Bloomreach Discovery

Discovery Cartridge Integration guide

Version 23.1.0

Table of Contents

1. Summary

The Bloomreach Discovery cartridge is built for Salesforce Commerce cloud and allows you to quickly kickstart integration of Bloomreach Discovery into a storefront. The cartridge includes instructions, code samples and best practices for an integration project.

Bloomreach Discovery offers AI-Driven product discovery focused on ROI. It leverages Loomi, our industry leading AI for ecommerce, to further drive product discovery. The Bloomreach Discovery x Shopify connector significantly simplifies your implementation to make sure pixel data from your storefront is fed appropriately into Bloomreach Discovery. Build your storefront quicker with a templated search UI and components for search results pages, facets and autosuggest.

Glossary

Term	Description
SFCC	Salesforce Commerce Cloud
Business Manager (BM)	The primary tool used to configure the SFCC platform and customer storefront SFRA
SFRA	Storefront Reference Architecture, a modern architecture for building the SFCC storefronts.
SFCC BM *or* Business Manager *or* BM	Salesforce Commerce Cloud Business Manages is the web application and the primary tool used to configure the SFCC ecommerce instances and customer storefronts.
UX Studio	An Eclipse UI extension for SFCC development and working with the sandboxes.

2. Component Overview

Functional Overview

The Bloomreach service allows the customer to use a search engine on your website.

Limitations, Constraints

Use of the Bloomreach cartridge requires credentials and API keys from Bloomreach. Please note that in order to have consistency of data flow and authorized request, Bloomreach Discovery integration requires having Bloomreach Core integration in place.

Contract

To work with the cartridge, you need to contact the Bloomreach support service to get your personal credentials to gain access to the Bloomreach services.

Compatibility

The cartridge is designed for Salesforce Commerce Cloud API version 23.9 (Compatibility Mode: 22.7).

Privacy, Payment

The Bloomreach cartridge doesn't collect and process user profile information or billing information. For additional privacy information, please contact your Bloomreach Account Manager.

3. Implementation Guide

Prerequisites

This Integration Guide is intended for developers with basic knowledge and skills of working with SFCC. Before starting work, make sure that:

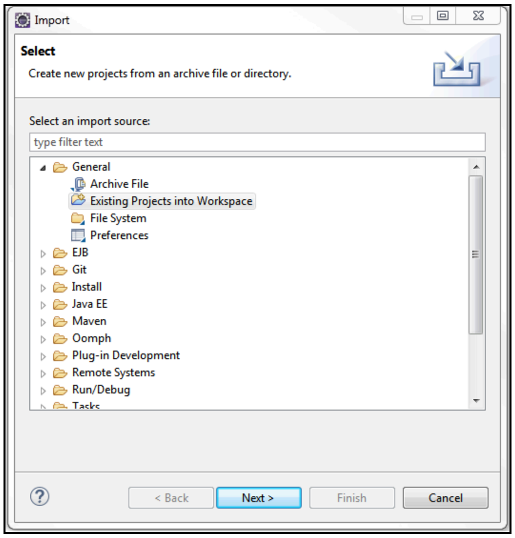
- You have access to a Salesforce Commerce cloud instance.
- Installed Eclipse IDE with Salesforce UX studio plugin.

The cartridge installation process is described for using Eclipse IDE with the Salesforce UX studio plugin, but you can use other IDEs as well, for example Visual Studio Code.

Setup of Business Manager

Importing cartridges

1. Download the cartridge source code from the github: https://github.com/bloomreach/discovery-sfcc-b2c
2. Establish a new digital server connection with your SFCC Instance
3. Import cartridges to a workspace in Salesforce UX Studio.



Import

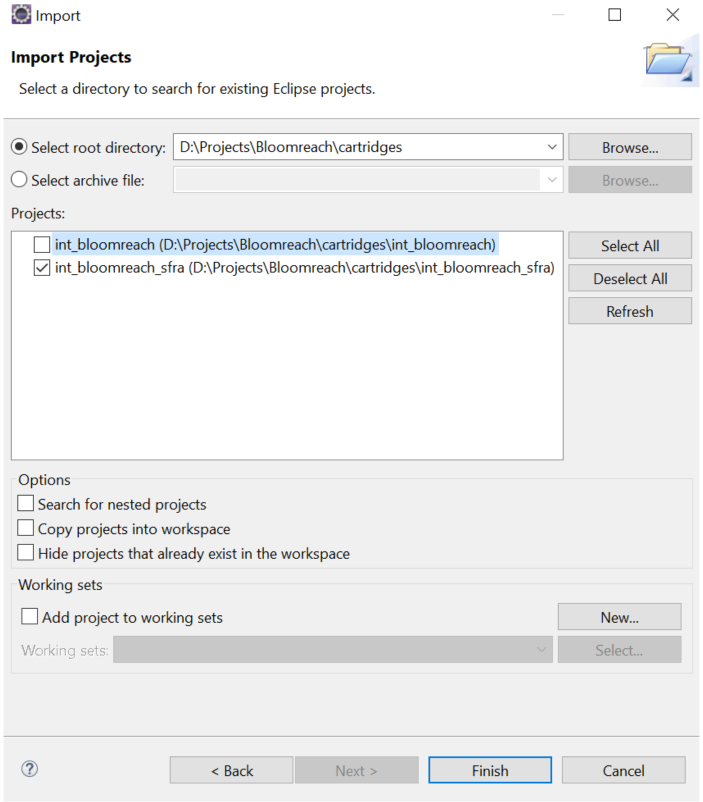
Select

Create new projects from an archive file or directory.

Select an import source:

type filter text

4. Add cartridges to Project Reference of Server Connection.



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: D:\Projects\Bloomreach\cartridges

○ Select archive file:

Projects:

□ int_bloomreach (D:\Projects\Bloomreach\cartridges\int_bloomreach)

☑ int_bloomreach_sfra (D:\Projects\Bloomreach\cartridges\int_bloomreach_sfra)

