

Product Management

Author: Josh S. Sakweli, Backend Lead Team

Last Updated: 2026-05-19

Version: v2.1

Short Description: The Product Management API provides comprehensive functionality for managing products within shops on the NextGate platform. It supports group buying, installment payment plans, color variations, specifications, digital product file delivery, product preview media (video, PDF, 3D, image), and comprehensive search/filter capabilities with role-based access control.

Hints:

- All shop-related endpoints use the prefix `api/v1/e-commerce/shops/{shopId}/products`
- Shop owners and system admins have full product management access
- Public users can only view active products from approved, active shops
- Products have a `productType`: `PHYSICAL` (shipping + delivery confirmation) or `DIGITAL` (instant download after payment)
- Action parameter (`SAVE_DRAFT` / `SAVE_PUBLISH`) determines product status on create/update
- Search supports multi-word queries with enhanced pattern matching across multiple fields
- Pagination is 1-indexed (page parameter starts from 1)
- Products use soft-delete (marked as deleted rather than permanently removed, except drafts)
- Product slugs are unique within each shop scope
- SKUs are auto-generated using shop, category, brand, and sequence information
- Group buying requires a maximum participant count, a group price, and a time limit
- Installment plans are managed separately via the Installment Plan Config endpoints
- Color variations can have individual price adjustments and separate image sets
- Digital products require uploading files via the Digital File Management endpoints before buyers can download
- Products can have a single preview (`previewType` + `previewUrl`) uploaded via the Product Preview endpoints — visible publicly before purchase, stored in a public bucket separate from private digital content
- Preview supports: `VIDEO`, `PDF`, `THREE_D`, `IMAGE`. Null means no preview. Managed via presign → upload → confirm flow

Endpoints

1. Create Product

Purpose: Creates a new product in a shop, supporting group buying, color variations, specifications, and digital product rules.

Endpoint: **POST** `api/v1/e-commerce/shops/{shopId}/products`

Access Level: Protected (Requires shop owner or system admin role)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token
Content-Type	string	Yes	application/json

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	ID of the shop

Query Parameters:

Parameter	Type	Required	Default	Description
action	ReqAction	Yes	—	<code>SAVE_DRAFT</code> or <code>SAVE_PUBLISH</code>

Request Body Parameters:

Parameter	Type	Required	Description	Validation
productType	ProductType	Yes	<code>PHYSICAL</code> or <code>DIGITAL</code>	Required
productName	string	Yes	Unique name within shop	Min: 2, Max: 100 chars
productDescription	string	Yes	Detailed description	Min: 10, Max: 1000 chars
price	decimal	Yes	Selling price	Min: 0.01, max 8 digits + 2 decimal places
stockQuantity	integer	Yes	Available stock	Min: 0
categoryId	UUID	Yes	Product category	Must exist and be active

Parameter	Type	Required	Description	Validation
comparePrice	decimal	No	Original price for discount display	Must be > price if provided
lowStockThreshold	integer	No	Low stock alert threshold	Min: 1, Max: 1000, Default: 5
condition	ProductCondition	No	Product condition	NEW, USED_LIKE_NEW, USED_GOOD, USED_FAIR, REFURBISHED, FOR_PARTS
status	ProductStatus	No	Initial status (overridden by action)	Default: ACTIVE
productImages	array	Yes	Product image URLs	Valid URLs, at least 1 required
specifications	object	No	Key-value specs	Key max 100 chars, Value max 500 chars
colors	array	No	Color variations	See Color object below
minOrderQuantity	integer	No	Minimum order qty	Min: 1, Default: 1
maxOrderQuantity	integer	No	Maximum order qty per order	Min: 1, must be ≥ minOrderQuantity
groupBuyingEnabled	boolean	No	Enable group buying	Default: false
groupMaxSize	integer	No	Maximum group participants	Min: 2, required if groupBuyingEnabled
groupPrice	decimal	No	Discounted group price	Must be < price
groupTimeLimitHours	integer	No	Group formation time limit	Min: 1, Max: 8760
downloadExpiryDays	integer	No	Download link expiry (DIGITAL only)	Min: 1, Default: 7
maxDownloadsPerBuyer	integer	No	Download attempts per buyer (DIGITAL only)	Min: 1
maxQuantityForDigital	integer	No	Purchase cap per buyer (DIGITAL only)	Min: 1

Color Object:

Field	Type	Required	Validation
name	string	Yes	Max: 50 chars
hex	string	Yes	Valid #RRGGBB format

Field	Type	Required	Validation
images	array	No	Valid URLs
priceAdjustment	decimal	No	Min: 0.0, Default: 0

