

UPC/EAN Product Code API

Implementation Guide

Revision history:

May 21, 2013	V 1.0	Initial publication
Aug 15, 2013	V 1.1	Added fields for GCP download, Thumbnail retrieval and categories
Sep 23, 2013	V 2.0	Major software release: Added export in JSON format, modified parameter names to be consistent across protocols, added structure to GCP and manufacturer sections in the XML, added requirement for language parameter to deletion request.
Oct 3, 2013	V 2.1	Added http status code documentation, Python code
Jan 8, 2014	V 2.1a	Corrected a typo in the name of the downloadable code library, added return codes 1 – 7 to the data request transaction
Jun 21, 2014	V 2.2a	Clarified the handling of HTTP status codes for JSON users
June 25, 2015	V 2.3	Documented the addition of the formatted nutrition data
January 6, 2016	V 2.3a	Corrections to the PHP code snippet, courtesy of Mark Hahs
June 15, 2020	V 3.0	Implementation of the V3 endpoint that allows the retrieval of price data for an item and which also allows the user to request system 4 (private) UPC numbers in the output.

CONTENTS

Digit-Eyes UPC/EAN Database API	4
Overview	4
JSON Single-Transaction API Functions.....	5
UPC/EAN Data Request	5
UPC/EAN Data Request Response.....	7
UPC/EAN Deletion/Correction Request	17
UPC/EAN Deletion/Correction Request Response.....	18
UPC/EAN Upsert Request.....	19
UPC/EAN Upsert Request Response	21
XML (SOAP) Single-Transaction API Functions.....	22
UPC/EAN Data Request	22
UPC/EAN Data Request Response.....	24
UPC/EAN Correction/Deletion Request	28
UPC/EAN Deletion/Correction Request Response.....	29
UPC/EAN Upsert Request.....	30
UPC/EAN Upsert Request Response	32
CSV Multiple-Transaction API Functions.....	33
UPC/EAN Data Request	33
UPC/EAN Data Request Response.....	36
UPC/EAN Upsert Request.....	37
UPC/EAN Upsert Request Response	39
Delayed Retrieval.....	40
Appendix A: A Few Useful Facts about UPC / EAN codes.....	41
Appendix B: UPC/EAN Country Codes	44
Appendix C: Error Messages	46
Data Retrieval Operation.....	46
Delete Operation	47

Upsert Operation.....	48
Delayed Retrieval Operation	48
Appendix D: Signing the request.....	49
Objective C Example	49
C# Example	49
PHP Example.....	49
perl Example	50
python Example.....	50
VBScript Example	50
Appendix E: Using and Formatting Nutrition Data	51

OVERVIEW

The Digit-Eyes UPC / EAN database contains over 250,000,000 items identified by UPC, GTIN, EAN, APN and JPN codes. It grows continually as a result of search activity by our customers. Listings are provided in 12 languages.

The API allows appropriately licensed, identified and authenticated processes to:

- Retrieve, update and delete individual records with our JSON or SOAP XML interface;
- Retrieve record groups using data in CSV (excel) format;
- Upload record groups using data in CSV (excel) format;

There are three versions of the JSON/XML API:

- V1 did not produce JSON, but the V2 endpoint provides XML output that is backward-compatible to the V1 specifications
- The V2 endpoint returns a single record per UPC number in either XML or JSON;
- The V3 endpoint produces either XML or JSON and in addition to the V2 data may, additionally:
 - If requested, provide price data
 - If requested, provide system 4 numbers (and, hence, multiple records per UPC number).

Data available about a code:

- always includes the product title and size;
- generally includes the address of the owner/issuer of the UPC/EAN code plus at least one reference to a product page where information can be found and the location of an image; and
- often includes ancillary information such as the manufacturer or vendor's romance description of the product, ingredients or nutritional information;
- if requested through the V3 endpoint, it may optionally include pricing information from various sources and data about system 4 (private) UPC numbers. For more about private UPC numbers, see Appendix A.

Revenue from licensing the digit-eyes database is used to provide low-cost digital product identification services for people without vision and who use our [digit-eyes app](#).

JSON SINGLE-TRANSACTION API FUNCTIONS

UPC/EAN DATA REQUEST

The request for information in the JSON format is submitted as a GET operation to the endpoint.