Options

□ Search for nested projects

□ Copy projects into workspace

□ Hide projects that already exist in the workspace

Working sets

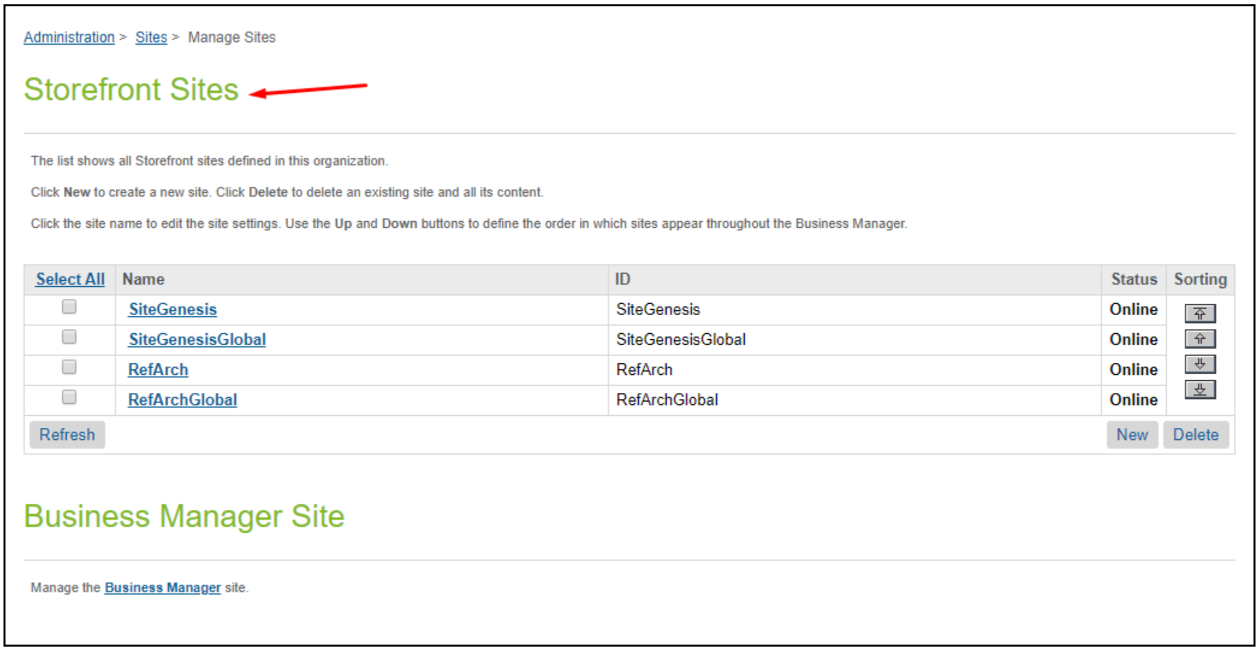
□ Add project to working sets

Working sets:

5. Wait until Studio completes the workspace build and uploads source codes to the sandbox.

Assigning cartridges to the site

Go to *Administration > Sites > Manage Sites > Select your site* from the list “Storefront Sites”.



Administration > Sites > Manage Sites

Storefront Sites

The list shows all Storefront sites defined in this organization.

Click **New** to create a new site. Click **Delete** to delete an existing site and all its content.

Click the site name to edit the site settings. Use the **Up** and **Down** buttons to define the order in which sites appear throughout the Business Manager.

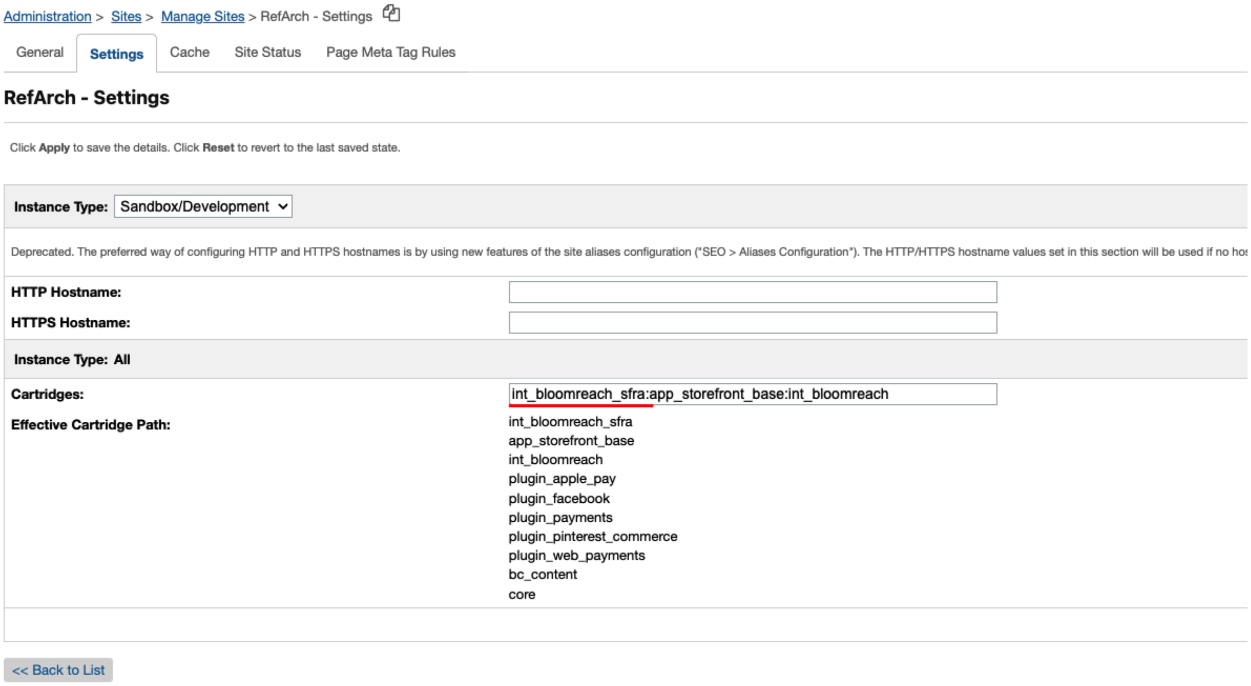
Select All	Name	ID	Status	Sorting
□	SiteGenesis	SiteGenesis	**Online**	⤒
□	SiteGenesisGlobal	SiteGenesisGlobal	**Online**	↑
□	RefArch	RefArch	**Online**	↓
□	RefArchGlobal	RefArchGlobal	**Online**	⤓

Business Manager Site

Manage the **Business Manager** site.

Select the Settings tab. In the beginning of the cartridge path add the following:

```
int_bloomreach_sfra:
```



RefArch - Settings

Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

Instance Type: Sandbox/Development

Deprecated. The preferred way of configuring HTTP and HTTPS hostnames is by using new features of the site aliases configuration ("SEO > Aliases Configuration"). The HTTP/HTTPS hostname values set in this section will be used if no hos

HTTP Hostname:

HTTPS Hostname:

