Search Engine

- Search API
- E-Commerce Search Engine

Search API

Author: Josh S. Sakweli, Backend Lead Team
Last Updated: 2025-12-23
Version: v1.0

Base URL: `https://api.nextgate.com/api/v1`

Short Description: Public search endpoints for discovering users and shops on the Nexgate platform. These endpoints power the @mention and $shop autocomplete features across the application.

Hints:

- Use `@` prefix for user search (automatically stripped)
- Use `$` prefix for shop search (automatically stripped)
- Results are ordered by relevance: exact match → starts with → contains
- All endpoints are public and do not require authentication
- Rate limiting applies to prevent abuse

Standard Response Format

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

Success Response Structure

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Operation completed successfully",
  "action_time": "2025-12-23T10:30:45",
  "data": {
    // Actual response data goes here
  }
}
```

Error Response Structure

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Error description",
  "action_time": "2025-12-23T10:30:45",
  "data": "Error description"
}
```

Endpoints

1. Search Users

Purpose: Search for users by username, first name, or last name for @mentions and user discovery.

Endpoint: `GET` `{base_url}/users/search`

Access Level: □□ Public (Rate limited)

Authentication: None

Query Parameters:

Parameter	Type	Required	Description	Validation	Default
q	string	No	Search query (username, first name, or last name). `@` prefix is automatically removed	Min: 1 character after trimming	-
page	integer	No	Page number (1-based)	Min: 1	1
size	integer	No	Number of results per page	Min: 1, Max: 50	10

Success Response JSON Sample:

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

```
"message": "Users retrieved successfully",
"action_time": "2025-12-23T11:14:05.293524449",
"data": {
  "content": [
    {
      "id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
      "userName": "john_doe",
      "displayName": "John Doe",
      "avatarUrl": "https://cdn.nextgate.com/avatars/john_doe.jpg",
      "isVerified": true
    },
    {
      "id": "b2c3d4e5-f6a7-8901-bcde-f12345678901",
      "userName": "johnny_b",
      "displayName": "Johnny Brown",
      "avatarUrl": null,
      "isVerified": false
    }
  ],
  "pageable": {
    "pageNumber": 0,
    "pageSize": 10,
    "sort": {
      "empty": true,
      "sorted": false,
      "unsorted": true
    },
    "offset": 0,
    "paged": true,
    "unpaged": false
  },
  "last": false,
  "totalPages": 3,
  "totalElements": 25,
  "first": true,
  "size": 10,
  "number": 0,
  "numberOfElements": 10,
  "sort": {
    "empty": true,
```

```
      "sorted": false,
      "unsorted": true
    },
    "empty": false
  }
}
```

Success Response Fields:

Field	Description
content	Array of user objects matching the search query
content[].id	Unique identifier of the user
content[].userName	Username of the user
content[].displayName	Full display name (firstName + lastName, or userName if not available)
content[].avatarUrl	URL to user's profile picture (null if not set)
content[].isVerified	Whether the user account is verified
totalElements	Total number of users matching the query
totalPages	Total number of pages available
first	Whether this is the first page
last	Whether this is the last page

Empty Result Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Users retrieved successfully",
  "action_time": "2025-12-23T11:14:05.293524449",
  "data": {
    "content": [],
    "pageable": {
      "pageNumber": 0,
      "pageSize": 10,
      "sort": {
        "empty": true,
        "sorted": false,
        "unsorted": true
      },
```

```
      "offset": 0,
      "paged": true,
      "unpaged": false
    },
    "last": true,
    "totalPages": 0,
    "totalElements": 0,
    "first": true,
    "size": 10,
    "number": 0,
    "numberOfElements": 0,
    "sort": {
      "empty": true,
      "sorted": false,
      "unsorted": true
    },
    "empty": true
  }
}
```


Example Requests:

Use Case	Request
Basic search	`/users/search?q=john`
Search with @ prefix	`/users/search?q=@john`
Paginated search	`/users/search?q=john&page=2&size=20`

---

2. Search Shops

Purpose: Search for shops by name, slug, or tagline for $mentions and shop discovery.

Endpoint: `GET` `{base_url}/shops/search`

Access Level: □□ Public (Rate limited)

Authentication: None

Query Parameters:

Parameter	Type	Required	Description	Validation	Default
q	string	No	Search query (shop name, slug, or tagline). `$` prefix is automatically removed	Min: 1 character after trimming	-
page	integer	No	Page number (1-based)	Min: 1	1
size	integer	No	Number of results per page	Min: 1, Max: 50	10

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Shops retrieved successfully",
  "action_time": "2025-12-23T11:20:30.123456789",
  "data": {
    "content": [
      {
        "shopId": "c3d4e5f6-a7b8-9012-cdef-123456789012",
        "shopName": "Mama's Kitchen",
        "shopSlug": "mamas-kitchen",
        "tagline": "Best local food in town",
        "logoUrl": "https://cdn.nextgate.com/shops/mamas-kitchen/logo.jpg",
        "isVerified": true,
        "verificationBadge": "GOLD",
        "city": "Dar es Salaam"
      },
      {
        "shopId": "d4e5f6a7-b8c9-0123-def0-234567890123",
        "shopName": "Mama Lishe",
        "shopSlug": "mama-lishe",
        "tagline": "Authentic Tanzanian cuisine",
        "logoUrl": null,
        "isVerified": false,
        "verificationBadge": null,
        "city": "Arusha"
      }
    ],
```

```
    "pageable": {
      "pageNumber": 0,
      "pageSize": 10,
      "sort": {
        "empty": true,
        "sorted": false,
        "unsorted": true
      },
      "offset": 0,
      "paged": true,
      "unpaged": false
    },
    "last": false,
    "totalPages": 2,
    "totalElements": 15,
    "first": true,
    "size": 10,
    "number": 0,
    "numberOfElements": 10,
    "sort": {
      "empty": true,
      "sorted": false,
      "unsorted": true
    },
    "empty": false
  }
}
```


