

Shops Management Service

The Shops Management Service provides comprehensive shop management capabilities for the NextGate social commerce platform. This service handles shop categories, shop creation and management, rating systems, and review functionality with role-based access controls and business logic enforcement.

Hints:

- Shop Approval: All shops are auto-approved upon creation (isApproved: true)
- Business Rules: Shop owners cannot rate/review their own shops
- One Per User: Each user can submit only one rating and one review per shop
- Admin Categories: Shop categories require SUPER_ADMIN role for management
- Soft Deletion: Shops use soft delete (isDeleted flag) to maintain data integrity
- Slug Generation: Shop slugs are auto-generated from names with uniqueness validation
- Rating Scale: Ratings use 1-5 star system with summary statistics
- Review Moderation: Reviews support status management (ACTIVE, HIDDEN, FLAGGED, UNDER_REVIEW)

- [Shop Management](#)
- [Shop Review](#)
- [Shop Subscription](#)

Shop Management

Author: Josh S. Sakweli, Backend Lead Team

Last Updated: 2026-05-19

Version: v2.0

Short Description: The Shop Management API provides endpoints for creating, managing, and retrieving shop information on the NextGate platform. Covers shop registration, updates, approvals, WABA (WhatsApp Business) integration, AI chatbot toggling, and conversation history.

Hints:

- All shop endpoints use the prefix `api/v1/e-commerce/shops`
- Shop creation requires authentication; public users can only read
- Pagination uses 1-based page numbering
- Shop approval operations require `ROLE_SUPER_ADMIN` or `ROLE_STAFF_ADMIN`
- Featured shops are randomized on each request
- Rating and review data is automatically included in shop responses (read-only — managed by separate review service)
- WABA = WhatsApp Business Account integration for the shop's AI-powered chatbot

Standard Response Format

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Operation completed successfully",
  "action_time": "2026-05-19T10:30:45",
  "data": { }
}
```

Error responses follow the same envelope with `"success": false` and `"data"` set to the error message string.

Endpoints

1. Create Shop

Endpoint: POST `api/v1/e-commerce/shops`

Access Level: Protected

Request Body:

Parameter	Type	Required	Description	Validation
shopName	string	Yes	Name of the shop	Min: 2, Max: 100 chars
shopDescription	string	Yes	Detailed description	Max: 1000 chars
phoneNumber	string	Yes	Contact phone	Pattern: <code>^\+?[0-9]{10,15}\$</code>
city	string	Yes	City	Min: 2, Max: 50 chars
region	string	Yes	Region/state	Min: 2, Max: 50 chars
logoUrl	string	No	Shop logo URL	Valid URL, max 1000 chars
bannerUrl	string	No	Shop banner URL	Valid URL, max 1000 chars
shopImages	array	No	Array of shop image URLs	Valid URLs, max 1000 chars each
email	string	No	Contact email	Valid email, max 100 chars
countryCode	string	No	Country code	Max: 3 chars, Default: <code>"TZ"</code>
streetAddress	string	No	Street address	Max: 255 chars
landmark	string	No	Landmark / location notes	Max: 300 chars
latitude	decimal	No	GPS latitude	Range: -90.0 to 90.0
longitude	decimal	No	GPS longitude	Range: -180.0 to 180.0

Request JSON Sample:

```
{
  "shopName": "Mama Lucy's Restaurant",
  "shopDescription": "Authentic Tanzanian cuisine in the heart of Dar es Salaam",
  "phoneNumber": "+255123456789",
  "city": "Dar es Salaam",
```

```
"region": "Dar es Salaam",
"logoUrl": "https://example.com/logo.jpg",
"bannerUrl": "https://example.com/banner.jpg",
"shopImages": ["https://example.com/shop1.jpg"],
"email": "info@mamalucy.co.tz",
"countryCode": "TZ",
"streetAddress": "Msimbazi Street, Block 45",
"landmark": "Near the main bus stop",
"latitude": -6.7924,
"longitude": 39.2083
}
```

Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Shop created successfully",
  "action_time": "2026-05-19T10:30:45",
  "data": {
    "shopId": "123e4567-e89b-12d3-a456-426614174000",
    "shopName": "Mama Lucy's Restaurant",
    "shopSlug": "mama-lucys-restaurant",
    "shopDescription": "Authentic Tanzanian cuisine...",
    "logoUrl": "https://example.com/logo.jpg",
    "bannerUrl": "https://example.com/banner.jpg",
    "shopImages": ["https://example.com/shop1.jpg"],
    "ownerId": "456e7890-e89b-12d3-a456-426614174001",
    "ownerName": "Lucy Mwalimu",
    "status": "PENDING",
    "phoneNumber": "+255123456789",
    "email": "info@mamalucy.co.tz",
    "streetAddress": "Msimbazi Street, Block 45",
    "city": "Dar es Salaam",
    "region": "Dar es Salaam",
    "countryCode": "TZ",
    "latitude": -6.7924,
    "longitude": 39.2083,
    "landmark": "Near the main bus stop",
    "isVerified": false,
  }
}
```

```
"verificationBadge": null,
"trustScore": 0.00,
"isApproved": true,
"createdAt": "2026-05-19T10:30:45",
"updatedAt": "2026-05-19T10:30:45",
"approvedAt": null,
"averageRating": null,
"totalRatings": 0,
"totalActiveReviews": 0,
"reviews": []
}
}
```

Error Responses:

- 400: Shop name already exists
- 401: Authentication required
- 422: Validation errors

2. Get All Shops

Endpoint: `GET` `api/v1/e-commerce/shops/all`

Access Level: Public

Returns a list of all shops with summary info and top 5 reviews per shop.

