Nutritionix API v2 Documentation [PUBLIC]

Table of Contents

Summary

The Nutritionix API v2 is geared towards making it as easy as possible for a user to track calories they consume while eating, and calories they expend while exercising.

The Nutritionix API v2 powers the Nutritionix Track App (ios, android and web). We recommend trying out the Track App before developing on this API to become familiar with its capabilities.

Please note, any user interface you develop that accesses the Nutritionix API is **required** to have attribution to Nutritionix.

Live API Demos

- Test the /v2/search/instant endpoint by using the search box at the top of nutritionix.com
- Test the natural API endpoint on our demo page
- Try our food logging app on iOS, Android, or Chrome: Nutritionix Track

Important Resources

- Comprehensive list of v2.0 API endpoints and parameters: https://trackapi.nutritionix.com/docs
- List of all nutrient attributes and nutrient IDs from API (full_nutrients array attr_id definitions)
- When starting out with our API, we **strongly** recommend using a standalone REST client like Postman to test your API calls before writing any application code.

Obtaining API Keys & Authenticating

1. Signup for a Nutritionix API developer account at developer.nutritionix.com
2. When logged into the Nutritionix developer console, get your **APP ID** and **APP Key** at this page: https://developer.nutritionix.com/admin/access_details
3. Required **HEADERS** when accessing Nutritionix V2 API endpoints:
    a. **x-app-id:** Your app ID issued from developer.nutritionix.com)
    b. **x-app-key:** Your app key issued from developer.nutritionix.com)

c. **x-remote-user-id:** A unique identifier to represent the end-user who is accessing the Nutritionix API. If in development mode, set this to 0. This is used for billing purposes to determine the number of active users your app has.

Please note, when authenticating with the API, you must send the x-app-id and x-app-key params as **headers**, and not as query string parameters.

Food Lookup Endpoints

/v2/natural/nutrients

Summary: Get detailed nutrient breakdown of any natural language text. Can also be used in combination with the /search/instant endpoint to provide nutrition information for common foods.
POST https://trackapi.nutritionix.com/v2/natural/nutrients
Example request to /v2/natural/nutrients and example response

Detailed API docs here.

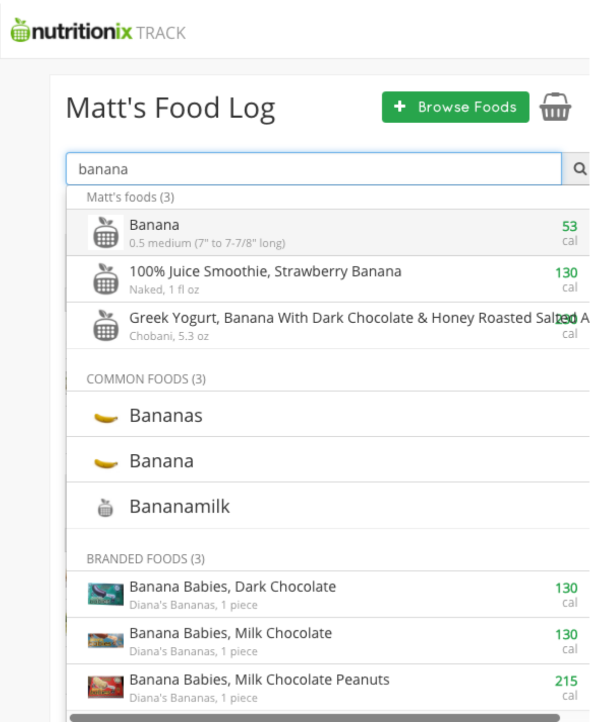
List of all nutrients and nutrient IDs from API (full_nutrients array attr_id definitions)

/v2/search/instant

Summary: Populate any search interface, including autocomplete, with common foods and branded foods from Nutritionix. This searches our entire database of 600K+ foods. Once a user selects the food from the autocomplete interface, make a separate API request to look up the nutrients of the food.
GET https://trackapi.nutritionix.com/v2/search/instant
Example request and response to /v2/search/instant endpoint

Expanded docs here.

