

# Bloomreach

## *Discovery Cartridge Integration guide*

*Version 23.1.0*



## [Table of Contents](#)

<b>1. Summary.....</b>	<b>3</b>
<b>2. Component Overview.....</b>	<b>4</b>
Functional Overview.....	4
Limitations, Constraints.....	4
Contract.....	4
Compatibility.....	4
Privacy, Payment.....	4
<b>3. Implementation Guide.....</b>	<b>5</b>
Setup of Business Manager.....	5
Importing cartridges.....	5
Assigning cartridges to the site.....	7
Import of metadata.....	8
Service.....	8
Configuration.....	10
Custom Site Preferences.....	10
Global Pixel.....	12
Enablement/disablement.....	12
Description.....	12
Customization.....	13
Recommendation & Pathways.....	14
Integration of Recommendation and Pathways widget.....	14
Adding a widget container to a page.....	14
Adding a client script for handling the widget.....	15
Updating the widget content for a single page application.....	15
An example of widget integration on a search page.....	16
Customisation.....	17
Dynamic Grouping.....	18
Field Stats.....	18
Log errors.....	19
Restrictions.....	19
Firewall Requirements.....	19
<b>4. Testing.....</b>	<b>19</b>
<b>5. Operations, Maintenance.....</b>	<b>19</b>
Data Storage.....	20
Availability.....	20
Failover/Recovery Process.....	20
Support.....	20
<b>6. Known Issues.....</b>	<b>20</b>
<b>7. Release History.....</b>	<b>20</b>

## 1. Summary

The Bloomreach Discovery cartridge allows you to quickly integrate a Bloomreach search service into a Salesforce Commerce Cloud website. The Bloomreach service allows the customer to use a search engine on your website.

This Discovery cartridge contains instructions that allow you to integrate with a reference Bloomreach application.

### Glossary

Term	Description
SFCC	Salesforce Commerce Cloud
Business Manager (BM)	The primary tool used to configure the SFCC platform and customer storefront SFRA

## 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

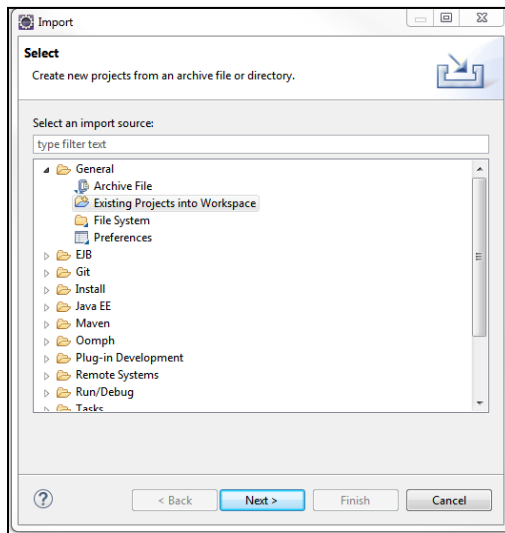
The Bloomreach cartridge doesn't collect and process user profile information or billing information. For additional privacy information, please contact your Bloomreach Account Manager.

