KLEVU

Klevu Personalisation API

(beta v1)

Overview

We are glad to introduce our brand new Personalisation API that has been built on powerful features of Search API (v2) (see https://developers.klevu.com/api#searching).

When a customer enters a physical shop, he or she may express preferences highlighting features of a product he or she wants to purchase. This is usually followed by an in-store assistant asking a few followup questions to establish a profile which he or she uses to align with what they have in-store. The assistant may be actually thinking of past experiences of him dealing with the customers who had similar requirements.

This simple exercise where customers are asked for their preferences and the assistants make their best efforts to present products matching their preferences is called personalisation.

In today's AI-driven digital world, these question answering sessions are possible via what we call shopping chatbots (and Klevu is coming up with one very soon), but otherwise, the task of profile creation is usually based on customers' activities on-site. For example,

- them visiting a certain category of products,
- clicking on a bunch of products with something common in them and/or
- searching with very specific search terms

In this document, our focus is on explaining how we utilise these activities to personalise search results and what features of our API, you can use to personalise search results.

We will look into the following four points related to personalising results:

1) Personalisation based on customer's recent search activity
2) Personalisation based on collaborative filtering method
3) Personalisation based on manual inputs
4) Impact of enabling personalisation on search relevance and manual promotions

But before doing so, we have highlighted a set of requirements that you must meet.

Prerequisites

Who is this guide for?

If you are using one of our plugins (e.g. Magento, Shopify, Bigcommerce) where the search results page and category pages (if the cat-nav add-on is enabled) are rendered using our Javascript based templates, the only step you need to take to enable personalisation is follow instructions on the following page: //TODO provide a link to support article. If you are using Klevu JS Library or a custom integration using our API, please read on for instructions to enable personalisation.

!	If you are on a Magento platform and using our preserve layout option for your search results landing page and/or category pages, unfortunately, the personalisation cannot be enabled. **Why?** Unless you disable full page caching by Magento, it will cache every page that is rendered and shown to your shoppers. Once a page is cached, any subsequent request to see that page is served from the cache. Here, if we enable personalisation, Magento will cache the first personalised page and continue to show the same to others. This, we think is not appropriate. **Is there any workaround?** It isn't recommended but should you want to enable personalisation with preserving your theme layout, the only possible way is to disable the full page caching. You should also follow instructions on the following page to get the personalisation activated. //TODO provide a link to the support article

What you need to get started?

- You must have a Klevu Search account
- You should have indexed your data with the Klevu Search engine
- For every store that you have configured with Klevu Search, you should have the following details:
    - Klevu API key
    - Klevu Search endpoint

    - Klevu Category Navigation endpoint (optional)
- You must have the personalisation feature enabled for your store (see TODO provide a link to the support article)

Klevu API Key

It has the prefix **klevu-** which is followed by a sequence of numbers unique to your search index (e.g. klevu-14666680139423664). If you have signed up for multiple stores, you should have a unique API key for each of your stores. Unless you have explicitly confirmed with the Klevu Search team, no two stores, will ever have the same Search API key.

Klevu Search Endpoint

The endpoint looks like the following:

https://<subdomain>.ksearchnet.com/cs/v2/search

Here, a subdomain is assigned to your store depending on the plan of your account and the country of your store.

!	As we maintain a separate index for each of your stores, it is totally possible that you have totally different sub-domains assigned to your other stores.

It is important to take a note of these parameters and use the relevant values when firing search queries to the Klevu Search engine.

Klevu Category Navigation Endpoint

If you have opted in to power your category pages with Klevu Search, you should have also received an endpoint where you can query to retrieve data for your category / collection pages. The endpoint may remain the same as the Klevu Search endpoint. When in doubt, please get in touch with the Klevu support team.

GET vs POST

The query can either be fired using a simple query parameter (i.e. a GET request)

GET https://<subdomain>.ksearchnet.com/cs/v2/search?q=<jsonBody>

or using the request body (i.e. a POST request)

POST https://<subdomain>.ksearchnet.com/cs/v2/search
Headers:
- Content-type: application/json

Body: <jsonBody>

Personalisation Based on Recent Search Activity

When a shopper enters an online shop, he or she usually browses through a catalog to find a relevant product of his or her choice. Many shoppers prefer to use search if they know exactly what they are looking for. If the customer is using search and has clicked on a few products, in most cases this is a good indicator that he or she may have spotted something of interest.