Success Response Fields:

Field	Description
content	Array of shop objects matching the search query
content[].shopId	Unique identifier of the shop
content[].shopName	Display name of the shop
content[].shopSlug	URL-friendly unique identifier
content[].tagline	Short promotional text for the shop
content[].logoUrl	URL to shop's logo image (null if not set)
content[].isVerified	Whether the shop is verified
content[].verificationBadge	Verification badge level (GOLD, SILVER, BRONZE, or null)

Field	Description
content[].city	City where the shop is located
totalElements	Total number of shops matching the query
totalPages	Total number of pages available
first	Whether this is the first page
last	Whether this is the last page

Empty Result Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Shops retrieved successfully",
  "action_time": "2025-12-23T11:20:30.123456789",
  "data": {
    "content": [],
    "pageable": {
      "pageNumber": 0,
      "pageSize": 10,
      "sort": {
        "empty": true,
        "sorted": false,
        "unsorted": true
      },
      "offset": 0,
      "paged": true,
      "unpaged": false
    },
    "last": true,
    "totalPages": 0,
    "totalElements": 0,
    "first": true,
    "size": 10,
    "number": 0,
    "numberOfElements": 0,
    "sort": {
      "empty": true,
      "sorted": false,
      "unsorted": true
```

```
    },
    "empty": true
  }
}
```

Example Requests:

Use Case	Request
Basic search	`/shops/search?q=mama`
Search with $ prefix	`/shops/search?q=$mama`
Paginated search	`/shops/search?q=kitchen&page=2&size=20`

Search Ordering Logic

Both search endpoints use smart ordering to return the most relevant results first:

Priority	Match Type	Example (query: "john")
1	Exact match	`john`
2	Starts with	`johnny`, `john_doe`
3	Contains	`big_john`, `_john_`

Results with the same priority are sorted alphabetically.

Error Responses

Standard Error Types:

Application-Level Exceptions (400-499)

- `400 BAD_REQUEST`: Invalid request parameters
- `429 TOO_MANY_REQUESTS`: Rate limit exceeded

Error Response Examples:

Rate Limit Exceeded (429):