### 3. Implementation Guide

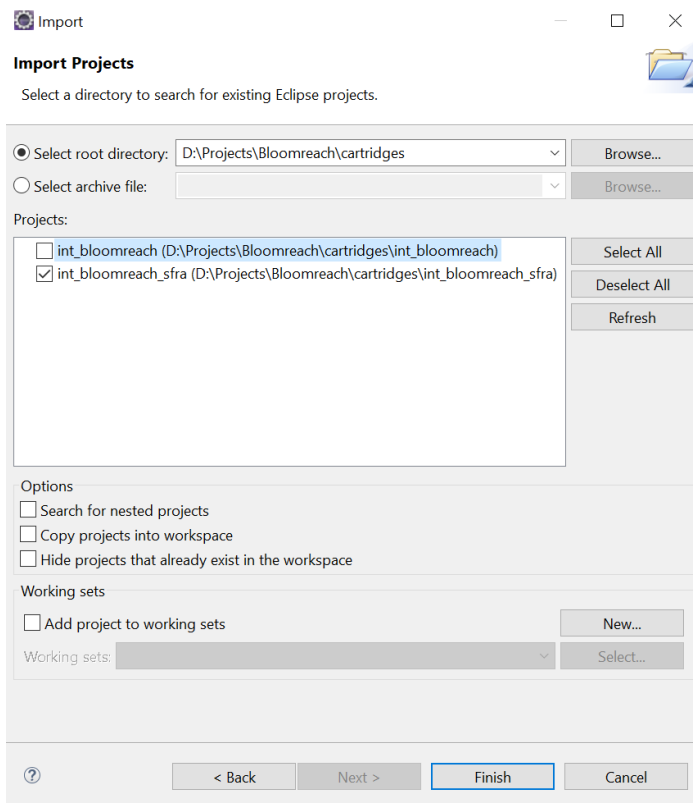
#### Setup of Business Manager

##### *Importing cartridges*

1. Download the cartridge source code.
2. Establish a new digital server connection with your SFCC Instance
3. Import cartridges to a workspace in Salesforce UX Studio.



4. Add cartridges to Project Reference of Server Connection.



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
<input type="checkbox"/>	<a href="#">SiteGenesis</a>	SiteGenesis	Online	
<input type="checkbox"/>	<a href="#">SiteGenesisGlobal</a>	SiteGenesisGlobal	Online	
<input type="checkbox"/>	<a href="#">RefArch</a>	RefArch	Online	
<input type="checkbox"/>	<a href="#">RefArchGlobal</a>	RefArchGlobal	Online	

[Refresh](#)[New](#)[Delete](#)

## 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:

[Administration](#) > [Sites](#) > [Manage Sites](#) > RefArch - Settings

[General](#) [Settings](#) [Cache](#) [Site Status](#) [Page Meta Tag Rules](#)

## 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 ho

**HTTP Hostname:**

**HTTPS Hostname:**

**Instance Type:** All

**Cartridges:**

**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

[<< Back to List](#)

## 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

No file chosen

Select	Name ▲	Location	File Size	Last Modified
<input type="radio"/>	<a href="#">/instances/24-01-2023-sandbox-site-date.zip</a>	global	153.77 MB	1/24/23 4:41:11 am
<input type="radio"/>	<a href="#">/instances/apparel-m-catalog.zip</a>	global	622.36 KB	2/16/23 8:57:53 am
<input checked="" type="radio"/>	<a href="#">instance/bloomreach.zip</a>	local	3.09 KB	9/8/23 12:27:31 am
<input type="radio"/>	<a href="#">/instances/default.zip</a>	global	76.92 MB	12/3/22 1:29:48 pm
<input type="radio"/>	<a href="#">/instances/Lab_11.zip</a>	global	589 B	12/2/22 9:49:55 am
<input type="radio"/>	<a href="#">/instances/lab_A_10_1.zip</a>	global	38.93 KB	7/20/22 9:46:10 am
<input type="radio"/>	<a href="#">/instances/lab_a_10_julian.zip</a>	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
<input type="checkbox"/>	<a href="#">bloomreach.http.credential</a>	<a href="https://api-staging.connect.bloomreach.com/dataconnect/api/v1/">https://api-staging.connect.bloomreach.com/dataconnect/api/v1/</a>	
<input checked="" type="checkbox"/>	<a href="#">bloomreach.http.search.api</a>	<a href="https://staging-core.dxpapi.com">https://staging-core.dxpapi.com</a>	
<input checked="" type="checkbox"/>	<a href="#">bloomreach.http.suggestions.api</a>	<a href="https://staging-suggest.dxpapi.com">https://staging-suggest.dxpapi.com</a>	
<input checked="" type="checkbox"/>	<a href="#">bloomreach.http.widgets.api</a>	<a href="https://pathways-staging.dxpapi.com/api/v2/widgets/">https://pathways-staging.dxpapi.com/api/v2/widgets/</a>	
<input type="checkbox"/>	<a href="#">bloomreach.sftp.sandbox</a>	<a href="https://sftp-staging.connect.bloomreach.com">sftp-staging.connect.bloomreach.com</a>	

