Federated Search – Interactive Module MediaSpace/KAF Administrator's Guide

Last Modified on 06/04/2026 9:18 am IDT

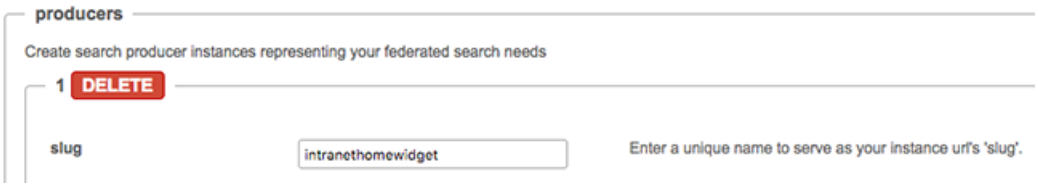
MediaSpace/KAF Admin – Setup

Your Kaltura PM / CSM will enable the “Federated Search – Interactive” module (“Federatedsearchinteractive”) in your MediaSpace/KAF environment, and also enable the “Federatedsearchcore” module, containing code shared by both Federated Search modules (Interactive and Indexing).

Once enabled, the Federatedsearchinteractive module includes three Search Producers by default. Each Producer corresponds to a common Intranet/Extranet use case, similar to the UI examples here.

You may delete the default Producers and create your own, but for starters, let’s review one of the default Producer's configuration settings to get better acquainted with available options. This should allow you, and your Kaltura team, to better collect requirements from your internal customer that intends to integrate Kaltura content into their portal.

The first default Producer, named “intranethomewidget” (“Intranet Home Widget”) produces data packaged to implement “Video News” widgets like in example #4.



1. It represents a saved search for content from a specific (non-private) gallery/channel,
2. It limits results to only 8 entries.
3. It is optimized for high-visibility, frequently-accessed areas, such as intranet portal homepages - by enabling local caching, which reduces the number of API calls and speeds up page loads.

By serving data into a "News" widget, the content is sorted by Most Recent and will always load the freshest content from that gallery/channel. The News widget often has a configured "sticky" entry that will always appear before (first) the freshest content. With this Producer, the team developing the News widget can easily load the data in JSON form, without needing to learn Kaltura or MediaSpace internals, and focus on

building their own UI. Beyond the App Token the CMS developers will need in order to authenticate, you can make their life easier by providing static JSON samples and “boilerplate” HTML code, both automatically generated in the module’s playground . See “What to Provide to Developers”.

MediaSpace/KAF Admin – Configuration

Let’s review the configuration setup of the first default Producer, to understand each parameter.

Field	Description	Tip
Enabled	Select Yes to enable the Federatedsearchinteractive module.	
slug	The Producer’s unique name. Passed as a URL parameter by the Consumer.	The module does not check for slug uniqueness. Please make sure to set descriptive, unique names.
searchFor	Select what objects the producer is searching for. Only “Entries” are supported for now.	
enableLocalCache	For optimal performance, if not using personalized results, set this to “Yes”.	Best to only switch this to “Yes” once integration is complete.

Field	Description	Tip
limitToItemsNum	Number of items to return. When building full-page video galleries instead of widgets, leave this field empty.	Results are limited to 50 items. A best practice would be to include a "Watch more videos" link in your UI, which will point back to your MediaSpace/KAF application site. When implementing a search field, concatenate "search/searchkeyword/", and the entered search term, to the "Watch more videos" link, to have the user continue their search there.
sortBy	Order of items to return.	When catering to “News” experiences go with Most Recent. When catering to interactive search experiences (see Example 3), go with Most Viewed or Most Liked. When catering to full page galleries, go with Alphabetical.

Field	Description	Tip
searchIn	Here it is set to “Only Specific Category/Channel”, to limit the content returned. Other options include “All Non-private Categories/Channels”, for returning no more than 50 entries from the entire portal catalog; “Only Categories/Channels Relevant to User”, for returning personalized results.	For personalized results, the user will need to be an actual member/owner/subscriber of channels in MediaSpace, and for the passed KS to include this user’s ID. Otherwise, results fallback to "All Non-private Categories/Channels" Make sure the developer passes, as part of App Token authentication, the SSO user ID, i.e. the user who authenticates to both the intranet portal and Video Portal.
limitToCategoryOrChannel	Allows to select which “Only Specific Category/Channel”.	
includeExtraMetadata	Should # of views and/or # of likes and/or custom metadata be returned as well.	Include the first two when promoting high-visibility content; Include custom metadata when building a richer experience, or when using an alternative title field for example, . “Title for Portal” or “Short Title”.
stickeyEntryId1-3	Set here entry ID(s) to always appear first.	