In V2 format:

```
http://digit-eyes.com/gtin/v2_0/?upc_code=x&app_key=x&signature=x&language=x&field_names=x
```

In V3 format:

```
http://digit-eyes.com/gtin/v3_0/?upc_code=x&app_key=x&signature=x&language=x&field_names=x
```

Price data (an additional charge item) is now available. Please note that if the data is requested, the charge is applied, even if the system is unable to obtain the price data. The value “,prices” should be added to the fieldname list:

```
http://digit-eyes.com/gtin/v3_0/?upc_code=x&app_key=x&signature=x&language=x&field_names=x,  
price
```

Required parameters:

1. **upc_code:** is the UPC, EAN, GTIN, APN or JPN code;
2. **app_key:** is the application identification key of the party making the request;
3. **signature:** created using the upc_code and auth_key. Code examples are in [Appendix D](#);
4. **language:** the ISO 639-1 2- character language code for the preferred language of response ("en," "es," etc.)

Optional parameters:

5. **field_names:** comma-separated list of the fields requested from the database.

Acceptable field requests for both V2 and V3 are:

- **description:** short product name;
- **uom:** unit of measure;
- **usage:** romance description / instructions (if available);
- **brand:** product labeling (if available);
- **language:** 2-character ISO code for the preferred language of listing;
- **website:** the authoritative website for the product (typically the seller or manufacturer website);
- **product_web_page:** address of the individual web page that describes the product, if available;
- **nutrition:** nutrition information in human-readable free text (if available);

JSON Single-transaction API Functions

- **formattedNutrition:** nutrition data in a structured format (US-format labels only at this time and only if the data is available)
- **ingredients:** ingredients of the product (if available);
- **manufacturer:** Manufacturer's name and address elements (if available);
- **gcp_name_address:** GCP owners name and address elements (if available). Please note that if you ask for GCP and the UPC information is not available, the system will return the GCP information and set http code 200. You should check the response_code to determine whether only GCP information was returned.
- **image:** the web address of the manufacturer's image of the product (if available)
- **thumbnail:** a small image of the item, as well as the dimensions
- **categories:** list of tags or descriptive categories associated with this item.

V3 supports the additional field requests (these field requests are accepted, but ignored by the V2 endpoint)

- **prices:** an array of prices, including the currency, vendor business name, vendor address, and date the price data was collected;
- **system4:** value 0 or 1; if 1 and a private/system 4 UPC number is provided. more than a single UPC record may be returned if the information is requested about a System 4 number. In V2, requests for information about System 4 numbers are ignored.

Demonstration: <http://www.digit-eyes.com/gtin/>

Signature calculator: <http://www.digit-eyes.com/gtin/>

Your keys: <https://www.digit-eyes.com/cgi-bin/digiteyes.fcgi?action=myAccount>

UPC/EAN DATA REQUEST RESPONSE

The http status code indicates the success or failure of the request (detail in appendix D). The JSON response for error transactions will include a return code and message that indicates the reason for the failure per Appendix C.

V2 RESULTS

The JSON response transaction for a successful UPC lookup to the V2 endpoint will have http status code 200 and a payload:

```
{
  "brand" : "Grey Poupon",
  "categories" : "Canned Dry & Packaged Foods, Condiment & Sauces, Condiments, Condiments & Salad Dressings, Condiments & Sandwich Toppings, Condiments & Sauces, Condiments Dressings, Condiments Mustard, Condiments Mustards, Condiments Pickles & Relishes, Condiments Sauces & Spices, Condiments Vinegars & Oils, Cooking & Baking, Cooking Sauces, Dressing, Food, Food Beverage, Grocery, Grocery & Gourmet Food, Grocery Food, Grypoupn, Ketchup & Mustard, Ketchups Mustards & Mayo, Kraft Foods, Meal Solutions Grains & Pasta, Mustard & Horseradish, Sandwich Sides, Sauces",
  "description" : "Grey Poupon Mustard Dijon",
  "formattedNutrition" : {
    "Biotin" : {},
    "Calcium" : {
      "dv" : "0%",
      "qty" : null
    },
    "Calories" : {
      "dv" : null,
      "qty" : "5"
    },
    "Calories From Fat" : {},
    "Calories Per Serving" : {},
    "Calories from Fat" : {
      "dv" : null,
      "qty" : "0"
    },
    "Carbohydrates" : {},
    "Chloride" : {},
    "Cholesterol" : {},
    "Chromium" : {},
    "Copper" : {},
    "Dietary Fiber" : {},
    "Fiber" : {},
    "Folic Acid" : {},
    "Insoluble Fiber" : {},
    "Iodine" : {},
    "Iron" : {
      "dv" : "0%",
      "qty" : null
    },
    "Magnesium" : {},
    "Manganese" : {},
    "Molybdenum" : {},
    "Monounsaturated Fat" : {},
    "Niacin" : {},
    "Other Carbohydrates" : {},
    "Pantothenic Acid" : {},
    "Phosphorous" : {},
    "Polyunsaturated Fat" : {},
    "Potassium" : {},
    "Protein" : {
      "dv" : null,
      "qty" : "0g"
    }
  }
}
```

