

Mục lục

1. THÔNG TIN ENDPOINT & GATEWAY

- 1.1. Cấu trúc URL Endpoint (Đồng bộ tiền tố /api)
- 1.2. Danh sách Gateway Base URL theo môi trường

2. REQUEST SPECIFICATION (THAM SỐ GỬI LÊN)

- 2.1. Cấu trúc JSON Request
- 2.2. Chi tiết các cách Filter (filters)
- 2.3. Chọn trường dữ liệu tối ưu (fields)

3. RESPONSE SPECIFICATION (GIẢI THÍCH CHI TIẾT CÁC FIELD)

- 3.1. Các trường Siêu dữ liệu (Metadata)
- 3.2. Danh sách sản phẩm (products[])
- 3.3. Dữ liệu bộ lọc (facets)

4. VÍ DỤ cURL VÀ RESPONSE THỰC TẾ

- 4.1. cURL: Tìm kiếm cơ bản (Trang đầu tiên)
- 4.2. cURL: Tìm kiếm kết hợp đầy đủ Bộ lọc (Filters + Sort + Fields)
- 4.3. cURL: Cuộn trang tiếp theo (Infinite Scroll bằng Cursor)
- 4.4. Response JSON mẫu thực tế (Trang 1):

5. MÃ LỖI HTTP & NGUYÊN NHÂN TỪ CHỐI (HTTP STATUS CODES)

6. ĐẶC TẢ TÌM KIẾM BẰNG HÌNH ẢNH (IMAGE SEARCH API)

- 6.1. Quy trình 2 pha (Two-phase Workflow)
- 6.2. Cấu trúc Tham số Image Search
- 6.3. Các trường Response bổ sung cho Image Search
- 6.4. Ví dụ cURL Image Search Thực Tế

ĐẶC TẢ KỸ THUẬT API: SEARCH & FILTER (v2.6)

Dành cho: Frontend Engineering Team (Web, Mobile App)
Dịch vụ: AI Search Engine API — LOTTE Mart Vietnam
Phiên bản: 2.6.0 | **Cập nhật:** 11/10/2026 | **Giao thức:** JSON over HTTP/2

1. THÔNG TIN ENDPOINT & GATEWAY

- HTTP Method:** POST
- Accept:** application/json

1.1. Cấu trúc URL Endpoint (Đồng bộ tiền tố /api)

Hệ thống cung cấp 2 dịch vụ tìm kiếm qua API Gateway với định dạng URL đồng bộ:

1. Tìm kiếm Từ khóa & Bộ lọc (Text Search):

- Canonical: POST /api/v2/{lang}/{storeId}/products/search
- Alias: POST /api/v2/p/mart/es/{storeId}/{lang}/products/search

2. Tìm kiếm bằng Hình ảnh (Image Search):

- Canonical: POST /api/v2/{lang}/{storeId}/products/search-by-image
- Alias: POST /api/v2/p/mart/es/{storeId}/{lang}/products/search-by-image

Ghi chú tham số URL: {lang} là mã ngôn ngữ (vi | en | ko), {storeId} là mã siêu thị / kho (vd: nsg — LOTTE Mart Nam Sài Gòn).

1.2. Danh sách Gateway Base URL theo môi trường

Môi trường	Base Gateway URL	URL Text Search	URL Image Search
DEV	https://dev-gateway.martonline.lotte.vn	/api/v2/vi/nsg/products/search	/api/v2/vi/nsg/products/search-by-image
STAGING	https://staging-gateway.martonline.lotte.vn	/api/v2/vi/nsg/products/search	/api/v2/vi/nsg/products/search-by-image
PRODUCTION	https://gateway.martonline.lotte.vn	/api/v2/vi/nsg/products/search	/api/v2/vi/nsg/products/search-by-image

(Trong mạng nội bộ backend k8s / direct service: POST /api/v1/search)