Request JSON Sample (PHYSICAL):

```
{
  "productType": "PHYSICAL",
  "productName": "iPhone 15 Pro Max 256GB",
  "productDescription": "The most advanced iPhone featuring the A17 Pro chip and titanium design.",
  "price": 1199.00,
  "comparePrice": 1299.00,
  "stockQuantity": 25,
  "lowStockThreshold": 5,
  "categoryId": "123e4567-e89b-12d3-a456-426614174000",
  "condition": "NEW",
  "productImages": [
    "https://example.com/images/iphone15-main.jpg"
  ],
  "specifications": {
    "Display": "6.7-inch Super Retina XDR OLED",
    "Chip": "A17 Pro"
  },
  "colors": [
    {
      "name": "Natural Titanium",
      "hex": "#F5F5DC",
      "images": ["https://example.com/colors/natural-titanium.jpg"],
      "priceAdjustment": 0.00
    }
  ],
  "minOrderQuantity": 1,
  "maxOrderQuantity": 3,
  "groupBuyingEnabled": true,
  "groupMaxSize": 50,
  "groupPrice": 1099.00,
  "groupTimeLimitHours": 72
}
```

Request JSON Sample (DIGITAL):

```
{
  "productType": "DIGITAL",
  "productName": "UI Design Kit Pro",
  "productDescription": "A comprehensive Figma component library with 500+ components.",
  "price": 49.00,
  "stockQuantity": 1000,
  "categoryId": "123e4567-e89b-12d3-a456-426614174000",
  "productImages": ["https://example.com/images/design-kit-preview.jpg"],
  "downloadExpiryDays": 30,
  "maxDownloadsPerBuyer": 5,
  "maxQuantityForDigital": 1
}
```

Response JSON Sample:

```
{
  "success": true,
  "message": "Product created successfully",
  "data": null
}
```

Business Rules:

- Product name must be unique within the shop
- `comparePrice` must be greater than `price`
- `groupPrice` must be less than `price`
- All group buying settings (`groupMaxSize`, `groupPrice`, `groupTimeLimitHours`) required when `groupBuyingEnabled=true`
- `maxOrderQuantity` must be \geq `minOrderQuantity`
- Digital download fields (`downloadExpiryDays`, `maxDownloadsPerBuyer`, `maxQuantityForDigital`) only apply to `DIGITAL` products
- After creation, add installment plans via [Installment Plan Config](#) and digital files via [Digital File Management](#)

Error Responses:

- `400`: Validation errors or business rule violations
- `401`: Authentication required
- `403`: Insufficient permissions
- `404`: Shop or category not found
- `409`: Product with same name already exists in shop

- `422`: Field-level validation errors

2. Update Product

Purpose: Updates an existing product. Only provided fields are updated.

Endpoint: `PUT` `api/v1/e-commerce/shops/{shopId}/products/{productId}`

Access Level: `🔒` Protected (Requires shop owner or system admin role)

Authentication: Bearer Token

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	ID of the shop
productId	UUID	Yes	ID of the product

Query Parameters:

Parameter	Type	Required	Default	Description
action	ReqAction	Yes	—	<code>SAVE_DRAFT</code> or <code>SAVE_PUBLISH</code>

Request Body Parameters (all optional):

Parameter	Type	Description	Validation
productName	string	Updated name	Min: 2, Max: 100 chars
productDescription	string	Updated description	Min: 10, Max: 1000 chars
price	decimal	Updated selling price	Min: 0.01
comparePrice	decimal	Updated compare price	Must be > price
stockQuantity	integer	Updated stock	Min: 0
lowStockThreshold	integer	Updated low stock threshold	Min: 1, Max: 1000
condition	ProductCondition	Updated condition	See enum values
status	ProductStatus	Updated status	Overridden by <code>action</code>
urgencyTag	UrgencyTag	Urgency badge on product	<code>NONE</code> , <code>NEW_ARRIVAL</code> , <code>LIMITED_EDITION</code> , <code>LIMITED_OFFER</code> , <code>FEW_REMAINS</code>

Parameter	Type	Description	Validation
categoryId	UUID	Updated category	Must exist and be active
productImages	array	Updated image URLs	Valid URLs, replaces existing
specifications	object	Updated specifications	Completely replaces existing
colors	array	Updated color variations	Completely replaces existing
minOrderQuantity	integer	Updated min order qty	Min: 1
maxOrderQuantity	integer	Updated max order qty	Min: 1, must be \geq min
groupBuyingEnabled	boolean	Enable/disable group buying	—
groupMaxSize	integer	Updated group max	Min: 2
groupPrice	decimal	Updated group price	Must be $<$ price
groupTimeLimitHours	integer	Updated group time limit	Min: 1, Max: 8760
installmentEnabled	boolean	Enable/disable installment feature toggle	—
maxQuantityForInstallment	integer	Max qty a buyer can purchase on installment	Min: 1
showStockAvailableToPublic	boolean	Show available stock count publicly	—
showSoldCountToPublic	boolean	Show sold count publicly	—
clearPreview	boolean	Set <code>true</code> to remove the product's preview file and type	—
previewDownloadable	boolean	Allow/disallow viewers from downloading the preview file	—

Response JSON Sample:

```
{
  "success": true,
  "message": "Product updated successfully and published",
  "data": {
    "productId": "456e7890-e89b-12d3-a456-426614174001",
    "productName": "iPhone 15 Pro Max 512GB",
    "productSlug": "iphone-15-pro-max-512gb",
    "price": 1399.00,
    "status": "ACTIVE",
```

```
"updatedAt": "2026-05-19T14:30:00Z"
}
}
```

Error Responses:

- `400`: Invalid update data or validation errors
- `401`: Authentication required
- `403`: Insufficient permissions
- `404`: Shop or product not found
- `409`: Updated product name already exists

3. Publish Product

Purpose: Publishes a draft product making it active and publicly available.

