

This is the API reference for the Bloomreach Product Search and Category APIs, which customers can use to retrieve data about products that match on a specific search query. This page provides essential information about the API endpoint and parameters.

Live at <https://documentation.bloomreach.com/discovery/reference/product-search-category-api>

Product Search and Category API

GET <https://core.dxpapi.com/api/v1/core/>


RECIPES

Build a Product Search Results Page - API Request
[Open Recipe >](#)

Build a Product Search Results Page - API Response
[Open Recipe >](#)

YOUR REQUEST HISTORY

0 Calls

7 Days 

Your API calls will appear here. Make a request to get started!



For detailed information on how Bloomreach Search works, how you can optimize your search quality, and how to troubleshoot search issues, go through the [Search Quality Guides](#).

What do the Product Search and Category APIs do?

The **Product Search API** sends a keyword query to retrieve results about products and other content on your site. You can apply additional filters and facets to return products and content that either include or exclude specified attributes like colors and brands.

The **Category API** sends a category query to retrieve results about products within the categories that you specify.

Endpoints

Production: <https://core.dxpapi.com/api/v1/core/>

Staging: <https://staging-core.dxpapi.com/api/v1/core/>

This reference takes advantage of Readme.io's API documentation features, such as displaying the API method and endpoint automatically below the title.

The recipes provided here walk through an example request and response.

The same API is used for both the Product Search and Category APIs, depending on the value defined in the `search_type`.

Therefore, I decided to combine them into a single reference page, and provided an overview of each API here.

Production and staging endpoints differ, so they are highlighted in their own section.

Example requests

Product Search request for "dresses"

Category request for "cat000922"

```
GET https://core.dxpapi.com/api/v1/core/?
account_id=<Bloomreach provided account ID>
&auth_key=jazzhands
&domain_key=example_com
&request_id=8438674018839
&_br_uid_2=uid=7797686432023:v=11.5:ts=1428617911187:hc=55
&ref_url=http://www.example.com/home
&url=http://www.example.com/index.html?q=dresses
&request_type=search #Value must be search for Product Search API
&search_type=keyword #Value must be keyword for Product Search API
&q=dresses
&fl=pid,title,brand,price,sale_price,thumb_image,url,description
&rows=10
&start=0
```



API requests should be limited to 16K Bytes. API requests above this length will throw [Error Code 414](#).

I provided one request each for the Product Search and Category APIs.

The error code link goes to the error code reference page, which provides more details.

I link to that page instead of reusing the content here for the sake of brevity.

Native mobile app and non-JS integration

API requests for native mobile apps or non-JS environments need different values for some parameters:

- **_br_uid_2**: The value should be the same as described on the [Cookie page](#). Generate a br_uid_2 for your app user and store it in the app permanently if a br_uid_2 does not exist. This br_uid_2 should be used on the corresponding API calls for the app. If the user is on iPhone and resets their IDFA, or if the user is on an Android device and resets their advertising settings: the br_uid_2 should be erased and regenerated the next time the app is set.
- **ref_url**: Leave your ref_url parameters empty.
- **url**: Use a dummy value for your url parameters, such as `exampledomain.app`.

Buy Online Pick-up In Store (BOPIS)

With this feature, you can allow your user to search products from across your sites (online) and have it picked up physically from any of your stores nearest to the user (offline).

BOPIS uses the following parameters:

- **ll** - BOPIS-specific parameter to specify the end-customer's latitude-longitude.
- **fl** - Returns the distance from the point specified in the ll parameter.
- **fq** - Allows filtering by distance from the point specified in the ll parameter.

BOPIS is highlighted here in order to provide an accompanying sample request, which cannot be implemented in the parameter table.

```
GET https://core.dxpapi.com/api/v1/core/?
account_id=<Bloomreach provided account ID>
&auth_key=jazzhands
&domain_key=example_com
&request_id=8438674518839
&br_uid_2=uid=7797686432023:v=11.5:ts=1428617911187:hc=55
&ref_url=http://www.example.com/home
&url=http://www.example.com/index.html?q=dresses
&request_type=search
&ll=38.880657,-77.396935 #latitude-longitude of the end customer
&search_type=keyword
&q=dresses
&fl=store_lat_lon, pid #Returns the distance from ll
&fq=store_lat_lon:"100" #Enables filtering by distance from ll
&rows=10
&start=0
```

Read more about this feature on the [BOPIS feature page](#).