2. REQUEST SPECIFICATION (THAM SỐ GỬI LÊN)

2.1. Cấu trúc JSON Request

Trường root	Kiểu dữ liệu	Bắt buộc	Mặc định	Mô tả & Ràng buộc
query	string	Có	—	Từ khóa tìm kiếm. Tối thiểu 1 ký tự sau trim. Chuỗi rỗng trả về HTTP 400.
pagination Mode	string	Không	"cursor"	Chế độ phân trang: "cursor" (cho infinite scroll) hoặc "page" (cho click số trang).
cursor	string	Không	null	Token phân trang Base64 lấy từ nextCursor của response trước đó (dùng cho mode "cursor").
page	number	Không	1	Số thứ tự trang (khi dùng paginationMode: "page", bắt đầu từ 1).
pageSize	number	Không	20	Số lượng sản phẩm trả về mỗi trang. Tối đa 100.
sort	string	Không	"relevance"	Tiêu chí sắp xếp: "relevance" (liên quan nhất), "price_asc" (giá tăng dần), "price_desc" (giá giảm dần), "newest" (mới nhất).
filters	object	Không	{}	Đối tượng chứa toàn bộ bộ lọc sản phẩm (Xem mục 2.2). Lưu ý: Không dùng tên cũ facetFilters.
fields	string []	Không	All	Danh sách các trường cần lấy để tối ưu băng thông (Xem mục 2.3).

2.2. Chi tiết các cách Filter (filters)

Tất cả các bộ lọc **bắt buộc nằm trong đối tượng filters** . Giữa các bộ lọc khác nhau áp dụng toán tử **AND**.

Trường trong filters	Kiểu dữ liệu	Toán tử	Cú pháp gửi lên	Mô tả hành vi & Quy tắc
viTags	string []	OR	["Sữa Tươi", "Ít Đường"]	Lọc theo tag tiếng Việt. Tối đa 10 items . Sản phẩm khớp nếu có ít nhất 1 tag trong mảng.
enTags	string []	OR	["Fresh Milk"]	Lọc theo tag tiếng Anh. Tối đa 10 items.
krTags	string []	OR	["우유"]	Lọc theo tag tiếng Hàn. Tối đa 10 items.
tags	string []	OR	["Organic"]	Tag alias chung đa ngữ. Tối đa 10 items.
brandId	string	EQUAL	"Vinamilk"	Lọc theo mã thương hiệu (Khớp chính xác).
categoryId	string	PREFIX	"110"	Lọc theo mã ngành hàng. Khớp tiền tố phân cấp (vd: "110" lấy cả con "11001" , "11002").
priceMin	number	≥	50000	Giá tối thiểu (VND). Bỏ qua nếu ≤ 0 .
priceMax	number	≤	200000	Giá tối đa (VND). Phải ≥ priceMin .
inStockOnly	boolean	EQUAL	true	Chỉ trả về sản phẩm còn tồn kho (inStock: true).
delivery	string []	OR	["express"]	Phương thức giao: "express" (siêu tốc), "standard" (tiêu chuẩn).
benefitFlags	string []	OR	["special_price"]	Ưu đãi: "special_price" (giá sốc), "gift" (quà tặng kèm).
ratingMin	number	≥	4.0	Đánh giá tối thiểu từ 1.0 đến 5.0 (vd: 4.0 tương ứng từ 4 sao trở lên).
attrs	object	OR (trong key)	{"brand_origin": ["Vietnam"]}	Thuộc tính động chuẩn hóa (EAV). Hỗ trợ 8 keys: brand_origin, pack_type, volume, weight, flavor, target_user, usage_type, storage_condition .

2.3. Chọn trường dữ liệu tối ưu (fields)

Frontend kiểm soát chính xác payload trả về bằng mảng **fields** :

- **Khối cấp 1:** "products" , "facets" , "suggestedFilters" .
- **Trường trong thẻ sản phẩm:** products.<field> (vd: products.id , products.name , products.price , products.imageUrl , products.inStock ...).
- **Quy tắc tối ưu bằng thông:**
 - **Trang 1:** Gửi fields: ["products" , "facets"] để lấy cả sản phẩm và cây bộ lọc.
 - **Trang 2 trở đi (Infinite scroll):** Gửi fields: ["products"] (bỏ qua tính toán facets, giảm 70% kích thước JSON).