Endpoint: `PATCH` `api/v1/e-commerce/shops/{shopId}/products/{productId}/publish`

Access Level: Protected (Requires shop owner or system admin role)

Authentication: Bearer Token

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	ID of the shop
productId	UUID	Yes	ID of the draft product

Response JSON Sample:

```
{
  "success": true,
  "message": "Product 'iPhone 15 Pro Max' published successfully",
  "data": {
    "productId": "456e7890-e89b-12d3-a456-426614174001",
    "productName": "iPhone 15 Pro Max",
    "status": "ACTIVE",
    "publishedAt": "2026-05-19T14:45:00Z"
  }
}
```

Publishing Requirements:

- Product name, description, price, stock quantity, category, and at least one image must be present
- Group buying settings complete if enabled
- Installment plans present if installment is enabled

Error Responses:

- 400: Product already published or missing required publish fields
- 401: Authentication required
- 403: Insufficient permissions
- 404: Shop or product not found

4. Delete Product

Purpose: Deletes a product. Draft products are hard-deleted; published products are soft-deleted.

Endpoint: DELETE `api/v1/e-commerce/shops/{shopId}/products/{productId}`

Access Level: Protected (Requires shop owner or system admin role)

Authentication: Bearer Token

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	ID of the shop
productId	UUID	Yes	ID of the product

Response JSON Sample (Soft Delete):

```
{
  "success": true,
  "message": "Product 'iPhone 15 Pro Max' has been deleted and will be permanently removed after 30 days",
  "data": {
    "productName": "iPhone 15 Pro Max",
    "productId": "456e7890-e89b-12d3-a456-426614174001",
    "previousStatus": "ACTIVE",
    "deletedAt": "2026-05-19T15:00:00Z",
  }
}
```

```
    "deletionType": "SOFT_DELETE"
  }
}
```

Response JSON Sample (Hard Delete):

```
{
  "success": true,
  "message": "Draft product 'iPhone 15 Pro Max' has been permanently deleted",
  "data": null
}
```

Deletion Logic:

- **Draft products:** Hard delete (permanently removed)
- **Published products:** Soft delete (status → ARCHIVED, 30-day recovery window)

Error Responses:

- **401:** Authentication required
- **403:** Insufficient permissions
- **404:** Shop or product not found

5. Restore Product

Purpose: Restores a soft-deleted product back to draft status.

Endpoint: **PATCH** `api/v1/e-commerce/shops/{shopId}/products/{productId}/restore`

Access Level: Protected (Requires shop owner or system admin role)

Authentication: Bearer Token

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	ID of the shop
productId	UUID	Yes	ID of the soft-deleted product

Response JSON Sample:

```
{
  "success": true,
  "message": "Product 'iPhone 15 Pro Max' has been restored successfully",
  "data": {
    "productId": "456e7890-e89b-12d3-a456-426614174001",
    "productName": "iPhone 15 Pro Max",
    "status": "DRAFT",
    "restoredAt": "2026-05-19T15:30:00Z",
    "note": "Product restored as draft. Publish to make it active again."
  }
}
```

Error Responses:

- 400: Product is not deleted
- 401: Authentication required
- 403: Insufficient permissions
- 404: Shop or product not found

6. Get Product Detailed (Owner/Admin View)

Purpose: Retrieves comprehensive product details including all management information.

Endpoint: `GET` `api/v1/e-commerce/shops/{shopId}/products/{productId}/detailed`

Access Level: Protected (Requires shop owner or system admin role)

Authentication: Bearer Token

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	ID of the shop
productId	UUID	Yes	ID of the product

Response JSON Sample:

```
{
  "success": true,
  "message": "Product details retrieved successfully",
  "data": {
    "productId": "456e7890-e89b-12d3-a456-426614174001",
    "productName": "iPhone 15 Pro Max 512GB",
    "productSlug": "iphone-15-pro-max-512gb",
    "productType": "PHYSICAL",
    "productDescription": "The most advanced iPhone ever...",
    "productImages": ["https://example.com/images/iphone15-main.jpg"],
    "price": 1199.00,
    "comparePrice": 1299.00,
    "discountAmount": 100.00,
    "discountPercentage": 7.69,
    "isOnSale": true,
    "stockQuantity": 25,
    "isInStock": true,
    "isLowStock": false,
    "sku": "SHP12345678-ELE-APP-512-0001",
    "condition": "NEW",
    "status": "ACTIVE",
    "urgencyTag": "NONE",
    "shopId": "123e4567-e89b-12d3-a456-426614174000",
    "shopName": "TechStore Pro",
    "categoryId": "789e0123-e89b-12d3-a456-426614174002",
    "categoryName": "Smartphones",
    "specifications": {
      "Display": "6.7-inch Super Retina XDR OLED",
      "Chip": "A17 Pro"
    },
  },
  "colors": [
    {
      "name": "Natural Titanium",
      "hex": "#F5F5DC",
      "images": ["https://example.com/colors/natural-titanium.jpg"],
      "priceAdjustment": 0.00,
      "finalPrice": 1199.00
    }
  ],
  "groupBuying": {
```

```
    "isEnabled": true,
    "groupMaxSize": 50,
    "groupPrice": 1099.00,
    "timeLimitHours": 72
  },
  "installmentOptions": {
    "isEnabled": true,
    "plans": []
  },
  "previewType": "VIDEO",
  "previewUrl": "https://minio.example.com/nextgate-preview-content/preview/shop-id/product-
id/uuid_trailer.mp4",
  "previewDownloadable": false,
  "createdAt": "2026-05-19T10:30:00Z",
  "updatedAt": "2026-05-19T14:30:00Z"
}
}
```

