"
      }
    }
  ]
}
```

Error Codes

Standard error codes across all endpoints:

Code	HTTP Status	Description
`VALIDATION_ERROR`	400	Request validation failed
`UNAUTHORIZED`	401	Authentication required
`INVALID_CREDENTIALS`	401	Invalid login credentials
`TOKEN_EXPIRED`	401	Auth token expired
`INVALID_API_KEY`	401	Partner API key invalid
`FORBIDDEN`	403	Insufficient permissions
`NOT_FOUND`	404	Resource not found
`CONFLICT`	409	Resource conflict
`UNPROCESSABLE_ENTITY`	422	Semantic validation error
`RATE_LIMIT_EXCEEDED`	429	Too many requests
`INTERNAL_ERROR`	500	Internal server error
`SERVICE_UNAVAILABLE`	503	Service temporarily down

Authentication Standards

User Authentication (Bearer Token)

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Token Characteristics:

- JWT format
- Expires after 24 hours
- Contains user ID and permissions
- Must be refreshed before expiry

Partner Authentication (API Key)

```
X-API-Key: your-api-key-here
X-Partner-ID: partner_123
```

API Key Characteristics:

- Long-lived credentials
- Tied to partner account
- Rate limited per key
- Can be rotated

Versioning

- **Current Version:** v2
- **Version in URL:** `/api/v2/`
- **Breaking Changes:** Require new version (v3)
- **Non-Breaking Changes:** Can be added to v2

Version Compatibility:

- v1 (legacy) - Deprecated, will be sunset in 6 months
- v2 (current) - Production, fully supported

- v3 (future) - Planned for major breaking changes

Legacy to V2 Complete Mapping

> **100% Coverage Achieved** ✅ | All 130+ endpoints mapped

📋 Quick Reference

Purpose: Complete 1:1 mapping of all legacy API endpoints to V2 RESTful API

🎯 V2 API Design Principles

✨ Core Rules:

- ✅ **Strict OpenAPI v3** compliance
- ✅ **URIs = Nouns only** (no verbs like `get`, `save`, `update`)
- ✅ **Actions = Query params** (`?action=resend`, `?action=cancel`)
- ✅ **Filters = Query params** (`?status=active`, `?type=diabetes`)
- ✅ **HTTP methods** properly used (GET, POST, PATCH, DELETE)
- ✅ **Kebab-case** naming convention

🔑 Key Changes at a Glance

Change Type	Legacy Example	V2 Example
🔹 **Remove Verbs**	`/api/v1/get-product`	`/api/v2/products`
🔹 **Actions as Params**	`/api/v1/resend-otp`	`/api/v2/phone-verifications?action=resend`
🔹 **Proper Methods**	`POST /update_profile`	`PATCH /profile`
🔹 **ID in Path**	`POST /product-details`	`GET /products/{id}`
🔹 **Filters as Params**	`/diabetes_package`	`/api/v2/packages?type=diabetes`
🔹 **User Context**	`/get_orders_by_user_id`	`/api/v2/orders` (auth token)

🔐 Authentication & User Management

🔑 Registration & Login

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/user/register`	POST	`/api/v2/users`	POST	Removed `/user` prefix
`/api/v1/user/login`	POST	`/api/v2/sessions`	POST	Session as resource
`/api/v1/user/me`	POST	`/api/v2/profile`	GET	Changed to GET
`/api/v1/send_otp`	POST	`/api/v2/phone-verifications`	POST	Noun-based naming

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/resend-otp`	POST	`/api/v2/phone-verifications?action=resend`	POST	Action as param ⚡
`/api/v1/verify-otp`	POST	`/api/v2/phone-verifications?action=verify`	POST	Action as param ⚡

👤 Profile Management

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/user_profile_image`	POST	`/profile/avatar`	POST	Clearer naming
`/api/v1/update_profile`	POST	`/api/v2/profile`	PATCH	Proper HTTP method ⚡
`/api/v1/beneficiaries`	GET/POST	`/api/v2/beneficiaries`	GET/POST	Already compliant ✅
`/api/v1/beneficiaries/{id}`	PUT	`/beneficiaries/{id}`	PATCH	PATCH for updates ⚡

🛒 E-Commerce & Products

📦 Product Catalog

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/get-product`	GET	`/api/v2/products`	GET	Removed verb `get` ⚡
`/api/v1/product-details`	POST	`/api/v2/products/{id}`	GET	GET with ID param ⚡
`/api/v1/get_product_by_category`	GET	`/api/v2/products?category_id={id}`	GET	Filter as query param
`/api/v1/search-products`	GET	`/api/v2/products?search={term}`	GET	Search as query param
`/api/v1/get-lifestyle-category`	GET	`/api/v2/lifestyle-categories`	GET	Removed verb
`/api/v1/categories/lifestyle`	GET	`/api/v2/categories/lifestyle`	GET	Already compliant ✅
`/api/v1/categories/medicine`	GET	`/api/v2/categories/medicine`	GET	Already compliant ✅

🛒 Shopping Cart

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/get_cart_products_by_user_id`	GET	`/api/v2/cart`	GET	Auth token provides user context ⚡
`/api/v1/add-to-cart`	POST	`/api/v2/cart/items`	POST	Sub-resource pattern
`/api/v1/update-cart-item`	PUT	`/api/v2/cart/items/{id}`	PATCH	PATCH for updates
`/api/v1/remove-from-cart`	DELETE	`/api/v2/cart/items/{id}`	DELETE	Already compliant ✅
`/api/v1/clear-cart`	POST	`/api/v2/cart`	DELETE	DELETE entire resource ⚡

📋 Order Management

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/product-order-place`	POST	`/api/v2/orders`	POST	Unified orders endpoint
`/api/v1/get_orders_by_user_id`	GET	`/api/v2/orders`	GET	Auth context ⚡
`/api/v1/get_order_by_id`	GET	`/api/v2/orders/{id}`	GET	Removed verb
`/api/v1/update_order_status`	POST	`/api/v2/orders/{id}`	PATCH	PATCH for updates ⚡
`/api/v1/cancel-order`	POST	`/api/v2/orders/{id}?action=cancel`	PATCH	Action + state update

🏥 Healthcare Services

👨‍⚕️ Doctors & Specialties

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/doctors`	GET	`/api/v2/doctors`	GET	Already compliant ✅

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/doctors/{id}`	GET	`/api/v2/doctors/{slug}`	GET	Supports slug or ID
`/api/v1/emergency_doctor_list`	GET	`/api/v2/doctors?specialty=emergency`	GET	Filter as query param
`/api/v1/specialties`	GET	`/api/v2/specialties`	GET	Already compliant ✅
`/api/v1/specialties/{id}`	GET	`/api/v2/specialties/{id}`	GET	Already compliant ✅

📅 Appointments

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/save-appointment`	POST	`/api/v2/appointments`	POST	Removed verb `save` ⚡
`/api/v1/get-appointment-by-user-id`	GET	`/api/v2/appointments`	GET	Auth context
`/api/v1/get-appointment-by-appointment-id`	GET	`/api/v2/appointments/{id}`	GET	Simplified
`/api/v1/update-appointment-payment`	POST	`/api/v2/appointments/{id}/payment`	PATCH	Sub-resource
`/api/v1/cancel-appointment`	POST	`/api/v2/appointments/{id}?action=cancel`	PATCH	Action param
`/api/v1/save_foreign_doctor_appointment`	POST	`/api/v2/foreign-doctor-appointments`	POST	Noun-based
`/api/v1/get-foreign-doctor-appointment-by-user-id`	GET	`/api/v2/foreign-doctor-appointments`	GET	Auth context

🎥 Telemedicine & Video Consultations

Legacy	Method	➡️ V2 Endpoint	Method	💡 Ch
`/api/v1/user/reserve_doctor_slot`	POST	`/api/v2/telemedicine/slots`	POST	No ba res
`/api/v1/cancel_reserve_doctor_slot`	GET	`/api/v2/telemedicine/slots/{id}`	DELETE	DE to ⚡
`/api/v1/generate-video-token`	POST	`/api/v2/video-calls/{id}/token`	POST	Su res pa

Legacy	Method	➡️ V2 Endpoint	Method	💡 Ch
`/api/v1/generate-video-token-delete`	POST	`/api/v2/video-calls/{id}/token`	DELETE	DE op ⚡
`/api/v1/get_telemedicine_token_user`	POST	`/api/v2/video-calls/{id}/token`	GET	GE op ⚡
`/api/v1/patient_video_call_prescription_upload`	POST	`/api/v2/video-calls/{id}/prescriptions`	POST	Su res
`/api/v1/doctor_video_call_prescription_upload`	POST	`/api/v2/video-calls/{id}/prescriptions`	POST	Sa en
`/api/v1/get-telemedicine-history-by-user-id`	GET	`/api/v2/telemedicine/history`	GET	Au co
`/api/v1/emergency-consultation-doctor`	GET	`/api/v2/doctors?service=emergency-consultation`	GET	Qu filt

🧪 Diagnostics & Lab Tests

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/diagnostics`	GET	`/api/v2/diagnostics`	GET	Already compliant ✅
`/api/v1/diagnostics/{id}`	GET	`/api/v2/diagnostics/{id}`	GET	Already compliant ✅
`/api/v1/diagnostic_categories`	GET	`/api/v2/diagnostic-categories`	GET	Kebab-case
`/api/v1/diagnostic_categories/{id}`	GET	`/api/v2/diagnostic-categories/{id}`	GET	Kebab-case
`/api/v1/diagnostic_orders`	POST/GET	`/api/v2/diagnostic-orders`	POST/GET	Noun-based
`/api/v1/diagnostic_orders/{id}`	GET	`/api/v2/diagnostic-orders/{id}`	GET	Noun-based

🏥 Hospitals

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/hospitals`	GET	`/api/v2/hospitals`	GET	Already compliant ✅
`/api/v1/hospitals/{id}`	GET	`/api/v2/hospitals/{id}`	GET	Already compliant ✅

📦 Packages & Insurance

💊 Health Packages

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/package`	GET	`/api/v2/packages`	GET	Plural naming
`/api/v1/package/{id}`	GET	`/api/v2/packages/{id}`	GET	Plural naming
`/api/v1/get_package_list`	GET	`/api/v2/packages`	GET	Removed verb
`/api/v1/diabetes_package`	GET	`/api/v2/packages?type=diabetes`	GET	Query param filter ⚡
`/api/v1/all_package_type`	GET	`/api/v2/package-types`	GET	Noun-based
`/api/v1/show_package_type/{id}/show`	GET	`/api/v2/package-types/{id}`	GET	Cleaned path
`/api/v1/save_package_type`	POST	`/api/v2/package-types`	POST	Removed verb
`/api/v1/update_package_type`	POST	`/api/v2/package-types/{id}`	PATCH	PATCH for updates ⚡
`/api/v1/delete_package_type`	POST	`/api/v2/package-types/{id}`	DELETE	DELETE operation ⚡

📝 Package Orders

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/package_order`	POST	`/api/v2/orders?order_type=package`	POST	Unified orders
`/api/v1/package_order/{id}`	GET	`/api/v2/orders/{id}`	GET	Unified orders
`/api/v1/get_package_orders_by_user_id`	POST	`/api/v2/orders?order_type=package`	GET	Query filter ⚡
`/api/v1/get_package_order_list`	GET	`/api/v2/orders?order_type=package`	GET	Query filter
`/api/v1/update_package_order`	POST	`/api/v2/orders/{id}`	PATCH	PATCH for updates
`/api/v1/enroll_corporate_package`	POST	`/api/v2/package-orders/corporate-enrollment`	POST	Noun-based DTO
`/api/v1/package_service_used`	POST	`/api/v2/package-orders/{id}/service-usage`	POST	Sub-resource
`/api/v1/package_order_claims`	POST	`/api/v2/package-orders/{id}/claims`	POST	Sub-resource

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/user_package_check`	POST	`/users/{id}/package-status`	GET	Changed to GET ⚡

💳 Health Cards

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/health_cards`	GET	`/api/v2/health-cards`	GET	Kebab-case
`/api/v1/health_cards/{id}`	GET	`/api/v2/health-cards/{id}`	GET	Kebab-case

🛡️ Insurance Policies

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/policies`	GET	`/api/v2/policies`	GET	Already compliant ✅
`/api/v1/policies/{id}`	GET	`/api/v2/policies/{id}`	GET	Already compliant ✅
`/api/v1/policy/search`	GET	`/api/v2/policies?search={term}`	GET	Query param
`/api/v1/policy_categories`	GET	`/api/v2/policy-categories`	GET	Kebab-case
`/api/v1/policy_orders`	POST/GET	`/api/v2/policy-orders`	POST/GET	Noun-based
`/api/v1/policy_orders/{id}`	GET	`/api/v2/policy-orders/{id}`	GET	Noun-based
`/api/v1/policy_orders/update_order_status`	POST	`/api/v2/policy-orders/{id}/status`	PATCH	Sub-resource
`/api/v1/policy_claim`	POST	`/api/v2/policy-orders/{id}/claim`	POST	Sub-resource

🏠 Home Services

🩺 Home Sample Collection

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/home-sample-orders`	POST/GET	`/api/v2/home-sample-orders`	POST/GET	Already compliant ✅
`/api/v1/home-sample-tests`	GET	`/api/v2/home-sample-tests`	GET	Already compliant ✅

🧑‍⚕️ Home Care (Nursing)

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
/api/v1/nurse-orders	POST	/api/v2/home-care/orders	POST	Renamed for clarity
/api/v1/get_nurse_order_list	GET	/api/v2/home-care/orders	GET	Removed verb
/api/v1/nurse-orders/{id}	GET	/api/v2/home-care/orders/{id}	GET	Noun-based

🚑 Ambulance Services

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
/api/v1/ambulance-areas	GET	/api/v2/ambulance/areas	GET	Grouped under ambulance
/api/v1/ambulance-price	POST	/api/v2/ambulance/price	POST	Price calculation
/api/v1/ambulance-orders	POST	/api/v2/ambulance/orders	POST	Noun-based
/api/v1/ambulance-orders/{id}	GET	/api/v2/ambulance/orders/{id}	GET	Noun-based

✈️ Wellbeing & Medical Tourism

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
/api/v1/wellbeing/order_create	POST	/api/v2/wellbeing/orders	POST	Removed verb
/api/v1/wellbeing/order_history	GET	/api/v2/wellbeing/orders	GET	Same endpoint
/api/v1/countries	GET	/api/v2/countries	GET	Already compliant ✅

🔔 Additional Features

💊 Medicine Reminders

Legacy	Method	➡️ V2 Endpoint	Method
/api/v1/save_medicine_reminder	POST	/api/v2/medicine-reminders	POST
/api/v1/get_medicine_reminder	GET	/api/v2/medicine-reminders	GET
/api/v1/medicine_reminder_time_status_change	POST	/api/v2/medicine-reminders/{id}/times/{timeId}	PATCH
/api/v1/medicine_reminder_medicine_remove	POST	/api/v2/medicine-reminders/{id}/medicines/{medicineId}	DELETE

📄 Prescriptions

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/upload_prescription`	POST	`/api/v2/prescriptions`	POST	Removed verb

🎟️ Promo Codes & Corporate

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/validate_promo_code`	POST	`/api/v2/promo-code-validations`	POST	Noun-based DTO
`/api/v1/validate_corporate_house_code`	POST	`/api/v2/corporate/code-validations`	POST	Noun-based DTO

⭐ Reviews & Ratings

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/user_review`	POST/GET	`/api/v2/reviews`	POST/GET	Simplified
`/api/v1/get_user_review_list`	GET	`/api/v2/reviews`	GET	Removed verb
`/api/v1/user_review/{id}`	GET	`/api/v2/reviews/{id}`	GET	Simplified

🔔 Notifications

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/save-notification`	POST	`/api/v2/notifications`	POST	Removed verb
`/api/v1/user_notifications`	GET	`/api/v2/notifications`	GET	Simplified

📝 Blog & Content

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/get-all-blog`	GET	`/api/v2/blogs`	GET	Removed verb
`/api/v1/get-blog-by-type-id`	POST	`/api/v2/blogs?type_id={id}`	GET	Query param ⚡
`/api/v1/get_all_diabetic_blogs`	GET	`/api/v2/blogs?category=diabetic`	GET	Query param
`/api/v1/blogs/{id}`	GET	`/api/v2/blogs/{id}`	GET	Already compliant ✅

⚙️ Settings & Configuration

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/slider-images`	GET	`/api/v2/slider-images`	GET	Already compliant ✅
`/api/v1/delivery-charges`	GET	`/api/v2/delivery-charges`	GET	Already compliant ✅
`/api/v1/settings`	GET	`/api/v2/settings`	GET	Already compliant ✅

📍 Address Management

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/addresses`	POST/GET	`/api/v2/addresses`	POST/GET	Already compliant ✅
`/api/v1/addresses/{id}`	GET/DELETE	`/api/v2/addresses/{id}`	GET/DELETE	Already compliant ✅
`/api/v1/addresses/{id}`	PUT	`/api/v2/addresses/{id}`	PATCH	PATCH for updates ⚡

🛠️ Utilities

Legacy	Method	➡️ V2 Endpoint	Method	💡 Change
`/api/v1/get-user-invoice-pdf`	GET	`/api/v2/orders/{id}/invoice`	GET	Sub-resource pattern
`/api/v1/save_hotline_request`	POST	`/api/v2/hotline-requests`	POST	Noun-based
`/api/v1/generics`	GET	`/api/v2/generics`	GET	Already compliant ✅
`/api/v1/manufacturers`	GET	`/api/v2/manufacturers`	GET	Already compliant ✅
`/api/v1/partners`	GET	`/api/v2/partners`	GET	Already compliant ✅
`/api/v1/partners/{id}`	GET	`/api/v2/partners/{id}`	GET	Already compliant ✅

🤝 Partner API (New in V2)

These endpoints are NEW - designed specifically for B2B partner integrations:

V2 Endpoint	Method	Purpose
✨`/partners/api-keys`	POST	Create API key for partner authentication
✨`/partners/api-keys`	GET	List all API keys
✨`/partners/api-keys/{keyId}`	GET	Get API key details
✨`/partners/api-keys/{keyId}`	PATCH	Update API key permissions
✨`/partners/api-keys/{keyId}`	DELETE	Revoke API key
✨`/partners/api-keys/validate`	POST	Validate API key authentication

🚀 Quick Migration Tips

💡 Common Patterns

1. Remove Verbs from URLs