3. Get All Shops (Paginated)

Endpoint: `GET` `api/v1/e-commerce/shops/all-paged`

Access Level: Public

Query Parameters:

Parameter	Type	Default	Description
page	integer	1	Page number (1-based)
size	integer	10	Items per page (max: 100)

Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Shops retrieved successfully",
  "action_time": "2026-05-19T10:30:45",
  "data": {
    "shops": [ { "...ShopSummary fields..." } ],
    "currentPage": 1,
    "pageSize": 10,
    "totalElements": 150,
    "totalPages": 15,
    "hasNext": true,
    "hasPrevious": false,
    "isFirst": true,
    "isLast": false
  }
}
```

4. Update Shop

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

Access Level: Protected (Shop Owner only)

Path Parameters:

Parameter	Type	Description
shopId	UUID	ID of the shop

All request body fields are optional — only provided fields are updated:

Parameter	Type	Validation
shopName	string	Min: 2, Max: 100 chars
shopDescription	string	Max: 1000 chars
logoUrl	string	Valid URL
bannerUrl	string	Valid URL

Parameter	Type	Validation
shopImages	array	Valid URLs
phoneNumber	string	<code>^\+?[0-9]{10,15}\$</code>
email	string	Valid email
streetAddress	string	Max: 255 chars
city	string	Min: 2, Max: 50 chars
region	string	Min: 2, Max: 50 chars
countryCode	string	Max: 3 chars
latitude	decimal	-90.0 to 90.0
longitude	decimal	-180.0 to 180.0
landmark	string	Max: 300 chars

Response: Full `ShopResponse` (same shape as Create response)

Error Responses:

- `400`: Not the shop owner, or shop is deleted
- `401`: Authentication required
- `404`: Shop not found

5. Get Shop by ID (Summary)

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

Access Level: `Public`

Returns `ShopSummaryListResponse` — public-facing fields including top 5 reviews and rating.

6. Get Shop by ID (Detailed)

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

Access Level: `Protected (Shop Owner or Admin)`

Returns full `ShopResponse` including all reviews and management fields.

7. Get My Shops

Endpoint: GET `api/v1/e-commerce/shops/my-shops`

Access Level: Protected

Returns all shops owned by the authenticated user (`ShopSummaryListResponse` list).

8. Get My Shops (Paginated)

Endpoint: GET `api/v1/e-commerce/shops/my-shops-paged`

Access Level: Protected

Query Parameters:

Parameter	Default	Description
page	1	Page number (1-based)
size	10	Items per page (max: 100)

9. Approve / Reject Shop

Endpoint: PATCH `api/v1/e-commerce/shops/{shopId}/approve-shop`

Access Level: Protected (`ROLE_SUPER_ADMIN` or `ROLE_STAFF_ADMIN`)

Query Parameters:

Parameter	Type	Required	Description
approve	boolean	Yes	<code>true</code> to approve, <code>false</code> to reject

Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Shop approval status changed successfully",
```

```
"action_time": "2026-05-19T10:30:45",
"data": {
  "shopId": "123e4567-e89b-12d3-a456-426614174000",
  "shopName": "Mama Lucy's Restaurant",
  "isApproved": true,
  "approvedAt": "2026-05-19T10:30:45"
}
```

10. Get Featured Shops

Endpoint: `GET` `api/v1/e-commerce/shops/featured`

Access Level: Public

Returns up to 20 randomly selected featured shops (`ShopSummaryListResponse` list).

11. Get Featured Shops (Paginated)

Endpoint: `GET` `api/v1/e-commerce/shops/featured-paged`

Access Level: Public

Query Parameters:

Parameter	Default	Description
page	1	Page number
size	10	Items per page (max: 100)

12. Search Shops

Endpoint: `GET` `api/v1/e-commerce/shops/search`

Access Level: Public

Query Parameters:

Parameter	Type	Required	Default	Description
q	string	No	—	Search query
page	integer	No	1	Page number
size	integer	No	10	Items per page

Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Shops retrieved successfully",
  "action_time": "2026-05-19T10:30:45",
  "data": {
    "content": [ { "...ShopSearchResponse fields..." } ],
    "totalElements": 8,
    "totalPages": 1,
    "currentPage": 1,
    "pageSize": 10,
    "hasNext": false,
    "hasPrevious": false
  }
}
```

13. Get Shop Summary Stats

Endpoint: `GET` `api/v1/e-commerce/shops/{shopId}/summary-stats`

Access Level: Public

Returns aggregated review and rating statistics for a shop, including per-user activity.

Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Shop summary stats retrieved successfully",
  "data": {
    "shopId": "123e4567-e89b-12d3-a456-426614174000",
  }
}
```

```
"shopName": "Mama Lucy's Restaurant",
"averageRating": 4.5,
"totalRatings": 25,
"ratingDistribution": { "1": 1, "2": 2, "3": 5, "4": 7, "5": 10 },
"totalReviews": 15,
"activeReviews": 12,
"hiddenReviews": 2,
"flaggedReviews": 1,
"userActivities": [
  {
    "userId": "user-123",
    "userName": "John Doe",
    "feedbackId": "rev-123",
    "reviewText": "Amazing food and service!",
    "reviewStatus": "ACTIVE",
    "ratingValue": 5,
    "date": "2026-05-19T14:30:00",
    "hasReview": true,
    "hasRating": true
  }
]
}
```

14. WABA (WhatsApp Business) Integration

WABA allows shops to receive and respond to WhatsApp customer messages via an AI-powered chatbot. The flow is: shop registers a WABA → admin approves with Meta credentials → shop toggles AI on/off and monitors conversation history.

Base path for all WABA endpoints: `api/v1/e-commerce/shops/{shopId}/waba`

Shared Path Parameter:

Parameter	Type	Description
shopId	UUID	ID of the shop

14a. Register WABA

Purpose: Shop owner submits a WhatsApp number and display name to begin WABA registration.

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

Access Level: Protected (Shop Owner)

Request Body:

Parameter	Type	Required	Validation
phoneNumber	string	Yes	Max: 20 chars
displayName	string	Yes	Max: 100 chars

Request JSON Sample:

```
{
  "phoneNumber": "+255712345678",
  "displayName": "Mama Lucy's Restaurant"
}
```

14b. Approve WABA

Purpose: Admin approves a pending WABA registration by supplying Meta WABA credentials.

Endpoint: **PATCH** `api/v1/e-commerce/shops/{shopId}/waba/approve`

Access Level: Protected (`ROLE_SUPER_ADMIN` or `ROLE_STAFF_ADMIN`)

Request Body:

Parameter	Type	Required	Validation
wabald	string	Yes	Max: 64 chars (Meta WABA ID)
phoneNumberId	string	Yes	Max: 64 chars (Meta Phone Number ID)
phoneNumber	string	Yes	Max: 20 chars

14c. Resubmit WABA

Purpose: Shop owner resubmits a rejected or pending WABA with corrected info.

Endpoint: `PATCH` `api/v1/e-commerce/shops/{shopId}/waba/resubmit`

Access Level: Protected (Shop Owner)

Request Body:

Parameter	Type	Validation
phoneNumber	string	Max: 20 chars
displayName	string	Max: 100 chars

14d. Admin Update WABA

Purpose: Admin updates Meta credentials on an existing WABA account.

Endpoint: `PATCH` `api/v1/e-commerce/shops/{shopId}/waba/admin-update`

Access Level: Protected (`ROLE_SUPER_ADMIN` or `ROLE_STAFF_ADMIN`)

Same request body as [Resubmit WABA](#).

14e. Update WABA Status

Purpose: Admin changes the status of a WABA account (e.g., suspend or reactivate).

Endpoint: `PATCH` `api/v1/e-commerce/shops/{shopId}/waba/status`

Access Level: Protected (`ROLE_SUPER_ADMIN` or `ROLE_STAFF_ADMIN`)

Query Parameters:

Parameter	Type	Required	Description
status	ShopWabaStatus	Yes	<code>PENDING</code> , <code>ACTIVE</code> , <code>SUSPENDED</code> , <code>REJECTED</code>

14f. Toggle AI Chatbot

Purpose: Shop owner enables or disables the AI chatbot for their WABA.

Endpoint: PATCH `api/v1/e-commerce/shops/{shopId}/waba/toggle-ai`

Access Level: Protected (Shop Owner)

Query Parameters:

Parameter	Type	Required	Description
enabled	boolean	Yes	<code>true</code> to enable, <code>false</code> to disable

Response JSON Sample:

```
{
  "success": true,
  "message": "AI enabled successfully",
  "data": {
    "shopId": "...",
    "aiEnabled": true,
    "updatedAt": "2026-05-19T11:00:00"
  }
}
```

14g. Get WABA Conversations

Purpose: Retrieves paginated WhatsApp conversation sessions for the shop.