You can see a functional demo on our Track web app here.



Matt's Food Log

banana

Matt's foods (3)

Banana 53 cal
0.5 medium (7" to 7-7/8" long)

100% Juice Smoothie, Strawberry Banana 130 cal
Naked, 1 fl oz

Greek Yogurt, Banana With Dark Chocolate & Honey Roasted Salted A 200 cal
Chobani, 5.3 oz

COMMON FOODS (3)

Bananas

Banana

Bananamilk

BRANDED FOODS (3)

Banana Babies, Dark Chocolate 130 cal
Diana's Bananas, 1 piece

Banana Babies, Milk Chocolate 130 cal
Diana's Bananas, 1 piece

Banana Babies, Milk Chocolate Peanuts 215 cal
Diana's Bananas, 1 piece

Example usage of /search/instant endpoint in action on Nutritionix Track web app

The results from the /v2/search/instant endpoint are separated into 3 arrays:

- **branded**: results from Nutritionix.com branded food database (restaurant brands and grocery brands)
- **self**: results from a user's own food log (only if header x-user-jwt is provided, otherwise this array is null)
- **common**: results from Nutritionix.com common food database (curated list of averaged food items that are not brand-specific)

/v2/search/instant supports the following **query string parameters**:

param	*description*	*default*
branded	boolean, determines whether to include branded foods (grocery and restaurant) in the results.	true
brand_ids	array[string], accepts array of Nutritionix brand IDs to restrict search results to only include those brands. Useful for the location endpoint	null
branded_region	integer, filters branded results to only those from a specific region. 1=US, 2=UK	null (US Only)
taxonomy	boolean, determines whether to include taxonomy node id in the results where one exists.	false
taxonomy_node_id	string, filters results to only those with the specified taxonomy node id.	null

More details about support params for /v2/search/instant available at:

https://trackapi.nutritionix.com/docs/#!/default/get_v2_search_instant

To filter search results by nutrient ranges (such as calories < 500), see this answer.

To look up nutrition data for results from the /instant/ endpoint, see this FAQ answer.

/v2/search/item

Summary: Look up the nutrition information for any *branded food* item by the nix_item_id (from /search/instant endpoint) or UPC scanned from a branded grocery product.
GET https://trackapi.nutritionix.com/v2/search/item
Resources:

- Expanded API docs for this endpoint.
- Example request and response

Implementing UPC scanning? Use this doc with example UPCs to scan for QA and development.
Please also see our best practices for UPC scanning.

Exercise Endpoints

/v2/natural/exercise

Summary: Estimate calories burned for various exercises using natural language. Developer can optionally include user demographics like age, gender, weight to make a more accurate estimate for calories burned.
POST https://trackapi.nutritionix.com/v2/natural/exercise
Resources:

- Example request and response
- Live Demo of Natural Exercise Endpoint.
- Expanded docs here.

Location Endpoint

/v2/locations

Summary: Returns a list of restaurant locations near a lat/long coordinate. Can be used in conjunction with the search endpoint. Supports “point distance” and “bounding box” queries
GET https://trackapi.nutritionix.com/v2/locations
Resources:

- Live demo
- Expanded API doc here.
- Request/response example
- JSFiddle example of using Location Endpoint with /v2/search/instant endpoint

At this time, we only support US restaurant locations, and we do not have a timeline for supporting locations outside of the US.

If you are trying to return restaurant menu items near a specific lat/long, submit the lat/long to the /locations endpoint to return a list of nearby brand IDs, then make a query to the /search/instant endpoint with the brand IDs that were returned from the /locations endpoint.

Track App End-User Endpoints (Enterprise Only)

Summary: Endpoints related to looking up your own user's Track app data via API. Note: this functionality is available as a paid-add on for enterprise plans only. End-user must have a Nutritionix Track user account, and it requires the email address and password for that Track account.

