

# Syncing a Custom Catalog Feed to Klaviyo



## Overview

If your catalog is not synced by one of our pre-built integrations or you have catalog information stored outside of your main integration, you can sync that information to Klaviyo as custom catalog source, and then populate information from this source inside of your emails. Our custom catalog feature can be used to populate the Product Block in Klaviyo, which supports easily inserting hand-picked catalog items in an email as well as using Klaviyo-generated product recommendations. Your custom catalog source can contain anything you would consider to be “items” you want to share or recommend to your email recipients, whether these are products, store locations, ticketed events, or blog articles.

This guide will walk you through

- The requirements for setting up a custom catalog feed
- Syncing a custom catalog feed to Klaviyo
- Using your custom catalog data inside your emails
- Some of our more advanced catalog features
  - Back-in-stock
  - Location-aware catalog items

Once your feed has been built and added/mapped in your Klaviyo account, you will be able to use our Product Feed, Product Block, and [Catalog Lookup Tag](#) features. We will refresh the information in the catalog every 6 hours.

## Custom catalog requirements

Klaviyo can sync catalog feeds that meet the following requirements:

- The catalog is a public, hosted feed of all the items in your catalog.
  - If you do not have your own feed hosting location, you can use a service like Google Merchant feeds
- The feed is in JSON or XML.
- The data in the feed is only one node deep (unless it is hosted on Google Merchant Feeds).
- The size of any one feed file should not exceed 100 MB. If it will exceed this amount, please split the feed into multiple files. For optimal performance, we recommend each feed file not exceed 50 MB. All files used as sources will feed into the same catalog in Klaviyo.
- Required fields must not be left blank. For unavailable numeric fields, pass a zero (0) and for unavailable text fields, pass something like “n/a” or “unavailable”.

**Note:** *If your feed will have more than 1 million items, please let us know before you get to the “Add feed to account” step of this setup. This will allow us to pre-allocate necessary resources on our end to better optimize for larger catalog sizes.*

**Note:** *If your feed is hosted with a caching layer, make sure to disable caching for the location where the feed is hosted. Otherwise, we will only pull updated data for the feed items after the cache is refreshed.*

## Mappable catalog properties

Similar to our [event](#) and [profile property](#) tracking, our catalog feed feature will accept any fields passed to it as metadata/custom fields but also has some special mappable properties which will unlock certain features of our platform. Below is a list of the required and optional fields available for our default catalog feed setup:

### Required Fields

In order for the mapping step of the catalog setup to work, some required fields must be included for each item in the feed:

- **Unique ID** (*\$id*): can be text, numeric, or alphanumeric, but must be unique
- **Title** (*\$title*): the name of the item
- **URL** (*\$link*): the website at which a person can find this item
- **Image URL** (*\$image\_link*): a URL at which the image for this item is hosted
- **Description** (*\$description*): a text description of the item

*Note: The **Unique ID** of an item in the feed must match the item ID included in the metric on which you want to base personalized recommendations. For example, this would be passed as "ProductID" in [the Ordered Product metric example here](#).*

### Optional Fields

In addition to the required fields above, we also recommend passing the following optional mappable fields:

- **Price** (*\$price*): numeric item price, no currency symbol
- **Categories** (*categories*): comma-separated list/array of categories, tags, or collections associated with an item

#### Notes

- *The only way to filter a feed is by inclusion or exclusion of category names. If there is something you want to filter your feed by, make sure to include it as a category.*
- *Make sure the first item in your feed contains all available fields (in case some of your items do not). The mapping step draws on the fields from the first item so it will be important that all fields are present.*

## Sync your catalog

The next step is to sync your catalog feed with your Klaviyo account.

### Add feed to account

Once your feed is set up and hosted at a url, you can add it to your account by following these steps:

1. Navigate to the [Catalog Source page](#) on your account (note, this is not the Data Feeds page).

## 2. Name your feed source and add the URL.

Catalog > Sources > Create

### Define Feed Source

Name your feed and specify a URL to fetch items from. Your feed source must be an XML or JSON feed. You will be prompted to configure field mappings in the next step.

Source Name	Source URL
<input type="text" value="Catalog_Feed"/>	<input type="text" value="https://website.com/catalog_feed.json"/>
Username (Optional)	Password (Optional)
<input type="text"/>	<input type="text"/>

[Define Source](#)

a. You can optionally add a username and password if access to the feed URL requires them.

