



Connecting Your Product to XOLogic

# Product API Guide

This guide provides everything you need to integrate with XOLogic's Product API, including authentication, endpoints, query parameters, and response formatting. Designed for third-party developers, it's packed with examples and best practices to get you up and running fast.

The Product API uses the REST standard over https for API calls. Each call to the API should be formatted as a URL with query string parameters and responses are returned as JSON by default but in some cases you can request returns as CSV. All examples are given in CURL for convenience. All requests should use IPv4, as IPv6 is not supported.



# Table of Contents

---

Quick Start .....	3
Accessing the API Documentation .....	3
Base URL for API Requests .....	3
Authentication .....	4
Product API Endpoints .....	4
Query Parameters .....	4
Retrieving Product Data with getAll and sort.....	5
Filtering by Recent Updates with lastMod .....	6
Formatting Responses with FlatFormat.....	7
StandardData vs. TaxonomyData.....	8
TaxonomyData (Hierarchical JSON).....	9
Additional Product Data .....	9
Key Identifiers.....	10
Example API Calls.....	10
Troubleshooting Common Errors.....	12
Best Practices .....	13
A Note About IPv6.....	13
Glossary.....	13
Support.....	13
Changelog.....	13

## Quick Start

New to the Product API? Here's a simple cURL request to fetch 5 products with basic fields:

```
"https://CLIENT_DATABASE_NAME.xologic.com/api/v2/  
product?getAll=1&fields=XOItemID,ItemName,Price&limit=5" \-H "Authorization: Bearer YOUR_ACCESS_  
TOKEN"
```

**Response:**

```
[  
  {  
    "XOItemID": "19-10",  
    "ItemName": "Chandelier 3Lt",  
    "Price": 299.99  
  },  
  {  
    "XOItemID": "19-11",  
    "ItemName": "Pendant Light",  
    "Price": 199.99  
  }  
]
```

Contact XOLogic Client Services at [clientservices@xologic.com](mailto:clientservices@xologic.com) to get your access token and whitelist your IP. For more details, see the Authentication section.

## Accessing the API Documentation

Each client accesses the API documentation via their unique database name as the subdomain::

```
https://YOUR_DATABASE_NAME.xologic.com/api/doc
```

This document is not meant to be an exhaustive list of all API endpoints available. A full list of these endpoints are available via the url above.

New customers: Your Implementation Manager or Client Services will provide your database name. Existing clients, contact [clientservices@xologic.com](mailto:clientservices@xologic.com) if you need it.

## Base URL for API Requests

Use your client database name in the base URL for all API requests:

```
https://YOUR_DATABASE_NAME.xologic.com
```

This ensures you interact with your specific data environment.

# Authentication

## Getting Your Access Token

The Product API requires a whitelisted IP address associated with a Bearer token for access. For convenience, this token does not expire. Contact [clientservices@xologic.com](mailto:clientservices@xologic.com) and list each IP address (IPv4 only) you intend to use to initiate this process.

## Using the Access Token

Include the token in the **Authorization** header of every request::

```
Authorization: Bearer YOUR_ACCESS_TOKEN
```

### Example:

```
curl  
"https://CLIENT_DATABASE_NAME.xologic.com/api/v2/product?getAll=1&fields=VendorName,ItemNum-  
ber&limit=20" \  
-H "Authorization: Bearer YOUR_ACCESS_TOKEN" \  

```

## Product API Endpoints

Parameter	Type	Description
GET	<a href="#">/api/v2/product</a>	Retrieve product data
GET	<a href="#">/api/v2/product/{productID}</a>	Retrieve a single product by ID
POST	<a href="#">/api/v2/product</a>	Create a new product
PUT	<a href="#">/api/v2/product/{productID}</a>	Update an existing product
DELETE	<a href="#">/api/v2/product</a>	Delete a product

## Query Parameters

Use query parameters to filter, sort, and refine GET requests.

### Required Parameters

- **fields**: Specifies fields to return (e.g., XOIItemID, ItemName, Price)
- **limit**: Number of records to return (max 500)

### Filtering & Search Parameters

Parameter	Type	Description
keyword	string	Search by name, SKU, or description
vendor	string	Filter by Vendor ID (e.g., 3442)
categoryPath	string	Filter by category path
minPrice	decimal	Products with price above this value
maxPrice	decimal	Products with price below this value

### Example:

```
GET
/api/v2/product?fields=VendorName,ItemNumber,Price&keyword=chandelier&minPrice=100&max-
Price=500
```

## Retrieving Product Data with getAll and sort

To fetch product data, use getAll=1 or sort=none. Each has distinct filtering and performance traits:

Option	Retrieves All Data	Filtering Options	Sorting Method	Use Case
getAll=1	Yes	Limited: vendor, catalogID, fields, limit, offset, includeDiscontinued, lastMod	MiniBeast (fast)	Fast bulk retrieval with minimal filtering
sort=none	Yes (in chunks)	Full: all filters (e.g., keyword, attribute)	Beast (flexible)	Large datasets with complex filtering
sort=FeatS	No (limited)	Some filters (e.g., keyword, minPrice)	Featured Sort (web-like)	Mimics website search results

- **getAll=1:** Fetches all products quickly using MiniBeast (optimized query engine) but limits filters. Invalid filters return an error.
- **sort=none:** Retrieves all data in chunks with Beast (flexible engine), supporting all filters. Best for complex queries.
- **sort=FeatS:** Prioritizes featured products, like website search. Doesn't guarantee all data.

### Example (getAll=1):

```
curl
"https://YOUR_DATABASE_NAME.xologic.com/api/v2/product?getAll=1&vendor=3442&fields=X-
OItemID,ItemName,Price&limit=100" \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

### Example (sort=none):

```
curl
"https://YOUR_DATABASE_NAME.xologic.com/api/v2/product?sort=none&attribute=48&grouping=all&-
fields=XOItemID,ItemName,Price&limit=100" \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

### Response:

```
[
  {
    "XOItemID": "19-10",
    "ItemName": "Chandelier 3Lt",
    "Price": 299.99
  },
  {
    "XOItemID": "19-11",
    "ItemName": "Pendant Light",
    "Price": 199.99
  }
]
```

For other `sort` options (e.g., `Price`, `PriceD`, `Width`), see [https://YOUR\\_DATABASE\\_NAME.xologic.com/api/doc](https://YOUR_DATABASE_NAME.xologic.com/api/doc).

## Filtering by Recent Updates with `lastMod`

The `lastMod` filter retrieves products modified within a specific time frame, checking three timestamps:

- **Availability Changed Date:** Stock or attribute updates.
- **Catalog Changed Date:** Catalog detail changes.
- **Price Changed Date:** Price updates.

**Note:** The API doesn't specify which timestamp triggered a match (due to query optimization). Include all three timestamps in fields for clarity. Timestamps are in UTC.

### Supported Time Expressions:

`second(s)`, `minute(s)`, `hour(s)`, `day(s)`, `week(s)`, `month(s)`, `year(s)`

### Example:

```
curl
"https://YOUR_DATABASE_NAME.xologic.com/api/v2/product?sort=none&fields=XOItemID,ItemName,-
Price,AvailabilityChangedDate,CatalogChangedDate,PriceChangedDate&limit=100&lastMod=-3hours" \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

### Response:

```
[
  {
    "XOItemID": "19-10",
    "ItemName": "Chandelier 3Lt",
    "Price": 299.99,
    "AvailabilityChangedDate": "2024-02-07T15:30:00Z",
    "CatalogChangedDate": "2024-01-25T10:15:00Z",
    "PriceChangedDate": "2024-02-05T08:00:00Z"
  }
]
```

## Formatting Responses with FlatFormat

The `FlatFormat` parameter customizes response structure for easier parsing. Apply it to the entire response or specific data types.

### Format Options

Type	Behavior
<code>o</code>	Original format (default, pipe-separated with with ' ')
<code>s</code>	Split labels and values into separate entries
<code>p</code>	Split labels (combine multiple values with with ' ')
<code>k</code>	Key-value pairs
<code>n</code>	Default JSON (no flattening)

### Supported Data Types

- StandardData
- ImportantData
- DimensionData
- VariantData
- DynamicData
- ExtraData
- FileData
- ImageData
- AttributeData
- IdentifierData

### Apply to Entire Response

Syntax:

```
&FlatFormat=p
```

Example:

```
GET /api/v2/product/3442-27913?fields=XOItemID,StandardData,FileData,ImageData&FlatFormat=p
```

Response:

```
[
  {
    "XOItemID": "3442-27913"
    "StandardData-Category": "Sconces",
    "StandardData-Finish": "Black | Nickel",
    "FileData": "FileName:https://example/SpecFiles/SP-284-1W.pdf | FileDescr:Spec Sheet",
    "ImageData": "FileName:284-1W-BPN-G165.jpg | FilePath:http://example.com/vendors/large/284-1W-BPN-G165.jpg"
  }
]
```

## Apply to Specific Data Types

### Syntax:

```
&FlatFormat[StandardData]=p
```

### Example:

```
GET /api/v2/product/3442-27913?fields=XOItemID,StandardData,FileData,ImageData&FlatFormat[StandardData]=p&FlatFormat[FileData]=n&FlatFormat[ImageData]=k
```

### Response:

```
[
  {
    "XOItemID": "3442-27913",
    "StandardData-Category": "Sconces",
    "StandardData-Finish": "Black | Nickel",
    "FileData": [
      {
        "FileName": "https://us1.hostedftp.com/SpecFiles/SP-284-1W.pdf",
        "FileDescr": "Spec Sheet"
      }
    ],
    "ImageData-1-FileName": "284-1W-BPN-G165.jpg",
    "ImageData-1-FilePath": "http://example.com/vendors/large/284-1W-BPN-G165.jpg"
  }
]
```

## StandardData vs. TaxonomyData

Product attributes can be retrieved in two formats:

### StandardData (Flat, Pipe-Separated)

- **Use Case:** Quick lookups, simple integrations.
- **Format:** Flat fields with | for multiple values.

### Example:

```
{
  "XOItemID": "3442-27913",
  "StandardData-Category": "Sconces",
  "StandardData-Finish": "Black | Nickel",
  "StandardData-Style": "Transitional | Industrial"
}
```

## TaxonomyData (Hierarchical JSON)

- **Use Case:** Detailed category hierarchies or relationships.
- **Format:** Nested JSON with category structure.

Example:

```
{
  "TaxonomyData": {
    "Category": {
      "Sconces": {
        "desc": "Sconces",
        "SortID": "15",
        "Subcategory": {
          "Wall Sconces": {
            "desc": "Wall Sconces",
            "SortID": "166"
          }
        }
      }
    }
  }
}
```

### Choosing Between Them:

Scenario	Recommended	Example Fields
Quick category access	StandardData	StandardData-Category
Full hierarchy	TaxonomyData	TaxonomyData.Category.Subcategory
Key identifiers	StandardData	XOItemID, ItemNumber

## Additional Product Data

### Pricing Information

Scenario	Recommended
ItemCost	Base cost from manufacturer
VendorDiscountedCost	Cost after vendor discounts
CostWithShipping	Cost including shipping fees
Price	Consumer sale price
WebPrice	Online sale price
MSRP	Manufacturer's Suggested Retail Price
IMAP	Internet Minimum Advertised Price

**Note:** Pricing is updated daily. Contact [clientservices@xologic.com](mailto:clientservices@xologic.com) for real-time update options.

## Stock & Availability

Field	Description
InStock	>0: Available, -1: In stock (unknown quantity), 0: Out of stock
BackOrderDate	Expected restock date (UTC)
Discontinued	Indicates if product is discontinued

## Key Identifiers

Option	Retrieves All Data	Filtering Options	Sorting Method
XOItemID	String	19-10	Primary unique product ID (VendorID + VItemID)
VendorID	integer	19	Unique vendor ID
VItemID	integer	10	Primary ID within vendor's catalog
ItemID	integer	123456	Unique identifier for product iteration
ItemNumber	string	2343NI	Vendor's assigned item number
BaseItemNumber	string	2343	Groups similar products
GTIN	string	00345678901234	Global Trade Item Number (UPC, EAN, etc.)

## Example API Calls

### Retrieve a List of Products

```
curl
"https://YOUR_DATABASE_NAME.xologic.com/api/v2/product?sort=none&getAll=1&fields=Vendor-
Name,ItemNumber,Price&limit=5" \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

### Response:

```
[
  {
    "VendorName": "Kichler",
    "ItemNumber": "2343NI",
    "Price": 199.99
  },
  {
    "VendorName": "Maxim Lighting",
    "ItemNumber": "5842MROI",
    "Price": 149.00
  }
]
```

## Retrieve a Single Product

```
curl -X GET "https://YOUR_DATABASE_NAME.xologic.com/api/v2/product/19-2088?fields=Vendor-Name,ItemNumber,Description" \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

### Response:

```
{
  "VendorName": "Kichler",
  "ItemNumber": "19-2088",
  "Description": "Chandelier 3Lt Brushed Nickel"
}
```

## Create a Product

```
curl -X POST "https://YOUR_DATABASE_NAME.xologic.com/api/v2/product" \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "VendorName": "Kichler",
  "ItemNumber": "9999ABC",
  "ItemName": "Example Item",
  "ShortDescription": "Example Description",
  "Price": 299.99
}'
```

### Response:

```
{
  "message": "Product created successfully",
  "XOItemID": "19-9999ABC"
}
```

## Update a Product