/v2/auth/signin

Summary: Use this endpoint to get the user-jwt, which is the authentication token that all other end-user endpoints require.
POST https://trackapi.nutritionix.com/v2/auth/signin
Resources:

- Expanded API doc here.

/v2/log

Summary: Get foods from user's food log. Note: this functionality is available as a paid-add on for enterprise plans only. Requires the x-user-jwt header param from the v2/auth/signin endpoint.
GET: https://trackapi.nutritionix.com/v2/log
Resources:

- Expanded API doc here.

Oauth Flow for Track User Data (Enterprise Only)

Summary: This method requires previous authorization by Nutritionix. For approved developers, you can allow existing Nutritionix Track users to link their Nutritionix account with a 3rd party application, and gain access to foods a user logs in the Nutritionix Track app.

- **Step 1:** Redirect users to https://www.nutritionix.com/authorize
  - must include "client_id", "response_type" (must equal 'code'), & "redirect_uri" (must match redirect uri for client_id). If you need to proxy stateful logic internal to your application, it must be included in the "state" param, which we will proxy for you. These parameters exist in the querystring.
- **Step 2:** Nutritionix redirects user's browser to the redirect_uri, along with a "code" to be used in exchange for a token.

- **Step 3:** Your server, listening at the specified redirect_uri , must then POST to the /v2/oauth/server/token route
  - You must set the content-type header to 'application/x-www-form-urlencoded'
  - You must include the "client_id", "client_secret", "grant_type" (must be "authorization_code" for this step), and "code" from step 2.
- **Step 4:** Assuming everything is valid, you will receive an access token & refresh token.
  - The access tokens have a limited lifetime (1 day).
  - The refresh tokens can be used to fetch new access tokens when needed (via the same endpoint: /v2/oauth/server/token), except instead of providing the "code" param, one provides the "refresh_token" param & the "grant_type" must be "refresh_token" instead of "authorization_code"

Localization (Enterprise Customers Only)

To see a live demo of our supported locales, please visit:
https://www.nutritionix.com/labs/localization

Current supported locales:

- de_DE (German)
- fr_FR (French)
- es_ES (Spanish - Spain)
- en_GB (UK)

The below endpoints are available with localization:
/v2/search/instant
Example request and response with localization
Full /v2/search/instant documentation (not specific to localization)

/v2/natural/nutrient
Example request and example response to /v2/natural/nutrients with localization
Full /v2/natural/nutrient documentation (not specific to localization)

For support with localization, please email enterprise@nutritionix.com

Attribution Requirement

All interfaces which hit the Nutritionix API are required to give attribution to Nutritionix. The format of the attribution is relatively flexible, but it must be reasonably visible to the end-user without necessitating scrolling.

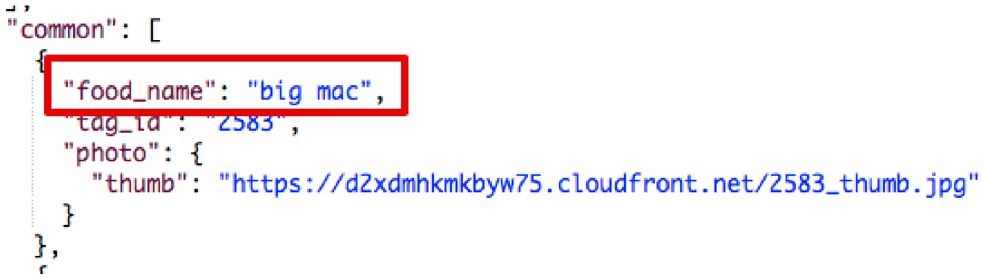
Click here to access downloadable attribution logo files. It is not a requirement to use these exact images, feel free to have your own team customize these logos to fit your app UI.

Frequently Asked Questions by Developers (FAQ)

How do you get nutrition information for foods in the /search/instant endpoint?