3. RESPONSE SPECIFICATION (GIẢI THÍCH CHI TIẾT CÁC FIELD)

3.1. Các trường Siêu dữ liệu (Metadata)

Các trường này **luôn luôn xuất hiện trong mọi response**, không bị ảnh hưởng bởi tham số `fields` :

Trường Response	Kiểu dữ liệu	Ý nghĩa & Mục đích sử dụng
<code>queryId</code>	<code>string</code>	UUID phiên tìm kiếm. Dùng để gửi kèm event theo dõi log / click-through tracking.
<code>totalHits</code>	<code>number</code>	Tổng số lượng sản phẩm thỏa mãn toàn bộ từ khóa và điều kiện lọc. Dùng hiển thị "Tìm thấy X sản phẩm".
<code>tookMs</code>	<code>number</code>	Thời gian OpenSearch thực thi truy vấn (tính bằng milliseconds).
<code>paginationMode</code>	<code>string</code>	Chế độ phân trang đang chạy (<code>"cursor"</code> hoặc <code>"page"</code>).
<code>page</code>	<code>number</code>	Số trang hiện tại (chỉ có khi <code>paginationMode: "page"</code>).
<code>pageSize</code>	<code>number</code>	Số sản phẩm được cấu hình trả về trên mỗi trang.
<code>hasMore</code>	<code>boolean</code>	<code>true</code> nếu vẫn còn sản phẩm ở các trang tiếp theo; <code>false</code> nếu đã đến trang cuối cùng. Dùng để ngắt trigger infinite scroll.
<code>nextCursor</code>	<code>string</code>	Chuỗi token Opaque Base64 đại diện cho vị trí tiếp theo. Frontend giữ nguyên giá trị này và truyền vào <code>cursor</code> ở request tiếp theo.
<code>resolvedMode</code>	<code>string</code>	Mode thuật toán thực thi nội bộ (<code>"text"</code> , <code>"hybrid"</code> ...).
<code>corrected</code>	<code>boolean</code>	<code>true</code> nếu từ khóa người dùng bị sai chính tả và hệ thống đã tự động sửa; <code>false</code> nếu từ khóa gốc được giữ nguyên.
<code>normalizedQuery</code>	<code>string</code>	Chuỗi từ khóa sau khi đã chuẩn hóa và tự động sửa lỗi chính tả. Dùng hiển thị: <i>"Kết quả cho từ khóa: ... "</i>
<code>didYouMean</code>	<code>string</code>	Từ khóa gợi ý nếu hệ thống phát hiện từ khóa khả dĩ khác. Nếu có, dùng hiển thị: <i>"Có phải bạn muốn tìm: ... "</i>

3.2. Danh sách sản phẩm (products[])

Mỗi phần tử trong mảng `products` đại diện cho một thẻ sản phẩm hoàn chỉnh:

Trường	Kiểu dữ liệu	Ý nghĩa
<code>id</code>	<code>string</code>	Mã SKU định danh duy nhất của sản phẩm.
<code>name</code>	<code>string</code>	Tên sản phẩm đầy đủ.
<code>price</code>	<code>number</code>	Giá bán thực tế hiện tại (VND).
<code>originalPrice</code>	<code>number</code>	Giá niêm yết ban đầu trước khi giảm giá (VND).
<code>discountRate</code>	<code>number</code>	Phần trăm giảm giá (vd: <code>15</code> tức là giảm 15%).
<code>imageUrl</code>	<code>string</code>	Đường dẫn ảnh đại diện sản phẩm (CDN).
<code>brandId</code>	<code>string</code>	Mã định danh thương hiệu.
<code>brandName</code>	<code>string</code>	Tên hiển thị của thương hiệu.
<code>categoryId</code>	<code>string</code>	Mã ngành hàng của sản phẩm.
<code>categoryName</code>	<code>string</code>	Tên ngành hàng phân cấp.
<code>inStock</code>	<code>boolean</code>	Trạng thái tồn kho: <code>true</code> = còn hàng, <code>false</code> = hết hàng.
<code>stockQty</code>	<code>number</code>	Số lượng tồn kho thực tế khả dụng.
<code>rating</code>	<code>number</code>	Điểm đánh giá trung bình của sản phẩm (thang điểm 1.0 đến 5.0).
<code>reviewCount</code>	<code>number</code>	Tổng số lượt khách hàng đánh giá sản phẩm.
<code>deliveryMethod</code>	<code>string[]</code>	Các hình thức giao hàng hỗ trợ (vd: <code>["express", "standard"]</code>).
<code>isNew</code>	<code>boolean</code>	<code>true</code> nếu là sản phẩm mới về. Dùng gắn badge "MỚI".
<code>isBest</code>	<code>boolean</code>	<code>true</code> nếu là sản phẩm bán chạy. Dùng gắn badge "BEST SELLER".
<code>isPromotion</code>	<code>boolean</code>	<code>true</code> nếu sản phẩm đang áp dụng chương trình khuyến mãi.
<code>tags</code>	<code>string[]</code>	Danh sách các tag nổi bật của sản phẩm.