JSON Single-transaction API Functions

```
    },
    "Riboflavin" : {},
    "Saturated Fat" : {},
    "Selenium" : {},
    "Serving Per Container" : {},
    "Serving Size" : {
      "dv" : null,
      "qty" : "1.0 tsp"
    },
    "Servings Per Container" : {
      "dv" : null,
      "qty" : "57"
    },
    "Sodium" : {
      "dv" : "5%",
      "qty" : "120mg"
    },
    "Soluble Fiber" : {},
    "Sugar" : {},
    "Sugar Alcohol" : {},
    "Sugars" : {},
    "Thiamin" : {},
    "Total Calories" : {},
    "Total Calories From Fat" : {},
    "Total Carbohydrate" : {},
    "Total Carbohydrates" : {},
    "Total Fat" : {
      "dv" : "0%",
      "qty" : "0g"
    },
    "Total carbohydrates" : {
      "dv" : "0%",
      "qty" : "0g"
    },
    "Trans Fat" : {},
    "Vitamin A" : {
      "dv" : "0%",
      "qty" : null
    },
    "Vitamin B12" : {},
    "Vitamin B6" : {},
    "Vitamin C" : {
      "dv" : "0%",
      "qty" : null
    },
    "Vitamin D" : {},
    "Vitamin E" : {},
    "Vitamin K" : {},
    "Vitamin K1" : {},
    "Zinc" : {}
  },
  "gcp" : {
    "address" : "3 Lakes Dr",
    "address2" : null,
    "city" : "Northfield",
    "company" : "Kraft Foods Group, Inc.",
    "contact" : null,
    "country" : "US",
    "fax" : null,
    "gcp" : "0054400",
    "gln" : "468000000000",
    "phone" : "(847) 646-2454",
    "postal_code" : "60093",
    "state" : "IL"
  },
  "gcp_gcp" : "54400",
  "image" : "http://images.salsify.com/image/upload/s--V03IoFC1--/h_500,w_500/xvvzvkltnnt3atmspm0h.jpg",
  "ingredients" : "Water, Vinegar, Mustard Seed, Salt, White Wine, Fruit Pectin, Citric Acid, Tartaric Acid, Sugar, Spice.",
  "language" : "en",
```


JSON Single-transaction API Functions

```
"manufacturer" : {
  "address" : null,
  "address2" : null,
  "city" : null,
  "company" : "Kraft Foods North America, Inc",
  "contact" : null,
  "country" : null,
  "phone" : null,
  "postal_code" : null,
  "state" : null
},
"nutrition" : "Serving Size 1.0 tsp Servings Per Container 57 Amount Per Serving Calories 5 Calories
from Fat 0 % Daily Value* Total Fat 0g 0% Sodium 120mg 5% Total Carbohydrate 0g 0% Protein 0g Vitamin A
0% Vitamin C 0% Calcium 0% Iron 0%",
"product_web_page" : "https://www.kraftheinz-foodservice.com/product/1005440000525/GREY-POUPON-
Dijon-Mustard-Squeeze-Bottle-10-oz-Bottle-Pack-of-12?categoryid=0000070248",
"return_code" : "000",
"return_message" : "Success",
"thumbnail" : {},
"uom" : "10.0 ounces Bottle",
"upc_code" : "0054400015034",
"usage" : "Grey Poupon Dijon Mustard. Made with white wine. Directions Refrigerate after opening. Do
not freeze. Shake well before using. Remove inner seal.",
"website" : "www.kraftheinz-foodservice.com/"
}
```

V3 RESULTS

The V3 endpoint allows access to System 4 numbers. System 4 numbers are not unique, which means that multiple products may be available for a single number. The UPC and a count of the products is provided before any product data. The product data follows, structured into a "products" segment. The GCP and manufacturer segments are meaningless in the context of System 4 numbers; instead, the "issuer" segment identifies the business entity that provided the information about the product. Thumbnail data is not available for System 4 products, but if the issuer provided an image, the address of that image is available.