Endpoint: GET `api/v1/e-commerce/shops/{shopId}/waba/conversations`

Access Level: Protected (Shop Owner or Admin)

Query Parameters:

Parameter	Default	Description
page	1	Page number
size	10	Items per page

Response JSON Sample:

```
{
  "success": true,
  "message": "Conversations retrieved",
}
```

```
"data": {
  "content": [ { "...WabaSessionResponse fields..." } ],
  "currentPage": 1,
  "pageSize": 10,
  "totalElements": 42,
  "totalPages": 5,
  "hasNext": true,
  "hasPrevious": false
}
```

14h. Get Session Messages

Purpose: Retrieves paginated messages within a specific conversation session.

Endpoint: `GET` `api/v1/e-commerce/shops/{shopId}/waba/conversations/{sessionId}/messages`

Access Level: Protected (Shop Owner or Admin)

Additional Path Parameter:

Parameter	Type	Description
sessionId	UUID	ID of the conversation session

Query Parameters:

Parameter	Default	Description
page	1	Page number
size	20	Items per page

14i. Get Session Messages by Date Range

Purpose: Retrieves messages in a session filtered by date range.

Endpoint: `GET` `api/v1/e-commerce/shops/{shopId}/waba/conversations/{sessionId}/messages/by-date`

Access Level: Protected (Shop Owner or Admin)

Additional Path Parameter:

Parameter	Type	Description
sessionId	UUID	ID of the conversation session

Query Parameters:

Parameter	Type	Required	Description
from	LocalDateTime	Yes	Start datetime (ISO 8601)
to	LocalDateTime	Yes	End datetime (ISO 8601)
page	integer	No	Default: 1
size	integer	No	Default: 20

Quick Reference

Enums

ShopStatus: PENDING, ACTIVE, SUSPENDED, CLOSED, UNDER_REVIEW

ShopType: PHYSICAL, ONLINE, HYBRID

VerificationBadge: BRONZE, SILVER, GOLD, PREMIUM

ShopWabaStatus: PENDING, ACTIVE, SUSPENDED, REJECTED

Error Response Codes

Code	Meaning
400	Business logic violation or item already exists
401	Authentication required or token invalid/expired
403	Insufficient permissions
404	Resource not found
422	Field-level validation errors

Validation Error (422):

```
{
  "success": false,
```

```
"httpStatus": "UNPROCESSABLE_ENTITY",
"message": "Validation failed",
"data": {
  "shopName": "Shop name must be between 2 and 100 characters",
  "phoneNumber": "Phone number must be between 10-15 digits and may start with +"
}
}
```

Access Control Summary

Role	Capabilities
Public	Read shops, search, featured, summary stats
Authenticated user	All public + create shop, view own shops
Shop Owner	All authenticated + update own shop, WABA management, view conversations
<code>ROLE_SUPER_ADMIN</code> / <code>ROLE_STAFF_ADMIN</code>	All + approve/reject shops, approve/update/status WABA

WABA Registration Flow

1. POST `/{shopId}/waba/register` – shop owner submits phone + display name
2. PATCH `/{shopId}/waba/approve` – admin supplies Meta wabaId + phoneNumberId
3. PATCH `/{shopId}/waba/toggle-ai` – shop owner enables AI chatbot
4. GET `/{shopId}/waba/conversations` – shop owner monitors customer chats

Shop Review

Author: Josh S. Sakweli, Backend Lead Team

Last Updated: 2026-05-19

Version: v2.0

Short Description: The Shop Review API allows users to write, manage, and retrieve reviews (text + star rating) for shops. Supports create, update, delete, listing with pagination, and summary statistics.

Hints:

- All endpoints use the prefix `api/v1/e-commerce/shops/reviews/{shopId}`
- All operations require Bearer token authentication
- One review per user per shop — use PUT to update an existing one
- Shop owners cannot review their own shops
- `reviewText` and `ratingValue` are both optional fields, but at least one should be provided
- `ratingValue` must be 1-5 if provided
- `reviewText` must be 10-1000 characters if provided
- `isMyReview` flag is only populated when the authenticated user is included in the response builder

Standard Response Format

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "...",
  "action_time": "2026-05-19T10:30:45",
  "data": { }
}
```

Endpoints

1. Create Review (Feedback)

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

Access Level: Protected

Path Parameters:

Parameter	Type	Description
shopId	UUID	ID of the shop to review

Request Body:

Parameter	Type	Required	Validation
reviewText	string	No	Min: 10, Max: 1000 chars
ratingValue	integer	No	Min: 1, Max: 5

Request JSON Sample:

```
{
  "reviewText": "Amazing food and excellent service! Highly recommend the local dishes.",
  "ratingValue": 5
}
```

Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Feedback submitted successfully",
  "action_time": "2026-05-19T10:30:45",
  "data": {
    "reviewId": "123e4567-e89b-12d3-a456-426614174000",
    "shopId": "456e7890-e89b-12d3-a456-426614174001",
    "shopName": "Mama Lucy's Restaurant",
    "userId": "789e0123-e89b-12d3-a456-426614174002",
    "userName": "John Doe",
    "reviewText": "Amazing food and excellent service! Highly recommend the local dishes.",
    "ratingValue": 5,
    "status": "ACTIVE",
    "createdAt": "2026-05-19T10:30:45",
    "updatedAt": "2026-05-19T10:30:45",
    "isMyReview": false
  }
}
```

```
}  
}
```