Error Responses:

- `401`: Authentication required
- `403`: Insufficient permissions
- `404`: Shop or product not found

7. Get Shop Products (Management View)

Purpose: Retrieves all products for a shop with summary statistics.

Endpoint: `GET` `api/v1/e-commerce/shops/{shopId}/products/all`

Access Level: `🔒` Protected (Requires shop owner or system admin role)

Authentication: Bearer Token

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	ID of the shop

Response JSON Sample:

```
{
  "success": true,
  "message": "Retrieved 47 products from shop: TechStore Pro",
  "data": {
    "shop": {
      "shopId": "123e4567-e89b-12d3-a456-426614174000",
      "shopName": "TechStore Pro",
      "isVerified": true,
      "isMyShop": true
    },
    "summary": {
      "totalProducts": 47,
      "activeProducts": 35,
      "draftProducts": 8,
      "outOfStockProducts": 4,
      "lowStockProducts": 6,
      "productsWithGroupBuying": 18,
      "productsWithInstallments": 25
    },
    "products": [
      {
        "productId": "456e7890-e89b-12d3-a456-426614174001",
        "productName": "iPhone 15 Pro Max 512GB",
        "price": 1199.00,
        "stockQuantity": 25,
        "status": "ACTIVE",
        "isInStock": true,
        "hasGroupBuying": true,
        "hasInstallments": true,
        "createdAt": "2026-05-19T10:30:00Z"
      }
    ],
    "totalProducts": 47
  }
}
```

Error Responses:

- `401`: Authentication required

- `403`: Insufficient permissions
- `404`: Shop not found

8. Get Shop Products Paginated (Management View)

Purpose: Retrieves shop products with pagination for management dashboard.

Endpoint: `GET` `api/v1/e-commerce/shops/{shopId}/products/all-paged`

Access Level: Protected (Requires shop owner or system admin role)

Authentication: Bearer Token

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	ID of the shop

Query Parameters:

Parameter	Type	Required	Default	Description
page	integer	No	1	Page number (1-indexed)
size	integer	No	10	Items per page (max: 100)

Response JSON Sample:

```
{
  "success": true,
  "message": "Retrieved 10 products from shop: TechStore Pro (Page 1 of 5)",
  "data": {
    "contents": { "...same structure as /all..." },
    "currentPage": 1,
    "pageSize": 10,
    "totalElements": 47,
    "totalPages": 5,
    "hasNext": true,
    "hasPrevious": false
  }
}
```

```
}  
}
```

Error Responses:

- 401: Authentication required
- 403: Insufficient permissions
- 404: Shop not found

9. Get Public Product by ID

Purpose: Retrieves a single active product for public viewing.

Endpoint: `GET` `api/v1/e-commerce/shops/{shopId}/products/{productId}`

Access Level: Public

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	Shop must be active and approved
productId	UUID	Yes	Product must be active

Response JSON Sample:

```
{  
  "success": true,  
  "message": "Product retrieved successfully",  
  "data": {  
    "productId": "456e7890-e89b-12d3-a456-426614174001",  
    "productName": "iPhone 15 Pro Max 512GB",  
    "productSlug": "iphone-15-pro-max-512gb",  
    "productType": "PHYSICAL",  
    "productDescription": "The most advanced iPhone ever...",  
    "price": 1199.00,  
    "comparePrice": 1299.00,  
    "discountAmount": 100.00,  
    "discountPercentage": 7.69,  
    "isOnSale": true,  
  }  
}
```

```
"isInStock": true,
"stockQuantity": 25,
"condition": "NEW",
"shopName": "TechStore Pro",
"categoryName": "Smartphones",
"specifications": { "Display": "6.7-inch OLED" },
"colors": [
  {
    "name": "Natural Titanium",
    "hex": "#F5F5DC",
    "priceAdjustment": 0.00,
    "finalPrice": 1199.00
  }
],
"groupBuying": {
  "isAvailable": true,
  "groupMaxSize": 50,
  "groupPrice": 1099.00,
  "timeLimitHours": 72
},
"installmentOptions": {
  "isAvailable": true,
  "plans": [
    {
      "planId": "...",
      "planName": "6-Month Interest-Free",
      "paymentFrequency": "MONTHLY",
      "numberOfPayments": 6,
      "apr": 0.00,
      "minDownPaymentPercent": 20
    }
  ]
},
"previewType": "PDF",
"previewUrl": "https://minio.example.com/nextgate-preview-content/preview/shop-id/product-
id/uuid_sample.pdf",
"previewDownloadable": true,
"createdAt": "2026-05-19T10:30:00Z"
}
}
```

Notes:

- `previewType` and `previewUrl` are `null` when no preview has been uploaded
- Preview files are publicly accessible without authentication — the URL is permanent and direct

Error Responses:

- `404`: Shop not found/not approved, or product not found/not active

10. Get Public Shop Products

Purpose: Retrieves all active products from a shop for public browsing.