When you hit the /search/instant endpoint it returns two arrays: common and branded foods.

To get the nutrition information for **common foods**, take the food_name that is returned from any item in the “common” array, e.g. “Big Mac” and POST that as the query to our /natural/nutrients endpoint.



```
"common": [
  {
    "food_name": "big mac",
    "tag_id": "2583",
    "photo": {
      "thumb": "https://d2xdmhkmkbyw75.cloudfront.net/2583_thumb.jpg"
    }
  },
```

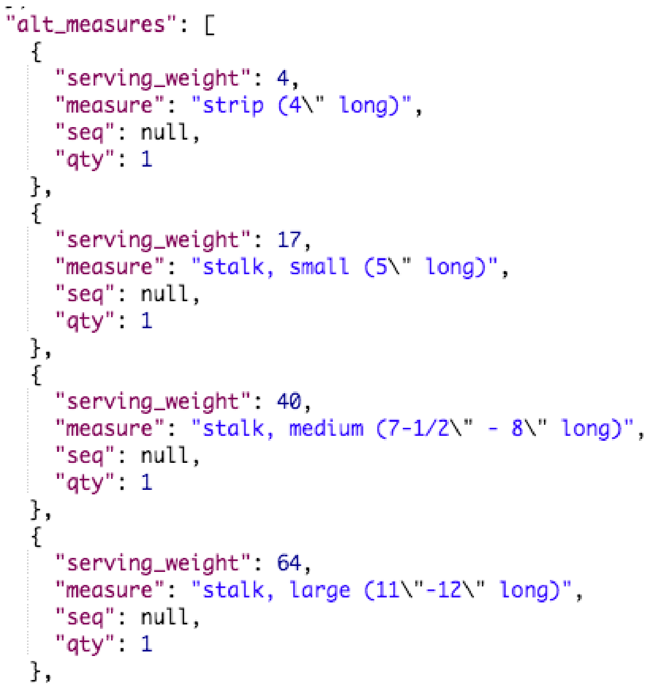
To get the nutrition information for **branded foods** take the value of the “nix_item_id” attribute from any of the objects in the “branded” array, e.g. 513fc9e73fe3ffd40300109f, and hit the /search/item endpoint.

How do you get a list of all of the definitions for the “full_nutrients” array?

This spreadsheet contains the list of all nutrients that are returned in the /search/item and /natural/nutrients endpoints.

How do you get the available serving sizes for a food?

POST https://trackapi.nutritionix.com/v2/natural/nutrients and look for the “alt_measures” array:

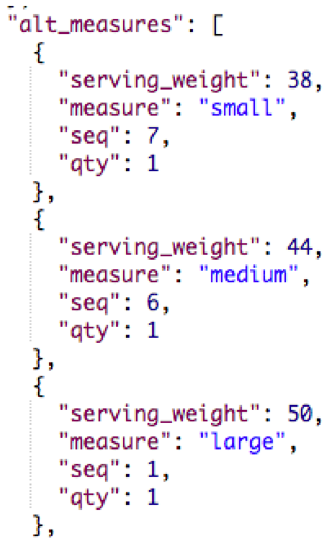


```
"alt_measures": [
  {
    "serving_weight": 4,
    "measure": "strip (4\" long)",
    "seq": null,
    "qty": 1
  },
  {
    "serving_weight": 17,
    "measure": "stalk, small (5\" long)",
    "seq": null,
    "qty": 1
  },
  {
    "serving_weight": 40,
    "measure": "stalk, medium (7-1/2\" - 8\" long)",
    "seq": null,
    "qty": 1
  },
  {
    "serving_weight": 64,
    "measure": "stalk, large (11\"-12\" long)",
    "seq": null,
    "qty": 1
  },
```

Using the serving_weight parameter value from the alt_measure array, you can derive the nutrients in each serving by dividing that serving weight by the default returned serving weight of the food object, and multiplying by the full_nutrients array:

(foods.alt_measures.serving_weight / foods.serving_weight_grams) * full_nutrients

For example, if you hit natural API with “3 large eggs”, the alt measures looks similar to this this:



```
"alt_measures": [
  {
    "serving_weight": 38,
    "measure": "small",
    "seq": 7,
    "qty": 1
  },
  {
    "serving_weight": 44,
    "measure": "medium",
    "seq": 6,
    "qty": 1
  },
  {
    "serving_weight": 50,
    "measure": "large",
    "seq": 1,
    "qty": 1
  },
```

The full_nutrients array will include the nutrient values for 3 large eggs, which weigh 150g. As a quality assurance check, we now know that 150g of eggs has 214.5 calories.

Now, let’s say the user only wants nutritionals for 1 small egg, instead of the original 3 large eggs. Instead of hitting the natural API again, you can calculate the new nutrition values by using the serving_weight in the alt_measures array. We know from alt_measures that 1 small egg is 38 grams. We know the nutrient values of 50g of eggs, so with some quick algebra we can calculate the nutrient values of any of the alt_measures.

There are a few different ways to do this and still get the same answer:

Method 1:
Take our original equation above:
(foods.alt_measures.serving_weight / foods.serving_weight_grams) * full_nutrients
So the calories of one 38g small egg would be, (38g/150g)*214.5kcal = 54.3kcal.

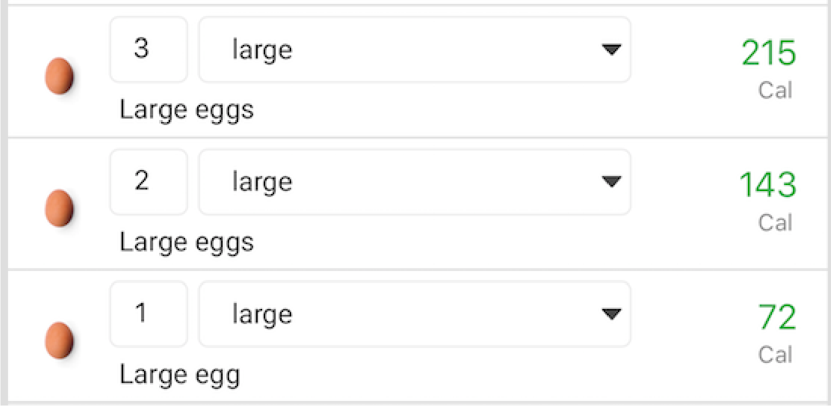
For development purposes, you can check your answer using our natural API and confirm that 38g egg is 54.3kcal.

Method 2:
If you want to have even more flexibility to calculate the nutrients for any arbitrary gram weight, divide the values in the full nutrients array by the foods.serving_weight_grams value, to get the nutrient values of 1 gram of the food (aka: “1g nutrients array”). In our 3 eggs example, you would divide the entire full nutrients array by 150g.

To calculate any arbitrary gram measures of eggs, you would multiply the desired gram value by the new “1g nutrients array”. All of these calculations can be done on-the-fly in the client, no need to make additional API calls.

Important note about serving sizes:
If your UI allows the user to change the quantity of an item after hitting our natural API, it is important to maintain the original quantity value from our API to properly calculate any future user-defined derivations. In our same example above, we start with 3 large eggs. If the user changes the quantity to 2 large eggs, in the

background you are generating a multiplier by dividing the user-defined quantity (2) by the original API quantity (3).