```
{
  "success": false,
  "httpStatus": "TOO_MANY_REQUESTS",
  "message": "Rate limit exceeded. Please try again later.",
  "action_time": "2025-12-23T11:30:45",
  "data": "Rate limit exceeded. Please try again later."
}
```

Quick Reference

Feature	User Search	Shop Search
Endpoint	`/users/search`	`/shops/search`
Prefix	`@` (optional)	`$` (optional)
Searches	userName, firstName, lastName	shopName, shopSlug, tagline
Filters	Non-locked users only	Active, non-deleted shops only
Auth	None	None

Future Scaling Strategy

As Nexgate grows, the search system will evolve to handle millions of users and shops while maintaining fast response times. Here's how we'll scale like Instagram and GitHub.

Phase 1: Current Implementation (MVP)

Scale: Up to 100K users/shops

How it works:

- Direct PostgreSQL queries with pattern matching
- Smart ordering: exact match → starts with → contains
- Simple and effective for early-stage growth

Limitations:

- Slows down with large datasets
- No typo tolerance

- No personalization

Phase 2: Database Optimization

Scale: 100K - 1M users/shops

What we'll add:

- **Trigram Indexing (pg_trgm)**: PostgreSQL extension that enables fast fuzzy search and handles typos (e.g., "jonh" finds "john")
- **Redis Caching**: Cache popular search queries and recent searches per user
- **Optimized Indexes**: GIN indexes for faster text matching

Benefits:

- 3-5x faster queries
- Typo tolerance
- Reduced database load

Phase 3: Dedicated Search Engine

Scale: 1M - 100M users/shops

What we'll add:

- **Elasticsearch Cluster**: Dedicated search infrastructure separate from main database
- **Real-time Sync**: Changes in PostgreSQL automatically sync to Elasticsearch
- **Advanced Features**: Autocomplete suggestions, fuzzy matching, relevance scoring

Benefits:

- Sub-10ms response times
- Horizontal scaling (add more nodes as needed)
- Rich search features without impacting main database

Phase 4: Personalized Search (Instagram/GitHub Style)

Scale: 100M+ users/shops

How Instagram does it: They don't search all 2 billion users. Instead, they search within YOUR world first.

Tiered Search Approach:

Priority	Who Gets Searched First	Why
1st	People you message frequently	Strongest signal of relationship
2nd	People you interact with (likes, comments)	Active engagement
3rd	People you follow	Explicit interest
4th	People who follow you	They know you
5th	Global search	Only if above tiers don't have enough results

What we'll add:

- **Interaction Graph**: Track who talks to whom, who views whose profile, who likes whose posts
- **Pre-computed Candidates**: For each user, maintain a list of likely search targets
- **ML Ranking**: Machine learning model to predict who you're most likely searching for
- **Prefix Caching**: Pre-compute results for common prefixes (typing "j" instantly shows your friend "John")

Relevance Scoring:

- 40% - How often you interact with this person/shop
- 30% - How well the query matches their name
- 20% - How recently you interacted
- 10% - Their popularity (followers, verification)

Performance Targets

Phase	Scale	Average Response Time
Phase 1 (Current)	<100K	~50ms
Phase 2 (Optimized DB)	<1M	~30ms
Phase 3 (Elasticsearch)	<100M	~10ms
Phase 4 (Personalized)	100M+	~5ms

Migration Triggers

When to Move	Action
50K users	Add pg_trgm + Redis caching
500K users	Introduce Elasticsearch
5M users	Build interaction graph + personalization

Key Principle: The API stays the same. Users won't notice any changes except faster, smarter results.

E-Commerce Search Engine

Author: Josh S. Sakweli, Backend Lead Team **Last Updated**: 2025-12-30 **Version**: v1.0

Base URL: `https://api.nextgate.com/api/v1`

Short Description: Public search endpoints for discovering shops, products, and active purchase groups on the Nexgate e-commerce platform. These endpoints power the main search functionality, product discovery, and group buying features across the application.

Hints:

- Search query must be at least 2 characters
- Results are ordered alphabetically by name
- Filter by type using the `type` parameter: `SHOP`, `PRODUCT`, or `PURCHASE_GROUP`
- Purchase groups are searched by both name and group code (e.g., `GP-ABC123`)
- All endpoints are public and do not require authentication
- Maximum page size is 50 results
- Rate limiting applies to prevent abuse

Standard Response Format

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

Success Response Structure

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Operation completed successfully",
  "action_time": "2025-12-30T10:30:45",
  "data": {
    // Actual response data goes here
  }
}
```

Error Response Structure

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Error description",
  "action_time": "2025-12-30T10:30:45",
  "data": null
}
```

Search Result Types

Each search result contains a `type` field indicating the result category:

Type	Description	Specific Fields
`SHOP`	Active shop/store	`shop` object
`PRODUCT`	Active product	`product` object
`PURCHASE_GROUP`	Open group purchase	`purchaseGroup` object

Endpoints

---

1. Global Search

Purpose: Search across all types (shops, products, and purchase groups) with optional type filtering

Endpoint: `GET` `{base_url}/api/v1/e-commerce/search`

Access Level: □□ Public (No Authentication Required)

Query Parameters:

Parameter	Type	Required	Default	Description	Validation
query	string	Yes	-	Search term	Min 2 characters
type	string	No	null (all)	Filter by type	SHOP, PRODUCT, PURCHASE_GROUP

Parameter	Type	Required	Default	Description	Validation
page	int	No	1	Page number	Min 1
size	int	No	20	Results per page	Max 50

Request Examples:

```
# Search all types
GET /api/v1/e-commerce/search?query=iphone&page=1&size=20

# Search shops only
GET /api/v1/e-commerce/search?query=fashion&type=SHOP

# Search products only
GET /api/v1/e-commerce/search?query=dress&type=PRODUCT

# Search purchase groups only
GET /api/v1/e-commerce/search?query=GP-ABC123&type=PURCHASE_GROUP
```

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Search results retrieved successfully",
  "action_time": "2025-12-30T10:30:45",
  "data": {
    "content": [
      {
        "type": "SHOP",
        "id": "550e8400-e29b-41d4-a716-446655440000",
        "name": "Fashion Hub TZ",
        "imageUrl": "https://storage.example.com/shops/logo1.jpg",
        "shop": {
          "slug": "fashion-hub-tz",
          "isVerified": true,
          "rating": 4.8,
          "productsCount": 156,
          "location": "Dar es Salaam"
        }
      },
```

```
    {
      "type": "PRODUCT",
      "id": "660e8400-e29b-41d4-a716-446655440001",
      "name": "Summer Fashion Dress",
      "imageUrl": "https://storage.example.com/products/dress1.jpg",
      "product": {
        "slug": "summer-fashion-dress",
        "price": 45000.00,
        "discountPrice": 38000.00,
        "currency": "TZS",
        "inStock": true,
        "rating": 4.5,
        "shopName": "Fashion Hub TZ",
        "shopIsVerified": true
      }
    },
    {
      "type": "PURCHASE_GROUP",
      "id": "770e8400-e29b-41d4-a716-446655440002",
      "name": "Fashion Week Special Deal",
      "imageUrl": "https://storage.example.com/products/bundle.jpg",
      "purchaseGroup": {
        "groupCode": "GP-FWS123",
        "regularPrice": 150000.00,
        "groupPrice": 120000.00,
        "savingsPercentage": 20,
        "seatsRemaining": 3,
        "totalSeats": 10,
        "expiresAt": "2025-12-31T23:59:59",
        "shopName": "Fashion Hub TZ"
      }
    }
  ],
  "page": 0,
  "size": 20,
  "totalElements": 3,
  "totalPages": 1,
  "hasNext": false,
  "hasPrevious": false,
  "shopsCount": 1,
```

```
    "productsCount": 1,
    "purchaseGroupsCount": 1
  }
}
```

Response Fields:

Field	Type	Description
content	array	List of search result items
page	int	Current page number (0-indexed)
size	int	Page size
totalElements	long	Total number of results
totalPages	int	Total number of pages
hasNext	boolean	Whether more pages exist
hasPrevious	boolean	Whether previous pages exist
shopsCount	long	Total shops matching query
productsCount	long	Total products matching query
purchaseGroupsCount	long	Total purchase groups matching query

Error Response JSON Samples:

Query too short:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Search query must be at least 2 characters",
  "action_time": "2025-12-30T10:30:45",
  "data": null
}
```

Invalid type:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Invalid type. Must be: SHOP, PRODUCT, or PURCHASE_GROUP",
  "action_time": "2025-12-30T10:30:45",
```

```
  "data": null
}
```

---

2. Search Shops

Purpose: Search for active shops by name

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

Access Level: □□ Public (No Authentication Required)

Query Parameters:

Parameter	Type	Required	Default	Description	Validation
query	string	Yes	-	Shop name search term	Min 2 characters
page	int	No	1	Page number	Min 1
size	int	No	20	Results per page	Max 50

Request Example:

```
GET /api/v1/e-commerce/search/shops?query=fashion&page=1&size=20
```

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Shops retrieved successfully",
  "action_time": "2025-12-30T10:30:45",
  "data": {
    "content": [
      {
        "type": "SHOP",
        "id": "550e8400-e29b-41d4-a716-446655440000",
        "name": "Fashion Hub TZ",
        "imageUrl": "https://storage.example.com/shops/logo1.jpg",
        "shop": {
          "slug": "fashion-hub-tz",
```

```
          "isVerified": true,
          "rating": 4.8,
          "productsCount": 156,
          "location": "Dar es Salaam"
        }
      },
      {
        "type": "SHOP",
        "id": "660e8400-e29b-41d4-a716-446655440001",
        "name": "Fashion Zone",
        "imageUrl": "https://storage.example.com/shops/logo2.jpg",
        "shop": {
          "slug": "fashion-zone",
          "isVerified": false,
          "rating": 4.2,
          "productsCount": 89,
          "location": "Arusha"
        }
      }
    ],
    "page": 0,
    "size": 20,
    "totalElements": 2,
    "totalPages": 1,
    "hasNext": false,
    "hasPrevious": false,
    "shopsCount": 2,
    "productsCount": 0,
    "purchaseGroupsCount": 0
  }
}
```


Shop Info Fields:

Field	Type	Description
slug	string	URL-friendly shop identifier
isVerified	boolean	Whether shop is verified
rating	double	Shop rating (0-5)
productsCount	int	Number of active products

Field	Type	Description
location	string	Shop location

---

3. Search Products

Purpose: Search for active products by name

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

Access Level: □□ Public (No Authentication Required)

Query Parameters:

Parameter	Type	Required	Default	Description	Validation
query	string	Yes	-	Product name search term	Min 2 characters
page	int	No	1	Page number	Min 1
size	int	No	20	Results per page	Max 50

Request Example:

```
GET /api/v1/e-commerce/search/products?query=dress&page=1&size=20
```

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Products retrieved successfully",
  "action_time": "2025-12-30T10:30:45",
  "data": {
    "content": [
      {
        "type": "PRODUCT",
        "id": "550e8400-e29b-41d4-a716-446655440000",
        "name": "Summer Dress Collection",
        "imageUrl": "https://storage.example.com/products/dress1.jpg",
        "product": {
          "slug": "summer-dress-collection",
```

```
      "price": 45000.00,
      "discountPrice": 38000.00,
      "currency": "TZS",
      "inStock": true,
      "rating": 4.5,
      "shopName": "Fashion Hub TZ",
      "shopIsVerified": true
    }
  },
  {
    "type": "PRODUCT",
    "id": "660e8400-e29b-41d4-a716-446655440001",
    "name": "Evening Dress",
    "imageUrl": "https://storage.example.com/products/dress2.jpg",
    "product": {
      "slug": "evening-dress",
      "price": 85000.00,
      "discountPrice": null,
      "currency": "TZS",
      "inStock": true,
      "rating": 4.8,
      "shopName": "Elegance Store",
      "shopIsVerified": true
    }
  },
  {
    "type": "PRODUCT",
    "id": "770e8400-e29b-41d4-a716-446655440002",
    "name": "Casual Dress",
    "imageUrl": "https://storage.example.com/products/dress3.jpg",
    "product": {
      "slug": "casual-dress",
      "price": 35000.00,
      "discountPrice": 30000.00,
      "currency": "TZS",
      "inStock": false,
      "rating": 4.0,
      "shopName": "Budget Fashion",
      "shopIsVerified": false
    }
```

```
      }
    ],
    "page": 0,
    "size": 20,
    "totalElements": 3,
    "totalPages": 1,
    "hasNext": false,
    "hasPrevious": false,
    "shopsCount": 0,
    "productsCount": 3,
    "purchaseGroupsCount": 0
  }
}
```

Product Info Fields:

Field	Type	Description
slug	string	URL-friendly product identifier
price	decimal	Regular price
discountPrice	decimal	Discounted price (null if no discount)
currency	string	Price currency (e.g., TZS)
inStock	boolean	Whether product is in stock
rating	double	Product rating (0-5)
shopName	string	Name of the shop selling this product
shopIsVerified	boolean	Whether the shop is verified

---

4. Search Purchase Groups

Purpose: Search for active (open, non-expired) purchase groups by name or group code

Endpoint: `GET` `{base_url}/api/v1/e-commerce/search/groups`

Access Level: □□ Public (No Authentication Required)

Query Parameters:

Parameter	Type	Required	Default	Description	Validation

query	string	Yes	-	Group name or code	Min 2 characters
page	int	No	1	Page number	Min 1
size	int	No	20	Results per page	Max 50

Request Examples:

```
# Search by group name
GET /api/v1/e-commerce/search/groups?query=iPhone Deal&page=1&size=20

# Search by group code
GET /api/v1/e-commerce/search/groups?query=GP-ABC123&page=1&size=20
```

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Purchase groups retrieved successfully",
  "action_time": "2025-12-30T10:30:45",
  "data": {
    "content": [
      {
        "type": "PURCHASE_GROUP",
        "id": "550e8400-e29b-41d4-a716-446655440000",
        "name": "iPhone 15 Pro - Dar es Salaam Deal",
        "imageUrl": "https://storage.example.com/products/iphone15.jpg",
        "purchaseGroup": {
          "groupCode": "GP-ABC123",
          "regularPrice": 2500000.00,
          "groupPrice": 2200000.00,
          "savingsPercentage": 12,
          "seatsRemaining": 2,
          "totalSeats": 5,
          "expiresAt": "2025-12-31T08:00:00",
          "shopName": "TechZone TZ"
        }
      },
      {
        "type": "PURCHASE_GROUP",
```