Field	Description	Tip
allowURLOverride	Allow the developers to override some aspects of the saved search programmatically.	Check “sortBy” when CMS developer plans to implement an interactive sort-by/rank-by UI. Check “limitToCategoryOrChannel” when developer plans to render more than one gallery, so can use a single producer. You will need to provide the developer with a list of category/channel ids.
limitToAppToken	Click Generate to create a developer-specific API token, limiting their API access to the task at hand. See “What to Provide to Developers”.	The App Token’s Id and value can be shared across Producers: Simply copy-paste the values generated from one, into the fields of additional Producers.

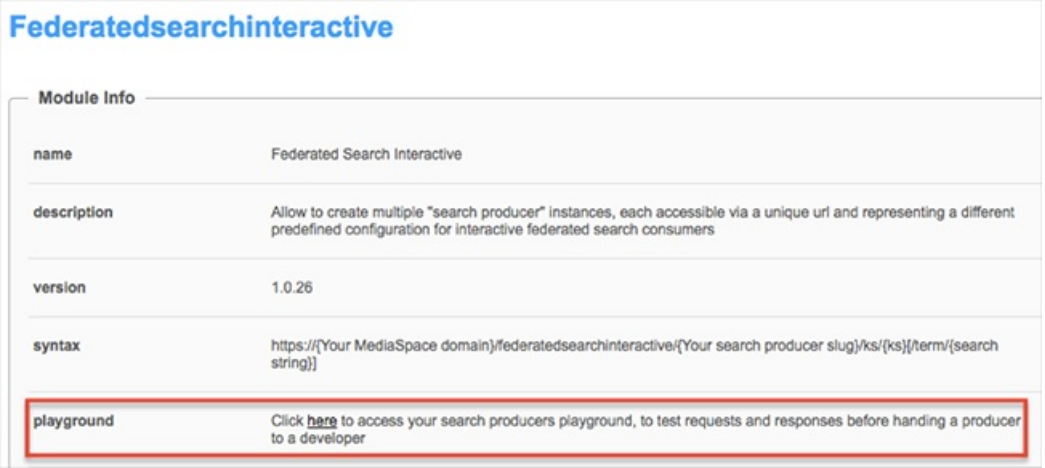
Click Save.

MediaSpace/KAF Admin – What to Provide to Developers

Since developers will not have access to the MediaSpace/KAF admin, “Federated Search – Interactive” includes a **playground** area. The playground area allows you, the admin, to first review the results returned from a Producer, then provide the developers not only with their App Token details, but also with static JSON content and “boilerplate” HTML/JavaScript so they can hit the ground running.