3 large
Large eggs
215 Cal

2 large
Large eggs
143 Cal

1 large
Large egg
72 Cal

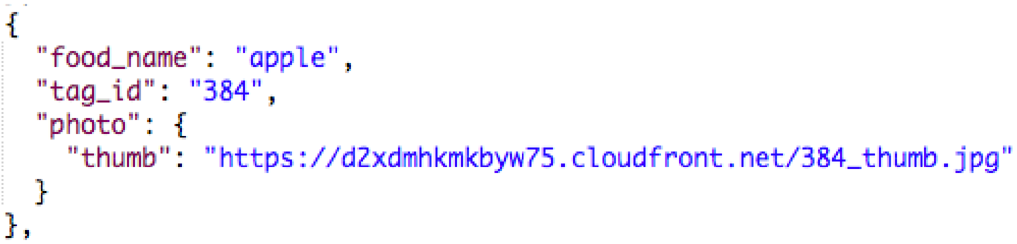
(example from the “view basket” page of the Nutritionix Track app)

So the multiplier in this example would be (⅔). If you multiply the full nutrients array of 3 large eggs by ⅔ you will get the nutrients for 2 large eggs.

This same logic can be applied when the user edits a quantity of any alt_measure. **Always perform a QA check by visiting our natural demo page and testing the values in your app against ours to make sure your math is right!**

How do you get an image for each returned food?

POST https://trackapi.nutritionix.com/v2/natural/nutrients and look for the “photo.thumb” object.



```
{
  "food_name": "apple",
  "tag_id": "384",
  "photo": {
    "thumb": "https://d2xdmhkmkbyw75.cloudfront.net/384_thumb.jpg"
  }
},
```

Check out our natural demo page to see what the photos look like.

How do I restrict search results to a specific brand or brands?

If you are hitting the /search/instant endpoint, you can restrict the results to a list of brands by appending an array of Nutritionix Brand IDs as the value of the “brand_ids” param on the /search/instant endpoint. More details available here.

This is particularly useful when used in conjunction with the restaurant location endpoint, so you can return menu items that exist nearby.

What are the dimensions of images that are returned from the API?

All images returned by the API will be constrained to a specific **width**, and slightly varying height that will not exceed the width. In the client, you should plan to scale down images proportionally, and vertically center the images in your UI to ensure they always look uniform.

/search/instant endpoint:
Small thumbnails of CPG products and common foods
150px width, variable height up to 150px

/natural/nutrient & /natural/exercise endpoints:
Small thumbnails and large images of foods and exercises.
small: 150px width, variable height up to 150px
large: 1080 px width, up to 1080px height

Here are 5 example sets of images from this natural language query:

highres: https://d2xdmhkmkbyw75.cloudfront.net/456_highres.jpg
thumb:https://d2xdmhkmkbyw75.cloudfront.net/456_thumb.jpg

highres: https://d2xdmhkmkbyw75.cloudfront.net/399_highres.jpg
thumb: https://d2xdmhkmkbyw75.cloudfront.net/399_thumb.jpg

highres: https://d2xdmhkmkbyw75.cloudfront.net/2583_highres.jpg
thumb: https://d2xdmhkmkbyw75.cloudfront.net/2583_thumb.jpg

highres: https://d2xdmhkmkbyw75.cloudfront.net/384_highres.jpg
thumb: https://d2xdmhkmkbyw75.cloudfront.net/384_thumb.jpg

highres: https://d2xdmhkmkbyw75.cloudfront.net/190_highres.jpg
thumb: https://d2xdmhkmkbyw75.cloudfront.net/190_thumb.jpg

How do I get nutrient values from the /search/instant endpoint?

As of September 2017, the /search/instant endpoint can now return nutrient values by adding the query string param “detailed=true”. For example: /search/instant?query=cheese&detailed=true

Example gist here.

I am having issues logging into my developer account.

- Make sure you are trying to login via this URL: https://developer.nutritionix.com/login
- Try the lost password function on the developer login above

If you still are having issues, please email your APP ID to support@nutritionix.com. Please note, we are not able to locate your account without the app ID, or the email address that is associated with your developer account.

Do you have any recommended barcode scanning libraries for iOS/Android or other best practices for barcode scanning?