We perform semantic and statistical analysis of this data to identify the intent and to build profiles of customers' preferences. As part of the semantic analysis, a relationship between the query and the products recently clicked by the customer is established. Only those products relevant to the current search query, **provided there are at least 3 of them**, are considered for further statistical analysis.

The process of analysing these records involves identifying common factors. It is possible that a few features are common and others have less similarly. Based on what is common, boosting weights for the individual feature are calculated and used when firing the original search query.

Here is an example of a search query explaining how to enable out of the box personalisation and provide recently clicked products as a context. This is the simplest way to enable personalisation, with Klevu doing all of the hard work in the background.

POST https://<your-website-specific-subdomain>.ksearchnet.com/cs/v2/search
Headers: contentType: application/json

```
{
  "Context":{ "apiKeys":["klevu-1234567890"] },
  "recordQueries": [{
    "id": "catInfo",
    "typeOfRequest": "SEARCH",
    "settings": {
      "typeOfRecords": ["KLEVU_PRODUCT"],
      "limit": 5,
      "query":{ "term":"door" },
      "personalisation": { "enablePersonalisation": true },
      "context": {
        "recentObjects":[{
          "typeOfRecord": "KLEVU_PRODUCT",
          "records": [{"id":"42971216518"},{"id":"42971216623"},{"id":"52934543"}]
        }]
      }
    }
  }]}
```

Here,

- **apiKeys**: please use your store specific API key
- **enablePersonalisation**: **true** instructs the backend engine to activate the personalisation engine
- **recentObjects**: this is where the products recently clicked by the customer should be provided. You may supply recently clicked records such as products, categories etc but only the recent objects of the same type as provided under the **settings** of the search query (***italicized*** *and* ***bold*** in the query above - i.e. KLEVU_PRODUCT) are considered for the personalisation purpose.

Personalisation Based on Collaborative Filtering

The word *collaboration* means *a joint effort*. In the context of personalisation, when there isn't enough data available (i.e. at least 3 products recently visited by the customer that are relevant to the search query), Klevu's personalisation engine has the ability to refer to other customers' activities and establish what this customer may also like to see.

For the collaborative filtering to kick-in, in a query, there must be at least two recently clicked records provided. Assuming our engine finds them relevant to the search query, these two products are then looked up in a table visited by other customers. Here, what appear as the most probable products to be liked by the customer are considered for statistical analysis.

How can you control when to invoke collaborative filtering?

You are required to enable personalisation and provide a set of recently clicked records in your request. There isn't any need to provide any additional parameter to enable collaborative filtering. You can set the threshold (i.e. the minimum number of recently clicked records) from the Klevu Merchant Center by visiting the **Customizations ⇒ Advanced Personalized Search ⇒ Min Clicks to Invoke Collaborative Personalization.**

Personalisation Based on Manual Inputs

We have identified at least two possible venues where merchants may like to provide their input:

1) They would like to use out-of-the-box (OOTB) personalisation but want to specify what attributes to consider when building personalisation profiles.

   Merchants know their domain very well. They know what attributes matter the most when it comes to shoppers choosing their products. For example,

- In a mobile shop, the brand of a mobile phone may be more relevant than the storage capacity. It may be that the merchants like to focus on a price range more than a color.
- In an apparel shop, gender and the age group may be more relevant than any other attribute.
- In a book shop, the genre of the book may be the most relevant, followed by an author.

The OOTB engine will anyway figure out what is common, however, if Merchants can participate in what is called Active Learning, where systems are given help to learn faster, it is always going to help converting customers faster. With API v2, merchants can give this input by using the "**fields**" option under the **personalisation** object (see below).