## 3. Click "Define Source"

## Map feed fields

After your feed has been added as a catalog source, you can configure the mapping of the fields in the feed.

1. From the [Catalog Source page](#), click on "Configure field mapping" to the right of the feed you just added.

Catalog\_Feed



Configure field mapping

2. We will try to automatically map the fields in your feed based on their name, but if they do not automatically map you may need to set them yourself. Using all of the required and optional properties above, your mapping should look like this:

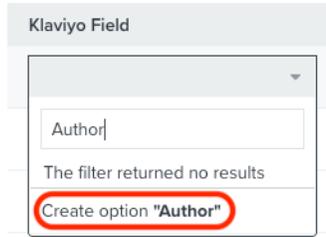
### Configure Field Mapping

All item fields in your feed must be mapped to Klaviyo fields. There are five required fields that must be present and mapped correctly: \$id, \$title, \$description, \$link, \$image\_link. Other optional fields may auto-map to standardized Klaviyo Fields, in which case the Field Type will also be pre-defined.

Feed Item Field	Klaviyo Field	Field Type	Required?
id	<input type="text" value="\$id"/>	<input type="text" value="Text"/>	<input checked="" type="checkbox"/>
title	<input type="text" value="\$title"/>	<input type="text" value="Text"/>	<input checked="" type="checkbox"/>
link	<input type="text" value="\$link"/>	<input type="text" value="URL"/>	<input checked="" type="checkbox"/>
image_link	<input type="text" value="\$image_link"/>	<input type="text" value="URL"/>	<input checked="" type="checkbox"/>
description	<input type="text" value="\$description"/>	<input type="text" value="Text"/>	<input checked="" type="checkbox"/>
price	<input type="text" value="\$price"/>	<input type="text" value="Number"/>	<input type="checkbox"/>
categories	<input type="text" value="categories"/>	<input type="text" value="Categories (List)"/>	<input type="checkbox"/>

[Complete Feed Setup](#)

- a. The “Required?” column will tell us to only sync items that include that designated field. This is auto-populated for fields we require by default and can be optionally toggled for additional fields.
- b. If you have extra metadata you would like to pass for catalog items, you can create a new field to map it to by typing the new field name into the “Klaviyo Field” section for that field and clicking “Create option \_\_\_\_\_”



## Finalizing your catalog

Once you've mapped the catalog feed in your account, the last step in the setup is to reach out to our Support Team with the following information:

1. The name of the metric on your account you would like to use for [personalized recommendations](#).
2. The name of the event property on that metric which corresponds to the unique ID (\$id) used in your catalog feed.

Once this step is complete, you will be able to access and use the Product Feed and Product Block features.

Our Support Team can be reached [via email](#) or in-app Live Chat.

## Using custom catalog items in Product Feeds and Product Blocks

Once our Support Team has finalized your catalog setup, you will be able to use our Product Feeds and Product Blocks in your emails. First, start by creating a Product Feed using the custom catalog feed data. To do this, follow the [steps here](#) and select the list item prefixed with "API" for the "Ecommerce Integration" and "Store Scope" dropdowns (pictured below)

Feed Name

All\_Products

The name you specify can consist of only letters, numbers and underscores and can be up to 32 characters long.

Weighting

Unweighted

Ecommerce Integration

API: Ordered Product

Store Scope

API: Default

Limit to Specific Categories

- No, include all categories
- Yes, only include these categories

Exclude Categories

- No, don't exclude any categories
- Yes, exclude these categories

Create Product Feed »

Now that you've set up a Product Feed using the custom catalog items, you can follow [the guide here](#) to set up a Product Block that uses this feed.

## Using the catalog lookup tag

Catalog lookup tags allow you to reference specific product information from your catalog within your emails using an item's unique ID (\$id). This is useful when you want to create custom product recommendations or cross-reference product information.

For example: if you're using your own product recommendation engine, you can pass a list of product IDs as a custom profile property for a given person. Then, you can iterate through that list in an email and perform a catalog lookup on each ID to pull in the information about that product. More information on this feature can be found [here](#).

## Note on tracking variant items

If your items have variants and you want these to appear in email recommendations, track each variant as a separate item in the feed using either the variant ID of that item or some unique combination of the parent and variant ID. There is currently no way to set up related or variant items in Klaviyo in a traditional parent-child sense so each item must be separately accessible and have a unique ID.