```
- GET  /api/v1/get-product
+ GET  /api/v2/products
```

2. Use Proper HTTP Methods

```
- POST /api/v1/update_profile
+ PATCH /api/v2/profile
```

3. Actions as Query Parameters

```
- POST /api/v1/resend-otp
+ POST /api/v2/phone-verifications?action=resend
```

4. User Context from Auth Token

```
- GET  /api/v1/get_orders_by_user_id
+ GET  /api/v2/orders  (uses Bearer token)
```

5. Filters as Query Parameters

```
- GET  /api/v1/diabetes_package
+ GET  /api/v2/packages?type=diabetes
```

API Overview

Base URLs

- **Production:** `https://api.lifeplusbd.com`
- **Staging:** `https://staging-api.lifeplusbd.com`
- **API Version:** `/api/v2`

Full Endpoint Example

```
https://api.lifeplusbd.com/api/v2/products?filter[category]=medicine&page[number]=1&page[size]=20
```

Common Query Parameters

Pagination

All list endpoints support pagination:

```
GET /api/v2/products?page[number]=1&page[size]=20
```

- `page[number]` - Page number (default: 1)
- `page[size]` - Items per page (default: 20, max: 100)

Response includes pagination metadata:

```
{
  "data": [...],
  "meta": {
    "total": 150,
    "page": {
      "number": 1,
      "size": 20
    }
  },
  "links": {
    "first": "/api/v2/products?page[number]=1",
    "prev": null,
    "self": "/api/v2/products?page[number]=1",
    "next": "/api/v2/products?page[number]=2",
    "last": "/api/v2/products?page[number]=8"
  }
}
```

Filtering

Use `filter[field]` to filter results:

```
GET /api/v2/orders?filter[status]=pending
GET /api/v2/products?filter[category]=medicine&filter[in_stock]=true
GET /api/v2/appointments?filter[date_from]=2025-12-26&filter[date_to]=2025-12-31
```

Sorting

Use `sort` parameter (prefix with `-` for descending):

```
GET /api/v2/orders?sort=created_at          # Ascending
GET /api/v2/orders?sort=-created_at         # Descending
GET /api/v2/products?sort=price,-created_at # Multiple fields
```

Search

Use `search` parameter for full-text search:

```
GET /api/v2/products?search=paracetamol
GET /api/v2/doctors?search=cardiology
```

Sparse Fieldsets

Request only specific fields:

```
GET /api/v2/users/123?fields[users]=name,email
```

Including Related Resources

Include related resources in response:

```
GET /api/v2/orders/123?include=items,user,address
```

Authentication & Authorization

Overview

LifePlus API v2 supports two authentication methods:

1. **Bearer Token Authentication** - For end-user applications (mobile, web)
2. **API Key Authentication** - For B2B partner integrations

Bearer Token Authentication (End Users)

Flow Overview

1. User requests OTP via phone verification
2. User verifies OTP
3. System returns session token
4. Client includes token in all subsequent requests

Step 1: Request OTP

Endpoint: `POST /api/v2/phone-verifications`

Request:

```
POST /api/v2/phone-verifications
Content-Type: application/json

{
  "phone": "+8801234567890"
}
```

Response: `201 Created`

```
{
  "data": {
    "type": "phone-verifications",
    "id": "verify_abc123",
    "attributes": {
      "phone": "+8801234567890",
      "expires_at": "2025-12-26T10:10:00Z",
      "status": "pending"
    }
  }
}
```

Step 2: Verify OTP

Endpoint: `POST /api/v2/phone-verifications?action=verify`

Request:

```
POST /api/v2/phone-verifications?action=verify
Content-Type: application/json

{
  "phone": "+8801234567890",
  "otp": "123456"
}
```

Response: `200 OK`

```
{
  "data": {
    "type": "phone-verifications",
    "id": "verify_abc123",
    "attributes": {
      "phone": "+8801234567890",
      "status": "verified",
      "verified_at": "2025-12-26T10:05:00Z"
    }
  }
}
```

Step 3: Create Session (Login)

Endpoint: `POST /api/v2/sessions`

Request:

```
POST /api/v2/sessions
Content-Type: application/json

{
  "phone": "+8801234567890",
  "otp": "123456"
}
```

Response: `201 Created`

```
{
  "data": {
    "type": "sessions",
    "id": "session_xyz789",
    "attributes": {
      "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
      "expires_at": "2025-12-27T10:05:00Z",
      "user": {
        "id": "user_123",
        "name": "John Doe",
        "phone": "+8801234567890",
        "email": "john@example.com"
      }
    }
  }
}
```

Step 4: Use Token in Requests

Add Authorization header to all requests:

```
GET /api/v2/users/me
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Resend OTP

Endpoint: `POST /api/v2/phone-verifications?action=resend`

Request:

```
POST /api/v2/phone-verifications?action=resend
Content-Type: application/json

{
  "phone": "+8801234567890"
}
```

Logout

Endpoint: `DELETE /api/v2/sessions/{session_id}`

Request:

```
DELETE /api/v2/sessions/session_xyz789
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Response: `204 No Content`

API Key Authentication (Partners)

Partners use API key authentication for B2B integrations.

Headers Required:

```
X-API-Key: your-api-key-here
X-Partner-ID: partner_123
Content-Type: application/json
```

Example Request:

```
GET /api/v2/partner/orders
X-API-Key: sk_live_abc123xyz789
X-Partner-ID: partner_123
```

Validate API Key:

```
POST /api/v2/partner/api-keys?action=validate
X-API-Key: sk_live_abc123xyz789
X-Partner-ID: partner_123
```

Response:

```
{
  "data": {
    "type": "api-key-validations",
    "attributes": {
      "valid": true,
      "partner_id": "partner_123",
      "partner_name": "ABC Healthcare",
      "permissions": [
        "orders:read",
        "orders:create",
        "orders:update"
      ],
      "rate_limit": 1000,
      "expires_at": "2026-12-31T23:59:59Z"
    }
  }
}
```

Security Best Practices

1. **Always use HTTPS** - Never send credentials over HTTP
2. **Store tokens securely** - Use secure storage mechanisms
3. **Never expose tokens** - Don't log or commit tokens
4. **Implement token refresh** - Refresh before expiration
5. **Rotate API keys** - Regular rotation for partner keys
6. **Use rate limiting** - Respect rate limits
7. **Validate on every request** - Server validates all tokens

---

API Endpoints Reference

This section provides detailed documentation for all API endpoints, organized by functional area.

Authentication Endpoints

Create Session (Login)

Create a new authenticated session for a user.

Endpoint: `POST /api/v2/sessions`

Authentication: None (public endpoint)

Request Body:

```
{
  "phone": "+8801234567890",
  "otp": "123456"
}
```

Success Response: `201 Created`

```
{
  "data": {
    "type": "sessions",
    "id": "session_xyz789",
    "attributes": {
      "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
      "expires_at": "2025-12-27T10:05:00Z",
      "user": {
        "id": "user_123",
        "name": "John Doe",
        "phone": "+8801234567890",
        "email": "john@example.com",
        "avatar_url": "https://cdn.lifeplusbd.com/avatars/user_123.jpg"
      }
    }
  }
}
```

Error Responses:

- `401 Unauthorized` - Invalid phone or OTP

- `422 Unprocessable Entity` - OTP expired or already used

Delete Session (Logout)

End an authenticated session.

Endpoint: `DELETE /api/v2/sessions/{session_id}`

Authentication: Required (Bearer token)

Success Response: `204 No Content`

Error Responses:

- `401 Unauthorized` - Invalid or expired token
- `404 Not Found` - Session not found

Request Phone Verification (OTP)

Send OTP to a phone number for verification.

Endpoint: `POST /api/v2/phone-verifications`

Authentication: None (public endpoint)

Request Body:

```json
{
  "phone": "+8801234567890"
}
```

Success Response: `201 Created`

```json
{
  "data": {
    "type": "phone-verifications",
    "id": "verify_abc123",
    "attributes": {
      "phone": "+8801234567890",
      "expires_at": "2025-12-26T10:10:00Z",
      "status": "pending",
      "attempts_remaining": 3
    }
  }
}
```

Error Responses:

- `429 Too Many Requests` - Rate limit exceeded (max 3 per hour)
- `422 Unprocessable Entity` - Invalid phone format

Verify Phone (OTP)

Verify a phone number using the OTP code.

Endpoint: `POST /api/v2/phone-verifications?action=verify`

Authentication: None (public endpoint)

Request Body:

```
{
  "phone": "+8801234567890",
  "otp": "123456"
}
```

Success Response: `200 OK`

```
{
  "data": {
    "type": "phone-verifications",
    "id": "verify_abc123",
    "attributes": {
      "phone": "+8801234567890",
      "status": "verified",
      "verified_at": "2025-12-26T10:05:00Z"
    }
  }
}
```

Error Responses:

- `401 Unauthorized` - Invalid OTP
- `422 Unprocessable Entity` - OTP expired or max attempts exceeded

Resend OTP

Resend OTP to a phone number.

Endpoint: `POST /api/v2/phone-verifications?action=resend`

Authentication: None (public endpoint)

Request Body:

```
{
  "phone": "+8801234567890"
}
```

Success Response: `200 OK`

```
{
  "data": {
    "type": "phone-verifications",
    "id": "verify_def456",
    "attributes": {
      "phone": "+8801234567890",
      "expires_at": "2025-12-26T10:15:00Z",
      "status": "pending",
      "attempts_remaining": 3
    }
  }
}
```

User Management

Register New User

Create a new user account.

Endpoint: `POST /api/v2/users`

Authentication: None (public endpoint, but requires verified phone)

Request Body:

```
{
  "phone": "+8801234567890",
  "name": "John Doe",
  "email": "john@example.com",
  "password": "SecureP@ssw0rd",
  "date_of_birth": "1990-01-15",
  "gender": "male"
}
```

Field Validations:

- `phone` - Required, E.164 format, must be verified
- `name` - Required, 2-100 characters
- `email` - Optional, valid email format
- `password` - Required, min 8 characters, must include uppercase, lowercase, number
- `date_of_birth` - Required, ISO 8601 date format
- `gender` - Required, one of: male, female, other

Success Response: `201 Created`

```
{
  "data": {
    "type": "users",
    "id": "user_123",
    "attributes": {
      "name": "John Doe",
      "phone": "+8801234567890",
      "email": "john@example.com",
      "date_of_birth": "1990-01-15",
      "gender": "male",
      "avatar_url": null,
      "created_at": "2025-12-26T10:00:00Z"
    }
  }
}
```

Error Responses:

- `409 Conflict` - Phone number already registered
- `422 Unprocessable Entity` - Validation errors

Get Current User Profile

Retrieve the authenticated user's profile.

Endpoint: `GET /api/v2/users/me`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "users",
    "id": "user_123",
    "attributes": {
      "name": "John Doe",
      "phone": "+8801234567890",
      "email": "john@example.com",
      "date_of_birth": "1990-01-15",
      "gender": "male",
      "avatar_url": "https://cdn.lifeplusbd.com/avatars/user_123.jpg",
      "created_at": "2025-12-26T10:00:00Z",
      "updated_at": "2025-12-26T10:00:00Z"
    }
  }
}
```

Update User Profile

Update the authenticated user's profile information.

Endpoint: `PATCH /api/v2/users/me`

Authentication: Required (Bearer token)

Request Body (partial update):

```
{
  "name": "John Updated",
  "email": "newemail@example.com"
}
```

Updatable Fields:

- `name` - User's full name
- `email` - Email address
- `date_of_birth` - Date of birth
- `gender` - Gender

Success Response: `200 OK`

```
{
  "data": {
    "type": "users",
    "id": "user_123",
    "attributes": {
      "name": "John Updated",
      "phone": "+8801234567890",
      "email": "newemail@example.com",
      "date_of_birth": "1990-01-15",
      "gender": "male",
      "avatar_url": "https://cdn.lifeplusbd.com/avatars/user_123.jpg",
      "updated_at": "2025-12-26T10:30:00Z"
    }
  }
}
```

Upload User Avatar

Upload or update user's profile picture.

Endpoint: `POST /api/v2/users/me/avatar`

Authentication: Required (Bearer token)

Content-Type: `multipart/form-data`

Form Fields:

- `file` - Image file (JPEG, PNG, max 5MB)

Success Response: `200 OK`

```
{
  "data": {
    "type": "avatars",
    "attributes": {
      "url": "https://cdn.lifeplusbd.com/avatars/user_123.jpg",
      "uploaded_at": "2025-12-26T10:35:00Z"
    }
  }
}
```

Error Responses:

- `400 Bad Request` - Invalid file format or size
- `413 Payload Too Large` - File exceeds 5MB

Change Password

Change the authenticated user's password.