POST https://<your-website-specific-subdomain>.ksearchnet.com/cs/v2/search
Headers: contentType: application/json

```
{
  "context":{ "apiKeys":["klevu-1234567890"] },
  "recordQueries": [{
    "id": "catInfo",
    "typeOfRequest": "SEARCH",
    "settings": {
      "typeOfRecords": ["KLEVU_PRODUCT"],
      "limit": 5,
      "query":{ "term":"door" },
      "personalisation": {
        "fields":["brand","klevu_price"]
        "enablePersonalisation": true
      },
      "context": {
        "recentObjects":[{
          "typeOfRecord": "KLEVU_PRODUCT",
          "records": [{"id":"42971216518"},{"id":"42971216623"},{"id":"52934543"}]
        }]
      }
    } }]}
```

Here,

fields: this is an array of strings where names of attributes that you want the Personalisation engine to consider (i.e. and no other attributes), should be provided.

2) Merchants already know what their customer would like to see and they can instruct the machine to boost certain products and/or attributes with specific values.

   For example, in an auto parts store, when a customer has logged in or has provided information about the vehicle they own (e.g. a Mercedes model), merchants may want to promote specific types of products (e.g. those within premium ranges or suitable to the Mercedes model).

   For such cases, Klevu API provides an object called **boost.** There are four ways products can be boosted:

   - one or more filter conditions with some weights (i.e. between 1 and 99)
   - one or more keywords or phrases with some weight
   - one or more product Ids with some weight
   - by firing another query and using its results to boost products

POST https://<your-website-specific-subdomain>.ksearchnet.com/cs/v2/search
Headers: contentType: application/json

```
{
  "Context":{ "apiKeys":["klevu-1234567890"] },
  "recordQueries": [{
    "id": "catInfo",
    "typeOfRequest": "SEARCH",
    "settings": {
      "typeOfRecords": ["KLEVU_PRODUCT"],
      "limit": 5,
      "query":{ "term":"wiper" }
    },
    "boost": {
      "filters":[{
        "key":"brand",
        "values":["Mercedes"],
        "weight":50
      }],
      "keywords": [{
        "phrase": "premium",
        "weight": 30.0
      }]
    }
  }}
```

Here,

boost: you can specify different keyword specific conditions and or keywords with appropriate weights. It is important to note that

- any boosts applied using the **boost** node may result in products being ranked differently, however, it does not change the overall result set itself.
- when using the **boost** object, you don't have to specify the **"enablePersonalisationPlease":true** field. In fact you should omit the personalisation node entirely.

Please see the following link to learn more about the **boost** node:
https://developers.klevu.com/api#the-boost-object-personalisation

Impact of Enabling Personalisation on Search Relevance and Manual Promotions

It is important to mention that personalisation is not about filtering products away. It is about boosting one or more types of products that are found to be in better alignment with the customer's preferences. However, within what we find relevant to the customer's preferences, it is also important that we boost products that are promoted by you. Below, we have highlighted the different merchandising features and how do they get affected by enabling personalisation:

1) Keyword specific product promotions

   Keyword specific promotions will be preserved and the products promoted to be shown at the top will be shown at the top of the search results. Personalisation will be applied only on the rest of the products.

2) Keyword specific product exclusions

   Products excluded for specific keywords will continue to be excluded from the search results, even though they may look very relevant to the customer's preferences.

3) Individual and rule-based product promotions

   Here, there are no hard rules but in general if the products boosted manually are matching customers' preferences, they will be boosted even higher. Here, we look at an example. Let us assume that

   - For a query, some products from brand A, B, C and D are relevant.
   - As a merchant, you have equally promoted products of brand A and brand B.
   - Thus, without personalisation, products relevant to the query that are from brand A and B will be promoted and shown towards the top of the search results.

- If we realise that the customer is interested in brand B (more than the brand A), the personalisation engine will boost products of brand B higher up.
- If we notice that brand C is what the customer is interested in, depending on the boosting scores of brand A and B products, it may happen that they will see products of brand C appearing higher up but also may be together with the products of brand A and brand B.

4) Category specific product promotion

   These are hero products which have been chosen by you to be displayed at the top of the category. We will respect it. After the top products have been displayed, personalisation will be applied to the rest of the products.

5) Category specific rule-based promotions

   Similar to how personalisation impacts results on search results pages, here too, there are no hard rules and in general if the products boosted manually are matching customers' preferences, they will be boosted even higher. The same example explained earlier is true here as well.