Error Responses:

- 400: Already reviewed this shop, or shop owner reviewing own shop
- 401: Authentication required
- 404: Shop not found

2. Update Review (Feedback)

Endpoint: **PUT** `api/v1/e-commerce/shops/reviews/{shopId}`

Access Level: Protected (Review Owner only)

Path Parameters:

Parameter	Type	Description
shopId	UUID	ID of the reviewed shop

Request Body (same as Create — all fields optional):

Parameter	Type	Validation
reviewText	string	Min: 10, Max: 1000 chars
ratingValue	integer	Min: 1, Max: 5

Response JSON Sample:

```
{  
  "success": true,  
  "httpStatus": "OK",  
  "message": "Feedback updated successfully",  
  "action_time": "2026-05-19T11:15:30",  
  "data": {  
    "reviewId": "123e4567-e89b-12d3-a456-426614174000",  
    "shopId": "456e7890-e89b-12d3-a456-426614174001",  
    "shopName": "Mama Lucy's Restaurant",  
    "userId": "789e0123-e89b-12d3-a456-426614174002",  
    "userName": "John Doe",  
    "reviewText": "Still amazing food. Local dishes are worth it.",  
  }  
}
```

```
"ratingValue": 4,
"status": "ACTIVE",
"createdAt": "2026-05-19T10:30:45",
"updatedAt": "2026-05-19T11:15:30",
"isMyReview": false
}
}
```

Error Responses:

- 401: Authentication required
- 404: Review not found (create one first)

3. Delete Review (Feedback)

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

Access Level: Protected (Review Owner only)

Path Parameters:

Parameter	Type	Description
shopId	UUID	ID of the reviewed shop

Soft-deletes the review. Deleted reviews no longer appear in listings or counts.

Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Feedback deleted successfully",
  "action_time": "2026-05-19T10:30:45",
  "data": null
}
```

Error Responses:

- 401: Authentication required
- 404: Review not found

4. Get Active Reviews for Shop

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

Access Level: `Protected` (auth required to populate `isMyReview`)

Path Parameters:

Parameter	Type	Description
shopId	UUID	ID of the shop

Returns all `ACTIVE` reviews. `isMyReview` is set to `true` for the authenticated user's own review.

Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Reviews retrieved successfully",
  "action_time": "2026-05-19T10:30:45",
  "data": [
    {
      "reviewId": "123e4567-e89b-12d3-a456-426614174000",
      "shopId": "456e7890-e89b-12d3-a456-426614174001",
      "shopName": "Mama Lucy's Restaurant",
      "userId": "789e0123-e89b-12d3-a456-426614174002",
      "userName": "John Doe",
      "reviewText": "Amazing food!",
      "ratingValue": 5,
      "status": "ACTIVE",
      "createdAt": "2026-05-19T10:30:45",
      "updatedAt": "2026-05-19T10:30:45",
      "isMyReview": true
    }
  ]
}
```

Error Responses:

- `401`: Authentication required
- `404`: Shop not found

5. Get Active Reviews (Paginated)

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

Access Level: Protected

Query Parameters:

Parameter	Default	Description
page	1	Page number (1-based)
size	10	Items per page (max: 100)

Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Reviews retrieved successfully",
  "action_time": "2026-05-19T10:30:45",
  "data": {
    "reviews": [
      {
        "reviewId": "123e4567-e89b-12d3-a456-426614174000",
        "shopName": "Mama Lucy's Restaurant",
        "userName": "John Doe",
        "reviewText": "Amazing food!",
        "ratingValue": 5,
        "status": "ACTIVE",
        "createdAt": "2026-05-19T10:30:45"
      }
    ],
    "currentPage": 1,
    "pageSize": 10,
    "totalElements": 25,
    "totalPages": 3,
    "hasNext": true,
    "hasPrevious": false,
    "isFirst": true,
    "isLast": false
  }
}
```

```
}  
}
```

Error Responses:

- `401`: Authentication required
- `404`: Shop not found

6. Get My Review for Shop

Endpoint: `GET` `api/v1/e-commerce/shops/reviews/{shopId}/my-review`

Access Level: `⏏` Protected

Returns the authenticated user's review for the shop, or `null` if they haven't reviewed it yet.

Response JSON Sample (has review):

```
{  
  "success": true,  
  "httpStatus": "OK",  
  "message": "Your feedback retrieved successfully",  
  "data": {  
    "reviewId": "123e4567-e89b-12d3-a456-426614174000",  
    "shopId": "456e7890-e89b-12d3-a456-426614174001",  
    "shopName": "Mama Lucy's Restaurant",  
    "userId": "789e0123-e89b-12d3-a456-426614174002",  
    "userName": "John Doe",  
    "reviewText": "Amazing food!",  
    "ratingValue": 5,  
    "status": "ACTIVE",  
    "createdAt": "2026-05-19T10:30:45",  
    "updatedAt": "2026-05-19T10:30:45",  
    "isMyReview": false  
  }  
}
```

Response when no review exists:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Your feedback retrieved successfully",
  "data": null
}
```

7. Get Shop Review Summary

Endpoint: `GET` `api/v1/e-commerce/shops/reviews/{shopId}/summary`

Access Level: Public

Returns aggregated rating and review counts for a shop.

Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Shop feedback summary retrieved successfully",
  "data": {
    "shopId": "456e7890-e89b-12d3-a456-426614174001",
    "shopName": "Mama Lucy's Restaurant",
    "averageRating": 4.5,
    "totalRatings": 25,
    "ratingDistribution": { "1": 1, "2": 2, "3": 5, "4": 7, "5": 10 },
    "totalReviews": 15,
    "activeReviews": 12,
    "hiddenReviews": 2,
    "flaggedReviews": 1
  }
}
```

Quick Reference

ReviewStatus Enum

Value	Description
ACTIVE	Visible in public listings
HIDDEN	Hidden from public view
FLAGGED	Flagged for admin review
UNDER_REVIEW	Under admin review

ReviewResponse Fields

Field	Type	Description
reviewId	UUID	Unique review ID
shopId	UUID	Shop that was reviewed
shopName	string	Shop name
userId	UUID	Reviewer's user ID
userName	string	Reviewer's display name
reviewText	string	Review text content
ratingValue	integer	Star rating (1-5), nullable
status	ReviewStatus	Current review status
createdAt	LocalDateTime	Creation timestamp
updatedAt	LocalDateTime	Last update timestamp
isMyReview	boolean	True if this is the current user's review

Error Codes

Code	Meaning
400	Business rule violation (duplicate review, owner self-review)
401	Authentication required
404	Shop or review not found
422	Field validation errors

Shop Subscription

Author: Josh S. Sakweli, Backend Lead Team

Last Updated: 2026-06-04

Version: v1.0

Base URL: `api/v1/e-commerce/shops`

Short Description: The Shop Subscription API allows users to subscribe and unsubscribe from shops on the Nexgate marketplace. Subscriptions drive personalised product discovery — subscribed shops receive a ranking boost in the Trending and For You feeds, and a dedicated "from your shops" signal in the relevance formula. Shop owners can view who has subscribed to their shop.

Hints:

- The action is called **Subscribe** — not Follow. This is intentional to distinguish from the social module's user→user **Follow** feature. Users follow people, they subscribe to shops.
- `POST /{shopId}/subscribe` is a **toggle** — calling it once subscribes, calling it again unsubscribes. No separate unsubscribe endpoint is needed.
- Subscription status (`isSubscribed`) is automatically included in `ShopResponse` for all shop detail endpoints. Authenticated users see their real status; anonymous users always receive `false`.
- Subscriptions directly feed into the Marketplace `FOR_YOU` and `TRENDING` personalisation — see `marketplace_api_doc.md` for formula details.
- Pages are **1-based** (`page=1` returns the first page).

Standard Response Format

All API responses follow a consistent structure using the Globe Response Builder pattern:

Success Response Structure

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Operation completed successfully",
  "action_time": "2026-06-04T10:30:45",
```

```
"data": { }  
}
```

Error Response Structure

```
{  
  "success": false,  
  "httpStatus": "BAD_REQUEST",  
  "message": "Error description",  
  "action_time": "2026-06-04T10:30:45",  
  "data": "Error description"  
}
```

Standard Response Fields

Field	Type	Description
success	boolean	true for success, false for errors
httpStatus	string	HTTP status name (OK, BAD_REQUEST, NOT_FOUND, etc.)
message	string	Human-readable operation result
action_time	string	ISO 8601 timestamp of response generation
data	object	Response payload for success, error detail for failures

HTTP Method Badge Standards

- GET — GET Green
- POST — POST Blue

Subscription State Reference

SUBSCRIPTION STATE FLOW

```

|
| User not subscribed
|   |
|   ▼ POST /{shopId}/subscribe
|
| User subscribed →
|   |
|   subscriberCount + 1
|   |
|   subscribed: true
|   |
|   ▼ POST /{shopId}/subscribe (toggle again)
|
| User not subscribed ←
|   |
|   subscriberCount - 1
|   |
|   subscribed: false
|
|
| Side effects:
| - Shop's subscriberCount is updated atomically
| - Marketplace FOR_YOU feed recalculates on next request
| - Marketplace TRENDING feed gives +0.25 boost to subscribed shops
|

```

How Subscriptions Affect the Marketplace

```

|
| SUBSCRIPTION → MARKETPLACE SIGNALS
|
|
| TRENDING feed:
|   personalizedScore = trendingScore + 0.25
|   (applied to every product from a subscribed shop)
|
| FOR YOU feed:
|   relevanceScore = categoryMatch × 0.40
|                   + favShopBoost × 0.40 ← subscription signal
|                   + trendingScore × 0.20
|   favShopBoost = 1.0 if subscribed, 0.0 otherwise
|
| Result: products from subscribed shops surface near the top of
|

```

| both feeds while still allowing genuinely trending products to
| compete (soft priority, not guaranteed top position). |

Endpoints

1. Toggle Subscribe / Unsubscribe

Purpose: Subscribes the authenticated user to a shop if they are not already subscribed, or unsubscribes them if they are. Returns the new subscription state and the updated subscriber count.