Endpoint: `PATCH /api/v2/users/me/password`

Authentication: Required (Bearer token)

Request Body:

```
{
  "current_password": "OldP@ssw0rd",
  "new_password": "NewSecureP@ssw0rd",
  "new_password_confirmation": "NewSecureP@ssw0rd"
}
```

Success Response: `200 OK`

```
{
  "data": {
    "type": "password-changes",
    "attributes": {
      "changed_at": "2025-12-26T10:40:00Z",
      "message": "Password changed successfully"
    }
  }
}
```

Error Responses:

- `401 Unauthorized` - Current password incorrect
- `422 Unprocessable Entity` - New password validation failed

List User Addresses

Get all addresses for the authenticated user.

Endpoint: `GET /api/v2/users/me/addresses`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": [
    {
      "type": "addresses",
      "id": "addr_123",
      "attributes": {
        "label": "Home",
        "address_line_1": "123 Main Street",
        "address_line_2": "Apt 4B",
        "city": "Dhaka",
        "state": "Dhaka Division",
        "postal_code": "1200",
        "country": "Bangladesh",
        "latitude": 23.8103,
        "longitude": 90.4125,
        "is_default": true,
        "created_at": "2025-12-26T10:00:00Z"
      }
    }
  ]
}
```

Create Address

Add a new address for the authenticated user.

Endpoint: `POST /api/v2/users/me/addresses`

Authentication: Required (Bearer token)

Request Body:

```
{
  "label": "Office",
  "address_line_1": "456 Business Ave",
  "address_line_2": "Suite 200",
  "city": "Dhaka",
  "state": "Dhaka Division",
  "postal_code": "1205",
  "country": "Bangladesh",
  "latitude": 23.7465,
  "longitude": 90.3763,
  "is_default": false
}
```

Field Validations:

- `label` - Required, e.g., "Home", "Office"
- `address_line_1` - Required
- `city` - Required
- `country` - Required
- `latitude`, `longitude` - Optional, for delivery optimization

Success Response: `201 Created`

```
{
  "data": {
    "type": "addresses",
    "id": "addr_456",
    "attributes": {
      "label": "Office",
      "address_line_1": "456 Business Ave",
      "address_line_2": "Suite 200",
      "city": "Dhaka",
      "state": "Dhaka Division",
      "postal_code": "1205",
      "country": "Bangladesh",
      "latitude": 23.7465,
      "longitude": 90.3763,
      "is_default": false,
      "created_at": "2025-12-26T10:45:00Z"
    }
  }
}
```

Update Address

Update an existing address.

Endpoint: `PATCH /api/v2/users/me/addresses/{address_id}`

Authentication: Required (Bearer token)

Request Body:

```
{
  "label": "Home (Updated)",
  "is_default": true
}
```

Success Response: `200 OK`

Delete Address

Remove an address from the user's account.

Endpoint: `DELETE /api/v2/users/me/addresses/{address_id}`

Authentication: Required (Bearer token)

Success Response: `204 No Content`

Error Responses:

- `404 Not Found` - Address doesn't exist
- `422 Unprocessable Entity` - Cannot delete the only address if user has active orders

---

Cart & Orders

Get Cart

Retrieve the current user's shopping cart.

Endpoint: `GET /api/v2/cart`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "carts",
    "id": "cart_123",
    "attributes": {
      "user_id": "user_123",
      "items": [
        {
          "id": "item_1",
          "product_id": "prod_456",
          "product_name": "Paracetamol 500mg",
          "product_image": "https://cdn.lifeplusbd.com/products/prod_456.jpg",
          "quantity": 2,
          "unit_price": 50.00,
          "subtotal": 100.00,
          "requires_prescription": false
        },
        {
          "id": "item_2",
          "product_id": "prod_789",
          "product_name": "Amoxicillin 250mg",
          "product_image": "https://cdn.lifeplusbd.com/products/prod_789.jpg",
          "quantity": 1,
          "unit_price": 150.00,
          "subtotal": 150.00,
          "requires_prescription": true
        }
      ],
      "subtotal": 250.00,
      "discount": 25.00,
      "promo_code": "SAVE10",
      "delivery_fee": 50.00,
      "total": 275.00,
      "requires_prescription": true,
      "updated_at": "2025-12-26T11:00:00Z"
    }
  }
}
```

Add Item to Cart

Add a product to the shopping cart.

Endpoint: `POST /api/v2/cart/items`

Authentication: Required (Bearer token)

Request Body:

```
{
  "product_id": "prod_456",
  "quantity": 2
}
```

Success Response: `201 Created`

```
{
  "data": {
    "type": "cart-items",
    "id": "item_3",
    "attributes": {
      "product_id": "prod_456",
      "product_name": "Paracetamol 500mg",
      "quantity": 2,
      "unit_price": 50.00,
      "subtotal": 100.00,
      "added_at": "2025-12-26T11:05:00Z"
    }
  }
}
```

Error Responses:

- `404 Not Found` - Product doesn't exist
- `422 Unprocessable Entity` - Product out of stock or quantity exceeds available stock

---

Update Cart Item

Update the quantity of an item in the cart.

Endpoint: `PATCH /api/v2/cart/items/{item_id}`

Authentication: Required (Bearer token)

Request Body:

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

Success Response: `200 OK`

```
{
  "data": {
    "type": "cart-items",
    "id": "item_1",
    "attributes": {
      "product_id": "prod_456",
      "product_name": "Paracetamol 500mg",
      "quantity": 3,
      "unit_price": 50.00,
      "subtotal": 150.00,
      "updated_at": "2025-12-26T11:10:00Z"
    }
  }
}
```

Error Responses:

- `404 Not Found` - Cart item doesn't exist
- `422 Unprocessable Entity` - Quantity exceeds available stock

Remove Item from Cart

Remove a specific item from the cart.

Endpoint: `DELETE /api/v2/cart/items/{item_id}`

Authentication: Required (Bearer token)

Success Response: `204 No Content`

Clear Cart

Remove all items from the cart.

Endpoint: `DELETE /api/v2/cart`

Authentication: Required (Bearer token)

Success Response: `204 No Content`

Apply Promo Code

Apply a promotional code to the cart.

Endpoint: `POST /api/v2/cart/promo-codes`

Authentication: Required (Bearer token)

Request Body:

```
{
  "code": "SAVE10"
}
```

Success Response: `200 OK`

```
{
  "data": {
    "type": "promo-code-applications",
    "attributes": {
      "code": "SAVE10",
      "discount_type": "percentage",
      "discount_value": 10,
      "discount_amount": 25.00,
      "applied_at": "2025-12-26T11:15:00Z",
      "cart_total": 275.00
    }
  }
}
```

Error Responses:

- `404 Not Found` - Promo code doesn't exist
- `422 Unprocessable Entity` - Promo code expired, invalid, or minimum order not met

---

Remove Promo Code

Remove the applied promo code from the cart.

Endpoint: `DELETE /api/v2/cart/promo-codes`

Authentication: Required (Bearer token)

Success Response: `204 No Content`

Create Order

Create a new order from the cart.

Endpoint: `POST /api/v2/orders`

Authentication: Required (Bearer token)

Request Body:

```
{
  "address_id": "addr_123",
  "payment_method": "cash_on_delivery",
  "notes": "Please call before delivery",
  "delivery_time_preference": "morning"
}
```

Field Validations:

- `address_id` - Required, must be user's address
- `payment_method` - Required, one of: `cash_on_delivery`, `bkash`, `card`, `nagad`
- `notes` - Optional, max 500 characters
- `delivery_time_preference` - Optional, one of: `morning`, `afternoon`, `evening`

Success Response: `201 Created`

```
{
  "data": {
    "type": "orders",
    "id": "order_789",
    "attributes": {
      "order_number": "LP-2025-001234",
      "user_id": "user_123",
      "status": "pending",
      "items": [
        {
          "product_id": "prod_456",
          "product_name": "Paracetamol 500mg",
          "quantity": 2,
          "unit_price": 50.00,
          "subtotal": 100.00
        }
      ],
      "subtotal": 250.00,
      "discount": 25.00,
      "delivery_fee": 50.00,
      "total": 275.00,
      "payment_method": "cash_on_delivery",
      "payment_status": "pending",
      "address": {
        "label": "Home",
        "address_line_1": "123 Main Street",
        "city": "Dhaka",
        "phone": "+8801234567890"
      },
      "requires_prescription": false,
      "notes": "Please call before delivery",
      "delivery_time_preference": "morning",
      "created_at": "2025-12-26T11:20:00Z",
      "estimated_delivery": "2025-12-27T15:00:00Z"
    }
  }
}
```

Error Responses:

- `400 Bad Request` - Empty cart
- `404 Not Found` - Address doesn't exist
- `422 Unprocessable Entity` - Cart contains items that require prescription but none uploaded

---

List Orders

Retrieve all orders for the authenticated user.

Endpoint: `GET /api/v2/orders`

Authentication: Required (Bearer token)

Query Parameters:

- `filter[status]` - Filter by status: `pending`, `confirmed`, `processing`, `shipped`, `delivered`, `cancelled`
- `filter[date_from]` - Filter by creation date (ISO 8601)
- `filter[date_to]` - Filter by creation date (ISO 8601)
- `sort` - Sort field: `created_at`, `-created_at` (default: `-created_at`)
- `page[number]` - Page number
- `page[size]` - Items per page

Success Response: `200 OK`

```
{
  "data": [
    {
      "type": "orders",
      "id": "order_789",
      "attributes": {
        "order_number": "LP-2025-001234",
        "status": "processing",
        "total": 275.00,
        "payment_method": "cash_on_delivery",
        "payment_status": "pending",
        "created_at": "2025-12-26T11:20:00Z",
        "estimated_delivery": "2025-12-27T15:00:00Z"
      }
    }
  ],
  "meta": {
    "total": 15,
    "page": {
      "number": 1,
      "size": 20
    }
  },
  "links": {
    "self": "/api/v2/orders?page[number]=1",
    "next": null
  }
}
```

Get Order Details

Retrieve detailed information about a specific order.

Endpoint: `GET /api/v2/orders/{order_id}`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "orders",
    "id": "order_789",
    "attributes": {
      "order_number": "LP-2025-001234",
      "user_id": "user_123",
      "status": "processing",
      "items": [
        {
          "id": "orderitem_1",
          "product_id": "prod_456",
          "product_name": "Paracetamol 500mg",
          "product_image": "https://cdn.lifeplusbd.com/products/prod_456.jpg",
          "quantity": 2,
          "unit_price": 50.00,
          "subtotal": 100.00
        }
      ],
      "subtotal": 250.00,
      "discount": 25.00,
      "delivery_fee": 50.00,
      "total": 275.00,
      "payment_method": "cash_on_delivery",
      "payment_status": "pending",
      "address": {
        "label": "Home",
        "address_line_1": "123 Main Street",
        "address_line_2": "Apt 4B",
        "city": "Dhaka",
        "postal_code": "1200",
        "phone": "+8801234567890"
      },
      "tracking_number": null,
      "status_history": [
        {
          "status": "pending",
          "timestamp": "2025-12-26T11:20:00Z",
          "note": "Order placed"
        },
        {
          "status": "confirmed",
          "timestamp": "2025-12-26T11:30:00Z",
          "note": "Order confirmed"
        },
        {
          "status": "processing",
          "timestamp": "2025-12-26T12:00:00Z",
          "note": "Order is being prepared"
        }
      ],
      "prescriptions": [],
      "notes": "Please call before delivery",
      "created_at": "2025-12-26T11:20:00Z",
      "updated_at": "2025-12-26T12:00:00Z",
      "estimated_delivery": "2025-12-27T15:00:00Z"
    }
  }
}
```

Error Responses:

- `404 Not Found` - Order doesn't exist or doesn't belong to user

Cancel Order

Cancel a pending or confirmed order.

Endpoint: `POST /api/v2/orders/{order_id}?action=cancel`

Authentication: Required (Bearer token)

Request Body (optional):

```
{
  "reason": "Ordered by mistake"
}
```

Success Response: `200 OK`

```
{
  "data": {
    "type": "orders",
    "id": "order_789",
    "attributes": {
      "order_number": "LP-2025-001234",
      "status": "cancelled",
      "cancelled_at": "2025-12-26T12:30:00Z",
      "cancellation_reason": "Ordered by mistake"
    }
  }
}
```

Error Responses:

- `422 Unprocessable Entity` - Order cannot be cancelled (already shipped or delivered)

Track Order

Get real-time tracking information for an order.

Endpoint: `GET /api/v2/orders/{order_id}/tracking`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "order-tracking",
    "attributes": {
      "order_id": "order_789",
      "order_number": "LP-2025-001234",
      "status": "shipped",
      "tracking_number": "TRK123456789",
      "current_location": "Dhaka Distribution Center",
      "estimated_delivery": "2025-12-27T15:00:00Z",
      "delivery_partner": "Pathao",
      "delivery_person": {
        "name": "Karim Delivery",
        "phone": "+8801987654321"
      },
      "timeline": [
        {
          "status": "pending",
          "timestamp": "2025-12-26T11:20:00Z",
          "location": "LifePlus Warehouse",
          "description": "Order placed"
        },
        {
          "status": "confirmed",
          "timestamp": "2025-12-26T11:30:00Z",
          "location": "LifePlus Warehouse",
          "description": "Order confirmed and being prepared"
        },
        {
          "status": "processing",
          "timestamp": "2025-12-26T12:00:00Z",
          "location": "LifePlus Warehouse",
          "description": "Order packaged"
        },
        {
          "status": "shipped",
          "timestamp": "2025-12-26T14:00:00Z",
          "location": "Dhaka Distribution Center",
          "description": "Out for delivery"
        }
      ]
    }
  }
}
```

Upload Prescription

Upload a prescription for an order that requires it.

Endpoint: `POST /api/v2/orders/{order_id}/prescriptions`

Authentication: Required (Bearer token)

Content-Type: `multipart/form-data`

Form Fields:

- `file` - Prescription image or PDF (max 10MB)
- `notes` - Optional notes about the prescription

Success Response: `201 Created`

```
{
  "data": {
    "type": "prescriptions",
    "id": "presc_123",
    "attributes": {
      "order_id": "order_789",
      "file_url": "https://cdn.lifeplusbd.com/prescriptions/presc_123.pdf",
      "status": "pending_review",
      "uploaded_at": "2025-12-26T12:45:00Z",
      "notes": "For chronic back pain"
    }
  }
}
```

Error Responses:

- `400 Bad Request` - Invalid file format (accept: JPEG, PNG, PDF)
- `413 Payload Too Large` - File exceeds 10MB
- `422 Unprocessable Entity` - Order doesn't require prescription

Telemedicine

List Medical Specialties

Get all available medical specialties.

Endpoint: `GET /api/v2/specialties`

Authentication: Required (Bearer token)

Query Parameters:

- `search` - Search by specialty name
- `sort` - Sort by: `name`, `-name`

Success Response: `200 OK`

```
{
  "data": [
    {
      "type": "specialties",
      "id": "spec_1",
      "attributes": {
        "name": "Cardiology",
        "description": "Heart and cardiovascular system specialists",
        "icon_url": "https://cdn.lifeplusbd.com/icons/cardiology.png",
        "doctor_count": 25
      }
    },
    {
      "type": "specialties",
      "id": "spec_2",
      "attributes": {
        "name": "Dermatology",
        "description": "Skin, hair, and nail specialists",
        "icon_url": "https://cdn.lifeplusbd.com/icons/dermatology.png",
        "doctor_count": 18
      }
    }
  ]
}
```

List Doctors

Browse available doctors with filtering options.