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:	<input type="text" value="bloomreach.http.search.api"/>
URL:	<input type="text" value="https://staging-core.dxpapi.com"/>
User:	<input type="text"/>
Password:	<input type="password" value="*****"/>

[Apply](#)

[<< Back to List](#)

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:	<input type="text" value="bloomreach.http.suggestions.api"/>
URL:	<input type="text" value="https://staging-suggest.dxpapi.com"/>
User:	<input type="text"/>
Password:	<input type="password" value="*****"/>

[Apply](#)

[<< Back to List](#)

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:	<input type="text" value="bloomreach.http.widgets.api"/>
URL:	<input type="text" value="https://pathways-staging.dxpapi.com/api/v2/widgets/"/>
User:	<input type="text"/>
Password:	<input type="password" value="*****"/>

[Apply](#)

[<< Back to List](#)

## Configuration

### Custom Site Preferences

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

Merchant Tools / Site Preferences /

**Custom Site Preference Groups** [View Across Sites](#) [New](#)

Q Search by ID [Find](#) [T](#)

1-2 of 2

ID	Name	Description	Preferences	View Across Sites
Bloomreach	Bloomreach		9	<a href="#">View</a>
BloomreachSearch	Bloomreach Search Settings		6	<a href="#">View</a>

Select BloomreachSearch and you should see fields for cartridge setup.

Merchant Tools / Site Preferences / Custom Site Preference Groups /

**Bloomreach Search Settings** [Cancel](#) [Apply to Other Sites](#) [Save](#)

Instance Type  
Sandbox

Search by ID [Q](#) [T](#)

1-6 of 6

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

#	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 <b>fl</b> parameter in the search API request. For more details, please use <a href="#">Bloomreach API Documentation</a>
3	Segment type	<p>This parameter defines which segment type is used in the search API request.</p> <p>Possible values:</p> <ul style="list-style-type: none"> <li> <b>Tier that the user belongs to.</b>  API parameter name - <b>customer_tier</b>.  API parameter value - <b>customer groups of which the user's profile is part of</b>. </li> <li> <b>Country that the user belongs to or is accessing the site from.</b> API parameter name - <b>customer_country</b>  API parameter value - <b>country code of the default (preferred) address of the registered user</b>. </li> <li> <b>Geography or region that the user belongs to.</b>  API parameter name - <b>customer_geo</b>  API parameter value - <b>country code of the geolocation of the user</b>. </li> <li> <b>Profile of the user.</b>  API parameter name - <b>customer_profile</b>  API parameter value - <b>customer number</b>. </li> <li> <b>Specific catalog view.</b>  API parameter name - <b>view_id</b>  API parameter value - <b>view ID</b>.  <b>Specific customer.</b>  API parameter name - <b>user_id</b>  API parameter value - <b>customer number</b>. </li> </ul> <p>To learn more about Relevance by Segment features, please use <a href="#">Bloomreach Relevance by Segment guide</a>.</p>
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 <a href="#">Bloomreach Relevance by Segment guide</a> .
5	Return Product IDs Only	This is a switcher that allows to enforce receiving product IDs ( <b>pid</b> ) parameter only from the product search API request by controlling <b>fl</b>

		parameter value of requests. Overrides behavior of the <i>Search Response Fields</i> 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 <b>fl</b> parameter in the search API request. For more details, please use <a href="#">Bloomreach API Documentation</a>

## 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 master site template footer, which affects every page on your site, including your conversion (HTTPS) pages. This will allow Bloomreach to gather the information needed from your site to support your Bloomreach content and optimize your user experience.