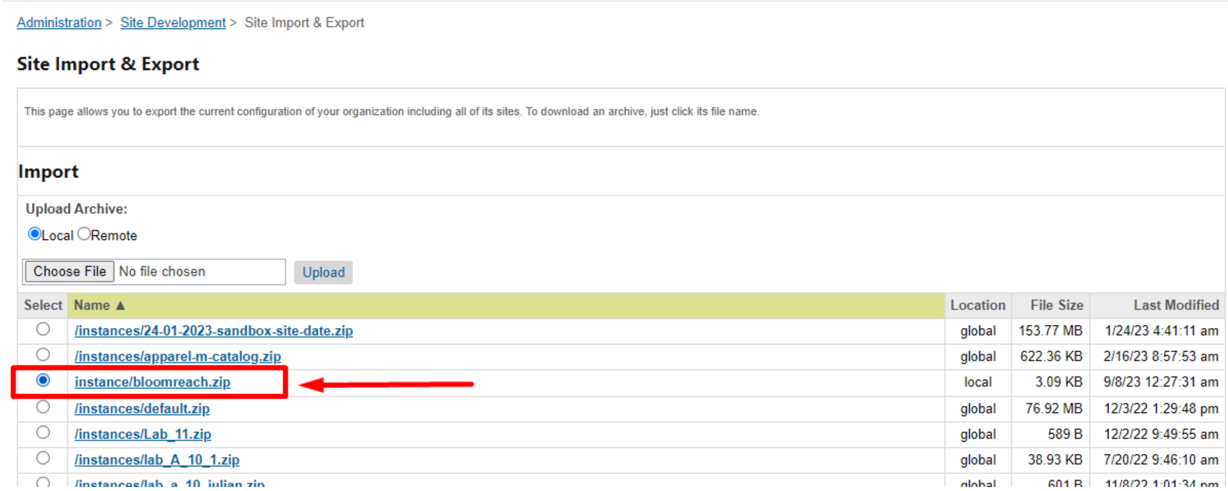
Instance Type: All

Cartridges: int_bloomreach_sfra:app_storefront_base:int_bloomreach

Effective Cartridge Path:
int_bloomreach_sfra
app_storefront_base
int_bloomreach
plugin_apple_pay
plugin_facebook
plugin_payments
plugin_pinterest_commerce
plugin_web_payments
bc_content
core

Import of metadata

Go to *Administration > Site Development > Site Import & Export.* Upload *bloomreach.zip* file from the metadata folder.



Administration > Site Development > Site Import & Export

Site Import & Export

This page allows you to export the current configuration of your organization including all of its sites. To download an archive, just click its file name.

Import

Upload Archive:

● Local ○ Remote

Choose File No file chosen Upload

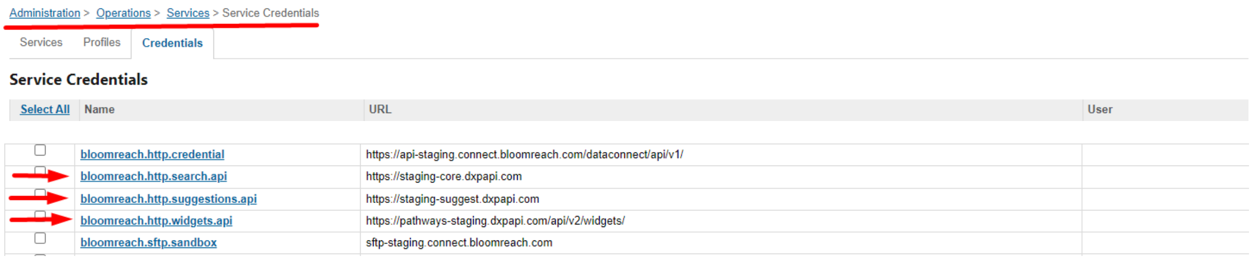
Select	Name ▲	Location	File Size	Last Modified
○	/instances/24-01-2023-sandbox-site-date.zip	global	153.77 MB	1/24/23 4:41:11 am
○	/instances/apparel-m-catalog.zip	global	622.36 KB	2/16/23 8:57:53 am
●	instance/bloomreach.zip	local	3.09 KB	9/8/23 12:27:31 am
○	/instances/default.zip	global	76.92 MB	12/3/22 1:29:48 pm
○	/instances/Lab_11.zip	global	589 B	12/2/22 9:49:55 am
○	/instances/lab_A_10_1.zip	global	38.93 KB	7/20/22 9:46:10 am
○	/instances/lab_a_10_julian.zip	global	601 B	11/8/22 1:01:34 pm

Select *bloomreach.zip*, click Import button and finish the import process

Service

You have to set the Bloomreach API endpoints URL.

Log in to the SFCC BM and go to *Administration > Operations > Services* and select the *Credentials* tab.



Administration > Operations > Services > Service Credentials

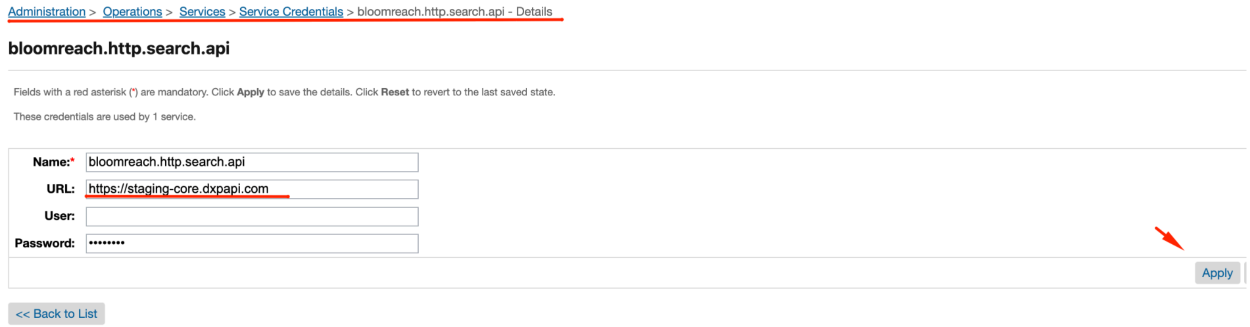
Services Profiles **Credentials**

Service Credentials

Select All	Name	URL	User
□	bloomreach.http.credential	https://api-staging.connect.bloomreach.com/dataconnect/api/v1/	
□	bloomreach.http.search.api	https://staging-core.dxpapi.com	
□	bloomreach.http.suggestions.api	https://staging-suggest.dxpapi.com	
□	bloomreach.http.widgets.api	https://pathways-staging.dxpapi.com/api/v2/widgets/	
□	bloomreach.sftp.sandbox	sftp-staging.connect.bloomreach.com	

Click to the *bloomreach.http.search.api*. Set the URL of the HTTP server for the Production or Staging environment and click the Apply button.

- Production: https://core.dxpapi.com
- Staging: https://staging-core.dxpapi.com


Administration > Operations > Services > Service Credentials > bloomreach.http.search.api - Details

bloomreach.http.search.api

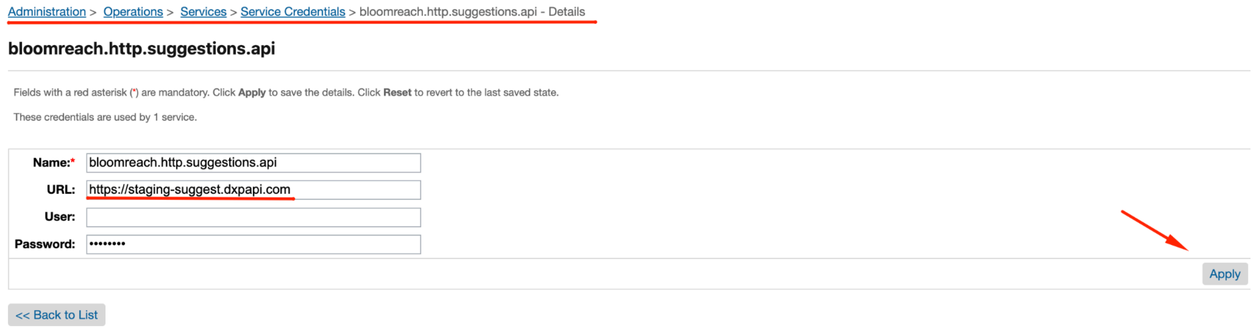
Fields with a red asterisk (*) are mandatory. Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

These credentials are used by 1 service.