```
curl -X PUT
"https://YOUR_DATABASE_NAME.xologic.com/api/v2/product/19-9999ABC" \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "Price": 279.99
}'
```

### Response:

```
{
  "message": "Product updated successfully"
}
```

## Delete a Product

```
curl -X DELETE "https://YOUR_DATABASE_NAME.xologic.com/api/v2/product/19-9999ABC" \
-H "Authorization: Bearer YOUR_ACCESS_TOKEN"
```

### Response:

```
{
  "message": "Product deleted successfully"
}
```

## Troubleshooting Common Errors

Error Code	Reason	Solution
401 Not Authorized	IPv6 IP used	Verify your request is not using your IPv6 interface (which would not match the whitelisted IP)
401 Unauthorized	Invalid/missing token	Verify <b>Authorization: Bearer YOUR_ACCESS_TOKEN</b>
403 Forbidden	IP not whitelisted	Contact <a href="mailto:clientservices@xologic.com">clientservices@xologic.com</a> to whitelist your IP
400 Bad Request	Invalid parameters	Check syntax, e.g., <b>fields=InvalidField</b> causes this. Ensure valid fields like <b>XOItemID</b> .
404 Not Found	Product not found	Verify productID exists, e.g., 19-2088
500 Internal Server Error	Server issue	Retry after 5 minutes; contact <a href="mailto:clientservices@xologic.com">clientservices@xologic.com</a> if persistent!

## Debugging Tips:

- Use **curl -v** for verbose output to inspect headers and responses.
- Test with Postman to validate query syntax.
- Adding **-H "Content-Type: application/json"** to the request is required when passing in a json body such as in a POST/PUT request.
- Example 400 Bad Request:

```
GET /api/v2/product?getAll=1&keyword=chandelier fails because keyword isn't allowed with getAll=1.
Use sort=none instead.
```

## Best Practices

- **Choose Retrieval Method:** Use `getAll=1` for fast bulk retrieval with `vendor` or `catalogID`. Use `sort=none` for complex filtering.
- **Filter Fields:** Specify only needed fields (e.g., `fields=VendorName,Price`) to boost performance.
- **Paginate:** Use `limit` (max 500) and `offset` for large datasets (e.g., `limit=100&offset=200`). This example would fetch the third set (or “page”) of results. To fetch the first set of results, set the offset to 0 and for the second set of results set the offset to 100, and so on.
- **Monitor Updates:** Use `lastMod` (e.g., `lastMod=-3hours`) with all timestamp fields.
- **Optimize Formatting:** Use `FlatFormat` (e.g., `&FlatFormat[StandardData]=p`) for your app’s needs.
- **Rate Limits:** Please do not exceed 20 requests per second for fast responses and 1 request per second for slow responses. Abuse will likely get your account throttled. Contact [clientservices@xologic.com](mailto:clientservices@xologic.com) for any questions.

## A Note About IPv6

Frequently there are issues with IPv6 IP addresses being used when not intended. This leads to a **401 Not Authorized** response. If the operating system is configured to use both, it is difficult to know which one is likely to be used. If you are using curl, you can add a `-4` parameter to test if this is happening because of IPv6 being used. If you are using Postman or similar tool, you can inspect the previous request and see what IP address was used. But many tools such as Postman do not allow you to force outbound calls to use the IPv4 interface, so many clients just end up disabling IPv6 on their Windows/Mac computers for troubleshooting.

## Glossary

- **XOItemID:** Primary product ID (VendorID + VItemID, e.g., `19-10`).
- **VendorID:** Unique vendor identifier (e.g., `19`).
- **XOItemID:** Vendor’s catalog-specific product ID (e.g., `10`).
- **MiniBeast:** Optimized query engine for fast bulk retrieval with `getAll=1`.
- **Beast:** Flexible query engine for `sort=none`, supporting all filters.
- **Featured Sort:** Search engine for `sort=FeatS`, using your Featured Sort options mimicking website results.

## Support

For assistance, contact XOLogic Client Services at [clientservices@xologic.com](mailto:clientservices@xologic.com) or visit the Swagger API Reference at [https://CLIENT\\_DATABASE\\_NAME.xologic.com/api/doc](https://CLIENT_DATABASE_NAME.xologic.com/api/doc). Join our developer community at <https://community.xologic.com> for tips and discussions.

## Changelog

**August 2025:** Added Quick Start, glossary, and enhanced troubleshooting. Consolidated `getAll/sort`, `FlatFormat`, and best practices. Fixed typos and standardized field names.

**Jan 2026:** Added details about issues with IPv6 addresses, clarified Rate Limits, Authorization, and various other sections.