There are many open-source barcode scanning plugins available, and we have had good experience with these:

- https://github.com/phonegap/phonegap-plugin-barcodescanner for Android and iOS
- https://github.com/tjwoon/csZBar for iOS only

1) **Calculate the check-digit** of a barcode to confirm the mobile device read the barcode correctly. You can utilize this script as a starting point. Sometimes in low light situations, or curved packages, the mobile phone will misread the barcode. If the incorrect barcode is sent to Nutritionix, it will incorrectly appear that the product does not exist in the database. In the event of a misread barcode event, we recommend building logic into your app to ask the user to scan the product one more time, or manually enter in the barcode number as a backup.

How do I utilize barcode/upc/gtin scanning on the v2 API?

You can pass a barcode to the /search/item endpoint to look up the nutrition facts for a UPC.

How do we calculate the daily value percentage for nutrients?

Our API does not calculate the daily value percentages (we list branded food items as they appear on the package, and convert those to nutrient amounts in the full_nutrients array). For common food results, you will have to have to calculate those percentages manually using the NIH daily value page. Please also see our comprehensive list of nutrients and units in this spreadsheet.

Am I allowed to cache data from the Nutritionix API?

The only pre-approved scenario in which we allow caching API data is for the purpose of recording a historical user food log transaction. For example, if your app allows users to track their food intake, you can cache the nutrients related to that food for the purpose of serving up calculations and history to that specific user.

We do not allow cached data requested by one user to be displayed to another user, or any similar types of caching that would circumvent the need to hit the Nutritionix API for every end user lookup for nutrition data.

If your app requires caching for any reason not listed here, please contact us at support@nutritionix.com to discuss enterprise caching options.

How do I filter search results based on their nutrients?

The /v2/search endpoint supports POST requests for filtering ranges of nutrients. For example, to filter items for matches with “burger” where calories (attr_id 208) is less than 500, you could submit a post query like this:

```
{
  "query": "burger",
  "detailed": true,
  "full_nutrients": {
    "208": {
      "lte": 500
    }  }
}
```

gte = greater than
lte = less than

See this gist for more details about filtering search results.
See this list of all nutrients and nutrient IDs from API (full_nutrients array attr_id definitions)

How do the API call limits work for paid plans?

The API plan limits are listed here: https://www.nutritionix.com/business/api

The Nutritionix API is billed based on the number of end-users that have accessed your application in the last 30 days. We track the number of end-users in your application using the x-remote-user-id header which is required when sending API requests to Nutritionix.

Unless otherwise specified in writing, Nutritionix requires that all API requests be initiated exclusively from an end-user’s action, such as searching for a food or viewing a food in your app interface.

Generally speaking, our system may throttle accounts that exceed 500 API queries for a single end-user in a 24 hour period. Apps that are correctly using the Nutritionix API should not exceed 500 API requests per end-user per 24 hour period.

What are the recommended debounce metrics when hitting the /v2/search/instant endpoint?

There are two important steps to ensure you are hitting the /search/instant endpoint effectively.

1. Requests to the /v2/search/instant endpoint should be debounced for at least 300ms. This will prevent an overload of requests to the endpoint as well as allow the client application some time before rendering the latest results.
2. You should not send requests to the instant endpoint until the user types at least 3 characters. Sending queries to instant at 1-2 characters can cause a substantially higher number of API queries to be made.

What are the food groups in the /natural/nutrient endpoint? [Experimental Feature]

On common foods within the /natural/nutrient endpoint, you may see a “food_group” attribute within the “tags” object. In general, this will indicate how many servings of a particular food group there are in this portion. **Please note, the food groups are an experimental feature**. The list of food groups and IDs:

ID	Food Group
1	Dairy
2	Protein
3	Fruit
4	Vegetable

5	Grain
6	Fat
7	Legume
8	Combination (multiple food groups, not discernable)
9	Not applicable

Standard serving amounts:

- **1 serving fruit** = 1 cup raw or cooked, .5 cup dried
- **1 serving vegetable** = 1 cup raw/cooked vegetables or 2 cups leafy salad greens

How do I filter out duplicate common foods from the /search/instant endpoint?