3.3. Dữ liệu bộ lọc (facets)

Đối tượng `facets` chứa số lượng đếm (aggregation count) của các bộ lọc để render sidebar và chip lọc:

Nhóm Facet	Cấu trúc dữ liệu	Chi tiết & Cách xử lý ở Frontend
<code>viTags</code> , <code>enTags</code> , <code>krTags</code> , <code>tags</code>	<code>FacetBucket[]</code>	Danh sách các tag và số lượng SP khớp. Cơ chế Self-exclusion: Tag đang được chọn vẫn hiển thị các tag anh em kèm số lượng nếu chọn thêm.
<code>brands</code>	<code>FacetBucket[]</code>	Danh sách thương hiệu khớp. Gồm <code>key</code> (mã gửi lên <code>filters.brandId</code>), <code>count</code> , và <code>label</code> .
<code>categories</code>	<code>FacetBucket[]</code>	Danh sách ngành hàng khớp. Gồm <code>key</code> (mã gửi lên <code>filters.categoryId</code>), <code>count</code> , và <code>label</code> .
<code>priceRanges</code>	<code>FacetBucket[]</code>	Các phân khúc giá được tính toán sẵn.
<code>inStock</code>	<code>FacetBucket[]</code>	Số lượng sản phẩm còn hàng (<code>key</code> : <code>"true"</code>).
<code>delivery</code>	<code>FacetBucket[]</code>	Số lượng sản phẩm theo từng hình thức giao (<code>key</code> : <code>"express"</code> , <code>"standard"</code>).
<code>benefitFlags</code>	<code>FacetBucket[]</code>	Số lượng sản phẩm có ưu đãi (<code>key</code> : <code>"special_price"</code> , <code>"gift"</code>).
<code>rating</code>	<code>FacetBucket[]</code>	Số lượng sản phẩm theo các mức sao.
<code>attrs</code>	<code>Record<string, FacetBucket[]></code>	LƯU Ý: Đây là Object Map , KHÔNG PHẢI MẢNG. Duyệt bằng <code>Object.entries(facets.attrs)</code> . Mỗi key là tên thuộc tính (vd: <code>"brand_origin"</code>), value là mảng <code>FacetBucket[]</code> .

Mỗi phần tử `FacetBucket` có cấu trúc:

```
{
  "key": "Vinamilk",
  "count": 45,
  "label": "Vinamilk"
}
```

4. VÍ DỤ cURL VÀ RESPONSE THỰC TẾ

4.1. cURL: Tìm kiếm cơ bản (Trang đầu tiên)

```
curl -X POST "https://dev-gateway.martonline.lotte.vn/api/v2/vi/nsg/products/search" \
-H "Content-Type: application/json" \
-d '{
  "query": "sữa tươi",
  "paginationMode": "cursor",
  "pageSize": 20,
  "fields": ["products", "facets"]
}'
```

4.2. cURL: Tìm kiếm kết hợp đầy đủ Bộ lọc (Filters + Sort + Fields)

```
curl -X POST "https://dev-gateway.martonline.lotte.vn/api/v2/vi/nsg/products/search" \
-H "Content-Type: application/json" \
-d '{
  "query": "sữa tươi",
  "paginationMode": "cursor",
  "pageSize": 20,
  "sort": "relevance",
  "filters": {
    "viTags": ["Sữa Tươi", "Ít Đường"],
    "brandId": "Vinamilk",
    "priceMin": 20000,
    "priceMax": 100000,
    "inStockOnly": true,
    "delivery": ["express"]
  },
  "fields": ["products", "facets"]
}'
```