Endpoint: `GET /api/v2/doctors`

Authentication: Required (Bearer token)

Query Parameters:

- `filter[specialty_id]` - Filter by specialty
- `filter[available]` - true/false (has slots available today)
- `filter[gender]` - male/female
- `filter[min_experience]` - Minimum years of experience
- `search` - Search by doctor name or qualifications
- `sort` - Sort by: `name`, `rating`, `consultation_fee`, `experience_years`
- `page[number]`, `page[size]` - Pagination

Success Response: `200 OK`

```
{
  "data": [
    {
      "type": "doctors",
      "id": "doc_123",
      "attributes": {
        "name": "Dr. Sarah Ahmed",
        "title": "MBBS, MD (Cardiology)",
        "specialty": {
          "id": "spec_1",
          "name": "Cardiology"
        },
        "qualifications": "MBBS, MD (Cardiology), FRCP",
        "experience_years": 15,
        "gender": "female",
        "languages": ["English", "Bengali"],
        "consultation_fee": 500.00,
        "video_consultation_fee": 400.00,
        "rating": 4.8,
        "total_reviews": 245,
        "total_consultations": 1250,
        "avatar_url": "https://cdn.lifeplusbd.com/doctors/doc_123.jpg",
        "bio": "Dr. Sarah Ahmed is a renowned cardiologist with 15 years of experience...",
        "chamber_address": "United Hospital, Dhaka",
        "available_today": true,
        "next_available_slot": "2025-12-26T15:00:00Z"
      }
    }
  ],
  "meta": {
    "total": 45,
    "page": {
      "number": 1,
      "size": 20
    }
  }
}
```

Get Doctor Details

Get detailed information about a specific doctor.

Endpoint: `GET /api/v2/doctors/{doctor_id}`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "doctors",
    "id": "doc_123",
    "attributes": {
      "name": "Dr. Sarah Ahmed",
      "title": "MBBS, MD (Cardiology)",
      "specialty": {
        "id": "spec_1",
        "name": "Cardiology"
      },
      "qualifications": "MBBS, MD (Cardiology), FRCP",
      "experience_years": 15,
      "gender": "female",
      "languages": ["English", "Bengali"],
      "consultation_fee": 500.00,
      "video_consultation_fee": 400.00,
      "rating": 4.8,
      "total_reviews": 245,
      "total_consultations": 1250,
      "avatar_url": "https://cdn.lifeplusbd.com/doctors/doc_123.jpg",
      "bio": "Dr. Sarah Ahmed is a renowned cardiologist with over 15 years of experience in treating cardiovascular diseases. She completed her MBBS from Dhaka Medical College and obtained her MD in Cardiology from BSMMU. Dr. Ahmed specializes in interventional cardiology and has performed over 500 successful procedures.",
      "education": [
        {
          "degree": "MBBS",
          "institution": "Dhaka Medical College",
          "year": 2005
        },
        {
          "degree": "MD (Cardiology)",
          "institution": "BSMMU",
          "year": 2010
        }
      ],
      "chamber_locations": [
        {
          "name": "United Hospital",
          "address": "Plot 15, Road 71, Gulshan, Dhaka",
          "schedule": "Saturday - Thursday: 3PM - 7PM"
        }
      ],
      "consultation_types": ["in-person", "video"],
      "available_today": true,
      "next_available_slot": "2025-12-26T15:00:00Z"
    }
  }
}
```

Get Doctor Available Slots

Retrieve available appointment slots for a doctor.

Endpoint: `GET /api/v2/doctors/{doctor_id}/slots`

Authentication: Required (Bearer token)

Query Parameters:

- `date` - Date in YYYY-MM-DD format (default: today)
- `type` - Consultation type: `video`, `in-person` (default: all)

Success Response: `200 OK`

```
{
  "data": [
    {
      "type": "doctor-slots",
      "id": "slot_1",
      "attributes": {
        "doctor_id": "doc_123",
        "start_time": "2025-12-26T15:00:00Z",
        "end_time": "2025-12-26T15:30:00Z",
        "consultation_type": "video",
        "available": true,
        "fee": 400.00
      }
    },
    {
      "type": "doctor-slots",
      "id": "slot_2",
      "attributes": {
        "doctor_id": "doc_123",
        "start_time": "2025-12-26T15:30:00Z",
        "end_time": "2025-12-26T16:00:00Z",
        "consultation_type": "video",
        "available": true,
        "fee": 400.00
      }
    },
    {
      "type": "doctor-slots",
      "id": "slot_3",
      "attributes": {
        "doctor_id": "doc_123",
        "start_time": "2025-12-26T16:00:00Z",
        "end_time": "2025-12-26T16:30:00Z",
        "consultation_type": "in-person",
        "available": false,
        "fee": 500.00
      }
    }
  ]
}
```

Book Appointment

Book an appointment with a doctor.

Endpoint: `POST /api/v2/appointments`

Authentication: Required (Bearer token)

Request Body:

```
{
  "doctor_id": "doc_123",
  "slot_id": "slot_1",
  "consultation_type": "video",
  "patient_name": "John Doe",
  "patient_age": 35,
  "patient_gender": "male",
  "patient_phone": "+8801234567890",
  "symptoms": "Chest pain and shortness of breath for the past week",
  "previous_medical_history": "No chronic conditions",
  "current_medications": "None"
}
```

Field Validations:

- `doctor_id` - Required
- `slot_id` - Required, slot must be available
- `consultation_type` - Required, one of: `video`, `in-person`
- `patient_name` - Required
- `patient_age` - Required, 1-120
- `symptoms` - Required, max 1000 characters

Success Response: `201 Created`

```
{
  "data": {
    "type": "appointments",
    "id": "appt_456",
    "attributes": {
      "appointment_number": "APPT-2025-001234",
      "doctor": {
        "id": "doc_123",
        "name": "Dr. Sarah Ahmed",
        "specialty": "Cardiology",
        "avatar_url": "https://cdn.lifeplusbd.com/doctors/doc_123.jpg"
      },
      "patient_name": "John Doe",
      "patient_age": 35,
      "patient_gender": "male",
      "patient_phone": "+8801234567890",
      "consultation_type": "video",
      "scheduled_time": "2025-12-26T15:00:00Z",
      "duration_minutes": 30,
      "status": "scheduled",
      "fee": 400.00,
      "payment_status": "pending",
      "symptoms": "Chest pain and shortness of breath for the past week",
      "created_at": "2025-12-26T13:00:00Z"
    }
  }
}
```

Error Responses:

- `404 Not Found` - Doctor or slot not found
- `422 Unprocessable Entity` - Slot no longer available

List Appointments

Get all appointments for the authenticated user.

Endpoint: `GET /api/v2/appointments`

Authentication: Required (Bearer token)

Query Parameters:

- `filter[status]` - Filter by status: `scheduled`, `completed`, `cancelled`, `no-show`
- `filter[type]` - Filter by type: `video`, `in-person`
- `filter[date_from]` - Start date (ISO 8601)
- `filter[date_to]` - End date (ISO 8601)
- `filter[upcoming]` - true (only future appointments)
- `sort` - Sort by: `scheduled_time`, `-scheduled_time`

- `page[number]`, `page[size]` - Pagination

Success Response: `200 OK`

```
{
  "data": [
    {
      "type": "appointments",
      "id": "appt_456",
      "attributes": {
        "appointment_number": "APPT-2025-001234",
        "doctor": {
          "id": "doc_123",
          "name": "Dr. Sarah Ahmed",
          "specialty": "Cardiology",
          "avatar_url": "https://cdn.lifeplusbd.com/doctors/doc_123.jpg"
        },
        "scheduled_time": "2025-12-26T15:00:00Z",
        "consultation_type": "video",
        "status": "scheduled",
        "fee": 400.00,
        "payment_status": "paid"
      }
    }
  ],
  "meta": {
    "total": 8,
    "page": {
      "number": 1,
      "size": 20
    }
  }
}
```

Get Appointment Details

Get detailed information about a specific appointment.

Endpoint: `GET /api/v2/appointments/{appointment_id}`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "appointments",
    "id": "appt_456",
    "attributes": {
      "appointment_number": "APPT-2025-001234",
      "doctor": {
        "id": "doc_123",
        "name": "Dr. Sarah Ahmed",
        "specialty": "Cardiology",
        "qualifications": "MBBS, MD (Cardiology)",
        "avatar_url": "https://cdn.lifeplusbd.com/doctors/doc_123.jpg"
      },
      "patient_name": "John Doe",
      "patient_age": 35,
      "patient_gender": "male",
      "patient_phone": "+8801234567890",
      "consultation_type": "video",
      "scheduled_time": "2025-12-26T15:00:00Z",
      "duration_minutes": 30,
      "status": "completed",
      "fee": 400.00,
      "payment_status": "paid",
      "payment_method": "bkash",
      "symptoms": "Chest pain and shortness of breath",
      "diagnosis": "Mild angina, stress-related",
      "prescription": {
        "id": "presc_789",
        "medications": [
          {
            "name": "Aspirin 75mg",
            "dosage": "1 tablet daily after breakfast",
            "duration": "30 days"
          }
        ],
        "advice": "Avoid stress, regular exercise, follow-up in 2 weeks",
        "file_url": "https://cdn.lifeplusbd.com/prescriptions/presc_789.pdf"
      },
      "video_call_duration": 25,
      "notes": "Patient advised lifestyle modifications",
      "follow_up_required": true,
      "follow_up_date": "2026-01-10T00:00:00Z",
      "created_at": "2025-12-26T13:00:00Z",
      "completed_at": "2025-12-26T15:28:00Z"
    }
  }
}
```

Cancel Appointment

Cancel a scheduled appointment.

Endpoint: `POST /api/v2/appointments/{appointment_id}?action=cancel`

Authentication: Required (Bearer token)

Request Body (optional):

```
{
  "reason": "Schedule conflict"
}
```

Success Response: `200 OK`

```
{
  "data": {
    "type": "appointments",
    "id": "appt_456",
    "attributes": {
      "appointment_number": "APPT-2025-001234",
      "status": "cancelled",
      "cancelled_at": "2025-12-26T13:30:00Z",
      "cancellation_reason": "Schedule conflict",
      "refund_status": "pending",
      "refund_amount": 400.00
    }
  }
}
```

Error Responses:

- `422 Unprocessable Entity` - Cannot cancel (less than 2 hours before appointment or already completed)

Reschedule Appointment

Reschedule an existing appointment to a new slot.

Endpoint: `POST /api/v2/appointments/{appointment_id}?action=reschedule`

Authentication: Required (Bearer token)

Request Body:

```
{
  "new_slot_id": "slot_5",
  "reason": "Original time not convenient"
}
```

Success Response: `200 OK`

```
{
  "data": {
    "type": "appointments",
    "id": "appt_456",
    "attributes": {
      "appointment_number": "APPT-2025-001234",
      "scheduled_time": "2025-12-27T10:00:00Z",
      "status": "scheduled",
      "rescheduled_at": "2025-12-26T13:40:00Z",
      "reschedule_reason": "Original time not convenient"
    }
  }
}
```

Join Video Consultation

Get video call credentials to join an appointment.

Endpoint: `GET /api/v2/appointments/{appointment_id}/video-call`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "video-calls",
    "attributes": {
      "appointment_id": "appt_456",
      "room_id": "room_abc123xyz",
      "room_name": "Dr. Sarah Ahmed - John Doe",
      "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
      "server_url": "https://video.lifeplusbd.com",
      "expires_at": "2025-12-26T16:00:00Z",
      "can_join": true,
      "join_allowed_from": "2025-12-26T14:50:00Z",
      "scheduled_time": "2025-12-26T15:00:00Z"
    }
  }
}
```

Error Responses:

- `422 Unprocessable Entity` - Cannot join yet (too early) or appointment expired

Get Prescription

Retrieve prescription for a completed appointment.

Endpoint: `GET /api/v2/appointments/{appointment_id}/prescription`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "prescriptions",
    "id": "presc_789",
    "attributes": {
      "appointment_id": "appt_456",
      "doctor": {
        "name": "Dr. Sarah Ahmed",
        "qualifications": "MBBS, MD (Cardiology)"
      },
      "patient_name": "John Doe",
      "diagnosis": "Mild angina, stress-related",
      "medications": [
        {
          "name": "Aspirin 75mg",
          "generic_name": "Acetylsalicylic Acid",
          "dosage": "1 tablet daily after breakfast",
          "duration": "30 days",
          "instructions": "Take with food"
        }
      ],
      "tests_recommended": [
        {
          "name": "Lipid Profile",
          "reason": "Check cholesterol levels"
        }
      ],
      "advice": "Avoid stress, maintain healthy diet, regular exercise 30 min daily",
      "follow_up_required": true,
      "follow_up_date": "2026-01-10T00:00:00Z",
      "file_url": "https://cdn.lifeplusbd.com/prescriptions/presc_789.pdf",
      "issued_at": "2025-12-26T15:28:00Z"
    }
  }
}
```

Error Responses:

- `404 Not Found` - Prescription not yet available (appointment not completed)

List Video Call History

Get history of all video consultations.

Endpoint: `GET /api/v2/video-call-history`

Authentication: Required (Bearer token)

Query Parameters:

- `page[number]`, `page[size]` - Pagination
- `sort` - Sort by: `created_at`, `-created_at`

Success Response: `200 OK`

```
{
  "data": [
    {
      "type": "video-call-history",
      "id": "call_123",
      "attributes": {
        "appointment_id": "appt_456",
        "doctor_name": "Dr. Sarah Ahmed",
        "started_at": "2025-12-26T15:02:00Z",
        "ended_at": "2025-12-26T15:27:00Z",
        "duration_minutes": 25,
        "quality_rating": 5,
        "recording_url": null
      }
    }
  ]
}
```

Diagnostics

List Diagnostic Tests

Browse available diagnostic tests and lab services.

Endpoint: `GET /api/v2/diagnostics`

Authentication: Required (Bearer token)

Query Parameters:

- `filter[category]` - Filter by category: `hematology`, `biochemistry`, `microbiology`, `radiology`, etc.
- `filter[available_for_home_sample]` - true/false
- `search` - Search by test name
- `sort` - Sort by: `name`, `price`, `-price`
- `page[number]`, `page[size]` - Pagination

Success Response: `200 OK`

```
{
  "data": [
    {
      "type": "diagnostics",
      "id": "diag_1",
      "attributes": {
        "name": "Complete Blood Count (CBC)",
        "description": "Comprehensive blood test measuring various components",
        "category": "Hematology",
        "price": 500.00,
        "sample_type": "Blood",
        "preparation_required": "Fasting not required",
        "turnaround_time": "24 hours",
        "available_for_home_sample": true,
        "home_sample_fee": 100.00
      }
    },
    {
      "type": "diagnostics",
      "id": "diag_2",
      "attributes": {
        "name": "Lipid Profile",
        "description": "Cholesterol and triglyceride levels",
        "category": "Biochemistry",
        "price": 800.00,
        "sample_type": "Blood",
        "preparation_required": "12 hours fasting required",
        "turnaround_time": "24 hours",
        "available_for_home_sample": true,
        "home_sample_fee": 100.00
      }
    }
  ],
  "meta": {
    "total": 250,
    "page": {
      "number": 1,
      "size": 20
    }
  }
}
```

Get Diagnostic Test Details

Get detailed information about a specific test.