Endpoint: `GET` `api/v1/e-commerce/shops/{shopId}/products/public-view/all`

Access Level: Public

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	Shop must be active and approved

Response JSON Sample:

```
{
  "success": true,
  "message": "Retrieved 23 products from TechStore Pro",
  "data": {
    "shop": {
      "shopId": "123e4567-e89b-12d3-a456-426614174000",
      "shopName": "TechStore Pro",
      "isVerified": true
    },
    "products": [
      {
        "productId": "456e7890-e89b-12d3-a456-426614174001",
        "productName": "iPhone 15 Pro Max",
        "price": 1199.00,
        "isOnSale": true,

```

```
    "isInStock": true,
    "hasGroupBuying": true,
    "hasInstallments": true
  }
],
"totalProducts": 23
}
```

Error Responses:

- `404`: Shop not found, not approved, or not active

11. Get Public Shop Products Paginated

Purpose: Retrieves active products from a shop with pagination for public browsing.

Endpoint: `GET` `api/v1/e-commerce/shops/{shopId}/products/public-view/all-paged`

Access Level: `Public`

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	Shop must be active and approved

Query Parameters:

Parameter	Type	Required	Default	Description
page	integer	No	1	Page number (1-indexed)
size	integer	No	10	Items per page (max: 50)

Error Responses:

- `404`: Shop not found, not approved, or not active

12. Search Products

Purpose: Searches products within a shop using multi-word query matching across multiple fields.

Endpoint: `GET` `api/v1/e-commerce/shops/{shopId}/products/search`

Access Level: `Public` (Enhanced features for authenticated users)

Authentication: Optional Bearer Token

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	ID of the shop

Query Parameters:

Parameter	Type	Required	Default	Description
q	string	Yes	—	Search query (min: 2, max: 100 chars)
status	ProductStatus[]	No	ACTIVE	Statuses to search (owners/admins only for non-ACTIVE)
page	integer	No	1	Page number
size	integer	No	10	Items per page (max: 50)
sortBy	string	No	relevance	relevance, createdAt, updatedAt, productName, price, stockQuantity, brand
sortDir	string	No	desc	asc or desc

Search Behavior:

Feature	Description
Multi-word	Searches for products containing ALL words
Partial match	"iph" matches "iPhone"
Cross-field	Matches against name, description, brand, tags, specifications
Case-insensitive	"APPLE" matches "apple"

User Access:

User Type	Searchable Statuses
Public / Authenticated	ACTIVE only
Shop Owner / Admin	All statuses

Response JSON Sample:

```
{
  "success": true,
  "message": "Found 12 products matching 'iphone'",
  "data": {
    "contents": {
      "shop": { "shopId": "...", "shopName": "TechStore Pro" },
      "products": [ { "...product summary fields..." } ],
      "totalProducts": 12,
      "searchMetadata": {
        "searchQuery": "iphone",
        "searchedStatuses": ["ACTIVE"],
        "userType": "PUBLIC"
      }
    },
    "currentPage": 1,
    "pageSize": 10,
    "totalElements": 12,
    "totalPages": 2,
    "hasNext": true,
    "hasPrevious": false
  }
}
```

Error Responses:

- `400`: Query too short or too long
- `404`: Shop not found or not accessible

13. Advanced Product Filter

Purpose: Filters products using multiple criteria with combined AND/OR logic.

Endpoint: `GET` `api/v1/e-commerce/shops/{shopId}/products/advanced-filter`

Access Level: Public (Enhanced features for authenticated users)

Authentication: Optional Bearer Token

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	ID of the shop

Query Parameters:

Parameter	Type	Required	Default	Description
minPrice	decimal	No	—	Minimum price
maxPrice	decimal	No	—	Maximum price (must be \geq minPrice)
condition	ProductCondition	No	—	Condition filter
categoryId	UUID	No	—	Category filter
inStock	boolean	No	—	Filter by availability
onSale	boolean	No	—	Filter by sale status
hasGroupBuying	boolean	No	—	Filter by group buying
hasInstallments	boolean	No	—	Filter by installments
hasMultipleColors	boolean	No	—	Filter by color variations
status	ProductStatus[]	No	ACTIVE	Status filter (owners/admins for non-ACTIVE)
page	integer	No	1	Page number
size	integer	No	10	Items per page (max: 50)
sortBy	string	No	createdAt	createdAt, updatedAt, productName, price, stockQuantity
sortDir	string	No	desc	asc OR desc

Filter Logic:

Filter Type	Logic
-------------	-------

Price range	AND (minPrice AND maxPrice)
Feature flags	AND (all must match)
Multiple statuses	OR

Error Responses:

- `400`: Invalid filter values or price range error
- `404`: Shop or category not found

14. Get Public Product by Slug

Purpose: Retrieves a single active product by its slug (same response shape as [Get Public Product by ID](#)).