To obtain system 4 results, the value “,prices” should be added to the fieldname list:

```
http://digit-eyes.com/gtin/v3_0/?upc_code=x&app_key=x&signature=x&language=x&field_names=x,
system4

{
  "upc_code" : "416000336108",
  "entries": 3
  "products":
  [
    {
      "brand" : null,
      "categories" : null,
      "description" : "Bgi Tomatogain Plant Food Granules 8-16-16 12ea / 2LB",
      "image" :
      "https://www.centralgarden.com/media/catalog/product/cache/1/image/229x151/9df78eab33525d08d6e5
      fb8d27136e95/1/0/100532892-a.jpg",
      "ingredients" : null,
      "issuer": {
        "address" : "http://www.centralgarden.com/contact-us/",
        "address2" : null,
        "city" : null,
        "company" : "Central Garden Distribution",
        "contact" : null,
        "country" : "US",
        "phone" : "(866) 811-1395",

```

JSON Single-transaction API Functions

```
        "postal_code" : null,
        "state" : null
    },
    "language" : "en",
    "nutrition" : null,
    "product_web_page" : "https://www.centralgarden.com/fertilizer/granules/bgi-tomatogain-tomato-
food-12ea-8-16-16.html",
    "return_code" : "000",
    "return_message" : "Success",
    "uom" : "2 lb",
    "usage" : "TomatoGain granular plant food is easy to use and formulated to provide immediately
available nutrients to promote vigorous growth and high yields while preventing blossom-end rot and
splitting. Works well on all vegetables and some fruits.",
    "website" : "www.centralgarden.com/"
},
{
    "brand" : null,
    "categories" : null,
    "description" : "Eno Leave No Trace Doublenest Hammock | Rei Co-op",
    "image" : "https://www.rei.com/media/product/886801g",
    "ingredients" : null,
    "issuer": {
        "address" : null,
        "address2" : null,
        "city" : "Seattle",
        "company" : "REI",
        "contact" : null,
        "country" : "US",
        "phone" : "(866) 811-1395",
        "postal_code" : null,
        "state" : "WA"
    },
    "language" : "en",
    "nutrition" : null,
    "product_web_page" : "https://www.rei.com/product/886801/eno-leave-no-trace-doublenest-
hammock",
    "return_code" : "000",
    "return_message" : "Success",
    "uom" : "",
    "usage" : null,
    "website" : "www.rei.com/"
},
{
    "brand" : "Locations",
    "categories" : "Orin Swift",
    "description" : "Locations Washington Red (750 ML)",
    "image" : "https://storage-
trianglewine.comcash.com/images/products/thumb_b425dea28f0060688b01570068d9c477.png",
    "ingredients" : null,
    "issuer": {
        "address" : "2613 Chatham Road",
        "address2" : null,
        "city" : "Springfield",
        "company" : "The Corkscrew",
        "contact" : null,
        "country" : "US",
        "phone" : null,
        "postal_code" : "62704",
        "state" : "IL"
    },
    "language" : "en",
    "nutrition" : null,
    "product_web_page" : "https://www.trianglewineco.com/catalog/product/12423/locations-wa-
washington-red.html",
    "return_code" : "000",
    "return_message" : "Success",
    "uom" : "750 ML",
    "usage" : "Wine maker notes A sixth generation vintner and friend of Locations winemaker, Dave
Phinney once said, 'if I were twenty-one, single, and could make wine anywhere in the world, it would
be Washington. ",
    "website" : "www.thecorkscrew.com/"
}
```

JSON Single-transaction API Functions

```
    }  
  ]  
}
```

Additionally, the V3 endpoint may return a price segment if prices are requested. The value “,prices” should be added to the fieldname list:

```
http://digit-eyes.com/gtin/v3_0/?upc_code=x&app_key=x&signature=x&language=x&field_names=x,  
price
```