4.3. cURL: Cuộn trang tiếp theo (Infinite Scroll bằng Cursor)

```
curl -X POST "https://dev-gateway.martonline.lotte.vn/api/v2/vi/nsg/products/search" \
-H "Content-Type: application/json" \
-d '{
  "query": "sữa tươi",
  "paginationMode": "cursor",
  "cursor": "ZXlKaGJHY2lPaUpTVXpVeE1pSXNJbXR...",
  "pageSize": 20,
  "fields": ["products"]
}'
```

4.4. Response JSON mẫu thực tế (Trang 1):

```

HTTP/2 200 OK
Content-Type: application/json; charset=utf-8

{
  "queryId": "e3b0c442-98fc-1c14-9afe-432100000000",
  "totalHits": 45,
  "tookMs": 18,
  "paginationMode": "cursor",
  "pageSize": 20,
  "hasMore": true,
  "nextCursor": "ZXlKaGJHY2lPaUpTVXpVeE1pSXNJbXR...",
  "resolvedMode": "hybrid",
  "corrected": false,
  "normalizedQuery": "sữa tươi",
  "products": [
    {
      "id": "8934673123456",
      "name": "Sữa Tươi Tiệt Trùng Vinamilk 100% Ít Đường 1L",
      "price": 36500,
      "originalPrice": 41000,
      "discountRate": 11,
      "imageUrl": "https://img.martonline.lotte.vn/p/8934673123456.jpg",
      "brandId": "Vinamilk",
      "brandName": "Vinamilk",
      "categoryId": "11001",
      "categoryName": "Sữa Tươi Tiệt Trùng",
      "inStock": true,
      "rating": 4.8,
      "reviewCount": 124,
      "deliveryMethod": ["express", "standard"],
      "isPromotion": true,
      "tags": ["Sữa Tươi", "Ít Đường"]
    }
  ],
  "facets": {
    "viTags": [
      { "key": "Sữa Tươi", "count": 45, "label": "Sữa Tươi" },
      { "key": "Ít Đường", "count": 18, "label": "Ít Đường" }
    ],
    "brands": [
      { "key": "Vinamilk", "count": 45, "label": "Vinamilk" }
    ],
    "attrs": {
      "brand_origin": [
        { "key": "Vietnam", "count": 45, "label": "Việt Nam" }
      ]
    }
  }
}

```

5. MÃ LỖI HTTP & NGUYÊN NHÂN TỪ CHỐI (HTTP STATUS CODES)

Mã HTTP	Tình huống	Thông điệp lỗi mẫu	Nguyên nhân & Cách khắc phục
200 OK	Thành công	Dữ liệu theo schema <code>SearchResponse</code> .	Xử lý bình thường.
400 Bad Request	Từ khóa rỗng	<code>{"error": "query must not be empty"}</code>	Giá trị <code>query</code> bị rỗng sau khi trim. Client phải kiểm tra <code>query.trim()</code> trước khi gửi.
400 Bad Request	Sai cấu trúc	<code>{"error": "unknown field \"facetFilters\"}"}</code>	Gửi tham số ngoài schema. Bắt buộc đổi tên thành <code>filters</code> . Không gửi trường dư thừa.
400 Bad Request	Quá giới hạn tag	<code>{"error": "filters.viTags accepts at most 10 items, got 11"}</code>	Mảng tag vượt quá giới hạn 10 phần tử. Không thể tối đa 10 tag được kích hoạt trên UI.
400 Bad Request	Sai cursor	<code>{"error": "invalid cursor token"}</code>	Token <code>cursor</code> bị lỗi hoặc sửa đổi. Gửi đúng chuỗi nguyên bản từ <code>nextCursor</code> .
500 Internal Error	Lỗi máy chủ	<code>{"error": "Internal search error"}</code>	Lỗi hạ tầng backend. Hiển thị thông báo thân thiện và nút thử lại.