Endpoint: `GET` `api/v1/e-commerce/shops/{shopId}/products/find-by-slug/{slug}`

Access Level: Public

Path Parameters:

Parameter	Type	Required	Description
shopId	UUID	Yes	Shop must be active and approved
slug	string	Yes	Product slug

Error Responses:

- `404`: Shop not found/not approved, or product not found/not active

15. Installment Plan Config

Purpose: CRUD for installment plans attached to a product. Plans are created separately after the product, and linked to it by `productId`.

Base URL: `api/v1/e-commerce/products/{shopId}/{productId}/installment-plans`

Access Level: Protected (Requires shop owner or system admin role)

Authentication: Bearer Token

Path Parameters (shared):

Parameter	Type	Required	Description
shopId	UUID	Yes	ID of the shop
productId	UUID	Yes	ID of the product

15a. Create Installment Plan

Endpoint: **POST** `api/v1/e-commerce/products/{shopId}/{productId}/installment-plans`

Request Body:

Parameter	Type	Required	Description	Validation
planName	string	Yes	Display name for the plan	Min: 3, Max: 100 chars
paymentFrequency	PaymentFrequency	Yes	Payment interval	DAILY, WEEKLY, BI_WEEKLY, SEMI_MONTHLY, MONTHLY, QUARTERLY, CUSTOM_DAYS
customFrequencyDays	integer	Conditional	Days between payments	Required when <code>paymentFrequency=CUSTOM_DAYS</code> , min: 1
numberOfPayments	integer	Yes	Total number of payments	Min: 2, Max: 120
apr	decimal	Yes	Annual percentage rate	Min: 0.0, Max: 36.0, 2 decimal places
minDownPaymentPercent	integer	Yes	Minimum down payment %	Min: 10, Max: 50
fulfillmentTiming	FulfillmentTiming	Yes	When to ship	IMMEDIATE (ship after down payment), AFTER_PAYMENT (layaway — ship after final payment)
displayOrder	integer	No	Sort order in UI	Min: 0, Default: 0
isFeatured	boolean	No	Highlight as recommended plan	Default: false
isActive	boolean	No	Plan is available to buyers	Default: true

Request JSON Sample:

```
{
  "planName": "6-Month Interest-Free",
  "paymentFrequency": "MONTHLY",
  "numberOfPayments": 6,
  "apr": 0.00,
  "minDownPaymentPercent": 20,
  "fulfillmentTiming": "IMMEDIATE",
  "displayOrder": 1,
  "isFeatured": true,
  "isActive": true
}
```

Error Responses:

- `400`: Validation errors
- `404`: Shop or product not found

15b. Get All Installment Plans

Endpoint: `GET` `api/v1/e-commerce/products/{shopId}/{productId}/installment-plans`

Returns a list of all installment plans for the product.

Error Responses:

- `404`: Shop or product not found

15c. Get Installment Plan by ID

Endpoint: `GET` `api/v1/e-commerce/products/{shopId}/{productId}/installment-plans/{planId}`

Additional Path Parameter:

Parameter	Type	Description
planId	UUID	ID of the installment plan

15d. Update Installment Plan

Endpoint: `PUT` `api/v1/e-commerce/products/{shopId}/{productId}/installment-plans/{planId}`

All fields are optional — only provided fields are updated. Same field structure as Create.

15e. Delete Installment Plan

Endpoint: **DELETE** `api/v1/e-commerce/products/{shopId}/{productId}/installment-plans/{planId}`

15f. Activate / Deactivate Plan

Activate: **PATCH** `api/v1/e-commerce/products/{shopId}/{productId}/installment-plans/{planId}/activate`

Deactivate: **PATCH** `api/v1/e-commerce/products/{shopId}/{productId}/installment-plans/{planId}/deactivate`

Toggles `isActive` on the plan without changing any other fields.

15g. Set Featured Plan

Endpoint: **PATCH** `api/v1/e-commerce/products/{shopId}/{productId}/installment-plans/{planId}/set-featured`

Marks the specified plan as the featured (recommended) plan for this product.

16. Digital File Management

Purpose: Manages downloadable files for `DIGITAL` products. Uses a presign → upload → confirm flow to upload files directly to object storage.

Base URL: `api/v1/e-commerce/shops/{shopId}/products/{productId}/digital-files`

Access Level: Protected (Requires shop owner or system admin role)

Authentication: Bearer Token

Path Parameters (shared):

Parameter	Type	Required	Description
shopId	UUID	Yes	ID of the shop
productId	UUID	Yes	ID of the digital product

16a. Presign Upload

Purpose: Generates a presigned PUT URL. The client uploads the file directly to this URL, then calls `/confirm`.