Name:*	bloomreach.http.search.api
URL:	https://staging-core.dxpapi.com
User:	
Password:	••••••••

Back to the credential tab and click to the *bloomreach.http.suggestions.api*. Set the URL of the API endpoint for the Production or Staging environment and click the Apply button.

- Production: https://suggest.dxpapi.com
- Staging: https://staging-suggest.dxpapi.com


Administration > Operations > Services > Service Credentials > bloomreach.http.suggestions.api - Details

bloomreach.http.suggestions.api

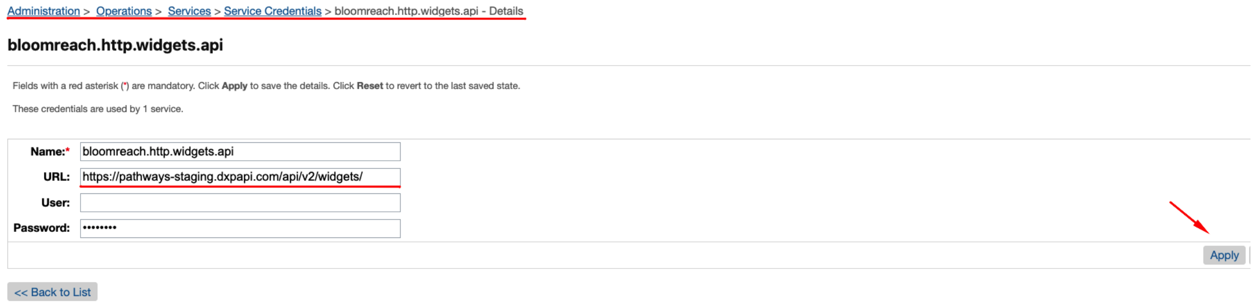
Fields with a red asterisk (*) are mandatory. Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

These credentials are used by 1 service.

Name:*	bloomreach.http.suggestions.api
URL:	https://staging-suggest.dxpapi.com
User:	
Password:	••••••••

Back to the credential tab and click to the *bloomreach.http.widgets.api*. Set the URL of the API endpoint for the Production or Staging environment and click the Apply button.

- Production: https://pathways.dxpapi.com/api/v2/widgets/
- Staging: https://pathways-staging.dxpapi.com/api/v2/widgets/


Administration > Operations > Services > Service Credentials > bloomreach.http.widgets.api - Details

bloomreach.http.widgets.api

Fields with a red asterisk (*) are mandatory. Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

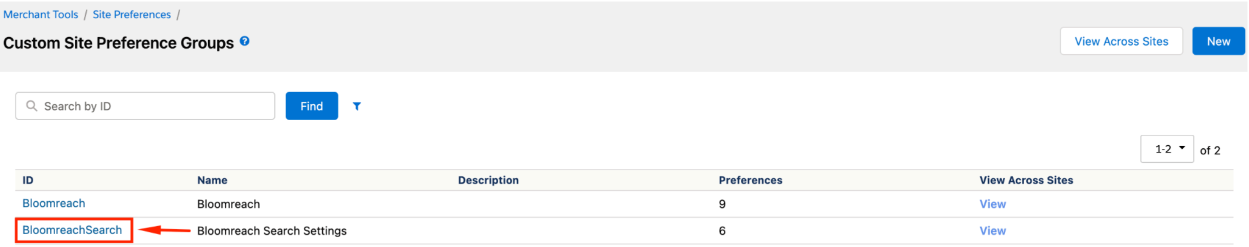
These credentials are used by 1 service.

Name:*	bloomreach.http.widgets.api
URL:	https://pathways-staging.dxpapi.com/api/v2/widgets/
User:	
Password:	••••••••

Configuration

Custom Site Preferences

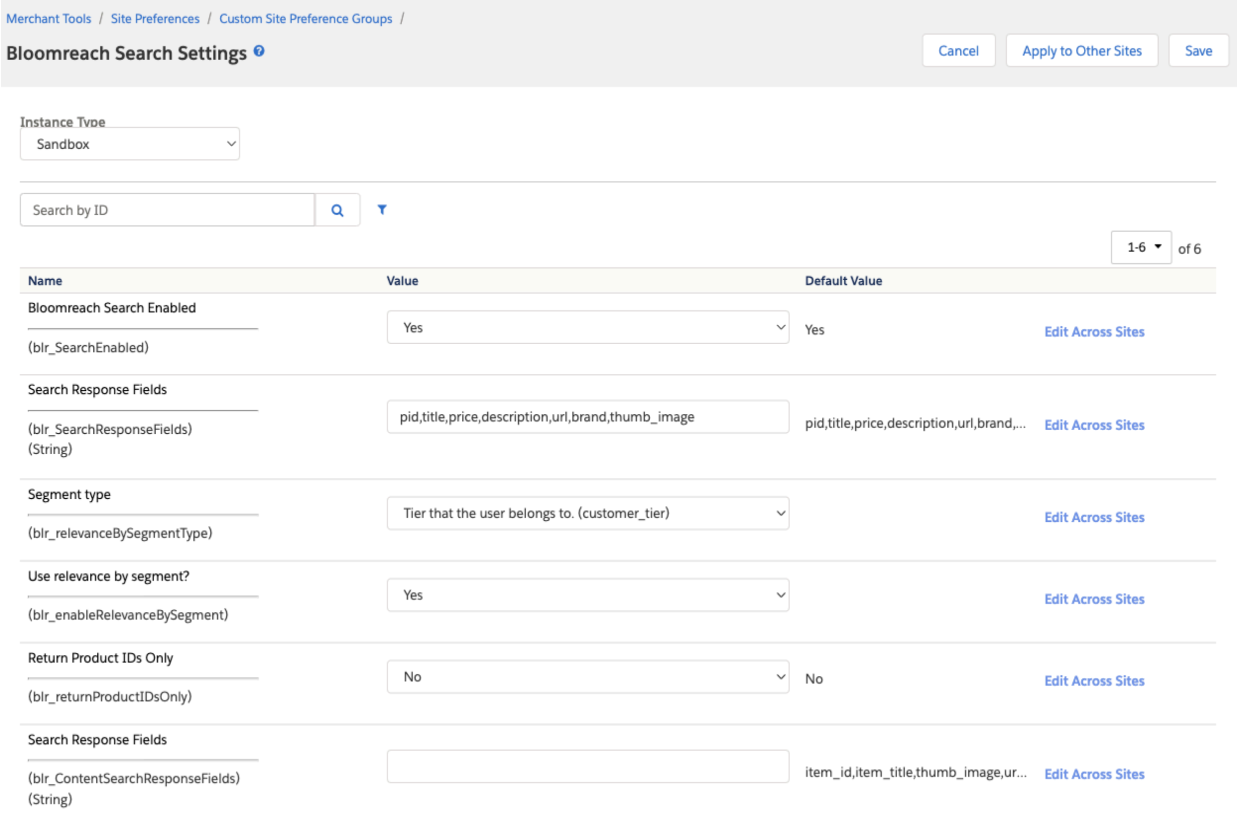
After import is successfully done, you should see Site Preferences in BM > Merchant Tools > Site Preferences > Custom Preferences.