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

Access Level: Protected (authentication required)

Authentication: Bearer Token (required)

Request Headers:

Header	Type	Required	Description
<code>Authorization</code>	string	Yes	<code>Bearer <token></code>

Path Parameters:

Parameter	Type	Required	Description	Validation
<code>shopId</code>	UUID	Yes	The shop to subscribe or unsubscribe from	Must be a valid UUID of an active, non-deleted shop

Success Response JSON Sample — Subscribe:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Subscribed to shop successfully",
  "action_time": "2026-06-04T10:30:45",
  "data": {
    "subscribed": true,
```

```
    "subscriberCount": 1284
  }
}
```

Success Response JSON Sample — Unsubscribe (same endpoint, called again):

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Unsubscribed from shop successfully",
  "action_time": "2026-06-04T10:30:45",
  "data": {
    "subscribed": false,
    "subscriberCount": 1283
  }
}
```

Success Response Fields:

Field	Type	Description
<code>subscribed</code>	boolean	Current subscription state after the toggle — <code>true</code> if now subscribed, <code>false</code> if now unsubscribed
<code>subscriberCount</code>	long	Updated total number of subscribers for this shop

Error Response JSON Samples:

Unauthorized (401):

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Token has expired",
  "action_time": "2026-06-04T10:30:45",
  "data": "Token has expired"
}
```

Shop Not Found (404):

```
{
  "success": false,
```

```
"httpStatus": "NOT_FOUND",
"message": "Shop not found",
"action_time": "2026-06-04T10:30:45",
"data": "Shop not found"
}
```

Standard Error Types:

- `401 UNAUTHORIZED`: Token missing, expired, or invalid
- `404 NOT_FOUND`: Shop does not exist or has been deleted
- `500 INTERNAL_SERVER_ERROR`: Unexpected server failure

2. My Subscriptions

Purpose: Returns a paginated list of all shops the authenticated user is currently subscribed to, ordered by most recently subscribed first. Each entry includes shop details and the current subscriber count.

Endpoint: `GET` `api/v1/e-commerce/shops/my-subscriptions`

Access Level: Protected (authentication required)

Authentication: Bearer Token (required)

Request Headers:

Header	Type	Required	Description
<code>Authorization</code>	string	Yes	<code>Bearer <token></code>

Query Parameters:

Parameter	Type	Required	Description	Default
<code>page</code>	integer	No	Page number (1-based)	<code>1</code>
<code>size</code>	integer	No	Items per page	<code>10</code>

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
```

```

"message": "My subscriptions retrieved successfully",
"action_time": "2026-06-04T10:30:45",
"data": {
  "content": [
    {
      "subscriptionId": "9b1deb4d-3b7d-4bad-9bdd-2b0d7b3dcb6d",
      "shopId": "7cb3a812-1234-4abc-b3fc-9d84f55bce12",
      "shopName": "TechStore Tanzania",
      "shopSlug": "techstore-tanzania",
      "logoUrl": "https://cdn.nexgate.com/shops/techstore-logo.jpg",
      "bannerUrl": "https://cdn.nexgate.com/shops/techstore-banner.jpg",
      "status": "ACTIVE",
      "isVerified": true,
      "verificationBadge": "GOLD",
      "trustScore": 4.80,
      "subscriberCount": 1284,
      "subscribedAt": "2026-05-20T14:32:00"
    }
  ],
  "currentPage": 1,
  "pageSize": 10,
  "totalElements": 7,
  "totalPages": 1,
  "hasNext": false,
  "hasPrevious": false
}
}

```

Success Response Fields:

Field	Description
<code>content[].subscriptionId</code>	Unique identifier of this subscription record
<code>content[].shopId</code>	Shop's unique identifier
<code>content[].shopName</code>	Shop display name
<code>content[].shopSlug</code>	Shop URL slug — use for navigating to the shop page
<code>content[].logoUrl</code>	Shop logo image URL
<code>content[].bannerUrl</code>	Shop banner image URL
<code>content[].status</code>	Shop status — <code>ACTIVE</code> , <code>TEMPORARILY_OFFLINE</code> , etc.
<code>content[].isVerified</code>	Whether the shop has passed platform verification