Endpoint: `POST` `api/v1/e-commerce/shops/{shopId}/products/{productId}/digital-files/presign-upload`

Request Body:

Parameter	Type	Required	Description
fileName	string	Yes	Original file name
contentType	string	Yes	MIME type (e.g. <code>application/pdf</code>)
fileSize	long	Yes	File size in bytes (must be positive)
displayOrder	integer	No	Sort order for multiple files

Request JSON Sample:

```
{
  "fileName": "design-kit-v2.fig",
  "contentType": "application/octet-stream",
  "fileSize": 52428800,
  "displayOrder": 1
}
```

Response JSON Sample:

```
{
  "success": true,
  "message": "Upload URL generated – upload directly to this URL then call /confirm",
  "data": {
    "uploadUrl": "https://s3.example.com/bucket/key?X-Amz-Signature=...",
    "objectKey": "digital-files/product-456/design-kit-v2.fig",
    "expiresAt": "2026-05-19T11:00:00"
  }
}
```

Upload Flow:

1. Call `POST /presign-upload` → receive `uploadUrl` and `objectKey`
2. `PUT {uploadUrl}` with binary file body (do not call the API for this step)
3. Call `POST /confirm` with `objectKey` to register the file

16b. Confirm Upload

Purpose: Registers a file after direct upload to object storage. Must be called after the actual file upload succeeds.

Endpoint: `POST` `api/v1/e-commerce/shops/{shopId}/products/{productId}/digital-files/confirm`

Request Body:

Parameter	Type	Required	Description
objectKey	string	Yes	Returned by presign-upload
fileName	string	Yes	Original file name
contentType	string	Yes	MIME type
fileSize	long	Yes	File size in bytes
displayOrder	integer	No	Sort order

Response JSON Sample:

```
{
  "success": true,
  "message": "File confirmed and linked to product",
  "data": {
    "fileId": "aaalbbbb-e89b-12d3-a456-426614174001",
    "productId": "456e7890-e89b-12d3-a456-426614174001",
    "fileName": "design-kit-v2.fig",
    "contentType": "application/octet-stream",
    "fileSize": 52428800,
    "fileVersion": 1,
    "displayOrder": 1,
    "isActive": true,
    "uploadedAt": "2026-05-19T10:45:00"
  }
}
```

16c. Get Product Files

Endpoint: `GET` `api/v1/e-commerce/shops/{shopId}/products/{productId}/digital-files`

Returns a list of `DigitalFileResponse` objects for all files linked to the product.

16d. Delete File

Endpoint: `DELETE` `api/v1/e-commerce/shops/{shopId}/products/{productId}/digital-files/{fileId}`

Additional Path Parameter:

Parameter	Type	Description
fileId	UUID	ID of the file to delete

16e. Toggle File Active Status

Endpoint: `PATCH` `api/v1/e-commerce/shops/{shopId}/products/{productId}/digital-files/{fileId}/toggle`

Query Parameters:

Parameter	Type	Required	Description
isActive	boolean	Yes	<code>true</code> to activate, <code>false</code> to deactivate

Deactivated files are hidden from buyers but not deleted.

17. Product Preview Management

Purpose: Manages a single preview file per product — a publicly accessible teaser shown to buyers before purchase. Distinct from private digital content files. Supports VIDEO, PDF, 3D models, and IMAGE previews.

Base URL: `api/v1/e-commerce/shops/{shopId}/products/{productId}/preview`

Access Level: Protected (Requires shop owner or system admin role)

Authentication: Bearer Token

Path Parameters (shared):

Parameter	Type	Required	Description
shopId	UUID	Yes	ID of the shop
productId	UUID	Yes	ID of the product (any type — PHYSICAL or DIGITAL)

Storage: Uploaded to `nextgate-preview-content` bucket (public read). The confirmed URL is permanent and requires no authentication to access.

17a. Presign Preview Upload

Purpose: Generates a presigned PUT URL. The client uploads the preview file directly to this URL, then calls `/confirm`.

Endpoint: `POST` `api/v1/e-commerce/shops/{shopId}/products/{productId}/preview/presign-upload`

Request Body:

Parameter	Type	Required	Description
fileName	string	Yes	Original file name (used to build the storage key)
contentType	string	Yes	MIME type (e.g. <code>video/mp4</code> , <code>application/pdf</code> , <code>image/jpeg</code> , <code>model/gltf-binary</code>)
fileSize	long	Yes	File size in bytes (must be positive)

Request JSON Sample:

```
{
  "fileName": "product-trailer.mp4",
  "contentType": "video/mp4",
  "fileSize": 52428800
}
```

Response JSON Sample:

```
{
  "success": true,
  "message": "Preview upload URL generated – upload directly then call /confirm",
}
```

```
"data": {
  "uploadUrl": "https://minio.example.com/nextgate-preview-content/preview/shop-id/product-id/uuid_product-trailer.mp4?X-Amz-Signature=...",
  "objectKey": "preview/shop-id/product-id/uuid_product-trailer.mp4",
  "expiresAt": "2026-05-19T11:30:00"
}
```

Upload Flow:

1. Call `POST /presign-upload` → receive `uploadUrl` and `objectKey`
2. `PUT {uploadUrl}` with binary file body (client-to-MinIO directly, not through the API)
3. Call `POST /confirm` with `objectKey` and `previewType` to link the file to the product

Error Responses:

- `401`: Authentication required
- `403`: Insufficient permissions
- `404`: Shop or product not found

17b. Confirm Preview Upload

Purpose: Links the uploaded file to the product. Sets `previewType` and stores the permanent public URL as `previewUrl`. If the product already has a preview, the old file is deleted from storage.