Custom Site Preference Groups

ID	Name	Description	Preferences	View Across Sites
Bloomreach	Bloomreach		9	View
BloomreachSearch	Bloomreach Search Settings		6	View

Select BloomreachSearch and you should see fields for cartridge setup.



Merchant Tools / Site Preferences / Custom Site Preference Groups /
Cancel
Apply to Other Sites
Save
Search by ID
1-6 of 6

Bloomreach Search Settings

Instance Type: Sandbox

Name	Value	Default Value	
Bloomreach Search Enabled (blr_SearchEnabled)	Yes	Yes	Edit Across Sites
Search Response Fields (blr_SearchResponseFields) (String)	pid,title,price,description,url,brand,thumb_image	pid,title,price,description,url,brand,...	Edit Across Sites
Segment type (blr_relevanceBySegmentType)	Tier that the user belongs to. (customer_tier)		Edit Across Sites
Use relevance by segment? (blr_enableRelevanceBySegment)	Yes		Edit Across Sites
Return Product IDs Only (blr_returnProductIDsOnly)	No	No	Edit Across Sites
Search Response Fields (blr_ContentSearchResponseFields) (String)		item_id,item_title,thumb_image,ur...	Edit Across Sites

#	Field Title	Description
1	Bloomreach Search Enabled	This is a switcher that allows you to enable/disable Bloomreach search integration. When it's disabled, controllers uses default SFRA search scripts
2	Search Response Fields	This setting allows you to control which attributes are returned in the Bloomreach product search API response. Value of this preference directly defines the value of the **fl** parameter in the search API request. For more details, please use Bloomreach API Documentation
3	Segment type	This parameter defines which segment type is used in the search API request. Possible values: ● **Tier that the user belongs to**. API parameter name - **customer_tier.** API parameter value - **customer groups of which the user's profile is part of.** ● **Country that the user belongs to or is accessing the site from**. API parameter name - **customer_country** API parameter value - **country code of the default (preferred) address of the registered user.** ● **Geography or region that the user belongs to**. API parameter name - **customer_geo** API parameter value - **country code of the geolocation of the user.** ● **Profile of the user**. API parameter name - **customer_profile** API parameter value - **customer number.** ● **Specific catalog view**. API parameter name - **view_id** API parameter value - **view ID.** **Specific customer**. API parameter name - **user_id** API parameter value - **customer number.** To learn more about Relevance by Segment features, please use Bloomreach Relevance by Segment guide.
4	Use relevance by segment?	This is a switcher that controls whether relevance by segment feature is enabled. To learn more about Relevance by Segment features, please use Bloomreach Relevance by Segment guide.
5	Return Product IDs Only	This is a switcher that allows to enforce receiving product IDs (**pid)** parameter only from the product search API request by controlling **fl**

		parameter value of requests. Overrides behavior of the *Search Response Fields* preference.
6	Content Search Response Fields	This setting allows you to control which attributes are returned in the Bloomreach content search API response. Value of this preference directly defines the value of the **fl** parameter in the search API request. For more details, please use Bloomreach API Documentation

Global Pixel

The Bloomreach Page View Pixels track views of your customers on your pages. Bloomreach recommends that you add the Page View Event Code to your main site template footer, which affects every page on your site, including your conversion (HTTPS) pages. This will allow Bloomreach to gather the analytics and clickstream events needed from your site. This ensures your Bloomreach Discovery powered search works as expected and allows optimization of your search results.

Enablement/disablement

Once you added int_bloomreach_sfra to your cartridge path, this functionality is enabled.

Implementation

Cartridge leverages the SFRA "app.template.afterFooter" hook to add the block of javascript code to the bottom of every html page. Javascript hook is executed in the context of the template engine(within Controller) on the B2C commerce server, not on the client.

As the result of the hook handler call, some client side javascript code(function and variables) is printed into the html page.

Example of the code:

```
<script>
var br_data = br_data || {};
br_data.test_data = "true";
br_data.ptype = "product";
br_data.locale = "en_US";
br_data.action = "Product-Show";
br_data.title = "Buy Canon PowerShot G10 Digital Point and Shoot Camera for USD 499.99 |
Bloomreach_SFRA";
br_data.acct_id = "7339";
br_data.domain_key = "sandbox_astound";
br_data.view_id = "usd";
br_data.user_id = "abXwjOsIwIGrdCdZLFoJLyZTU0";
br_data.prod_id = "canon-powershot-g10M";
br_data.prod_name = "Canon PowerShot G10 Digital Point and Shoot Camera";

(function() {
    var brtrk = document.createElement('script');
    brtrk.type = 'text/javascript';
    brtrk.async = true;
    brtrk.src = "//cdn.brcdn.com/v1/br-trk-7339.js";
    var s = document.getElementsByTagName('script')[0];
    s.parentNode.insertBefore(brtrk, s);
})();
</script>
```

Customization

There are two points of customization of the data being tracked by the Bloomreach Pixel: PageType and Pixel Data.

Page Type

By default the cartridge maps the following controllers to the Bloomreach page types:

Controller - endpoint	Page Type value
Home-Show	homepage
Product-Show	product
Page-Show	content
Order-Confirm	conversion
Search-Show with **cgid** parameter	category
Search-Show without **cgid** parameter	search
All other pages	other

If you need to add more granularity to the page types or change the existing mapping modify the following file:

int_bloomreach_sfra/cartridge/scripts/customizable/blrPageType.js

Parameter 'action' holds the controller-endpoint pair, you just have to modify the switch clause to your needs.

Pixel Data

Cartridge provides a set of decorator functions to collect data corresponding to the page type. Decorator functions are special functions that enhance or modify the behavior of properties and methods in Lightning Web Components without directly altering their original code.