The results:

```
{  
  "entries" : "1",  
  "products" : [  
    {  
      "brand" : "Grey Poupon",  
      "description" : "Grey Poupon Mustard Dijon",  
      "directions" : "http://www.digit-eyes.com/cgi-  
bin/digiteyes.fcgi?action=quickScan&k=/0Sd9/4LiohG&iP=3&upcCode=0054400015034&l=en",  
      "formattedNutrition" : {  
        "Calcium" : {  
          "dv" : "0%",  
          "qty" : null  
        },  
        "Calories" : {  
          "dv" : null,  
          "qty" : "5"  
        },  
        "Calories from Fat" : {  
          "dv" : null,  
          "qty" : "0"  
        },  
        "Iron" : {  
          "dv" : "0%",  
          "qty" : null  
        },  
        "Protein" : {  
          "dv" : null,  
          "qty" : "0g"  
        },  
        "Serving Size" : {  
          "dv" : null,  
          "qty" : "1.0 tsp"  
        },  
        "Servings Per Container" : {  
          "dv" : null,  
          "qty" : "57"  
        },  
        "Sodium" : {  
          "dv" : "5%",  
          "qty" : "120mg"  
        },  
        "Total Fat" : {  
          "dv" : "0%",  
          "qty" : "0g"  
        },  
        "Total carbohydrates" : {  
          "dv" : "0%",  
          "qty" : "0g"  
        }  
      }  
    }  
  ]  
}
```

JSON Single-transaction API Functions

```
    },
    "Vitamin A" : {
      "dv" : "0%",
      "qty" : null
    },
    "Vitamin C" : {
      "dv" : "0%",
      "qty" : null
    }
  },
  "gcp" : {
    "address" : "3 Lakes Dr",
    "address2" : null,
    "city" : "Northfield",
    "company" : "Kraft Foods Group, Inc.",
    "contact" : null,
    "country" : "US",
    "fax" : null,
    "gcp" : "0054400",
    "gln" : "46800000000",
    "phone" : "(847) 646-2454",
    "postal_code" : "60093",
    "state" : "IL"
  },
  "gcp_gcp" : "54400",
  "image" : "https://i5.walmartimages.com/asr/666478f6-aea4-4641-9f76-
b239eda920cc_2.3d0ae576c416d4422021782ae547f671.jpeg",
  "ingredients" : "Water, Vinegar, Mustard Seed, Salt, White Wine, Fruit Pectin, Citric Acid,
Tartaric Acid, Sugar, Spice.",
  "language" : "en",
  "manufacturer" : {
    "address" : null,
    "address2" : null,
    "city" : null,
    "company" : "Kraft Foods North America, Inc",
    "contact" : null,
    "country" : null,
    "phone" : null,
    "postal_code" : null,
    "state" : null
  },
  "nutrition" : "Serving Size 1.0 tsp Servings Per Container 57 Amount Per Serving Calories 5
Calories from Fat 0 % Daily Value* Total Fat 0g 0% Sodium 120mg 5% Total Carbohydrate 0g 0% Protein 0g
Vitamin A 0% Vitamin C 0% Calcium 0% Iron 0%",
  "prices" : {
    "entries" : 5,
    "offers" : [
      {
        "currencyCode" : "USD",
        "date" : "2020-06-14 09:11:46",
        "price" : "3.6900",
        "url" : "http://grocery.harristeeter.com/pd/Grey-Poupon/Mustard-Dijon/10-
oz/054400015034/"
      },
      {
        "currencyCode" : "AUD",
        "date" : "2020-06-09 15:13:42",
        "price" : "52.1600",
        "url" : "https://www.fishpond.com.au/Kitchen/Grey-Poupon-Dijon-Mustard-10-
oz/0054400015034"
      }
    ]
  }
}
```

JSON Single-transaction API Functions

```
{
  "currencyCode" : "USD",
  "date" : "2020-06-05 09:57:40",
  "price" : "4.5900",
  "url" : "https://shopfoodex.com/grey-poupon-dijon-mustard-squeeze-p-7575.html"
},
{
  "currencyCode" : "USD",
  "date" : "2020-06-02 04:03:13",
  "price" : "29.5900",
  "url" : "https://www.fishpond.com/Kitchen/Grey-Poupon-Dijon-Mustard-10-oz/0054400015034"
},
{
  "currencyCode" : "USD",
  "date" : "2020-04-12 09:08:07",
  "price" : "2.9800",
  "url" : "https://www.walmart.com/ip/Grey-Poupon-Dijon-Mustard-10-0-oz-Squeeze-Bottle/10801675"
}
],
"product_web_page" : "http://directionsforme.org/item/3306684",
"return_code" : 0,
"return_message" : "Success",
"uom" : "10.0 ounces Bottle",
"upc_code" : "0054400015034",
"usage" : "Grey Poupon Dijon Mustard. Made with white wine. Directions Refrigerate after opening. Do not freeze. Shake well before using. Remove inner seal.",
"website" : "directionsforme.org/"
}
],
"return_code" : "000",
"return_message" : "Success",
"upc_code" : "0054400015034"
}
```

Usage, Ingredients, and Nutrition will not contain HTML. A complete list of possible data fields in the "formattedNutrition" element appears in Appendix E of this document. The formatted data includes allergy information and cautions.