Timeout

We recommend that you configure the application client timeout value to *at least 2 seconds*. If needed, you can set the client timeout value to more than 2 seconds depending on your use case.

If there's a timeout during a search or category page load, then we recommend that you redirect your customer with an HTTP 302 response code to your site's home page. Append this specific tracking parameter to the redirected URL: `?_thto=1`. This parameter serves a few purposes:

- This ensures that your customers always receive a page load.
- It also collects data to assist Bloomreach in troubleshooting timeouts.

For example, let's say that the following page encounters a timeout error when loading:

<http://www.example.com/search?q=shoes>

Then the site redirects the customer to the home page with a 302 response code and uses the following URL:

http://www.example.com/?_thto=1



Please note that **the correct domain_key is mandatory** in the API request. Starting March 2023, sending API requests without domain_key will return a 4xx error response.

QUERY PARAMS

account_id int32 required	6702
<p>Your site's numerical Bloomreach account ID. Your Bloomreach representative gives your site's account ID to you before or during your integration kickoff meeting.</p> <p>The example value shown here, 6702, is included for your convenience to send a request with Try It.</p> <p><div background-color="red">Callout</p>	
auth_key string required	
<p>The Bloomreach-provided authentication key for the Bloomreach account that's sending the request.</p> <p>Pass the auth_key with an empty value in client-side calls. The auth_key value is a private authorization key. If you include your valid auth_key value in client-side calls, then you inadvertently expose that private information to everybody.</p>	
_br_uid_2 string required	uid%3D779768643
<p>A first-party cookie created by the Bloomreach tracking pixel library (BrTrk). This cookie creates a unique, anonymous identifier for every browser or device.</p> <p>Use the default value provided, which is already encoded.</p>	
domain_key string required	documentation_sit
<p>Your site domain's ID, which is provided by Bloomreach. This ID is for the domain that you want to receive your Bloomreach API requests. This parameter identifies the specific site version when the one account ID hosts multiple site versions with unique characteristics, such as language versions. Bloomreach uses your domain_key parameter value to ensure that only the data that pertains to that site version is used for query and analytics features, such as autosuggestions.</p> <p>The example value shown here, <i>documentation_site</i>, is included for your convenience to send a request with Try It.</p>	
fl string required	pid
<p>The attributes that you want returned in your API response, such as product IDs and prices.</p> <p>All fl parameters for Product Search or Category requests must include pid as one of their values. Any attribute from your product feed may be used as a value for fl.</p> <p>Multiple values should be comma separated, such as fl=pid,price.</p>	

This table contains the parameter information, and is automatically generated by Readme.io. The parameter name, type, and required tag are all generated automatically.

The parameter descriptions were originally written before I joined, so as part of our migration to Readme.io, I rewrote them for clarity and consistency across our API reference pages.

Note that Readme.io has a "Try It" feature to send a request and receive a response within the documentation. To ensure a valid response is returned, some parameters contain prefilled values.

Do note that it is possible to get a response without providing values for every required parameter.