```
        "id": "660e8400-e29b-41d4-a716-446655440001",
        "name": "iPhone 15 Plus Group Buy",
        "imageUrl": "https://storage.example.com/products/iphone15plus.jpg",
        "purchaseGroup": {
          "groupCode": "GP-XYZ789",
          "regularPrice": 2200000.00,
          "groupPrice": 1950000.00,
          "savingsPercentage": 11,
          "seatsRemaining": 4,
          "totalSeats": 8,
          "expiresAt": "2025-12-30T18:00:00",
          "shopName": "Mobile World"
        }
      }
    ],
    "page": 0,
    "size": 20,
    "totalElements": 2,
    "totalPages": 1,
    "hasNext": false,
    "hasPrevious": false,
    "shopsCount": 0,
    "productsCount": 0,
    "purchaseGroupsCount": 2
  }
}
```


Purchase Group Info Fields:

Field	Type	Description
groupCode	string	Unique group code (e.g., GP-ABC123)
regularPrice	decimal	Original product price
groupPrice	decimal	Discounted group price
savingsPercentage	int	Percentage saved
seatsRemaining	int	Available seats
totalSeats	int	Total seats in group
expiresAt	datetime	Group expiration time
shopName	string	Name of the shop

Note: Only returns groups that are:

- Status: `OPEN`
- Not expired (`expiresAt` > now)
- Not deleted

Endpoints Summary

#	Method	Endpoint	Purpose
1	GET	`/api/v1/e-commerce/search`	Search all types
2	GET	`/api/v1/e-commerce/search/shops`	Search shops only
3	GET	`/api/v1/e-commerce/search/products`	Search products only
4	GET	`/api/v1/e-commerce/search/groups`	Search purchase groups only

Common Error Responses

Status	Message	Cause
`400 BAD_REQUEST`	Search query must be at least 2 characters	Query too short
`400 BAD_REQUEST`	Invalid type. Must be: SHOP, PRODUCT, or PURCHASE_GROUP	Invalid type parameter
`429 TOO_MANY_REQUESTS`	Rate limit exceeded	Too many requests

Frontend Usage Examples

JavaScript/TypeScript

```
// Global search
const searchAll = async (query, page = 1, size = 20) => {
  const response = await fetch(
    `/api/v1/e-commerce/search?query=${encodeURIComponent(query)}&page=${page}&size=${size}`
```

```
  );
  return response.json();
};

// Search with type filter
const searchByType = async (query, type, page = 1) => {
  const response = await fetch(
    `/api/v1/e-commerce/search?query=${encodeURIComponent(query)}&type=${type}&page=${page}`
  );
  return response.json();
};

// Search shops only
const searchShops = async (query) => {
  const response = await fetch(
    `/api/v1/e-commerce/search/shops?query=${encodeURIComponent(query)}`
  );
  return response.json();
};

// Search products only
const searchProducts = async (query) => {
  const response = await fetch(
    `/api/v1/e-commerce/search/products?query=${encodeURIComponent(query)}`
  );
  return response.json();
};

// Search groups by name or code
const searchGroups = async (query) => {
  const response = await fetch(
    `/api/v1/e-commerce/search/groups?query=${encodeURIComponent(query)}`
  );
  return response.json();
};
```

Rendering Results

```
const renderResults = (results) => {
  results.content.forEach(item => {
    switch (item.type) {
      case 'SHOP':
        renderShopCard(item);
        break;
      case 'PRODUCT':
        renderProductCard(item);
        break;
      case 'PURCHASE_GROUP':
        renderGroupCard(item);
        break;
    }
  });

  // Show counts
  console.log(`Found: ${results.shopsCount} shops, ${results.productsCount} products,
${results.purchaseGroupsCount} groups`);
};
```

5. Get Product Details by ID

Purpose: Retrieve detailed product information by product ID for public viewing

Endpoint: `GET` `{base_url}/api/v1/e-commerce/search/id/{productId}`

Access Level: □□ Public (No Authentication Required)

Path Parameters:

Parameter	Type	Required	Description	Validation
productId	UUID	Yes	Unique product identifier	Valid UUID format

Request Example:

```
GET /api/v1/e-commerce/search/id/550e8400-e29b-41d4-a716-446655440000
```

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Product retrieved successfully",
  "action_time": "2025-12-30T10:30:45",
  "data": {
    "productId": "550e8400-e29b-41d4-a716-446655440000",
    "productName": "iPhone 15 Pro Max",
    "productSlug": "iphone-15-pro-max",
    "productDescription": "The latest iPhone with advanced features, A17 Pro chip, and titanium design.",
    "shortDescription": "Latest iPhone with A17 Pro chip",
    "productImages": [
      "https://storage.example.com/products/iphone15-1.jpg",
      "https://storage.example.com/products/iphone15-2.jpg",
      "https://storage.example.com/products/iphone15-3.jpg"
    ],
    "price": 2500000.00,
    "comparePrice": 2800000.00,
    "discountPercentage": 10.71,
    "isOnSale": true,
    "stockQuantity": 25,
    "lowStockThreshold": 5,
    "isInStock": true,
    "isLowStock": false,
    "trackInventory": true,
    "brand": "Apple",
    "sku": "IPHONE15PM-256-BLK",
    "condition": "NEW",
    "status": "ACTIVE",
    "tags": ["smartphone", "apple", "iphone", "5g"],
    "metaTitle": "iPhone 15 Pro Max - Buy Online",
    "metaDescription": "Buy the latest iPhone 15 Pro Max with free shipping",
    "shopId": "660e8400-e29b-41d4-a716-446655440001",
    "shopName": "TechZone TZ",
    "shopSlug": "techzone-tz",
    "categoryId": "770e8400-e29b-41d4-a716-446655440002",
    "categoryName": "Smartphones",
    "isFeatured": true,
    "isDigital": false,
```

```
"requiresShipping": true,
"createdAt": "2025-12-01T08:00:00",
"updatedAt": "2025-12-30T10:00:00",
"specifications": {
  "Display": "6.7-inch Super Retina XDR",
  "Chip": "A17 Pro",
  "Storage": "256GB",
  "Camera": "48MP Main + 12MP Ultra Wide + 12MP Telephoto",
  "Battery": "4422 mAh",
  "5G": "Yes"
},
"hasSpecifications": true,
"specificationCount": 6,
"colors": [
  {
    "name": "Natural Titanium",
    "hex": "#9A9A98",
    "images": ["https://storage.example.com/products/iphone15-titanium.jpg"],
    "priceAdjustment": 0.00,
    "finalPrice": 2500000.00,
    "hasExtraFee": false,
    "extraFeeReason": null
  },
  {
    "name": "Blue Titanium",
    "hex": "#394C5F",
    "images": ["https://storage.example.com/products/iphone15-blue.jpg"],
    "priceAdjustment": 50000.00,
    "finalPrice": 2550000.00,
    "hasExtraFee": true,
    "extraFeeReason": "Limited edition color"
  }
],
"hasMultipleColors": true,
"colorCount": 2,
"priceRange": {
  "minPrice": 2500000.00,
  "maxPrice": 2550000.00,
  "priceStartsFrom": 2500000.00,
  "hasPriceVariations": true
```

```
  },
  "orderingLimits": {
    "minOrderQuantity": 1,
    "maxOrderQuantity": 5,
    "canOrderQuantity": 5,
    "maxAllowedQuantity": 5,
    "hasOrderingLimits": true
  },
  "groupBuying": {
    "isEnabled": true,
    "maxGroupSize": 5,
    "groupPrice": 2200000.00,
    "groupDiscount": 300000.00,
    "groupDiscountPercentage": 12.00,
    "timeLimitHours": 48,
    "canJoinGroup": true
  },
  "installmentOptions": {
    "isEnabled": true,
    "isAvailable": true,
    "downPaymentRequired": true,
    "minDownPaymentPercentage": 20.00,
    "plans": [
      {
        "planId": "3-months",
        "duration": 3,
        "interval": "MONTHLY",
        "interestRate": 0.00,
        "description": "3 months interest-free",
        "calculations": {
          "downPayment": 500000.00,
          "remainingAmount": 2000000.00,
          "totalInterest": 0.00,
          "paymentAmount": 666667.00,
          "totalAmount": 2500000.00
        },
        "paymentSchedule": [
          {
            "paymentNumber": 1,
            "amount": 666667.00,
```

```
          "dueDate": "2026-01-30T10:30:45",
          "description": "Payment 1 of 3"
        },
        {
          "paymentNumber": 2,
          "amount": 666667.00,
          "dueDate": "2026-02-28T10:30:45",
          "description": "Payment 2 of 3"
        },
        {
          "paymentNumber": 3,
          "amount": 666666.00,
          "dueDate": "2026-03-30T10:30:45",
          "description": "Payment 3 of 3"
        }
      ],
      "isPopular": true
    },
    {
      "planId": "6-months",
      "duration": 6,
      "interval": "MONTHLY",
      "interestRate": 5.00,
      "description": "6 months with 5% interest",
      "calculations": {
        "downPayment": 500000.00,
        "remainingAmount": 2000000.00,
        "totalInterest": 100000.00,
        "paymentAmount": 350000.00,
        "totalAmount": 2600000.00
      },
      "isPopular": false
    }
  ],
  "eligibilityStatus": "ELIGIBLE",
  "creditCheckRequired": false
},
"purchaseOptions": {
  "canBuyNow": true,
  "canJoinGroup": true,
```

```
      "canPayInstallment": true,
      "recommendedOption": "GROUP_BUYING",
      "bestDeal": {
        "option": "GROUP_BUYING",
        "savings": 300000.00,
        "finalPrice": 2200000.00
      }
    }
  }
}
```

Response Fields:

Basic Information

Field	Type	Description
productId	UUID	Unique product identifier
productName	string	Product name
productSlug	string	URL-friendly product identifier
productDescription	string	Full product description
shortDescription	string	Short product summary
productImages	array	List of product image URLs