To provide the best experience to the end user, it is recommended to filter common food results on "tag_id". Common foods with the same tag_id have identical nutrition (ex. "Blackberry" and "Blackberries"). For best results, filter the results to show only one food per tag_id. Generally, the first one will be the most relevant based on the user search query.

How do I submit foods to the Nutritionix database?

To submit a grocery food that is not found in the database please send the following instructions to the end user.

To submit a grocery product to the Nutritionix database, please email the below three photos to support@nutritionix.com:

Photo 1: Photo of the products barcode (that would get scanned at the checkout counter at grocery store).
Photo 2: Photo of the front of the package (so we can determine the name of product)
Photo 3: Photo of the nutrition label.

To submit a restaurant item to be added or modified, please send us the name of the menu item, and a link to the restaurant's nutrition information page on the restaurant website. This can be submitted to support@nutritionix.com.

Please note, that a clear version of all three of these photos will be needed in order to get the product added to our database. Generally, it will take 7-10 days to get the product added after submitting the photos. Thank you!

How do you toggle between restaurant and grocery foods in branded results of the /search/instant endpoint?

To filter the branded results section to only show restaurant or only grocery foods, set the "branded_type" request attribute to 1 (restaurant) or 2 (grocery). If no value is specified, both restaurant and grocery foods will appear in the response. Example GET request with brand_type parameter can be found here.

How do I filter branded foods out of the common food results on the search endpoint?

To filter branded foods out of the common food results on the search endpoint, set the “common_grocery” and “common_restaurant” request attributes to False. This will remove branded (grocery and restaurant) search results from the response. If you want to hide restaurant *or* grocery results from the search, set only the respective request attribute to False.

How do I get a list of the restaurant brands available in the API?

You can use this endpoint (production-ready) to get a list of available restaurant brands with the brand ID: https://d1gvlspmcma3iu.cloudfront.net/restaurants-3d-party.json.gz

How do I retrieve “sub recipe” information for common foods?

The Natural endpoint (v2/natural/nutrients) accepts an optional boolean parameter, “include_subrecipe”, that should be set to True. With this attribute enabled, sub-recipe information will be returned in the response. Further detail can be found here.

Example API v2 Requests

- Making a standard natural language request (for food logging)
- Building a Recipe

API v2 Errors

400 - Validation Error, Invalid input parameters, Invalid request
401 - Unauthorized, Invalid auth keys, Usage limits exceeded, Missing tokens
403 - Forbidden, Disallowed entity
404 - Resource not found
409 - Resource conflict, Resource already exists
500 - Base error, internal server error, request failed.

Change Log

- June 2023
  - V1 API permanently shutdown
- March 2018
  - Added localization support (Enterprise customers only, not available on Starter/Pro plans)
- February 2018
  - Improved instructions on how to make nutrient data lookup once a user selects their food from the /v2/search/instant endpoint results
- November 2017
  - Added filtering based on nutrient ranges to search. See here.
- August 14, 2017
  - Added docs for restaurant location endpoint
- May 12, 2017
  - Added Oauth user flow for enterprise clients.
- March 21, 2017
  - Added and clarified Nutritionix API attribution requirements.

- March 7, 2017
    - Added documentation for Track app end-user endpoints.
- January 5, 2017
    - Added documentation for the /search/item endpoint.
    - Added more resources to the FAQ section
- December 13, 2016
    - Added natural exercise endpoint documentation
- November 1, 2016
    - Added branded food search (restaurant and grocery) via the /v2/search/instant endpoint
- September 8, 2016
    - API endpoint URLs now refer to trackapi.nutritionix.com/**v2** instead of **/v1**
- August 1, 2016
    - Added images to nutrient endpoint response. See here.

Contact Us

Please note: Email and phone support are **only available on paid API plans**.

For clients on paid plans, please email us at support@nutritionix.com

To purchase a paid API plan, visit:
https://www.nutritionix.com/business/api