Endpoint: `POST` `api/v1/e-commerce/shops/{shopId}/products/{productId}/preview/confirm`

Request Body:

Parameter	Type	Required	Description
<code>objectKey</code>	string	Yes	Returned by <code>presign-upload</code>
<code>previewType</code>	PreviewType	Yes	<code>VIDEO</code> , <code>PDF</code> , <code>THREE_D</code> , or <code>IMAGE</code>
<code>previewDownloadable</code>	boolean	No	Whether viewers can download the file. Default: <code>false</code> (view/stream only)

Request JSON Sample:

```
{
  "objectKey": "preview/shop-id/product-id/uuid_product-trailer.mp4",
  "previewType": "VIDEO",
```

```
"previewDownloadable": false
}
```

Response JSON Sample:

```
{
  "success": true,
  "message": "Preview confirmed and linked to product",
  "data": null
}
```

After confirm, the product's public response will include:

```
{
  "previewType": "VIDEO",
  "previewUrl": "https://minio.example.com/nextgate-preview-content/preview/shop-id/product-
id/uuid_product-trailer.mp4",
  "previewDownloadable": false
}
```

Error Responses:

- `400`: Missing objectKey or previewType
- `401`: Authentication required
- `403`: Insufficient permissions
- `404`: Shop or product not found

17c. Remove Preview

Purpose: Deletes the product's preview file from storage and clears `previewType` and `previewUrl` on the product.

Endpoint: `DELETE` `api/v1/e-commerce/shops/{shopId}/products/{productId}/preview`

Response JSON Sample:

```
{
  "success": true,
  "message": "Preview removed from product",
  "data": null
}
```

Error Responses:

- `401`: Authentication required
- `403`: Insufficient permissions
- `404`: Shop or product not found

Quick Reference

Common HTTP Status Codes

Code	Meaning
<code>200 OK</code>	Successful GET/PUT/PATCH
<code>201 Created</code>	Successful POST (resource created)
<code>400 Bad Request</code>	Invalid data, validation errors, business rule violations
<code>401 Unauthorized</code>	Authentication required or invalid token
<code>403 Forbidden</code>	Insufficient permissions
<code>404 Not Found</code>	Resource not found or not accessible
<code>409 Conflict</code>	Duplicate product name or business constraint violation
<code>422 Unprocessable Entity</code>	Field-level validation errors
<code>500 Internal Server Error</code>	Server error

User Access Levels

User Type	Product Management	Status Access
Public	View active products only	ACTIVE only
Authenticated	View active products only	ACTIVE only
Shop Owner	Full CRUD on own shop	All statuses
System Admin	Full CRUD on all shops	All statuses

Product Type Fulfillment Flows

Type	Flow
<code>PHYSICAL</code>	Payment → <code>PENDING_SHIPMENT</code> → Seller ships → Buyer confirms with 6-digit code → Escrow releases → <code>COMPLETED</code>

Example: SHP12345678-ELE-APP-512-0001

Data Format Standards

- **Dates:** ISO 8601 (2026-05-19T14:30:00Z)
- **Prices:** Decimal with 2 decimal places, stored as BigDecimal
- **UUIDs:** Standard UUID format
- **Pagination:** 1-indexed page parameter
- **Colors:** Hex format #RRGGBB
- **Percentages:** Decimal format (20.00 = 20%)

Error Response Format

```
{
  "success": false,
  "message": "Human-readable error message",
  "error": {
    "code": "ERROR_CODE",
    "details": "Detailed information",
    "field": "fieldName (if field-specific)",
    "timestamp": "2026-05-19T14:30:00Z"
  }
}
```

Product Creation Flow

1. POST /shops/{shopId}/products?action=SAVE_DRAFT
– create the product shell
2. POST /products/{shopId}/{productId}/installment-plans
– add installment plans (if installmentEnabled)
3. POST /shops/{shopId}/products/{productId}/digital-files/presign-upload
PUT {uploadUrl} (direct to storage)
POST /shops/{shopId}/products/{productId}/digital-files/confirm
– upload private digital files (DIGITAL products only)
4. POST /shops/{shopId}/products/{productId}/preview/presign-upload
PUT {uploadUrl} (direct to storage)

POST /shops/{shopId}/products/{productId}/preview/confirm

– upload preview teaser (any product type, optional)

– preview is stored in public bucket; URL is immediately accessible

5. PATCH /shops/{shopId}/products/{productId}/publish

– publish when ready

Preview vs Digital Files

	Preview	Digital Files
Who sees it	Everyone (before purchase)	Buyers only (after payment)
Bucket	nextgate-preview-content (public)	nextgate-digital-content (private)
Access	Permanent public URL	Presigned URL, expires per downloadExpiryDays
Count	One per product	Multiple files per product
Products	PHYSICAL or DIGITAL	DIGITAL only
Purpose	Teaser/sample before buying	Actual purchased content

Revision #9

Created 23 September 2025 08:15:01 by Admin Qbit

Updated 19 May 2026 11:27:14 by Admin Qbit