PageType	Decorator Function
global page data for all pages	globalPage
product	productPage
category	categoryPage
search	searchPage
conversion	confirmationPage
content, homepage	globalPage

If you need to modify the data, find the appropriate decorator and edit the function.

int_bloomreach_sfra/cartridge/scripts/customizable/pixelDecorators.js

Recommendation & Pathways

Before beginning to integrate the Recommendation and Pathways widget, you have to integrate the Bloomreach pixel, as well as create and configure widgets on the Bloomreach dashboard.

The Bloomreach documentation describes how to create and configure a widget.

How the Recommendation & Pathways widget integration works

Loading the widget is carried out in two stages. First, the widget container is rendered on the page along with the content of the main page. Second, after the page loads, an AJAX request is sent to the Bloomreach-Recommendations controller. The controller sends a request to the Bloomreach API, gets response data for the widget, renders the widget's content, and returns it to the storefront to display the content in the widget container. Loading the widget in two stages allows to exclude the influence of the widget loading delay time on the loading time of the main page content and improve site performance.

Creating an AJAX request, loading the widget content into the container, and processing events is done using a client-side script that is loaded along with the main page template.

Integration of Recommendation and Pathways widget

Adding a widget container to a page

In your project page templates, add the Recommendation & Pathways widget container template using remote include. Using remote include allows you to add the necessary parameters for the widget directly into the template and simplify integration. The format for adding a widget container template is:

```
<isinclude url="${URLUtils.url('Bloomreach-Widget', <param1>, <value1>,
<param2>, <value2>...}"/>
```

You have to add required parameters to the container template to initialize the widget. The set of required parameters depends on the widget type, but for all widget types there must be two required parameters:

- *widget_id* - widget ID;
- *wty* - widget type.

The values of the *widget_id* and *wty* parameters are indicated on the Bloomreach Discovery dashboard in the widget settings. The Bloomreach documentation describes how to get these parameters

The list of widget parameters can be found in the Bloomreach documentation.

Below is an example of a widget template for Category-based widget with widget ID= 'nl5dw1jq' on the search page:

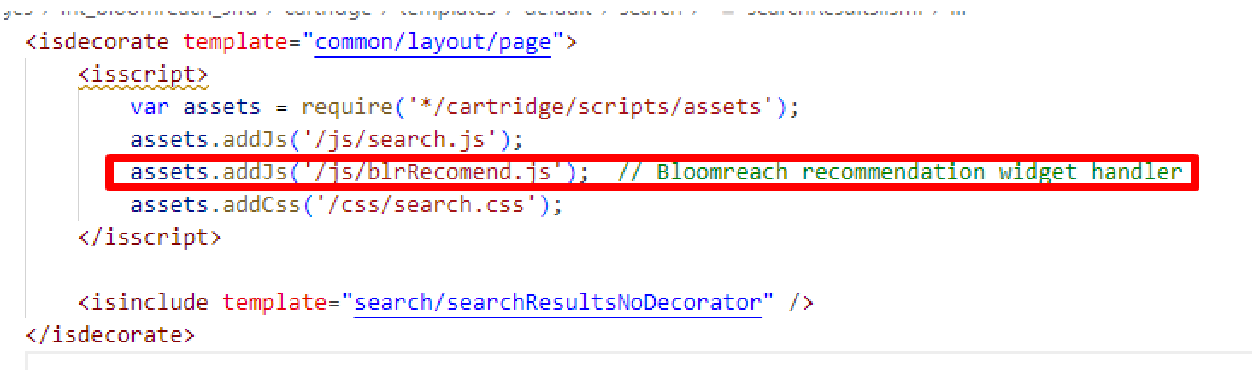
```
<isinclude url="${URLUtils.url('Bloomreach-Widget', 'widget_id', 'nl5dw1jq',
'wty', 'category', 'cat_id', pdict.productSearch.category.id)}"/>
```

Adding a client script for handling the widget

The client script *int_bloomreach_sfra/cartridge/client/default/js/blrRecomend.js* handles loading content into the widget container and handling View Event Tracking Pixel and Click Event Tracking Pixel.

Note: ***before start, compile the client script.***

The client script */js/blrRecomend.js* must be added to the template in the loading scripts and styles section of the page on which the widget containers are added. Below is an example of adding a client script to a page.



```
<isdecorate template="common/layout/page">
    <isscript>
        var assets = require('*/cartridge/scripts/assets');
        assets.addJs('/js/search.js');
        assets.addJs('/js/blrRecomend.js');  // Bloomreach recommendation widget handler
        assets.addCss('/css/search.css');
    </isscript>

    <isinclude template="search/searchResultsNoDecorator" />
</isdecorate>
```

At this stage, the integration of the Recommendation & Pathways widget for static pages is completed.

Updating the widget content for a single page application

For a single page application, for which the content is updated dynamically using AJAX requests, it is necessary to update the widget content as well. To do this, add a call to the widget update handler in the client script.

```
$('body').trigger('widget:update', data);
```

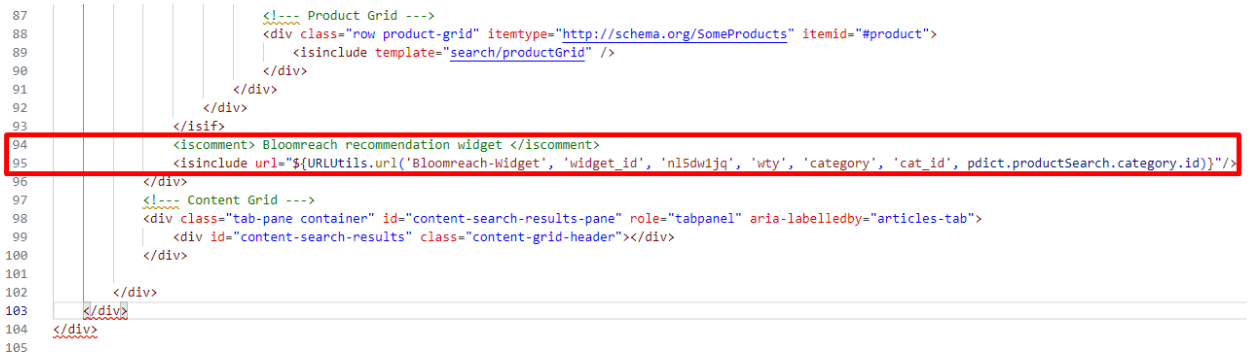
In the `data` object, new parameters for the widget must be passed when calling the widget update handler. The parameters for the widget are listed in the [Bloomreach documentation](). The data object can only specify parameters that have been changed. If any parameter is missing in the `data` object, then the previous value of the parameter for this widget is taken.

An example of widget integration on a search page

Below is an example of adding a Category-based widget with ID = 'nl5dw1jq' to the search page. Previously, a Category-based Widget with ID = 'nl5dw1jq' was created on the Bloomreach dashboard.