## Advanced Feature - Back in Stock

In addition to being able to use Product Feeds and Product Blocks, you can also enable Back in Stock tracking/alerting for items based on their inventory quantity.

### Feed setup

In order to enable Back in Stock, complete the default catalog steps above and include an additional field for:

- **Inventory** (*\$inventory\_quantity*): a number representing the stock for a given item

You can also include the following optional parameter:

- **Inventory Policy** (*\$inventory\_policy*): a number that controls how we treat product visibility in Product Feeds/Blocks. This field has two available options:
  - **1**: If the policy is set to 1, a product **will not** appear in dynamic product recommendation feeds/blocks if it is out of inventory quantity.
  - **2**: If the policy is set to 2, a product **can** appear in dynamic product recommendation feeds/blocks regardless of inventory quantity. This is also the behavior when this field is not set.

During the field mapping step, these fields should look like:

Feed Item Field	Klaviyo Field	Field Type	Required?
inventory_quantity	<input type="text" value="\$inventory_quantity"/>	<input type="text" value="Number"/>	<input type="checkbox"/>
inventory_policy	<input type="text" value="\$inventory_policy"/>	<input type="text" value="Number"/>	<input type="checkbox"/>

### Account settings & flow triggering

Once the feed setup is done, a new setting will appear in the [Settings > Email Settings](#) on your account. Use this to determine how you would like to trigger Back in Stock triggers

#### Back in Stock Settings

##### Minimum Inventory Rules

Notify customers if  or more units are restocked.

If you set this value to 10, at least 10 units must come into stock for Klaviyo to notify customers. This setting will impact any Back in Stock Flows you have setup in your account. [Learn more.](#)

##### Customer Notification Rules

Notify All Customers

Notify  customers for each unit in stock.

Wait   before notifying the next group of customers.

This delay period determines how quickly groups of customers are notified. Based on your settings, 5 subscribers will be notified for each unit in stock every 1 hours, until either all subscribers are notified or until the number of units drops below the setting threshold defined above.

[Update Back in Stock Settings »](#)

To set up a Back in Stock alert flow, take a look at our [Create a Back in Stock Flow](#) guide.

## Subscribing customers to Back in Stock alerts

When a customer subscribes to be alerted when an item returns in stock, send a POST request to our Back in Stock endpoint. There are server-side and front-end versions of this request.

### Server-side request

This request passes the subscription information as form data in a POST request. The required properties for this request are:

- **a**: your public API key
- **email**: the email address of the person subscribing to alerts for this item
- **variant**: the ID of the item (must correspond with *\$id* in the field mapping step)
- **platform**: this should be set to "api"

An example curl request:

```
curl -X POST \  
  https://a.klaviyo.com/api/v1/catalog/subscribe \  
  -F a=abc123 \  
  -F email=customer@email.com \  
  -F variant=64814-0004 \  
  -F platform=api
```

### Front-end request

A Back in Stock subscription request can also be sent using JavaScript. To do this, include the following parameters:

- **a**: your public API key
- **email**: the email address of the person subscribing to alerts for this item
- **variant**: the ID of the item (must correspond with *\$id* in the field mapping step)
- **platform**: this should be set to "api"

And make an ajax request like the following example:

```
$.ajax({  
  type: "POST",  
  url: "https://a.klaviyo.com/onsite/components/back-in-stock/subscribe",  
  data: {  
    a: "abc123",  
    email: "customer@email.com",  
    variant: "64814-0004",  
    platform: "api"  
  },  
  success: function(response){  
    console.log(response)  
  }  
})
```

### API responses

Successful Back in Stock subscription:

```
{"email": "test@email.com", "success": true}
```

Invalid email address or API key:

```
{"errors": ["There was something wrong with your request. Please try again."], "data": {},  
"success": false}
```

Missing formdata parameters:

```
{"status": 400, "message": "There was something wrong with your request: * email * This field  
is required."}
```

## Invalid variants & duplicate subscriptions

- If an invalid variant ID is included in a request, the API will return a standard success response as the API does not validate variant IDs. No event will be created for this profile, and the customer will not be queued to receive a back in stock message.
- If multiple requests are submitted for the same email address on the same variant, only the first will be accepted. It is not possible to submit multiple subscriptions for a profile on the same variant.
  - There is no limit to the number of unique variants that a profile can subscribe to.
  - After an individual receives an email notifying them that a specific variant is back in stock, their existing subscription is cleared. They are then able to sign up for another subscription on that same variant.