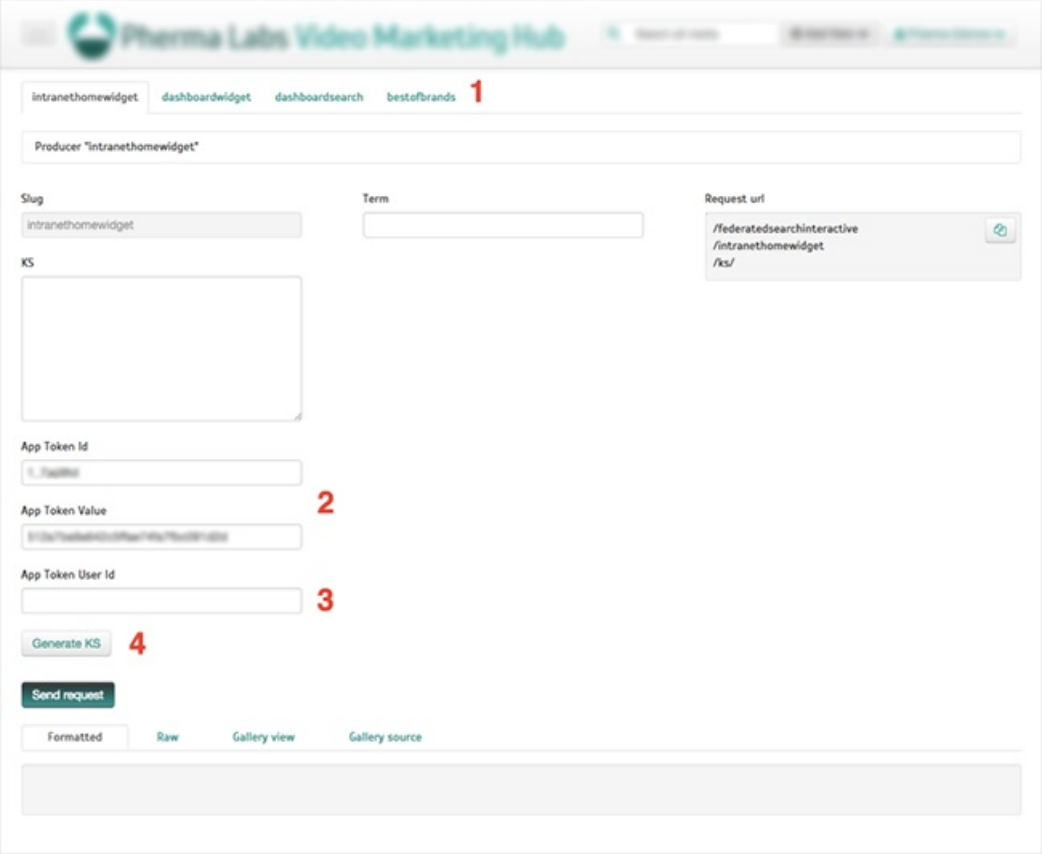
The following lists a common step-by-step workflow for working with developers, and lists the deliverables they will need:

1. After you finish creating a Producer, click Save.
2. Access the playground via the link in the header of the module.



Each Producer has its own tab (1), auto-generated when you add new Producers. Each Producer's App Token Id and value are already populated (2).

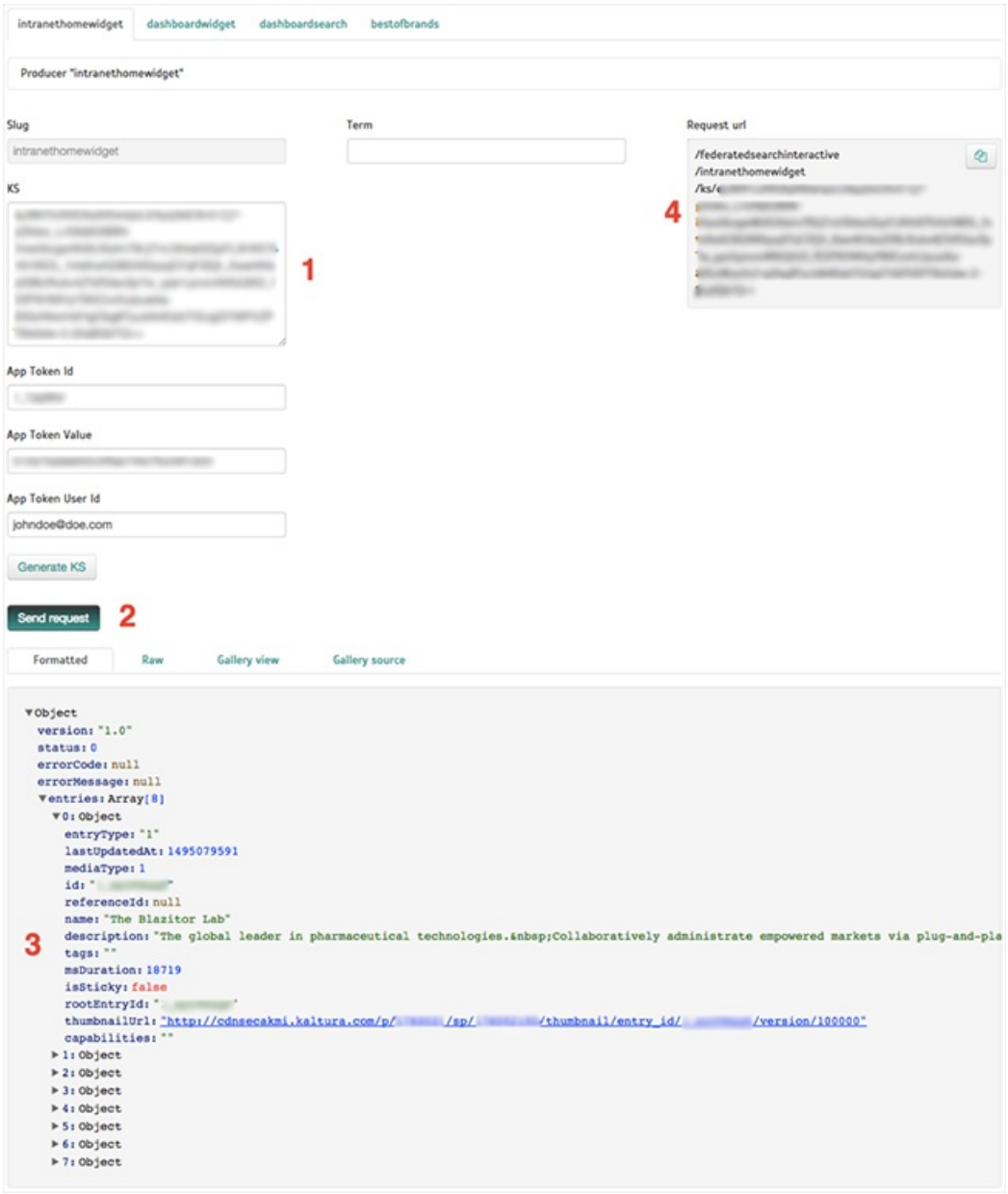
3. Enter a user App Token User Id string (any user string will do, even if the User ID does not exist in Kaltura yet) (3) and click "Generate KS" (4).