1. Add a widget container to your search page. In the template */search/searchResultsNoDecorator.isml* add the following rows in line 94:

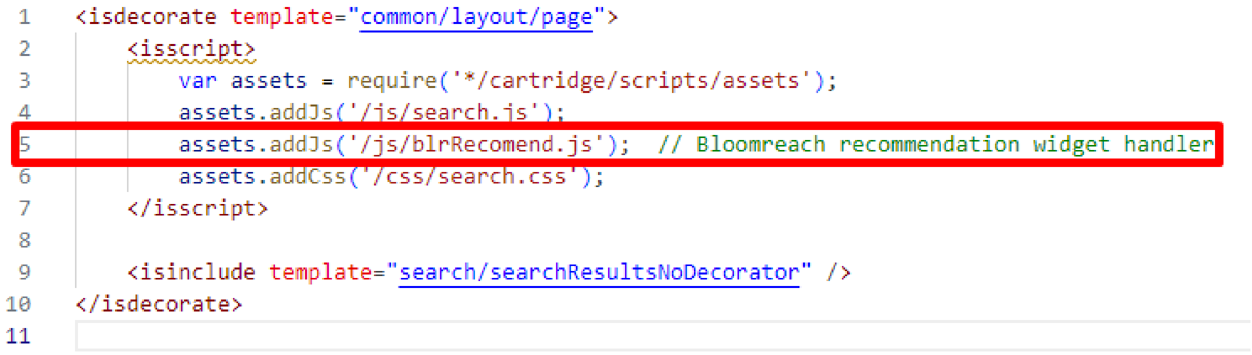
```
<iscomment> Bloomreach recommendation widget </iscomment>
<isinclude url="${URLUtils.url('Bloomreach-Widget', 'widget_id', 'nl5dw1jq',
'wty', 'category', 'cat_id', pdict.productSearch.category.id)}"/>
```



```
87                              <!--- Product Grid --->
88                              <div class="row product-grid" itemtype="http://schema.org/SomeProducts" itemid="#product">
89                                  <isinclude template="search/productGrid" />
90                              </div>
91                          </div>
92                      </div>
93                  </isif>
94                  <iscomment> Bloomreach recommendation widget </iscomment>
95                  <isinclude url="${URLUtils.url('Bloomreach-Widget', 'widget_id', 'nl5dw1jq', 'wty', 'category', 'cat_id', pdict.productSearch.category.id)}"/>
96              </div>
97              <!--- Content Grid --->
98              <div class="tab-pane container" id="content-search-results-pane" role="tabpanel" aria-labelledby="articles-tab">
99                  <div id="content-search-results" class="content-grid-header"></div>
100             </div>
101
102         </div>
103     </div>
104 </div>
105
```

2. Add the client script blrRecomend.js to the search page. In the template */search/searchResults.isml* add the following row in line 5:

```
assets.addJs('/js/blrRecomend.js');
```



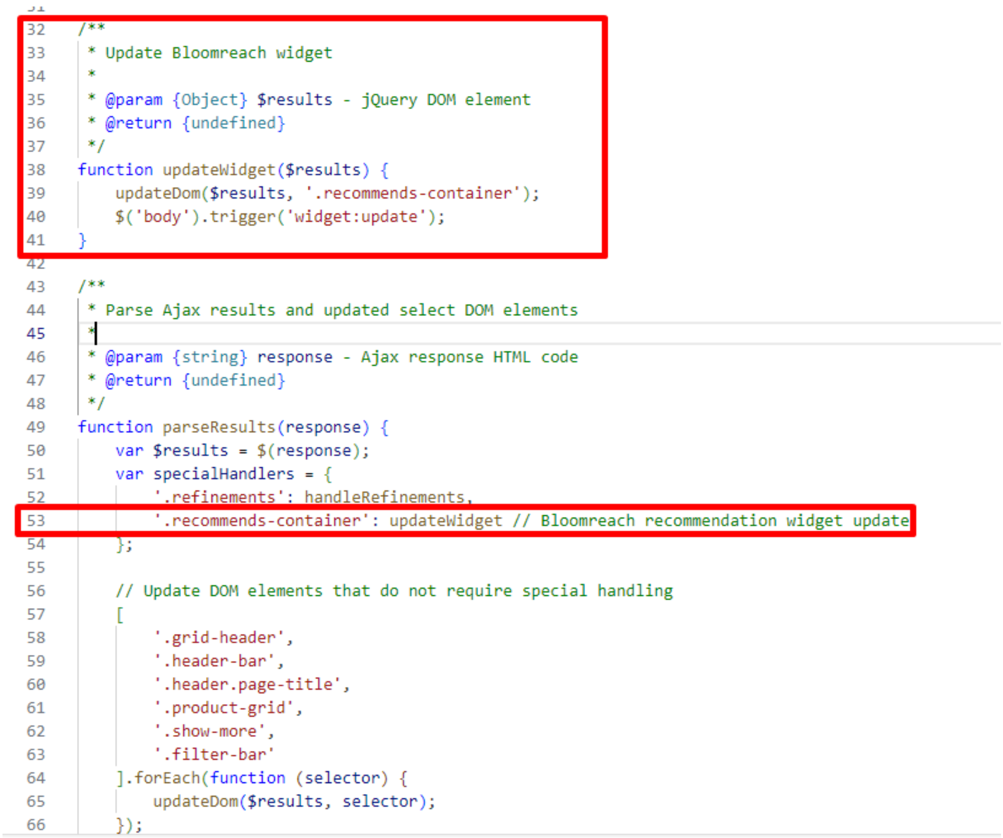
```
<isdecorate template="common/layout/page">
    <isscript>
        var assets = require('*/cartridge/scripts/assets');
        assets.addJs('/js/search.js');
        assets.addJs('/js/blrRecomend.js');  // Bloomreach recommendation widget handler
        assets.addCss('/css/search.css');
    </isscript>

    <isinclude template="search/searchResultsNoDecorator" />
</isdecorate>
```

3. Add a call to the widget update handler in the client script.
   In the client's *script /search/search.js* add the updateWidget function line 32:

```
/**
 * Update Bloomreach widget
 *
 * @param {Object} $results - jQuery DOM element
 * @return {undefined}
 */
function updateWidget($results) {
    updateDom($results, '.recommends-container');
    $('body').trigger('widget:update');
}
```

4. In the client's script /search/search.js add the bloomreach widget handler to the specialHandlers object in line 53:

```
'.recommends-container': updateWidget // Bloomreach recommendation widget
update
```



```
/**
 * Update Bloomreach widget
 *
 * @param {Object} $results - jQuery DOM element
 * @return {undefined}
 */
function updateWidget($results) {
    updateDom($results, '.recommends-container');
    $('body').trigger('widget:update');
}

/**
 * Parse Ajax results and updated select DOM elements
 *
 * @param {string} response - Ajax response HTML code
 * @return {undefined}
 */
function parseResults(response) {
    var $results = $(response);
    var specialHandlers = {
        '.refinements': handleRefinements,
        '.recommends-container': updateWidget // Bloomreach recommendation widget update
    };

    // Update DOM elements that do not require special handling
    [
        '.grid-header',
        '.header-bar',
        '.header.page-title',
        '.product-grid',
        '.show-more',
        '.filter-bar'
    ].forEach(function (selector) {
        updateDom($results, selector);
    });
```