q string required Your site visitor's search query. Search queries are composed of one or more terms. For Category queries, the value is the category ID. You can percent encode spaces between terms as %20, or you can leave the spaces unencoded. If you use q=*, the latency of the response will vary depending on your catalog size and it may not adhere to Bloomreach's standard SLA. Additionally, most merchandising operations do not work on * query parameters, except for include/exclude operations. The example value shown here, <i>cable</i> , is included for your convenience to send a request with Try It .	<input type="text" value="cable"/>
ref_url string required The URL of the page or HTTP referrer where the request is started.	<input type="text"/>
request_id string required An ID to track site visitor clicks. We recommend that you generate unique, random values of 13 digits to enable click-tracking. Bloomreach doesn't automatically enforce the requirements for this parameter. For example, you can enter test as your value for each instance of the request_id parameter without triggering an error message. However, using a unique value allows us to help you if you encounter a problem.	<input type="text"/>
request_type string required The type of API request. Value should be search for Product Search or Category requests.	<input type="text" value="search"/>
rows int32 required The number of matching items to return per results page in the API response. The maximum value is 200. To enhance performance, limit this value to the number of items that you think is reasonable for a single page of search results.	<input type="text"/>
search_type string required The type of search. Value should be keyword for Product Search requests, category for Category requests.	<input type="text" value="keyword"/>
start int32 required The number of the first item on a page of results. For example, the first item on the first page is 0, making the start value also 0. The maximum value is 10000.	<input type="text"/>
url string required The absolute URL of the page where the request is initiated. Do not use a relative URL. The example value shown here, https://www.documentation-site.com , is included for your convenience to send a request with Try It .	<input type="text" value="https://www.docur"/>

I listed the parameters in alphabetical order, with required parameters first. (Readme.io does not automatically order them).

search_type is an enumerated string that determines whether the API request is for Product Search or for Category.

callback string

Indicates whether to return data wrapped in the function for cross-origin requests.

For server-side requests, use the value **br_server**. For native-app requests, use the value **br_app**.

efq string

Applies a complex boolean filter to search results to include or exclude items that fit your parameter values. Any product attribute in your product feed is valid, such as brand names and sizes.

Typically, the efq parameter is used for custom attributes that you include in your product feed to support additional business logic that you might need to filter. Read more about using the efq parameter in the "Complex boolean filtering" section in the [Faceting and filtering page](#).

facet.range string

Return a count of ranged facets, such as price and sale price. Use numeric attributes only.

You need to parse the values that are in the facets_counts section of the response. The facet_queries section has custom range facets for numeric fields that you define in your request. The facet_fields section gives you facets that you can display to your site's users, such as brands and colors.

fq string

The fq parameter applies a faceted filter to the returned products, searching for products that fit your parameter values.

Any facet that you want to filter must be in your feed. Read more about using the fq parameter in the "Simple Filtering" section in the [Faceting and filtering page](#).

Attributes must be enabled and mapped by Bloomreach. Let your Bloomreach representative know which attributes in your product feed you want to apply as filters to search results.

ll string

The latitude-longitude of the end-customer used for the [Buy Online Pick-up In Store \(BOPIS\) feature](#).

Value should be provided as latitude,longitude. For example, **ll=11.09,10.018**.

Use **fl=store_lat_lon** to return the distance from ll. In the response body, this value is returned as an additional field with the suffix 'distance'.

Use **fq=store_lat_lon** to enable filtering by distance.

sort string

Sorts results based on the field value in ascending, descending, or another combination of orders. You can sort any fl field.

Value is a field name, followed by **asc/desc** for ascending/descending order respectively. For example, **sort=sale_price desc** sorts in descending order of the sale price.

stats.field string

This parameter allows you to display the maximum and minimum values of any numeric field in your data set for a user query. With this parameter, you can get all the documents matching a query and display the minimum and maximum values of single-valued, numeric attributes such as price, sale_price, length, width, reviews, etc.

Values are returned in the response as **stats_field**.

user_id string

The universal customer ID of the user. You only need to pass this field if your particular integration tracks customers this way. The parameter captures user IDs from the customer side, and reuses the information when powering apps or enhancing cross-device linking. In this way, Bloomreach recognizes users in a way that's aligned with your system.

Use an anonymous string. Don't use email or other personally identifiable information. If you do not track users this way, then omit this field.

view_id string

A unique identifier for a specific view of your product catalog. If you have multiple versions of your site, each with their own product catalog characteristics like product titles and prices, then add view_id to your call.

Bloomreach uses your view_id parameter value to display the right product information for your customers based on their individual site views. You can enter any string value to identify the specific site catalog view. This string must be consistent in your pixel, API, and product catalog.

widget_id string

The widget_id provided in the Dashboard for the Dynamic Widgets feature, which is used to provided curated results.

This is an optional feature that can be enabled by discussing with your CSM.