Field	Description
<code>content[].verificationBadge</code>	Verification badge level — <code>BRONZE</code> , <code>SILVER</code> , <code>GOLD</code>
<code>content[].trustScore</code>	Shop trust rating <code>0.00–5.00</code>
<code>content[].subscriberCount</code>	Total subscribers this shop currently has
<code>content[].subscribedAt</code>	ISO 8601 timestamp of when the user subscribed
<code>currentPage</code>	Current page number (1-based)
<code>totalElements</code>	Total shops the user is subscribed to
<code>hasNext</code> / <code>hasPrevious</code>	Pagination navigation flags

Error Response JSON Samples:

Unauthorized (401):

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Token has expired",
  "action_time": "2026-06-04T10:30:45",
  "data": "Token has expired"
}
```

Standard Error Types:

- `401 UNAUTHORIZED`: Token missing, expired, or invalid
- `500 INTERNAL_SERVER_ERROR`: Unexpected server failure

3. Shop Subscribers (Owner Only)

Purpose: Returns a paginated list of users who have subscribed to a specific shop, ordered by most recently subscribed first. Only the shop owner can access this endpoint — calling it for a shop you do not own returns a 403 error.

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

Access Level: `🔒` Protected (shop owner only)

Authentication: Bearer Token (required — must be the owner of the shop)

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer <token>

Path Parameters:

Parameter	Type	Required	Description	Validation
shopId	UUID	Yes	The shop whose subscriber list to retrieve	Caller must own this shop

Query Parameters:

Parameter	Type	Required	Description	Default
page	integer	No	Page number (1-based)	1
size	integer	No	Items per page	10

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Subscribers retrieved successfully",
  "action_time": "2026-06-04T10:30:45",
  "data": {
    "content": [
      {
        "userId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "fullName": "Amina Hassan",
        "userName": "amina.h",
        "avatarUrl": "https://cdn.nexgate.com/avatars/amina-h.jpg",
        "subscribedAt": "2026-06-01T09:45:00"
      },
      {
        "userId": "1cb2e847-3f1a-4bcd-a3fc-9e84f22bce99",
        "fullName": "John Mwangi",
        "userName": "jmwangi",
        "avatarUrl": null,
        "subscribedAt": "2026-05-29T16:12:00"
      }
    ]
  }
}
```

```

    "currentPage": 1,
    "pageSize": 10,
    "totalElements": 1284,
    "totalPages": 129,
    "hasNext": true,
    "hasPrevious": false
  }
}

```

Success Response Fields:

Field	Description
<code>content[].userId</code>	Subscriber's unique account identifier
<code>content[].fullName</code>	Subscriber's first + last name
<code>content[].userName</code>	Subscriber's public username
<code>content[].avatarUrl</code>	Subscriber's profile picture URL — <code>null</code> if no avatar set
<code>content[].subscribedAt</code>	ISO 8601 timestamp of when this user subscribed
<code>currentPage</code>	Current page number (1-based)
<code>totalElements</code>	Total subscriber count for this shop
<code>hasNext</code> / <code>hasPrevious</code>	Pagination navigation flags

Error Response JSON Samples:

Unauthorized (401):

```

{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Token has expired",
  "action_time": "2026-06-04T10:30:45",
  "data": "Token has expired"
}

```

Forbidden — Not the shop owner (403):

```

{
  "success": false,
  "httpStatus": "FORBIDDEN",
  "message": "You do not own this shop",
}

```

```
"action_time": "2026-06-04T10:30:45",
"data": "You do not own this shop"
}
```

Shop Not Found (404):

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Shop not found",
  "action_time": "2026-06-04T10:30:45",
  "data": "Shop not found"
}
```

Standard Error Types:

- `401 UNAUTHORIZED`: Token missing, expired, or invalid
- `403 FORBIDDEN`: Authenticated user does not own this shop
- `404 NOT_FOUND`: Shop does not exist or has been deleted
- `500 INTERNAL_SERVER_ERROR`: Unexpected server failure

isSubscribed & subscriberCount in ShopResponse

Subscription state is automatically embedded in the standard `ShopResponse` returned by all shop detail endpoints — no separate call needed.

```
{
  "shopId": "7cb3a812-...",
  "shopName": "TechStore Tanzania",
  "isVerified": true,
  "trustScore": 4.80,
  "isSubscribed": true,
  "subscriberCount": 1284,
  "productCount": 47
}
```

Field	Type	Description
-------	------	-------------

<code>isSubscribed</code>	boolean	<code>true</code> if the requesting authenticated user is subscribed to this shop. Always <code>false</code> for anonymous users
<code>subscriberCount</code>	long	Total number of subscribers this shop currently has
<code>productCount</code>	long	Number of active (published) products in this shop

Quick Reference — All Subscription Endpoints

#	Endpoint	Method	Auth	Who can call
1	<code>api/v1/e-commerce/shops/{shopId}/subscribe</code>	POST	Required	Any authenticated user
2	<code>api/v1/e-commerce/shops/my-subscriptions</code>	GET	Required	Any authenticated user
3	<code>api/v1/e-commerce/shops/{shopId}/subscribers</code>	GET	Required	Shop owner only