Customisation

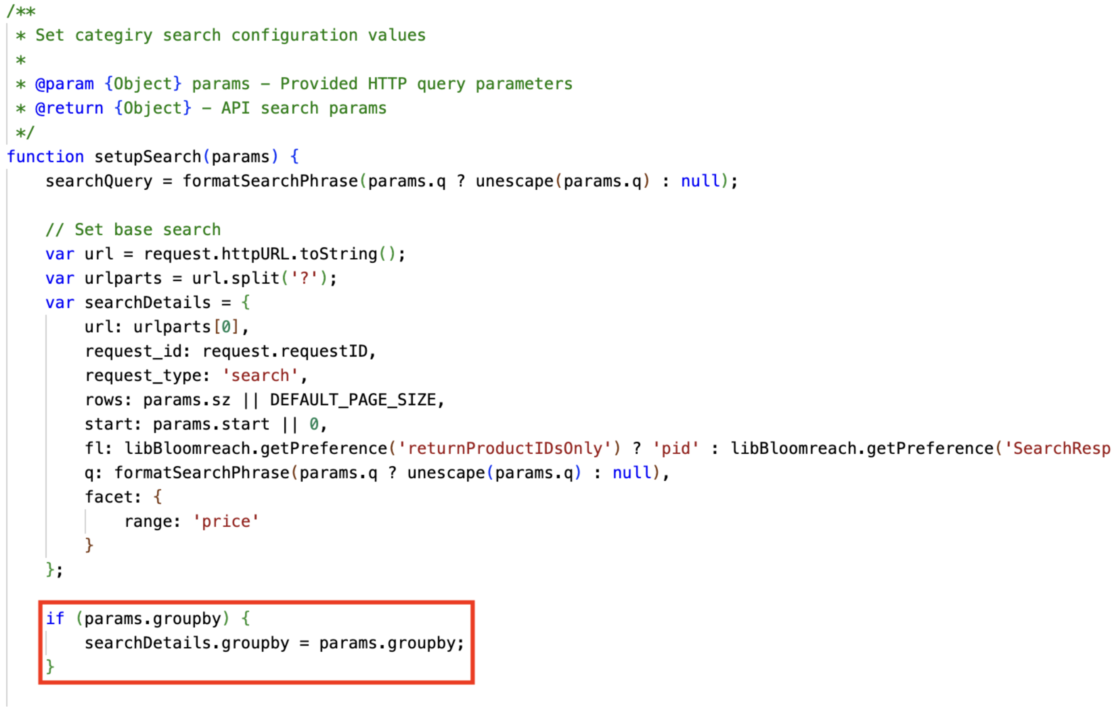
List of possible code changes that allows you to personalize your experience with Bloomreach search.

Dynamic Grouping

Dynamic Grouping allows grouping of similar products together to reduce the size of the recall. Note that searches can only group on a single attribute at a time.

In order to use dynamic grouping add the **groupby** parameter to the request in the *bmSelarchHelper.js* by customising *setupSearch* function.

Example of the code customisation



```
/**
 * Set categiry search configuration values
 *
 * @param {Object} params - Provided HTTP query parameters
 * @return {Object} - API search params
 */
function setupSearch(params) {
    searchQuery = formatSearchPhrase(params.q ? unescape(params.q) : null);

    // Set base search
    var url = request.httpURL.toString();
    var urlparts = url.split('?');
    var searchDetails = {
        url: urlparts[0],
        request_id: request.requestID,
        request_type: 'search',
        rows: params.sz || DEFAULT_PAGE_SIZE,
        start: params.start || 0,
        fl: libBloomreach.getPreference('returnProductIDsOnly') ? 'pid' : libBloomreach.getPreference('SearchResp
        q: formatSearchPhrase(params.q ? unescape(params.q) : null),
        facet: {
            range: 'price'
        }
    };

    if (params.groupby) {
        searchDetails.groupby = params.groupby;
    }
```

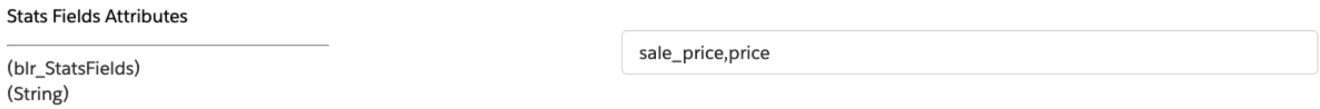
Please note that response structure is different from product search. Products in the response divided into groups, "*doclist*" and "*docs*" section is moved under the "*groups*".

To learn more about Dynamic Grouping, please use Bloomreach Dynamic Grouping guide.

Field Stats

The **stats.field** 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 specific attributes such as **price**, **sale_price**, **length**, **width**, **reviews**, and so on.
In order to use field stats feature add the corresponding parameter to the custom preference Search Response Fields with the list of attributes separated by comma.

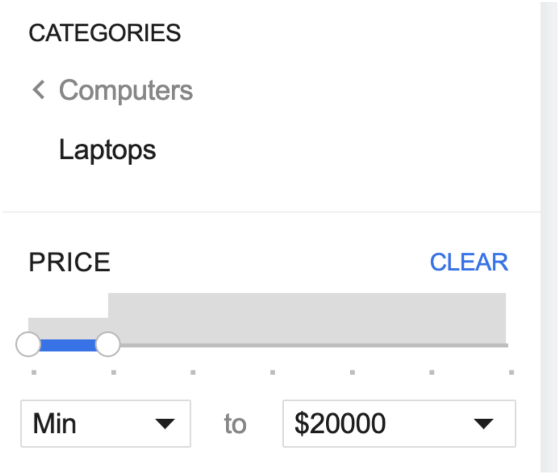


Stats Fields Attributes

(blr_StatsFields)
(String)

sale_price,price

As a result in the API response you will get a list of requested attributes with the specified max and min values. Use this data to render your custom filter component. Example of usage below.



For more information, use [Bloomreach Field Stats documentation]()

Log errors

All errors that may appear during a job run are stored in the error log file with name *customlog-Bloomreach…log*.

Restrictions

Intentionally left blank

Firewall Requirements

No special firewall requirements

4. Testing

Intentionally left blank

5. Operations, Maintenance

Data Storage

Intentionally left blank

Availability

If the Bloomreach service becomes unavailable, the storefront continues to work without the service. All errors work of the service are saved in the log file.

Failover/Recovery Process

Intentionally left blank

Support

For questions please contact your Bloomreach Success Manager.

6. Known Issues

7. Release History

Version	**Date**	**Changes**
23.1.0	2023-09-01	Initial release