Different scenarios for retrieving product information using the PS-API

This document describes the most common scenarios for retrieving product information using the PS-API. These scenarios are an addition to the general API documentation (see for more information on the API website https://webapi.psinfoodservice.com/v50/prod). The scenarios are described as a step by step tutorial.

This document assumes that you have basic knowledge about REST (GET and POST) calls and that you are in the possession of a valid PS-API login.

Scenario 1: Retrieve new and updated product information

This scenario is useful if you want to retrieve all available (and accessible) product information with regular updates. This scenario is for all users, except for those who want information based on GTINs (see scenario 2)

The described steps can be repeated once a day, once a week or an interval of you chosen.

Used API calls:
https://webapi.psinfoodservice.com/v50/prod/api/Product/Search (Search)
The call is a POST call and parameters are send with Content-Type: application/x-www-form-urlencoded

Local administration:

- Date of the last sync (*LastSyncDate*), with an initial date (first run) in the far past (2000-01-01)
- A queue with PS-IDs for products which are not yet downloaded (*DownloadQueue*)

Step 1: Retrieve available PS-IDs which are changed since the last sync

Call the Search API with the following parameters:

- ShowPublicProductSet = true
- ShowMyBrandsProductSet = true (for producers and private label owners)
- ShowSubscribedProductSet = true
- IncludeAllTargetmarkets (if you want all target markets TM)
- FilterOnTargetmarketIds (if you want a specific TM NL=1, BE=2, or NL+BE = 1;2)
- IdOnly = true (this result in a list of available PS-IDs)
- Skip = 0
- Take = 1000000 (large number)
- LastModifiedAfter = LastSyncDate (because the initial date is in the far past, all products information is retrieved at the first time)

Result: a list of all PS-IDs (; token separated). E.g. 44;60;61;62;65;144;146;149;150;154;

Step 2: Save the PS-IDs in the DownloadQueue and update the LastSyncDate

You now have a list of PS-ID's which are available, accessible and need an update. Store these IDs in the *DownloadQueue* and update the *LastSyncDate* with the DateTime of the start of the process

Step 3: Download product information until the DownloadQueue is empty

Now that we have all the PS-IDs which need an update in the DownloadQueue, we can download them in batches of max 50 products at once. Therefore, we fill the FilterOnPSIds parameter with (max 50) PS-IDs from the DownloadQueue. FilterOnPSIds accepts an ; token separated string. We don't use the LastModifiedAfter filter.

Call the Search API with the following parameters (because you PS-IDs are the result of the previous step with filters, the same filters are now obsolete)

- ShowPublicProductSet = true
- ShowMyBrandsProductSet = true (for producers)
- ShowSubscribedProductSet = true
- IncludeAllTargetmarkets = true
- IdOnly = false
- Skip = 0
- Take = 1000000 (large number)
- FilterOnPSIds = 44;60;61;62;65;144;146;149;150;154; (**MAX 50 IDs**)

Result: an XML with multiple products.

Store these products locally and process them in your own system (the PS-ID can be used as reference).

Repeat this step until the DownloadQueue is empty.

Step 4: Remove deleted products

When a product is removed in the PS database we cannot send you a message (we do not know which products you store locally), so you have check it yourself. Therefore, we will retrieve all available and accessible PS-IDs and if there are products in your local database with a PS-ID that is not in this list you know that this product is removed and you will not receive any updates.

Note: NEVO products may need some extra attention in your local database

Call: https://webapi.psinfoodservice.nl/v50/prod/api/product/ProductIdList?TokenSeparated=true

Result: a list of PS-IDs (; token separated). E.g. 44;60;61;62;65;144;146;149;150;154;

Reference the PS-IDs with your local store and remove PS-IDs which are no longer available at PS

Scenario 2: Retrieve product information based on your own assortment

If you have an assortment of your own, you can execute a match (based on GTINs or a producer GLN with producer article number combination) and see which of the products are available in the PS database. Next, you will collect all the matches and download them.

The best result is achieved with a match based on GTINs

Used API calls:
Search: https://webapi.psinfoodservice.com/v50/prod/api/Product/Search
The call is a POST call and parameters are send with Content-Type: application/x-www-form-urlencoded

Match: https://webapi.psinfoodservice.nl/v50/prod/api/conversion/MatchRaw
The call is a POST and send a raw XML in the body (no application/x-www-form-urlencoded). An example of such a XML can be found in Appendix A.

Local administration:

- Date of the last sync (*LastSyncDate*), with an initial date in the far past (2000-01-01)
- A queue with PS-IDs for products which are not yet downloaded (*DownloadQueue*)
- A *MatchList*, with all desired gtins and the corresponding PS-ID of previous matches

Step 1: Retrieve updates of previous matches

Check if there are updates of previous downloaded products (PS-IDs in the MatchList) since the last run (LastSyncDate). The first run the matchlist will be empty.