Endpoint: `GET /api/v2/diagnostics/{diagnostic_id}`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "diagnostics",
    "id": "diag_1",
    "attributes": {
      "name": "Complete Blood Count (CBC)",
      "description": "A CBC measures several components of blood including red blood cells, white blood cells, hemoglobin, hematocrit, and platelets.",
      "category": "Hematology",
      "price": 500.00,
      "sample_type": "Blood",
      "sample_volume": "2-3 ml",
      "preparation_required": "Fasting not required",
      "preparation_instructions": "No special preparation needed. You can eat and drink normally.",
      "turnaround_time": "24 hours",
      "parameters_tested": [
        "Red Blood Cell Count",
        "White Blood Cell Count",
        "Hemoglobin",
        "Hematocrit",
        "Platelet Count"
      ],
      "available_for_home_sample": true,
      "home_sample_fee": 100.00,
      "available_labs": [
        {
          "name": "LifePlus Diagnostics Gulshan",
          "address": "House 45, Road 11, Gulshan, Dhaka"
        }
      ]
    }
  }
}
```

Book Home Sample Collection

Schedule home sample collection for diagnostic tests.

Endpoint: `POST /api/v2/home-sample-orders`

Authentication: Required (Bearer token)

Request Body:

```
{
  "test_ids": ["diag_1", "diag_2", "diag_5"],
  "address_id": "addr_123",
  "preferred_date": "2025-12-27",
  "preferred_time_slot": "morning",
  "patient_name": "John Doe",
  "patient_age": 35,
  "patient_gender": "male",
  "patient_phone": "+8801234567890",
  "notes": "Please call 30 minutes before arrival"
}
```

Field Validations:

- `test_ids` - Required, array of test IDs
- `address_id` - Required
- `preferred_date` - Required, YYYY-MM-DD format, must be today or future
- `preferred_time_slot` - Required, one of: `morning` (8AM-12PM), `afternoon` (12PM-4PM), `evening` (4PM-8PM)
- `patient_name` - Required

Success Response: `201 Created`

```
{
  "data": {
    "type": "home-sample-orders",
    "id": "hso_789",
    "attributes": {
      "order_number": "HSO-2025-001234",
      "tests": [
        {
          "id": "diag_1",
          "name": "Complete Blood Count (CBC)",
          "price": 500.00
        },
        {
          "id": "diag_2",
          "name": "Lipid Profile",
          "price": 800.00
        }
      ],
      "patient": {
        "name": "John Doe",
        "age": 35,
        "gender": "male",
        "phone": "+8801234567890"
      },
      "address": {
        "label": "Home",
        "full_address": "123 Main Street, Apt 4B, Dhaka 1200"
      },
      "preferred_date": "2025-12-27",
      "preferred_time_slot": "morning",
      "scheduled_time": "2025-12-27T09:00:00Z",
      "status": "scheduled",
      "test_fee": 1300.00,
      "home_sample_fee": 100.00,
      "total": 1400.00,
      "payment_status": "pending",
      "created_at": "2025-12-26T14:00:00Z"
    }
  }
}
```

List Home Sample Orders

Get all home sample collection orders.

Endpoint: `GET /api/v2/home-sample-orders`

Authentication: Required (Bearer token)

Query Parameters:

- `filter[status]` - Filter by: `scheduled`, `sample_collected`, `processing`, `completed`, `cancelled`
- `filter[date_from]`, `filter[date_to]` - Date range
- `sort` - Sort by: `scheduled_time`, `-scheduled_time`
- `page[number]`, `page[size]` - Pagination

Success Response: `200 OK`

```
{
  "data": [
    {
      "type": "home-sample-orders",
      "id": "hso_789",
      "attributes": {
        "order_number": "HSO-2025-001234",
        "scheduled_time": "2025-12-27T09:00:00Z",
        "status": "scheduled",
        "total": 1400.00,
        "test_count": 2
      }
    }
  ]
}
```

Get Home Sample Order Details

Get detailed information about a home sample order.

Endpoint: `GET /api/v2/home-sample-orders/{order_id}`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "home-sample-orders",
    "id": "hso_789",
    "attributes": {
      "order_number": "HSO-2025-001234",
      "tests": [
        {
          "id": "diag_1",
          "name": "Complete Blood Count (CBC)",
          "price": 500.00,
          "status": "completed",
          "report_url": "https://cdn.lifeplusbd.com/reports/report_123.pdf"
        }
      ],
      "patient": {
        "name": "John Doe",
        "age": 35,
        "gender": "male"
      },
      "scheduled_time": "2025-12-27T09:00:00Z",
      "sample_collected_at": "2025-12-27T09:15:00Z",
      "status": "processing",
      "phlebotomist": {
        "name": "Ahmed Hassan",
        "phone": "+8801987654321"
      },
      "estimated_report_time": "2025-12-28T09:00:00Z",
      "status_history": [
        {
          "status": "scheduled",
          "timestamp": "2025-12-26T14:00:00Z"
        },
        {
          "status": "sample_collected",
          "timestamp": "2025-12-27T09:15:00Z"
        },
        {
          "status": "processing",
          "timestamp": "2025-12-27T10:00:00Z"
        }
      ]
    }
  }
}
```

Cancel Home Sample Order

Cancel a scheduled home sample collection.

Endpoint: `POST /api/v2/home-sample-orders/{order_id}?action=cancel`

Authentication: Required (Bearer token)

Request Body (optional):

```
{
  "reason": "Need to reschedule"
}
```

Success Response: `200 OK`

Error Responses:

- `422 Unprocessable Entity` - Cannot cancel (sample already collected)

Packages

List Health Packages

Browse available health packages and plans.

Endpoint: `GET /api/v2/packages`

Authentication: Required (Bearer token)

Query Parameters:

- `filter[type]` - Filter by: `health_checkup`, `telemedicine`, `insurance`, `corporate`
- `filter[price_min]`, `filter[price_max]` - Price range
- `sort` - Sort by: `name`, `price`, `-price`, `popularity`
- `page[number]`, `page[size]` - Pagination

Success Response: `200 OK`

```
{
  "data": [
    {
      "type": "packages",
      "id": "pkg_1",
      "attributes": {
        "name": "Basic Health Checkup",
        "description": "Comprehensive annual health screening package",
        "type": "health_checkup",
        "price": 5000.00,
        "validity_days": 365,
        "services": [
          "3 Doctor Consultations (Video/In-person)",
          "10 Diagnostic Tests",
          "1 Annual Health Checkup",
          "24/7 Hotline Support",
          "Health Records Management"
        ],
        "included_tests": [
          "Complete Blood Count",
          "Lipid Profile",
          "Blood Sugar",
          "Liver Function Test"
        ],
        "popular": true,
        "discount_percentage": 20
      }
    }
  ]
}
```

Get Package Details

Get detailed information about a specific package.

Endpoint: `GET /api/v2/packages/{package_id}`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "packages",
    "id": "pkg_1",
    "attributes": {
      "name": "Basic Health Checkup",
      "description": "Our most popular health checkup package designed for annual wellness
screening...",
      "type": "health_checkup",
      "price": 5000.00,
      "original_price": 6250.00,
      "discount_percentage": 20,
      "validity_days": 365,
      "max_beneficiaries": 4,
      "services_included": [
        {
          "category": "Doctor Consultations",
          "quantity": 3,
          "description": "Video or in-person consultations with general physicians"
        },
        {
          "category": "Diagnostic Tests",
          "quantity": 10,
          "description": "Any diagnostic tests up to 10 in a year"
        }
      ],
      "included_tests": [
        "Complete Blood Count (CBC)",
        "Lipid Profile",
        "Blood Sugar (Fasting)",
        "Liver Function Test",
        "Kidney Function Test",
        "Thyroid Profile"
      ],
      "terms_and_conditions": [
        "Package valid for 365 days from purchase",
        "Services must be availed within validity period",
        "Non-refundable after activation"
      ],
      "popular": true,
      "total_sold": 1250
    }
  }
}
```

Book Package

Purchase a health package.

Endpoint: `POST /api/v2/package-orders`

Authentication: Required (Bearer token)

Request Body:

```
{
  "package_id": "pkg_1",
  "payment_method": "bkash",
  "beneficiaries": [
    {
      "name": "John Doe",
      "age": 35,
      "gender": "male",
      "relation": "self",
      "phone": "+8801234567890"
    },
    {
      "name": "Jane Doe",
      "age": 32,
      "gender": "female",
      "relation": "spouse",
      "phone": "+8801234567891"
    }
  ]
}
```

Success Response: `201 Created`

```
{
  "data": {
    "type": "package-orders",
    "id": "pkgord_123",
    "attributes": {
      "order_number": "PKG-2025-001234",
      "package": {
        "id": "pkg_1",
        "name": "Basic Health Checkup"
      },
      "price": 5000.00,
      "payment_method": "bkash",
      "payment_status": "pending",
      "status": "active",
      "validity_starts": "2025-12-26T00:00:00Z",
      "validity_ends": "2026-12-26T23:59:59Z",
      "beneficiaries": [
        {
          "id": "ben_1",
          "name": "John Doe",
          "relation": "self"
        },
        {
          "id": "ben_2",
          "name": "Jane Doe",
          "relation": "spouse"
        }
      ],
      "created_at": "2025-12-26T14:30:00Z"
    }
  }
}
```

List Package Orders

Get all package orders for the authenticated user.

Endpoint: `GET /api/v2/package-orders`

Authentication: Required (Bearer token)

Query Parameters:

- `filter[status]` - Filter by: `active`, `expired`, `cancelled`
- `sort` - Sort by: `created_at`, `-created_at`
- `page[number]`, `page[size]` - Pagination

Success Response: `200 OK`

Get Package Order Details

Get detailed information about a package order.

Endpoint: `GET /api/v2/package-orders/{order_id}`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "package-orders",
    "id": "pkgord_123",
    "attributes": {
      "order_number": "PKG-2025-001234",
      "package": {
        "id": "pkg_1",
        "name": "Basic Health Checkup",
        "description": "Comprehensive health screening"
      },
      "status": "active",
      "validity_starts": "2025-12-26T00:00:00Z",
      "validity_ends": "2026-12-26T23:59:59Z",
      "days_remaining": 365,
      "beneficiaries": [
        {
          "id": "ben_1",
          "name": "John Doe",
          "age": 35,
          "gender": "male",
          "relation": "self"
        }
      ],
      "service_usage": {
        "doctor_consultations": {
          "total": 3,
          "used": 1,
          "remaining": 2
        },
        "diagnostic_tests": {
          "total": 10,
          "used": 3,
          "remaining": 7
        }
      },
      "created_at": "2025-12-26T14:30:00Z"
    }
  }
}
```

Get Package Service Usage

Track service usage for a package order.

Endpoint: `GET /api/v2/package-orders/{order_id}/usage`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "package-usage",
    "attributes": {
      "package_order_id": "pkgord_123",
      "services": [
        {
          "service_type": "doctor_consultation",
          "total_allowed": 3,
          "used": 1,
          "remaining": 2,
          "usage_history": [
            {
              "date": "2025-12-20T10:00:00Z",
              "doctor": "Dr. Sarah Ahmed",
              "type": "video"
            }
          ]
        },
        {
          "service_type": "diagnostic_test",
          "total_allowed": 10,
          "used": 3,
          "remaining": 7,
          "usage_history": [
            {
              "date": "2025-12-15T09:00:00Z",
              "test": "Complete Blood Count"
            }
          ]
        }
      ]
    }
  }
}
```

Corporate Enrollment

Enroll employees in corporate health packages.

Endpoint: `POST /api/v2/package-orders/corporate-enrollment`

Authentication: Required (Bearer token or Partner API key)

Request Body:

```json
{
  "package_id": "pkg_5",
  "company_name": "ABC Corporation",
  "employees": [
    {
      "name": "John Doe",
      "email": "john@abccorp.com",
      "phone": "+8801234567890",
      "employee_id": "EMP001"
    }
  ]
}
```

Success Response: `201 Created`

Ambulance Services

List Ambulance Service Areas

Get available service areas for ambulance booking.

Endpoint: `GET /api/v2/ambulance-areas`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```json
{
  "data": [
    {
      "type": "ambulance-areas",
      "id": "area_1",
      "attributes": {
        "name": "Dhaka Central",
        "coverage_area": "Gulshan, Banani, Mohakhali, Tejgaon",
        "base_fare": 1000.00,
        "per_km_rate": 50.00,
        "available_24_7": true,
        "average_response_time_minutes": 15
      }
    },
    {
      "type": "ambulance-areas",
      "id": "area_2",
      "attributes": {
        "name": "Dhaka South",
        "coverage_area": "Dhanmondi, Lalmatia, Mohammadpur",
        "base_fare": 1000.00,
        "per_km_rate": 50.00,
        "available_24_7": true,
        "average_response_time_minutes": 18
      }
    }
  ]
}
```

Book Ambulance

Request an ambulance for emergency or scheduled transport.

Endpoint: `POST /api/v2/ambulance-orders`

Authentication: Required (Bearer token)

Request Body:

```
{
  "pickup_address": "123 Main Street, Gulshan, Dhaka",
  "pickup_latitude": 23.8103,
  "pickup_longitude": 90.4125,
  "destination_address": "United Hospital, Gulshan, Dhaka",
  "destination_latitude": 23.7465,
  "destination_longitude": 90.3763,
  "patient_name": "John Doe",
  "patient_age": 35,
  "patient_gender": "male",
  "patient_condition": "Emergency - Chest pain, difficulty breathing",
  "contact_phone": "+8801234567890",
  "ambulance_type": "basic",
  "service_type": "emergency",
  "additional_requirements": "Oxygen support needed"
}
```

Field Validations:

- `pickup_address`, `destination_address` - Required
- `pickup_latitude`, `pickup_longitude` - Required for accurate location
- `patient_name`, `patient_condition`, `contact_phone` - Required
- `ambulance_type` - Required, one of: `basic`, `advanced`, `icu`
- `service_type` - Required, one of: `emergency`, `scheduled`, `inter_city`

Success Response: `201 Created`

```
{
  "data": {
    "type": "ambulance-orders",
    "id": "amb_456",
    "attributes": {
      "order_number": "AMB-2025-001234",
      "pickup_address": "123 Main Street, Gulshan, Dhaka",
      "destination_address": "United Hospital, Gulshan, Dhaka",
      "distance_km": 5.2,
      "patient": {
        "name": "John Doe",
        "age": 35,
        "gender": "male",
        "condition": "Emergency - Chest pain, difficulty breathing"
      },
      "contact_phone": "+8801234567890",
      "ambulance_type": "basic",
      "service_type": "emergency",
      "status": "assigned",
      "estimated_arrival_time": "2025-12-26T15:15:00Z",
      "base_fare": 1000.00,
      "distance_fare": 260.00,
      "total_fare": 1260.00,
      "ambulance": {
        "vehicle_number": "DHA-1234",
        "driver_name": "Karim Ahmed",
        "driver_phone": "+8801987654321"
      },
      "created_at": "2025-12-26T15:00:00Z"
    }
  }
}
```

List Ambulance Orders

Get all ambulance booking history.

Endpoint: `GET /api/v2/ambulance-orders`

Authentication: Required (Bearer token)

Query Parameters:

- `filter[status]` - Filter by: `pending`, `assigned`, `en_route`, `completed`, `cancelled`
- `filter[service_type]` - Filter by: `emergency`, `scheduled`, `inter_city`
- `sort` - Sort by: `created_at`, `-created_at`
- `page[number]`, `page[size]` - Pagination

Success Response: `200 OK`

Get Ambulance Order Details

Get detailed information and real-time tracking for an ambulance order.