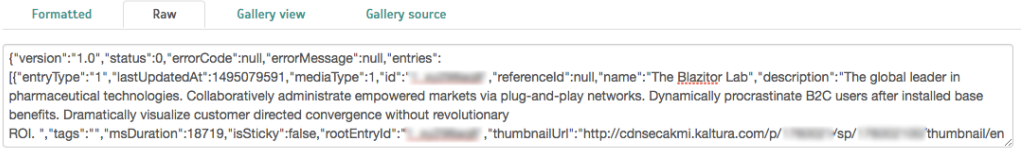
NOTE: Content from galleries/categories do not require a User Id to be part of the KS (Kaltura Session). However, when creating Producers that should include content from channels, a User Id, (any User Id), should be included in the App Token authentication. Please relay this to the CMS developer.

The playground will generate a session token, allowing you to test requests/responses.
TIP: If you are also using Federated Search – Indexing, you can use the playground as a shortcut to creating app token KSs, needed by the Index module's feeds (see "Federated Search – Indexing Module".)

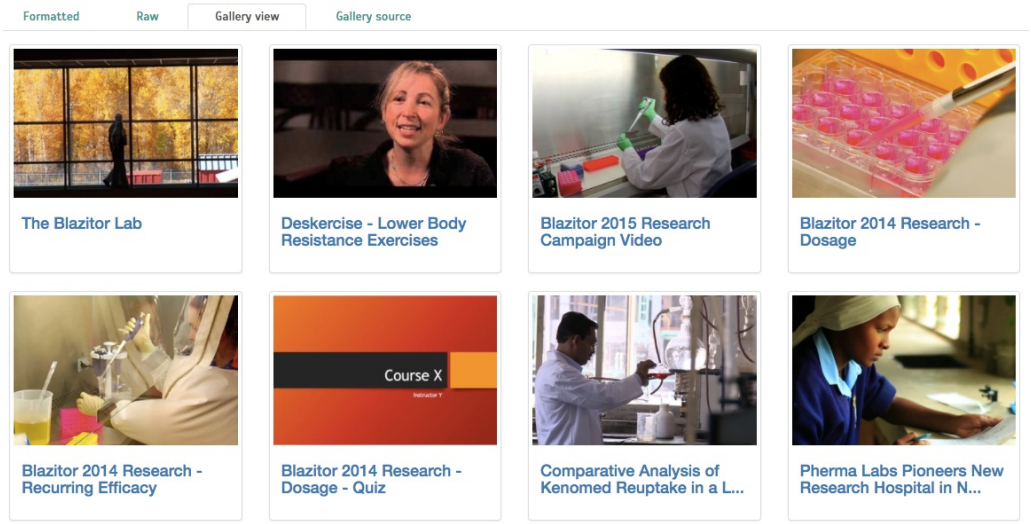
4. Once a KS is created (1), you can click "Send request" (2), to simulate a request to your producer and return fresh JSON data (3). The "Formatted" tab allows you to interactively examine each entry returned, by clicking the expand/contract arrows. You can also see an example of the Request url (4) the developer will use to generate this same request. This box is meant to help convey to a developer how a request would look like when passing override parameters or search terms (see Step 6).