Call the Search API with the following parameters:

- ShowPublicProductSet = true
- ShowMyBrandsProductSet = true (for producers)
- ShowSubscribedProductSet = true
- IncludeAllTargetmarkets (if you want all targetmarkets TM)
- FilterOnTargetmarketIds (if you want a specific TM NL=1, BE=2, or NL+BE = 1;2)
- IdOnly = true
- Skip = 0
- Take = 1000000 (large number)
- FilterOnPSIds = 44;65 (all matched PS-IDs from previous runs)
- LastModifiedAfter = LastSyncDate (because the initial date is in the far past, all products information is retrieved)

Result: a list of available PS-IDs (; token separated). E.g. 44;60;61;62;65;144;146;149;150;154;

Step 2: Update download queue and set LastSyncDate

You now have a list of PS-ID's which are available, accessible and need an update. Store these IDs in the *DownloadQueue* and update the *LastSyncDate* with the DateTime of the start of the process

Step 3: Put all GTINs without a match in the match procedure

Put all GTINs which are new or unmatched in the previous runs in the input XML as described in Appendix A and call the Match API. Although, there is no limit on the number GTINs it is best to do this in batches of max 2000 GTINs.

Result: the same XML as the input, with a PSID element if there is a match (See in Appendix B for an example)

Step 4: Add new matches to the MatchTable and DownloadQueue

Add the matches found in the previous step to the MatchTable and DownloadQueue.

Step 5: Get products based on the DownloadQueue

See scenario 1 – step 3

Step 6: Remove deleted products

See scenario 1 – step 4

Additional you need to update the MatchTable with deleted products and remove those matches

Appendix A: example of an input for the match procedure

Repeat the element psidmatch for each GTIN.

```
<?xml version="1.0" encoding="UTF-8"?>
<envelope>
  <header>
    <origin>PSXML</origin>
    <version>1.0.0.0</version>
    <provider>PS In Foodservice</provider>
    <updateproductinfo>0</updateproductinfo>
    <updatespecificationinfo>0</updatespecificationinfo>
    <updatecommercialinfo>0</updatecommercialinfo>
    <updatelogisticinfo>0</updatelogisticinfo>
    <actiontype>GET</actiontype>
  </header>
  <body>
    <psidmatchlist>
      <psidmatch>
        <gtince>4003659011164</gtince>
        <gtinhe>4003659011164</gtinhe>
      </psidmatch>
      <psidmatch>
        <gtince>4008728255128</gtince>
        <gtinhe>4008728255128</gtinhe>
      </psidmatch>
      <psidmatch>
        <producergln>8713883999997</producergln>
        <producerarticlenumber>PS00267</producerarticlenumber>
      </psidmatch>
    </psidmatchlist>
  </body>
</envelope>
```

Appendix B: example of a match result

Example of a match result. It contains all GTINs of the input with a PSID element when there is a match. The element matchingrulename contains the criteria where the match is based on.

```
<?xml version="1.0" encoding="utf-8"?>
<envelope>
  <header>
    <provider>PS In Foodservice</provider>
    <disclaimer>PS In Foodservice</disclaimer>
    <version>5.1.9</version>
    <totalrowcount>0</totalrowcount>
    <origin>PSXML</origin>
    <actiontype>GET</actiontype>
    <updateproductinfo>0</updateproductinfo>
    <updatespecificationinfo>0</updatespecificationinfo>
    <updatecommercialinfo>0</updatecommercialinfo>
    <updatelogisticinfo>0</updatelogisticinfo>
    <executiontime>0001595</executiontime>
  </header>
  <body>
    <psidmatchlist>
      <psidmatch>
        <psid>185700</psid>
        <catalogownergln>8714406000008</catalogownergln>
        <catalogownerarticlenumber>01000008</catalogownerarticlenumber>
        <matchingrulename>Match Supplier GLN +
CatalogOwnerArticleNumber</matchingrulename>
      </psidmatch>
      <psidmatch>
        <producergln>8713883999997</producergln>
        <producerarticlenumber>PS00267</producerarticlenumber>
      </psidmatch>
      <psidmatch>
        <psid>65</psid>
        <gtinhe>4008728255128</gtinhe>
        <matchingrulename>Match EAN(HE)</matchingrulename>
      </psidmatch>
      <psidmatch>
        <psid>44</psid>
        <gtince>4003659011164</gtince>
        <matchingrulename>Match EAN(CE)</matchingrulename>
      </psidmatch>
      <psidmatch>
        <gtince>8712423002593</gtince>
      </psidmatch>
    </psidmatchlist>
```

```
   </body>
</envelope>
```