Endpoint: `GET /api/v2/ambulance-orders/{order_id}`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "ambulance-orders",
    "id": "amb_456",
    "attributes": {
      "order_number": "AMB-2025-001234",
      "pickup_address": "123 Main Street, Gulshan, Dhaka",
      "destination_address": "United Hospital, Gulshan, Dhaka",
      "patient": {
        "name": "John Doe",
        "age": 35,
        "condition": "Emergency - Chest pain"
      },
      "status": "en_route",
      "ambulance": {
        "vehicle_number": "DHA-1234",
        "type": "basic",
        "driver_name": "Karim Ahmed",
        "driver_phone": "+8801987654321",
        "current_location": {
          "latitude": 23.7950,
          "longitude": 90.4050
        }
      },
      "estimated_arrival_time": "2025-12-26T15:15:00Z",
      "total_fare": 1260.00,
      "status_history": [
        {
          "status": "pending",
          "timestamp": "2025-12-26T15:00:00Z"
        },
        {
          "status": "assigned",
          "timestamp": "2025-12-26T15:02:00Z"
        },
        {
          "status": "en_route",
          "timestamp": "2025-12-26T15:05:00Z"
        }
      ],
      "tracking_url": "https://lifeplusbd.com/track/amb_456"
    }
  }
}
```

Track Ambulance

Get real-time tracking for an ambulance.

Endpoint: `GET /api/v2/ambulance-orders/{order_id}/tracking`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "ambulance-tracking",
    "attributes": {
      "order_id": "amb_456",
      "status": "en_route",
      "ambulance_location": {
        "latitude": 23.7950,
        "longitude": 90.4050,
        "updated_at": "2025-12-26T15:08:00Z"
      },
      "pickup_location": {
        "latitude": 23.8103,
        "longitude": 90.4125
      },
      "destination_location": {
        "latitude": 23.7465,
        "longitude": 90.3763
      },
      "estimated_arrival_time": "2025-12-26T15:12:00Z",
      "distance_remaining_km": 2.1,
      "driver": {
        "name": "Karim Ahmed",
        "phone": "+8801987654321"
      }
    }
  }
}
```

Cancel Ambulance Order

Cancel an ambulance booking.

Endpoint: `POST /api/v2/ambulance-orders/{order_id}?action=cancel`

Authentication: Required (Bearer token)

Request Body (optional):

```
{
  "reason": "Patient condition improved"
}
```

Success Response: `200 OK`

Error Responses:

- `422 Unprocessable Entity` - Cannot cancel (ambulance already en route or arrived)

Products & Pharmacy

List Products

Browse medicines and health products.

Endpoint: `GET /api/v2/products`

Authentication: Required (Bearer token)

Query Parameters:

- `filter[category]` - Filter by: `medicine`, `personal_care`, `medical_equipment`, `supplements`, `lifestyle`
- `filter[manufacturer_id]` - Filter by manufacturer
- `filter[requires_prescription]` - true/false

- `filter[in_stock]` - true/false
- `search` - Search by product name or generic name
- `sort` - Sort by: `name`, `price`, `-price`, `popularity`
- `page[number]`, `page[size]` - Pagination

Success Response: `200 OK`

```
{
  "data": [
    {
      "type": "products",
      "id": "prod_123",
      "attributes": {
        "name": "Paracetamol 500mg",
        "generic_name": "Acetaminophen",
        "brand": "Napa",
        "manufacturer": {
          "id": "mfr_1",
          "name": "Square Pharmaceuticals"
        },
        "category": "medicine",
        "strength": "500mg",
        "dosage_form": "Tablet",
        "package_size": "10 tablets",
        "price": 2.50,
        "mrp": 3.00,
        "discount_percentage": 17,
        "in_stock": true,
        "stock_quantity": 500,
        "requires_prescription": false,
        "image_url": "https://cdn.lifeplusbd.com/products/prod_123.jpg",
        "description": "Pain reliever and fever reducer"
      }
    }
  ],
  "meta": {
    "total": 2500,
    "page": {
      "number": 1,
      "size": 20
    }
  }
}
```

Get Product Details

Get detailed information about a specific product.

Endpoint: `GET /api/v2/products/{product_id}`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": {
    "type": "products",
    "id": "prod_123",
    "attributes": {
      "name": "Paracetamol 500mg",
      "generic_name": "Acetaminophen",
      "brand": "Napa",
      "manufacturer": {
        "id": "mfr_1",
        "name": "Square Pharmaceuticals Ltd.",
        "country": "Bangladesh"
      },
      "category": "medicine",
      "strength": "500mg",
      "dosage_form": "Tablet",
      "package_size": "10 tablets per strip",
      "price": 2.50,
      "mrp": 3.00,
      "discount_percentage": 17,
      "in_stock": true,
      "stock_quantity": 500,
      "requires_prescription": false,
      "images": [
        "https://cdn.lifeplusbd.com/products/prod_123_1.jpg",
        "https://cdn.lifeplusbd.com/products/prod_123_2.jpg"
      ],
      "description": "Napa is used to relieve mild to moderate pain from headaches, muscle aches, menstrual periods, colds and sore throats, toothaches, backaches, and reactions to vaccinations, and to reduce fever.",
      "indications": [
        "Pain relief",
        "Fever reduction",
        "Headache",
        "Toothache"
      ],
      "dosage_instructions": "Adults: 1-2 tablets every 4-6 hours as needed. Do not exceed 8 tablets in 24 hours.",
      "side_effects": [
        "Rare: Allergic reactions",
        "Nausea (rare)",
        "Liver damage with overdose"
      ],
      "contraindications": [
        "Severe liver disease",
        "Allergy to paracetamol"
      ],
      "storage_instructions": "Store below 30°C in a dry place away from light",
      "active_ingredients": "Paracetamol 500mg",
      "rating": 4.5,
      "review_count": 1250,
      "total_sold": 15000
    }
  }
}
```

List Product Categories

Get all product categories.

Endpoint: `GET /api/v2/product-categories`

Authentication: Required (Bearer token)

Success Response: `200 OK`

```
{
  "data": [
    {
      "type": "product-categories",
      "id": "cat_1",
      "attributes": {
        "name": "Medicine",
        "slug": "medicine",
        "icon_url": "https://cdn.lifeplusbd.com/icons/medicine.png",
        "product_count": 2500
      }
    },
    {
      "type": "product-categories",
      "id": "cat_2",
      "attributes": {
        "name": "Personal Care",
        "slug": "personal_care",
        "icon_url": "https://cdn.lifeplusbd.com/icons/personal_care.png",
        "product_count": 850
      }
    }
  ]
}
```

List Manufacturers

Get all product manufacturers.

Endpoint: `GET /api/v2/manufacturers`

Authentication: Required (Bearer token)

Query Parameters:

- `search` - Search by manufacturer name
- `sort` - Sort by: `name`, `product_count`
- `page[number]`, `page[size]` - Pagination

Success Response: `200 OK`

```
{
  "data": [
    {
      "type": "manufacturers",
      "id": "mfr_1",
      "attributes": {
        "name": "Square Pharmaceuticals Ltd.",
        "country": "Bangladesh",
        "logo_url": "https://cdn.lifeplusbd.com/manufacturers/mfr_1.png",
        "product_count": 450,
        "established_year": 1958
      }
    }
  ]
}
```

Partner APIs

These endpoints are for B2B partner integrations using API key authentication.

Validate API Key

Validate partner API key and check permissions.

Endpoint: `POST /api/v2/partner/api-keys?action=validate`

Authentication: Partner API Key

Headers:

```
X-API-Key: sk_live_abc123xyz789
X-Partner-ID: partner_123
```

Success Response: `200 OK`

```
{
  "data": {
    "type": "api-key-validations",
    "attributes": {
      "valid": true,
      "partner_id": "partner_123",
      "partner_name": "ABC Healthcare Services",
      "permissions": [
        "orders:read",
        "orders:create",
        "orders:update",
        "products:read",
        "diagnostics:read"
      ],
      "rate_limit": {
        "requests_per_hour": 1000,
        "burst_limit": 100
      },
      "expires_at": "2026-12-31T23:59:59Z",
      "environment": "production"
    }
  }
}
```

Create Order (Partner)

Create an order on behalf of a customer.

Endpoint: `POST /api/v2/partner/orders`

Authentication: Partner API Key

Request Body:

```
{
  "external_order_id": "partner_order_12345",
  "customer": {
    "name": "John Doe",
    "phone": "+8801234567890",
    "email": "john@example.com",
    "address": {
      "line_1": "123 Main Street",
      "line_2": "Apt 4B",
      "city": "Dhaka",
      "postal_code": "1200",
      "country": "Bangladesh"
    }
  },
  "items": [
    {
      "product_id": "prod_123",
      "quantity": 2,
      "unit_price": 50.00
    },
    {
      "product_id": "prod_456",
      "quantity": 1,
      "unit_price": 100.00
    }
  ],
  "subtotal": 200.00,
  "delivery_fee": 50.00,
  "total": 250.00,
  "payment_method": "partner_account",
  "notes": "Deliver before 5 PM",
  "webhook_url": "https://partner.com/webhooks/lifeplus"
}
```

Success Response: `201 Created`

```
{
  "data": {
    "type": "orders",
    "id": "order_789",
    "attributes": {
      "order_number": "LP-2025-001234",
      "external_order_id": "partner_order_12345",
      "status": "pending",
      "customer": {
        "name": "John Doe",
        "phone": "+8801234567890"
      },
      "total": 250.00,
      "payment_status": "pending",
      "created_at": "2025-12-26T16:00:00Z",
      "estimated_delivery": "2025-12-27T17:00:00Z"
    }
  }
}
```

Get Order Status (Partner)

Check order status and tracking information.

Endpoint: `GET /api/v2/partner/orders/{order_id}`

Authentication: Partner API Key

Success Response: `200 OK`

```
{
  "data": {
    "type": "orders",
    "id": "order_789",
    "attributes": {
      "order_number": "LP-2025-001234",
      "external_order_id": "partner_order_12345",
      "status": "processing",
      "tracking_number": "TRK123456",
      "payment_status": "paid",
      "status_history": [
        {
          "status": "pending",
          "timestamp": "2025-12-26T16:00:00Z"
        },
        {
          "status": "confirmed",
          "timestamp": "2025-12-26T16:15:00Z"
        },
        {
          "status": "processing",
          "timestamp": "2025-12-26T17:00:00Z"
        }
      ],
      "estimated_delivery": "2025-12-27T17:00:00Z"
    }
  }
}
```

Update Order Status (Partner)

Update order status (for partners with fulfillment permissions).

Endpoint: `PATCH /api/v2/partner/orders/{order_id}/status`

Authentication: Partner API Key (requires `orders:update` permission)

Request Body:

```
{
  "status": "delivered",
  "delivered_at": "2025-12-27T16:30:00Z",
  "notes": "Delivered successfully to customer",
  "delivery_proof_url": "https://partner.com/proofs/delivery_123.jpg"
}
```

Success Response: `200 OK`

List Products (Partner)

Get product catalog for partner integrations.

Endpoint: `GET /api/v2/partner/products`

Authentication: Partner API Key

Query Parameters: Same as regular product listing

Success Response: `200 OK` (includes wholesale pricing if applicable)

Webhook Configuration

Partners can configure webhooks to receive real-time updates.

Event Types:

- `order.created` - New order created
- `order.confirmed` - Order confirmed
- `order.processing` - Order being processed
- `order.shipped` - Order shipped
- `order.delivered` - Order delivered
- `order.cancelled` - Order cancelled
- `payment.completed` - Payment received
- `payment.failed` - Payment failed

Webhook Payload Example:

```json
{
  "event": "order.delivered",
  "event_id": "evt_abc123",
  "timestamp": "2025-12-27T16:30:00Z",
  "data": {
    "order_id": "order_789",
    "order_number": "LP-2025-001234",
    "external_order_id": "partner_order_12345",
    "status": "delivered",
    "delivered_at": "2025-12-27T16:30:00Z"
  }
}
```

Webhook Signature Verification: All webhooks include `X-LifePlus-Signature` header for verification:

```javascript
const crypto = require('crypto');

function verifyWebhook(payload, signature, secret) {
  const hmac = crypto.createHmac('sha256', secret);
  const digest = hmac.update(payload).digest('hex');
  return crypto.timingSafeEqual(
    Buffer.from(signature),
    Buffer.from(digest)
  );
}
```

Architecture & Implementation

This section describes the technical architecture and implementation details of the LifePlus Healthcare Platform API.

System Architecture Overview

The LifePlus API follows a **microservices-oriented architecture** with clear separation of concerns across multiple layers.

Architecture Layers

1. **Microservices Layer** - Domain-specific services
2. **Service Layer** - Business logic and orchestration
3. **Package Layer** - Reusable components
4. **Plugin Layer** - External integrations

Microservices Architecture

The platform is organized into independent microservices, each owning its domain logic and data.

Auth Microservice

Purpose: User authentication, authorization, and session management

Responsibilities:

- User registration and login
- JWT token generation and validation
- Password reset and OTP verification
- Role-based access control (RBAC)

Technology:

- Laravel JWT (tymon/jwt-auth)
- Redis for session storage
- SMS gateway for OTP delivery

User Microservice

Purpose: User profile and beneficiary management

Responsibilities:

- User profile CRUD operations
- Profile image upload
- Beneficiary (dependent) management
- User address management

Database Tables:

- `users`
- `patients` (beneficiaries)
- `user_addresses`

Order Microservice

Purpose: Unified order processing for all order types

Responsibilities:

- Order creation and validation
- Payment integration
- Order status tracking
- Invoice generation
- Vendor order distribution

Order Types Handled:

1. **Product Orders** - Medicine, Lifestyle products
2. **Appointment Orders** - Doctor consultations
3. **Diagnostic Orders** - Lab tests
4. **Package Orders** - Healthcare subscriptions
5. **Ambulance Orders** - Emergency transport
6. **Home Care Orders** - Nursing services
7. **Home Sample Orders** - Sample collection

8. **Wellbeing Orders** - Medical tourism
9. **Policy Orders** - Insurance policies

Key Design Decision: All order types share a unified `orders` table with polymorphic relationships, enabling consistent order tracking and processing across all services.

Product Microservice

Purpose: Product catalog and inventory management

Responsibilities:

- Product catalog (medicines, lifestyle products)
- Category management
- Search and filtering
- Stock management
- Pricing and discounts

Database Tables:

- `products`
- `product_variants`
- `product_vendors`
- `categories`, `sub_categories`
- `brands`

Cart Microservice

Purpose: Shopping cart management

Responsibilities:

- Add/remove items from cart
- Cart quantity updates
- Cart persistence
- Multi-vendor cart splitting

Payment Microservice

Purpose: Payment processing and gateway integration

Integrated Services:

- **SSL Commerz** - Bangladesh payment gateway
- **bKash, Nagad, Rocket** - Mobile banking

Responsibilities:

- Payment gateway integration
- Payment validation
- Refund processing
- Transaction tracking

Notification Microservice

Purpose: Multi-channel notification delivery

Channels:

- Push notifications (Firebase Cloud Messaging)
- SMS notifications
- Email notifications
- In-app notifications

Integration Layer

Payment Gateway Integration

SSL Commerz Plugin Location: `app/Library/SslCommerz/`

Components:

- `AbstractSslCommerz` - Base class
- `SslCommerzInterface` - Contract
- `SslCommerzNotification` - Webhook handler

Usage Example:

```php
$ssl = new SslCommerz();
$ssl->setAmount($amount);
$ssl->setCurrency('BDT');
$paymentUrl = $ssl->initiate();
```

Video Consultation Integration

Agora.io Plugin Location: `app/Library/Agoraio/`

Purpose: Real-time video consultation for telemedicine

Responsibilities:

- Generate Agora tokens for video calls
- Manage video call channels
- Handle token expiry and refresh

SDK Architecture

SDK Structure

All official SDKs follow a consistent structure:

```
sdk/
├── src/
│   ├── Client                    # Main client class
│   ├── Configuration             # Configuration
│   ├── api/
│   │   ├── OrdersApi
│   │   ├── ProductsApi
│   │   ├── AppointmentsApi
│   │   └── ...
│   ├── models/
│   │   ├── Order
│   │   ├── Product
│   │   ├── Appointment
│   │   └── ...
│   └── utils/
│       ├── HttpClient
│       ├── AuthHandler
│       └── ErrorHandler
├── tests/
├── docs/
└── README.md
```

SDK Features (Full Implementation, Not Stubs)

✅ Complete Implementation Includes:

1. **HTTP Client** - Complete HTTP request/response handling
2. **Authentication** - Automatic JWT token management
3. **Error Handling** - Typed exceptions for different error codes
4. **Retry Logic** - Automatic retry with exponential backoff
5. **Request Validation** - Client-side validation before API calls
6. **Response Parsing** - Automatic JSON parsing to typed models
7. **Pagination Helpers** - Easy pagination iteration
8. **File Upload** - Multipart form-data support
9. **Logging** - Optional request/response logging
10. **Type Safety** - Full type definitions (TypeScript) or annotations (PHP)

❌ NOT Stub Implementation:

- No empty method bodies
- No placeholder comments
- No "TODO: Implement this"
- Full error handling included
- Complete model serialization/deserialization

Deployment Architecture

Production Environment

```
┌─────────────────────────────────────┐
│    Load Balancer (DigitalOcean)    │
│         NGINX Load Balancer          │
└──────────────┬──────────────────────┘
               │
    ┌──────────┴─────────┐
    │                    │
┌───▼────┐        ┌──────▼───┐
│ Web1   │        │  Web2    │
│Laravel │        │ Laravel  │
└───┬────┘        └──────┬───┘
    │                    │
    └─────────┬──────────┘
              │
    ┌─────────▼───────┐
    │ Redis Cluster   │
    │ Session/Cache   │
    └─────────┬───────┘
              │
    ┌─────────▼───────┐
    │   MySQL RDS     │
    │   (Multi-AZ)    │
    └─────────┬───────┘
              │
    ┌─────────▼───────┐
    │   S3 Bucket     │
    │  File Storage   │
    └─────────────────┘
```

Components:

- **Load Balancer** - Distributes traffic across web servers
- **Web Servers** - Laravel application (horizontal scaling)
- **Redis** - Session storage and caching

- **MySQL** - Primary database with multi-AZ replication
- **S3** - Static file storage (images, documents, prescriptions)

Future Microservices Deployment

When transitioning to true distributed microservices:

```
┌─────────────────────────────────────┐
│    API Gateway (Kong/AWS Gateway)   │
└──────────────┬──────────────────────┘
               │
    ┌──────────┴──────────────┐
    │                          │
┌───▼────┐  ┌──────▼─────┐  ┌─▼──────┐
│Auth Svc│  │Order Svc   │  │User Svc│
│Node.js │  │    Go      │  │  PHP   │
└───┬────┘  └──────┬─────┘  └─┬──────┘
    │              │           │
    └──────┬───────┴───────────┘
           │
    ┌──────▼──────┐
    │Message Queue│
    │  RabbitMQ   │
    └─────────────┘
```

Security Architecture

Authentication Flow

1. User sends credentials to `POST /api/v2/sessions`
2. Server validates credentials
3. Server generates JWT token (expiry: 24 hours)
4. Client stores token securely
5. Client includes token in `Authorization: Bearer {token}` header
6. Server validates token on each request

Authorization

- **Role-Based Access Control (RBAC)** - Users have roles: customer, admin, partner
- **Resource Ownership** - Users can only access their own resources
- **Partner API Keys** - Partners use separate API keys with limited scope

Data Privacy & Security

- **PII Encryption** - Sensitive data encrypted at rest
- **HTTPS Only** - All API calls require TLS 1.2+
- **Input Sanitization** - All inputs validated and sanitized
- **Output Encoding** - Responses properly encoded
- **HIPAA Considerations** - Healthcare data handled per compliance requirements

---

Performance Optimization

Caching Strategy

- **Redis Caching**
  - Session data (5-15 minute TTL)
  - Frequently accessed data (product catalogs, categories)
  - API response caching for GET endpoints

- **CDN Caching**
  - Static assets (images, documents)
  - Public API responses (with proper cache headers)
- **Database Query Caching**
  - Complex query results cached
  - Cache invalidation on data updates

Database Optimization

- **Indexes** - All foreign keys and frequently queried columns indexed
- **Query Optimization** - N+1 queries eliminated using eager loading
- **Connection Pooling** - Database connection reuse
- **Read Replicas** - Separate read/write operations for scaling

Rate Limiting

- **Per IP** - 60 requests/minute (unauthenticated)
- **Per User** - 120 requests/minute (authenticated)
- **Partner API** - Custom limits based on agreement
- **Burst Protection** - 100 requests in 10 seconds

Rate Limit Headers:

```
X-RateLimit-Limit: 120
X-RateLimit-Remaining: 95
X-RateLimit-Reset: 1640534400
```

Monitoring & Observability

Key Metrics

- **Request Rate** - Requests per second
- **Error Rate** - 4xx and 5xx responses
- **Latency** - p50, p95, p99 response times
- **Database Performance** - Query execution time
- **Cache Hit Rate** - Redis cache effectiveness
- **Queue Depth** - Background job backlog

Logging Strategy

- **Application Logs** - Laravel logs (storage/logs)
- **Access Logs** - Nginx/Apache access logs
- **Error Tracking** - Sentry for exception tracking
- **Audit Logs** - User action tracking for compliance

Alerting Rules

- **High Error Rate** - Alert if > 5% of responses are 5xx
- **Slow Responses** - Alert if p95 latency > 2 seconds
- **Database Issues** - Alert on connection pool exhaustion
- **Service Downtime** - Alert on health check failures
- **Security Events** - Alert on suspicious patterns

Data Model Schema

Core Order Schema Example

```
Order:
  type: object
  required:
    - id
    - user_id
    - order_type
    - status
    - total_amount
  properties:
    id:
      type: integer
      example: 12345
    user_id:
      type: integer
    order_type:
      type: string
      enum: [product, appointment, diagnostic, package, ambulance, home_care, home_sample, wellbeing, policy]
    status:
      type: string
      enum: [pending, confirmed, processing, shipped, delivered, cancelled, refunded]
    items:
      type: array
      items:
        $ref: '#/components/schemas/OrderItem'
    total_amount:
      type: number
      format: float
    payment_method:
      type: string
      enum: [cod, online, bkash, nagad, card, wallet]
    payment_status:
      type: string
      enum: [unpaid, pending, paid, refunded]
    created_at:
      type: string
      format: date-time
    updated_at:
      type: string
      format: date-time
```

Migration Guide

This guide helps developers migrate from the legacy API to API v2. Follow these steps to update your integration.

Migration Steps

Step 1: Update Base URL

Change all API calls from `/api/*` to `/api/v2/*`:

```
Legacy: https://api.lifeplusbd.com/api/login
V2:     https://api.lifeplusbd.com/api/v2/sessions
```

Step 2: Update Authentication

Switch to standard Bearer token authentication:

```
// Before
headers: { 'X-Auth-Token': token }
```

```
// After
headers: { 'Authorization': `Bearer ${token}` }
```

Step 3: Update HTTP Methods

Use proper REST methods instead of POST for everything:

```
Legacy: POST /api/orders/cancel/123
V2:     POST /api/v2/orders/123?action=cancel

Legacy: POST /api/user/update
V2:     PATCH /api/v2/users/me
```

Step 4: Update Response Parsing

Handle JSON:API response format:

```
// Before
const userId = response.data.id;

// After
const userId = response.data.id;
const userName = response.data.attributes.name;
```

Step 5: Update Error Handling

Handle new error format:

```
// Before
if (response.error) { console.error(response.error); }

// After
if (response.errors) {
  response.errors.forEach(err => {
    console.error(`${err.title}: ${err.detail}`);
  });
}
```

Step 6: Test Thoroughly

- Test all API endpoints in staging environment
- Verify authentication flows work correctly
- Test error scenarios and edge cases
- Validate data format changes

Breaking Changes

1. Authentication

Legacy:

```
// Custom header
headers: {
  'X-Auth-Token': token
}
```

V2:

```
// Standard Bearer token
headers: {
  'Authorization': `Bearer ${token}`
}
```

2. Response Format

Legacy:

```
{
  "success": true,
  "data": { "id": 123, "name": "John" },
  "message": "Success"
}
```

V2 (JSON:API):

```
{
  "data": {
    "type": "users",
    "id": "123",
    "attributes": {
      "name": "John"
    }
  }
}
```

3. Error Format

Legacy:

```
{
  "success": false,
  "error": "Invalid credentials"
}
```

V2:

```
{
  "errors": [
    {
      "status": "401",
      "code": "INVALID_CREDENTIALS",
      "title": "Authentication Failed",
      "detail": "Invalid credentials provided"
    }
  ]
}
```

4. Pagination

Legacy:

```
GET /api/orders?page=1&per_page=20
```

V2:

```
GET /api/v2/orders?page[number]=1&page[size]=20
```

5. Filtering

Legacy:

```
GET /api/orders/by-status?status=pending
```

V2:

```
GET /api/v2/orders?filter[status]=pending
```

6. Actions

Legacy:

```
POST /api/orders/cancel/123
```

V2:

```
POST /api/v2/orders/123?action=cancel
```

Code Examples

JavaScript/TypeScript Migration

Legacy Code:

```
// Login
const response = await fetch('https://api.lifeplusbd.com/api/login', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    phone: '+8801234567890',
    otp: '123456'
  })
});

const data = await response.json();
const token = data.data.token;

// Get orders
const ordersResponse = await fetch('https://api.lifeplusbd.com/api/orders', {
  headers: {
    'X-Auth-Token': token
  }
});

const orders = await ordersResponse.json();
```

V2 Code:

```
// Login
const response = await fetch('https://api.lifeplusbd.com/api/v2/sessions', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    phone: '+8801234567890',
    otp: '123456'
  })
});

const data = await response.json();
const token = data.data.attributes.token;
```

```
// Get orders
const ordersResponse = await fetch('https://api.lifeplusbd.com/api/v2/orders', {
  headers: {
    'Authorization': `Bearer ${token}`
  }
});

const ordersData = await ordersResponse.json();
const orders = ordersData.data; // Array of order objects
```

Best Practice - Use SDK:

```
import { LifePlusClient } from '@lifeplus/sdk';

const client = new LifePlusClient({
  baseUrl: 'https://api.lifeplusbd.com'
});

// Login
const session = await client.auth.createSession({
  phone: '+8801234567890',
  otp: '123456'
});

// Get orders (automatically handles pagination, filtering, etc.)
const orders = await client.orders.list({
  filter: { status: 'pending' },
  page: { number: 1, size: 20 }
});
```

PHP Migration

Legacy Code:

```
<?php
// Login
$response = Http::post('https://api.lifeplusbd.com/api/login', [
    'phone' => '+8801234567890',
    'otp' => '123456'
]);

$token = $response['data']['token'];

// Get orders
$orders = Http::withHeaders([
    'X-Auth-Token' => $token
])->get('https://api.lifeplusbd.com/api/orders');
```

V2 Code:

```
<?php
use LifePlus\SDK\Client;

$client = new Client([
    'base_url' => 'https://api.lifeplusbd.com'
]);

// Login
$session = $client->auth->createSession([
    'phone' => '+8801234567890',
    'otp' => '123456'
]);

$token = $session->attributes->token;

// Get orders
$orders = $client->orders->list([
    'filter' => ['status' => 'pending'],
    'page' => ['number' => 1, 'size' => 20]
]);
```

Go Migration

Legacy Code:

```go
// Login
reqBody := map[string]string{
    "phone": "+8801234567890",
    "otp": "123456",
}
resp, _ := http.Post("https://api.lifeplusbd.com/api/login", "application/json", body)

var loginData map[string]interface{}
json.NewDecoder(resp.Body).Decode(&loginData)
token := loginData["data"].(map[string]interface{})["token"].(string)

// Get orders
req, _ := http.NewRequest("GET", "https://api.lifeplusbd.com/api/orders", nil)
req.Header.Set("X-Auth-Token", token)
```

V2 Code:

```go
import "github.com/LifeplusBangladesh/lifeplus-go-sdk"

client := lifeplus.NewClient(&lifeplus.Config{
    BaseURL: "https://api.lifeplusbd.com",
})

// Login
session, err := client.Auth.CreateSession(context.Background(), &lifeplus.SessionRequest{
    Phone: "+8801234567890",
    OTP: "123456",
})

// Get orders
orders, err := client.Orders.List(context.Background(), &lifeplus.OrderListParams{
    Filter: lifeplus.OrderFilter{
        Status: "pending",
    },
    Page: lifeplus.PageParams{
        Number: 1,
        Size: 20,
    },
})
```

Common Migration Patterns

Pattern 1: List Resources

```
Legacy: GET /api/products
V2:     GET /api/v2/products
```

Pattern 2: Get Resource by ID

```
Legacy: GET /api/products/{id}
V2:     GET /api/v2/products/{id}
```

Pattern 3: Create Resource

```
Legacy: POST /api/products/create
V2:     POST /api/v2/products
```

Pattern 4: Update Resource

```
Legacy: POST /api/products/update/{id}
V2:     PATCH /api/v2/products/{id}
```

Pattern 5: Delete Resource

```
Legacy: POST /api/products/delete/{id}
V2:     DELETE /api/v2/products/{id}
```

Pattern 6: Resource Actions

```
Legacy: POST /api/orders/cancel/{id}
V2:     POST /api/v2/orders/{id}?action=cancel
```

Pattern 7: Filter by Status

```
Legacy: GET /api/orders/by-status?status=pending
V2:     GET /api/v2/orders?filter[status]=pending
```

Pattern 8: Search

```
Legacy: GET /api/products/search?query=paracetamol
V2:     GET /api/v2/products?search=paracetamol
```

Partner Integration Guide

This guide is for B2B partners integrating with the LifePlus API.

Getting Started

1. **Contact LifePlus** to get your Partner credentials:
   - Partner ID
   - API Key
   - API Secret
2. **Review the API documentation** and understand the endpoints you'll need
3. **Set up your development environment** with test credentials
4. **Implement authentication** using API keys
5. **Test your integration** in the staging environment
6. **Go live** after approval from LifePlus team

Authentication

Partners use API Key authentication instead of Bearer tokens.

Headers Required:

```
X-API-Key: your-api-key-here
X-Partner-ID: partner_123
Content-Type: application/json
```

Validate Your API Key:

```
POST /api/v2/partner/api-keys?action=validate
X-API-Key: your-api-key-here
X-Partner-ID: partner_123
```

Response:

```
{
  "data": {
    "type": "api-key-validations",
    "attributes": {
      "valid": true,
      "partner_id": "partner_123",
      "partner_name": "ABC Healthcare",
      "permissions": [
        "orders:read",
        "orders:create",
        "orders:update",
        "products:read"
      ],
      "rate_limit": 1000,
      "expires_at": "2026-12-31T23:59:59Z"
    }
  }
}
```

Partner Endpoints

1. Create Order (Partner)

Create an order on behalf of a customer.

```
POST /api/v2/partner/orders
X-API-Key: your-api-key-here
X-Partner-ID: partner_123
Content-Type: application/json

{
  "external_order_id": "partner_order_12345",
  "customer": {
    "name": "John Doe",
    "phone": "+8801234567890",
    "email": "john@example.com",
    "address": {
      "line_1": "123 Main Street",
      "line_2": "Apt 4B",
      "city": "Dhaka",
      "postal_code": "1200"
    }
  },
  "items": [
    {
      "product_id": "prod_456",
      "quantity": 2,
      "unit_price": 50.00
    },
    {
      "product_id": "prod_789",
      "quantity": 1,
      "unit_price": 100.00
    }
  ],
  "delivery_fee": 50.00,
  "notes": "Please deliver before 5 PM"
}
```

Response:

```
{
  "data": {
    "type": "orders",
    "id": "order_123",
    "attributes": {
      "order_number": "LP-2025-001234",
      "external_order_id": "partner_order_12345",
      "status": "pending",
      "subtotal": 200.00,
      "delivery_fee": 50.00,
      "total": 250.00,
      "created_at": "2025-12-26T03:42:00Z"
    }
  }
}
```

2. Get Order Status

Check the status of an order.