6. ĐẶC TẢ TÌM KIẾM BẰNG HÌNH ẢNH (IMAGE SEARCH API)

6.1. Quy trình 2 pha (Two-phase Workflow)

Quy trình tìm kiếm ảnh được thiết kế tối ưu bằng thông mạng và độ trễ theo 2 pha:

- Pha 1 (Trang đầu tiên):** Client gửi request dạng `multipart/form-data` chứa file ảnh `image` và tham số lọc `params` (JSON). Response trả về danh sách sản phẩm cùng mã `imageRef` (vector cache key).
- Pha 2 (Trang 2+ Infinite Scroll):** Client gửi request dạng `application/json` thông thường mang theo `imageRef` và `cursor`. **Tuyệt đối không cần upload lại file ảnh**, máy chủ tự động tra cứu vector đã tính toán trong bộ nhớ cache.

6.2. Cấu trúc Tham số Image Search

Pha 1: Upload ảnh (`multipart/form-data`)

- Part `image` (Bắt buộc):** Binary dữ liệu ảnh (hỗ trợ `image/jpeg`, `image/png`, `image/webp`). Dung lượng tối đa 5MB (*khuyến nghị Frontend / App resize cạnh dài về khoảng 800px trước khi upload*).
- Part `params` (Tùy chọn):** Chuỗi JSON định nghĩa các tiêu chí hỗ trợ:

```
{
  "query": "loại không đường",           // (Tùy chọn) Text đi kèm để lọc hẹp trong kết quả ảnh
  "paginationMode": "cursor",             // Mặc định: "cursor"
  "pageSize": 20,                         // Số sản phẩm mỗi trang (mặc định 20, max 100)
  "filters": {                             // Bộ lọc (y hệt cấu trúc filters của Text Search)
    "inStockOnly": true,
    "brandId": "Vinamilk"
  },
  "fields": ["products", "facets"]        // Chọn trường cần nhận
}
```

Pha 2: Phân trang tiếp theo (application/json)

Khi người dùng cuộn trang tiếp theo, gửi body JSON:

```
{
  "imageRef": "sha256:9f2a4c8e71b2d3...", // Token nhận từ response Pha 1 (BẮT BUỘC)
  "paginationMode": "cursor",
  "cursor": "ZXlKaGJHY2lPaUpTVXpVeE1pSXNJbXR...", // nextCursor từ trang trước
  "pageSize": 20,
  "fields": ["products"] // Tiết kiệm băng thông: chỉ lấy products
}
```

6.3. Các trường Response bổ sung cho Image Search

Response của Image Search kế thừa toàn bộ cấu trúc của Text Search (products[] , facets , totalHits , tookMs , nextCursor , hasMore), kèm theo các trường đặc thù:

Trường Response	Kiểu dữ liệu	Ý nghĩa & Cách xử lý
imageRef	string	Mã hash SHA256 định danh vector của bức ảnh trong cache. Client lưu chuỗi này để gửi lên ở request Pha 2+.
resolvedImage Mode	string	Thuật toán vector ảnh thực thi nội bộ (vd: "vector_titan_multi").
caption	object null	Thông tin chú thích nếu hệ thống chạy ở chế độ AI Caption (gồm query , outcome , modelId). Thường là null ở chế độ vector thông thường.

6.4. Ví dụ cURL Image Search Thực Tế

cURL Pha 1: Upload ảnh tìm kiếm lần đầu (Multipart)

```
curl -X POST "https://dev-gateway.martonline.lotte.vn/api/v2/vi/nsg/products/search-by-image" \
-F "image=@san-pham.jpg;type=image/jpeg" \
-F 'params={
  "paginationMode": "cursor",
  "pageSize": 20,
  "filters": {
    "inStockOnly": true
  },
  "fields": ["products", "facets"]
}'
```

cURL Pha 2: Cuộn xem trang tiếp theo (JSON dùng `imageRef`)

```
curl -X POST "https://dev-gateway.martonline.lotte.vn/api/v2/vi/nsg/products/search-by-image" \
-H "Content-Type: application/json" \
-d '{
  "imageRef": "sha256:9f2a4c8e71b2d3e4f5a6b7c8d9e0f1a2b3c4d5e6f7a8b9c0d1e2f3a4b5c6d7e8",
  "paginationMode": "cursor",
  "cursor": "ZXlKaGJHY2lPaUpTVXpVeE1pSXNJbXR...",
  "pageSize": 20,
  "fields": ["products"]
}'
```