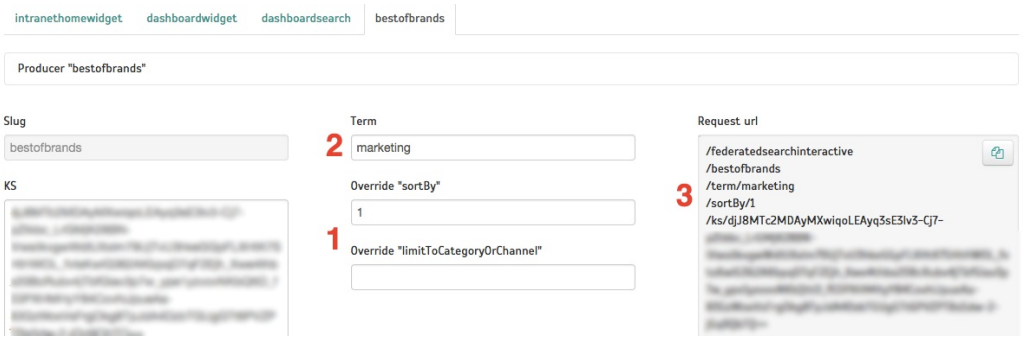
5. Select the Raw tab to access a developer-friendly version of the returned JSON. Provide this as early as possible to the developer so that they can start coding the front-end of their widget using this “stub” (static) data:



6. Select the Gallery view tab for quick visualization of the returned content. Use this tab to quickly examine returned results:



TIP: If you've enabled override parameters, you can enter values in their corresponding fields (1 below), or enter a search term in the Term field (2 below), and click "Send request" again to quickly test the results. You can also make changes to the producer, Click Save, switch back to the playground and continue to "Send request" to quickly examine your changes. Notice how the Request url (3) changes when entering these values.



7. Select the Gallery source tab to access the automatically-generated, "boilerplate" HTML code used to render the gallery in Gallery view. Clicking on the "Copy code" button (1 below) copies the entire code, which you can save in a text file (save it as <somename>.html) and provide to the developer. The code is standalone, depends only on common web libraries such as Bootstrap, and provides a quick way for the developer to learn how to query the producer and how to use the JSON data, simply by placing their file in their local hosting, running it, and reviewing the code.
   NOTE: The "boilerplate" code statically includes the generated KS. This KS will expire after a while, commonly after 24 hours. If still required by the developers past that time, repeat steps 2-4 to re-generate fresh code.

8. Provide the developer with the link to the App Token Authentication recipe on the Kaltura Developers Portal, along with the generated App Token Id and Value. Provide your Partner id. The recipe provides instructions on how to code an authentication request and allows running it interactively, using the provided values. The developer can then grab the generated code, available in multiple coding languages and use it in their back-end code, usually a CMS template, to generate an authentication token each time the portal page loads. The generated token is to be passed as the /ks url parameter to the Producer. See Request url (3) in the screenshot above.

Checklist

Here’s a summary of what you will need to provide the developer:

1. App Token ID and value
2. Partner ID
3. Static JSON from the “Raw” tab (Optional. This is useful if different developers handle front-end and back-end).
4. “Boilerplate” HTML code from the “Gallery source“ tab.
5. Link to the App Token Authentication recipe.