```
GET /api/v2/partner/orders/{order_id}
X-API-Key: your-api-key-here
X-Partner-ID: partner_123
```

Response:

```
{
  "data": {
    "type": "orders",
    "id": "order_123",
    "attributes": {
      "order_number": "LP-2025-001234",
      "external_order_id": "partner_order_12345",
      "status": "processing",
      "tracking_number": "TRK123456",
      "estimated_delivery": "2025-12-27T15:00:00Z",
      "status_history": [
        {
          "status": "pending",
          "timestamp": "2025-12-26T03:42:00Z"
        },
        {
          "status": "confirmed",
          "timestamp": "2025-12-26T04:00:00Z"
        },
        {
          "status": "processing",
          "timestamp": "2025-12-26T05:00:00Z"
        }
      ]
    }
  }
}
```

3. Update Order Status

Update the status of an order (if you have permission).

```
PATCH /api/v2/partner/orders/{order_id}/status
X-API-Key: your-api-key-here
X-Partner-ID: partner_123
Content-Type: application/json

{
  "status": "delivered",
  "delivered_at": "2025-12-27T14:30:00Z",
  "notes": "Delivered successfully to customer"
}
```

4. List Products

Get the product catalog.

```
GET /api/v2/partner/products?page[size]=100
X-API-Key: your-api-key-here
X-Partner-ID: partner_123
```

5. Check Product Availability

```
GET /api/v2/partner/products/{product_id}/availability
X-API-Key: your-api-key-here
X-Partner-ID: partner_123
```

Response:

```
{
  "data": {
    "type": "product-availability",
    "attributes": {
      "product_id": "prod_456",
      "in_stock": true,
      "quantity_available": 150,
      "estimated_restock_date": null
    }
  }
}
```

Webhooks

LifePlus can send webhook notifications for order status updates.

Webhook Event Types:

- `order.created` - New order created
- `order.confirmed` - Order confirmed
- `order.processing` - Order is being processed
- `order.shipped` - Order has been shipped
- `order.delivered` - Order delivered
- `order.cancelled` - Order cancelled

Webhook Payload:

```
{
  "event": "order.delivered",
  "timestamp": "2025-12-27T14:30:00Z",
  "data": {
    "order_id": "order_123",
    "external_order_id": "partner_order_12345",
    "status": "delivered",
    "delivered_at": "2025-12-27T14:30:00Z"
  }
}
```

Webhook Signature Verification: All webhooks include an `X-LifePlus-Signature` header that you should verify:

```
const crypto = require('crypto');

function verifyWebhook(payload, signature, secret) {
  const hmac = crypto.createHmac('sha256', secret);
  const digest = hmac.update(payload).digest('hex');
  return crypto.timingSafeEqual(
    Buffer.from(signature),
    Buffer.from(digest)
  );
}
```

Rate Limits

- **Default:** 1000 requests per hour
- **Burst:** 100 requests per minute
- **Headers:** Response includes rate limit info

```
X-RateLimit-Limit: 1000
X-RateLimit-Remaining: 950
X-RateLimit-Reset: 1640534400
```

Error Handling

Common Error Codes:

- `401` - Invalid API key or unauthorized
- `403` - Forbidden - insufficient permissions
- `429` - Rate limit exceeded
- `500` - Internal server error

Example Error Response:

```
{
  "errors": [
    {
      "status": "401",
      "code": "INVALID_API_KEY",
      "title": "Authentication Failed",
      "detail": "The provided API key is invalid or expired"
    }
  ]
}
```

Best Practices

1. **Always validate your API key** before starting operations
2. **Implement retry logic** with exponential backoff for failed requests
3. **Use webhooks** instead of polling for status updates
4. **Cache product data** to reduce API calls
5. **Log all requests and responses** for debugging
6. **Monitor rate limits** and adjust request frequency
7. **Use batch operations** when available
8. **Implement proper error handling** for all API calls
9. **Keep credentials secure** - never commit to version control
10. **Test thoroughly** in staging before going to production

Support

For partner integration support:

- **Email:** mamun@lifeplusbd.com
- **Phone:** +880 1913705269
- **Secondary:** sagor@lifeplusbd.com
- **Phone:** +880 1681408185

SDKs & Code Examples

Official SDKs

LifePlus provides official SDKs for multiple languages:

JavaScript/TypeScript

```
npm install @lifeplus/sdk
```

```
import { LifePlusClient } from '@lifeplus/sdk';

const client = new LifePlusClient({
  baseUrl: 'https://api.lifeplusbd.com',
  apiKey: 'your-api-key' // For partner integrations
});

// Authenticate
const session = await client.auth.createSession({
  phone: '+8801234567890',
  otp: '123456'
});

// Set token for subsequent requests
client.setToken(session.data.attributes.token);

// List products
const products = await client.products.list({
  search: 'paracetamol',
  page: { size: 20 }
});

// Create order
const order = await client.orders.create({
  cart_id: 'cart_123',
  address_id: 'addr_456'
});
```

Go

```
go get github.com/LifeplusBangladesh/lifeplus-go-sdk
```

```
import "github.com/LifeplusBangladesh/lifeplus-go-sdk"

client := lifeplus.NewClient(&lifeplus.Config{
    BaseURL: "https://api.lifeplusbd.com",
})

// Authenticate
session, err := client.Auth.CreateSession(ctx, &lifeplus.SessionRequest{
    Phone: "+8801234567890",
    OTP: "123456",
})

// Set token
client.SetToken(session.Data.Attributes.Token)

// List products
products, err := client.Products.List(ctx, &lifeplus.ProductListParams{
    Search: "paracetamol",
    Page: lifeplus.PageParams{Size: 20},
})

// Create order
order, err := client.Orders.Create(ctx, &lifeplus.OrderCreation{
    CartID: "cart_123",
    AddressID: "addr_456",
})
```

PHP

```
composer require lifeplus/php-sdk
```

```php
<?php
use LifePlus\SDK\Client;

$client = new Client([
    'base_url' => 'https://api.lifeplusbd.com'
]);

// Authenticate
$session = $client->auth->createSession([
    'phone' => '+8801234567890',
    'otp' => '123456'
]);

// Set token
$client->setToken($session->attributes->token);

// List products
$products = $client->products->list([
    'search' => 'paracetamol',
    'page' => ['size' => 20]
]);

// Create order
$order = $client->orders->create([
    'cart_id' => 'cart_123',
    'address_id' => 'addr_456'
]);
```

Error Handling

Error Response Format

All errors follow the JSON:API error specification:

```json
{
  "errors": [
    {
      "id": "error_unique_id",
      "status": "400",
      "code": "VALIDATION_ERROR",
      "title": "Validation Failed",
      "detail": "The phone number format is invalid",
      "source": {
        "pointer": "/data/attributes/phone"
      },
      "meta": {
        "timestamp": "2025-12-26T03:42:00Z"
      }
    }
  ]
}
```

Common Error Codes

Status	Code	Description
400	VALIDATION_ERROR	Request validation failed
401	UNAUTHORIZED	Authentication required
401	INVALID_CREDENTIALS	Invalid phone/OTP combination
401	TOKEN_EXPIRED	Authentication token has expired
401	INVALID_API_KEY	Partner API key is invalid
403	FORBIDDEN	Insufficient permissions
404	NOT_FOUND	Resource not found

Status	Code	Description
409	CONFLICT	Resource conflict (e.g., duplicate)
422	UNPROCESSABLE_ENTITY	Request semantically invalid
429	RATE_LIMIT_EXCEEDED	Too many requests
500	INTERNAL_ERROR	Internal server error
503	SERVICE_UNAVAILABLE	Service temporarily unavailable

Error Handling Examples

JavaScript/TypeScript

```
try {
  const order = await client.orders.create(orderData);
} catch (error) {
  if (error.status === 401) {
    // Token expired, refresh authentication
    await refreshAuth();
  } else if (error.status === 400) {
    // Validation error
    error.errors.forEach(err => {
      console.error(`${err.title}: ${err.detail}`);
      console.error(`Field: ${err.source?.pointer}`);
    });
  } else if (error.status === 429) {
    // Rate limit exceeded, implement backoff
    await wait(error.headers['Retry-After'] * 1000);
    // Retry request
  } else {
    // Handle other errors
    console.error('Error:', error);
  }
}
```

Go

```
order, err := client.Orders.Create(ctx, orderData)
if err != nil {
    if apiErr, ok := err.(*lifeplus.APIError); ok {
        switch apiErr.Status {
        case 401:
            // Handle authentication error
            return refreshAuth()
        case 400:
            // Handle validation errors
            for _, e := range apiErr.Errors {
                log.Printf("%s: %s (field: %s)", e.Title, e.Detail, e.Source.Pointer)
            }
        case 429:
            // Handle rate limit
            time.Sleep(time.Duration(apiErr.RetryAfter) * time.Second)
            // Retry
        default:
            log.Printf("API error: %v", apiErr)
        }
    }
    return err
}
```

PHP

```
<?php
try {
    $order = $client->orders->create($orderData);
} catch (LifePlus\SDK\ValidationException $e) {
    // Validation errors
    foreach ($e->getErrors() as $error) {
        echo "{$error->title}: {$error->detail}\n";
        echo "Field: {$error->source->pointer}\n";
    }
} catch (LifePlus\SDK\AuthenticationException $e) {
    // Authentication error
    refreshAuth();
} catch (LifePlus\SDK\RateLimitException $e) {
    // Rate limit
    sleep($e->getRetryAfter());
    // Retry
} catch (LifePlus\SDK\APIException $e) {
    // Other API errors
    echo "Error: " . $e->getMessage();
}
```

Best Practices

1. Authentication & Security

✅ DO:

- Use HTTPS for all API requests
- Store tokens securely (e.g., secure storage, environment variables)
- Implement token refresh logic
- Use Bearer token authentication
- Validate API key before operations (partners)
- Keep API keys secret and rotate regularly

❌ DON'T:

- Hardcode tokens or API keys in source code
- Share tokens between users
- Store tokens in localStorage (web apps - use httpOnly cookies)
- Commit credentials to version control
- Use HTTP instead of HTTPS

2. API Design & Usage

✅ DO:

- Use proper HTTP methods (GET, POST, PATCH, DELETE)
- Use nouns in URLs (`/api/v2/orders`, not `/api/v2/getOrders`)
- Use query parameters for actions (`?action=cancel`)
- Use filters for querying (`?filter[status]=pending`)
- Implement pagination for list endpoints
- Handle errors gracefully with retry logic

❌ DON'T:

- Use verbs in endpoint URLs
- Use POST for everything
- Fetch all records without pagination
- Ignore error responses
- Make unnecessary API calls

3. Performance Optimization

✅ DO:

- Cache frequently accessed data (products, categories)
- Use pagination to limit response sizes
- Implement request throttling/debouncing
- Use webhooks instead of polling
- Batch operations when possible
- Use CDN for static assets

❌ **DON'T:**

- Fetch entire product catalog repeatedly
- Poll for status updates every second
- Make redundant API calls
- Ignore rate limits
- Load all data at once

4. Error Handling

✅ **DO:**

- Implement proper error handling for all requests
- Log errors with context for debugging
- Show user-friendly error messages
- Implement retry logic with exponential backoff
- Handle network failures gracefully
- Monitor error rates

❌ **DON'T:**

- Ignore error responses
- Show technical errors to end users
- Retry immediately without backoff
- Assume requests always succeed
- Fail silently

5. Data Validation

✅ **DO:**

- Validate input on client side before sending
- Handle validation errors from API
- Use proper data types
- Format phone numbers correctly (+880...)
- Validate required fields

❌ **DON'T:**

- Send invalid data to API
- Ignore validation errors
- Use incorrect data types
- Skip client-side validation

6. Testing

✅ **DO:**

- Test in staging environment first
- Test error scenarios
- Test with invalid data

- Test rate limiting behavior
- Test authentication flows
- Use mock servers for development

❌ **DON'T:**

- Test directly in production
- Test only happy paths
- Skip edge cases
- Ignore rate limits during testing

7. Monitoring & Logging

✅ **DO:**

- Log all API requests and responses
- Monitor API response times
- Track error rates
- Set up alerts for failures
- Monitor rate limit usage
- Keep audit trail

❌ **DON'T:**

- Log sensitive data (passwords, tokens)
- Ignore performance degradation
- Operate without monitoring
- Skip logging

Version History

v2.0.0 (December 2025)

- Complete API redesign following REST principles
- JSON:API specification for responses
- Proper HTTP method usage (GET, POST, PATCH, DELETE)
- Resource-based URLs with nouns only
- Standardized authentication (Bearer tokens)
- Comprehensive error handling
- Official SDKs (Go, PHP, TypeScript)
- Partner/B2B APIs
- Webhook support
- OpenAPI 3.0 specification

Breaking Changes from Legacy API

- URL structure changed (`/api/*` → `/api/v2/*`)
- Authentication header changed (custom → Bearer)
- Response format changed (custom → JSON:API)
- HTTP methods changed (mostly POST → proper REST)
- Error format standardized
- Pagination format changed
- All endpoints now require authentication

Support & Resources

Documentation

- **API Reference:** This document
- **OpenAPI Spec:** `/api/v2/openapi.json`
- **Swagger UI:** https://api.lifeplusbd.com/docs

SDKs

- **Go:** https://github.com/LifeplusBangladesh/lifeplus-go-sdk
- **PHP:** https://packagist.org/packages/lifeplus/php-sdk
- **JavaScript/TypeScript:** https://www.npmjs.com/package/@lifeplus/sdk

Support Contacts

- **Primary:** mamun@lifeplusbd.com | +880 1913705269
- **Secondary:** sagor@lifeplusbd.com | +880 1681408185

Legacy Resources

- **Legacy Backend:** https://github.com/LifeplusBangladesh/life-plus-laravel
- **Legacy Docs:** Available in archived documentation

Appendix

HTTP Status Codes Reference

Code	Name	Usage
200	OK	Successful GET, PATCH, DELETE
201	Created	Successful POST (resource created)
204	No Content	Successful DELETE (no response body)
400	Bad Request	Invalid request format/validation error
401	Unauthorized	Authentication required or failed
403	Forbidden	Authenticated but insufficient permissions
404	Not Found	Resource doesn't exist
409	Conflict	Resource conflict (duplicate)
422	Unprocessable Entity	Semantic validation error
429	Too Many Requests	Rate limit exceeded
500	Internal Server Error	Server error
503	Service Unavailable	Service temporarily down

Data Type Reference

Type	Format	Example
Phone	E.164	+8801234567890
Date	ISO 8601	2025-12-26
DateTime	ISO 8601	2025-12-26T03:42:00Z
Currency	Decimal	99.99
Boolean	true/false	true
UUID	UUID v4	550e8400-e29b-41d4-a716-446655440000

Reserved Query Parameters

Parameter	Purpose	Example
`page[number]`	Page number	`?page[number]=1`
`page[size]`	Items per page	`?page[size]=20`
`filter[field]`	Filter by field	`?filter[status]=active`
`search`	Full-text search	`?search=keyword`
`sort`	Sort field	`?sort=-created_at`
`include`	Include relations	`?include=user,items`
`action`	Resource action	`?action=cancel`

End of API V2 Complete Reference

This document consolidates all API documentation for LifePlus Healthcare Platform API v2.0