### *Enablement/disablement*

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

### *Description*

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 = "abXwj0sIwIGrdCdZLFoJLyZTU0";
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 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 <b>cgid</b> parameter	category
Search-Show without <b>cgid</b> 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.

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 the begin integrating 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 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',
'wt', 'category', 'cat_id', pdict.productSearch.category.id)}"/>

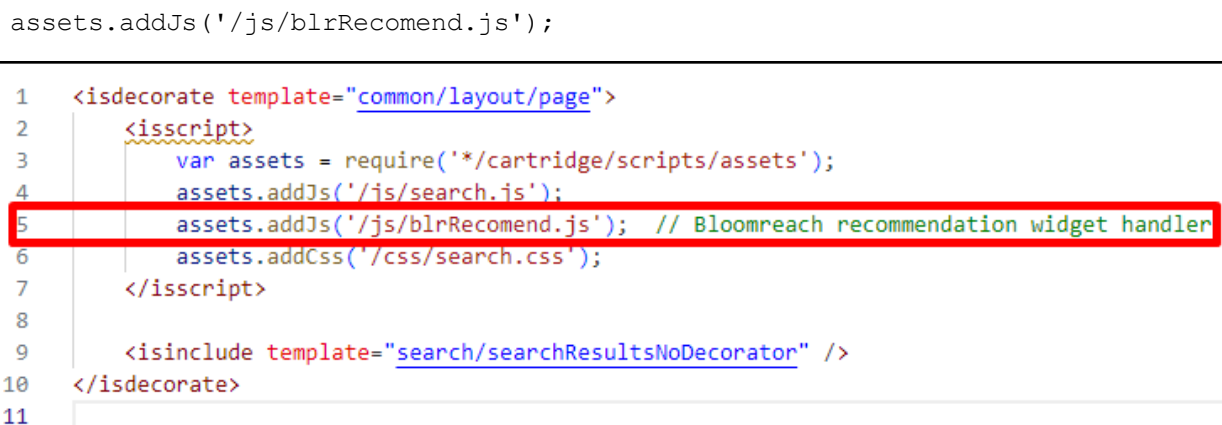
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
```



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');

1
2
3
4
5
6
7
8
9
10
11
```



3. Add a call to the widget update handler in the client script.  
In the clients 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 clients script /search/search.js add the bloomreach widget handler to the specialHandlers object in line 53:

```

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

```

```

32 /**
33  * Update Bloomreach widget
34  *
35  * @param {Object} $results - jQuery DOM element
36  * @return {undefined}
37  */
38 function updateWidget($results) {
39     updateDom($results, '.recommends-container');
40     $('body').trigger('widget:update');
41 }
42
43 /**
44  * Parse Ajax results and updated select DOM elements
45  *
46  * @param {string} response - Ajax response HTML code
47  * @return {undefined}
48  */
49 function parseResults(response) {
50     var $results = $(response);
51     var specialHandlers = {
52         '.refinements': handleRefinements,
53         '.recommends-container': updateWidget // Bloomreach recommendation widget update
54     };
55
56     // Update DOM elements that do not require special handling
57     [
58         '.grid-header',
59         '.header-bar',
60         '.header.page-title',
61         '.product-grid',
62         '.show-more',
63         '.filter-bar'
64     ].forEach(function (selector) {
65         updateDom($results, selector);
66     });

```

## Customisation

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

## Dynamic Grouping

Dynamic Grouping allows to group 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 **groupby** parameter to the request in the *blmSearchHelper.js* by customising *setupSearch* function.

Example of the code customisation

```
/**
 * Set category 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.

The screenshot shows a filter interface. At the top, it says 'CATEGORIES' with a back arrow and 'Computers'. Below that, 'Laptops' is selected. The main section is titled 'PRICE' and has a 'CLEAR' button. It features a horizontal slider with two white handles on a grey track. Below the slider, there are two dropdown menus: the first is labeled 'Min' and the second is labeled '\$20000', with a 'to' label between them. The interface is clean and modern, with a light grey background.

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