Pricing Information

Field	Type	Description
price	decimal	Current selling price
comparePrice	decimal	Original/compare price
discountPercentage	decimal	Discount percentage
isOnSale	boolean	Whether product is on sale

Inventory Information

Field	Type	Description
stockQuantity	int	Available stock
lowStockThreshold	int	Low stock alert threshold

Field	Type	Description
isInStock	boolean	Whether product is in stock
isLowStock	boolean	Whether stock is low
trackInventory	boolean	Whether inventory is tracked

Product Details

Field	Type	Description
brand	string	Product brand
sku	string	Stock keeping unit
condition	enum	NEW, USED, REFURBISHED
status	enum	ACTIVE, INACTIVE, DRAFT

Shop & Category

Field	Type	Description
shopId	UUID	Shop identifier
shopName	string	Shop name
shopSlug	string	Shop URL slug
categoryId	UUID	Category identifier
categoryName	string	Category name

Specifications

Field	Type	Description
specifications	object	Key-value pairs of product specs
hasSpecifications	boolean	Whether product has specs
specificationCount	int	Number of specifications

Colors

Field	Type	Description
colors	array	Available color options
colors[].name	string	Color name

Field	Type	Description
colors[].hex	string	Hex color code
colors[].images	array	Color-specific images
colors[].priceAdjustment	decimal	Price adjustment for color
colors[].finalPrice	decimal	Final price with adjustment
colors[].hasExtraFee	boolean	Whether color has extra fee
hasMultipleColors	boolean	Whether multiple colors exist
colorCount	int	Number of color options

Price Range

Field	Type	Description
priceRange.minPrice	decimal	Minimum price across variants
priceRange.maxPrice	decimal	Maximum price across variants
priceRange.priceStartsFrom	decimal	Starting price
priceRange.hasPriceVariations	boolean	Whether prices vary

Ordering Limits

Field	Type	Description
orderingLimits.minOrderQuantity	int	Minimum order quantity
orderingLimits.maxOrderQuantity	int	Maximum order quantity
orderingLimits.canOrderQuantity	int	Quantity user can order
orderingLimits.hasOrderingLimits	boolean	Whether limits apply

Group Buying

Field	Type	Description
groupBuying.isEnabled	boolean	Whether group buying is enabled
groupBuying.maxGroupSize	int	Maximum group participants
groupBuying.groupPrice	decimal	Group purchase price
groupBuying.groupDiscount	decimal	Savings amount
groupBuying.groupDiscountPercentage	decimal	Savings percentage

Field	Type	Description
groupBuying.timeLimitHours	int	Hours to fill group
groupBuying.canJoinGroup	boolean	Whether user can join

Installment Options

Field	Type	Description
installmentOptions.isEnabled	boolean	Whether installments enabled
installmentOptions.isAvailable	boolean	Whether available for product
installmentOptions.downPaymentRequired	boolean	Whether down payment needed
installmentOptions.minDownPaymentPercentage	decimal	Minimum down payment %
installmentOptions.plans	array	Available payment plans
installmentOptions.eligibilityStatus	string	User eligibility status

Purchase Options Summary

Field	Type	Description
purchaseOptions.canBuyNow	boolean	Can purchase immediately
purchaseOptions.canJoinGroup	boolean	Can join group purchase
purchaseOptions.canPayInstallment	boolean	Can use installments
purchaseOptions.recommendedOption	string	Best option for user
purchaseOptions.bestDeal.option	string	Best deal type
purchaseOptions.bestDeal.savings	decimal	Amount saved
purchaseOptions.bestDeal.finalPrice	decimal	Final price with deal

Error Response JSON Sample:


```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Product not found",
  "action_time": "2025-12-30T10:30:45",
  "data": null
}
```

Standard Error Types:

Status	Message	Cause
`404 NOT_FOUND`	Product not found	Invalid product ID
`400 BAD_REQUEST`	Invalid product ID format	Malformed UUID

Endpoints Summary (Updated)

#	Method	Endpoint	Purpose
1	GET	`/api/v1/e-commerce/search`	Search all types
2	GET	`/api/v1/e-commerce/search/shops`	Search shops only
3	GET	`/api/v1/e-commerce/search/products`	Search products only
4	GET	`/api/v1/e-commerce/search/groups`	Search purchase groups only
5	GET	`/api/v1/e-commerce/search/id/{productId}`	Get product details by ID

Frontend Usage Example

```
// Get full product details after search result click
const getProductDetails = async (productId) => {
  const response = await fetch(
    `/api/v1/e-commerce/search/id/${productId}`
  );
  return response.json();
};

// Usage
const handleSearchResultClick = async (item) => {
  if (item.type === 'PRODUCT') {
    const productDetails = await getProductDetails(item.id);
    showProductModal(productDetails.data);
  }
};
```