Finally, V3 allows retrieval of private ("System4") numbers. Since these numbers are unique only to the business that creates them:

- additional fields are provided that have the name of the business that is using the number.
- It is possible that there will be more than a single product response. When handling these numbers, the designer should plan to present all as choices to the user and allow them to select the choice that best matches their product.

See Appendix A for more information on these numbers.

The following example shows an example where the same private number has been used for (a) plant food, (b) a private-label brand of vodka and (c) a particular vintage of wine. Because our data comes from many sources, note that the same item may be listed under multiple names. In this

JSON Single-transaction API Functions

case, the State of Idaho has a formal naming convention for the “Locations” red wine made by Orin Swift "Wa Washington Red Blend NV" that is not consistent with the descriptive name used for marketing:

```
{
  "entries" : "5",
  "return_code" : "000",
  "return_message" : "Success",
  "upc_code" : "0416000336108"
  "products" :   "products" : [
    {
      "brand" : null,
      "businessName" : "Central Garden Distribution",
      "businessUrl" : "www.centralgarden.com/",
      "description" : "Bgi Tomatogain Plant Food Granules 8-16-16 12ea / 2LB",
      "formattedNutrition" : null,
      "image" :
      "https://www.centralgarden.com/media/catalog/product/cache/1/image/229x151/9df78eab33525d08d6e5fb8d27136e95/1/0/100532892-a.jpg",
      "ingredients" : null,
      "language" : "en",
      "nutrition" : null,
      "prices" : {
        "entries" : 0,
        "offers" : []
      },
      "product_web_page" : "https://www.centralgarden.com/fertilizer/granules/bgi-tomatogain-tomato-food-12ea-8-16-16.html",
      "uom" : null,
      "upc_code" : "0416000336108",
      "usage" : "TomatoGain granular plant food is easy to use and formulated to provide immediately available nutrients to promote vigorous growth and high yields while preventing blossom-end rot and splitting. Works well on all vegetables and some fruits, such as but no",
      "website" : "www.centralgarden.com/"
    },
    {
      "brand" : null,
      "businessName" : "Kosher Wine Cellar",
      "businessUrl" : "www.kosherwinecellar.co.uk/",
      "description" : "Checkmate Vodka",
      "formattedNutrition" : null,
      "image" : null,
      "ingredients" : null,
      "language" : "en",
      "nutrition" : null,
      "prices" : {
        "entries" : 1,
        "offers" : [
          {
            "currencyCode" : "GBP",
            "date" : "2020-04-02 03:37:45",
            "price" : "26.9900",
            "url" : "https://kosherwinecellar.co.uk/vodka/2306-checkmate-vodka-416000336108.html"
          }
        ]
      },
      "product_web_page" : "https://kosherwinecellar.co.uk/vodka/2306-checkmate-vodka-416000336108.html",
      "uom" : null,
    }
  ]
}
```

JSON Single-transaction API Functions

```
    "upc_code" : "0416000336108",
    "usage" : "Made from super premium distilled vodka with entirely natural ingredients.
Checkmate offers the true taste of smooth vodka in every sip. So pull out a glass and enjoy!",
    "website" : "kosherwinecellar.co.uk/"
  },
  {
    "brand" : null,
    "businessName" : "Triangle Wine Company",
    "businessUrl" : "www.trianglewineco.com/",
    "description" : "Locations Wa Washington Red (750ML)",
    "formattedNutrition" : null,
    "image" : "https://storage-
trianglewine.comcash.com/images/products/thumb_b425dea28f0060688b01570068d9c477.png",
    "ingredients" : null,
    "language" : "en",
    "nutrition" : null,
    "prices" : {
      "entries" : 0,
      "offers" : []
    },
    "product_web_page" : "https://www.trianglewineco.com/catalog/product/12423/locations-wa-
washington-red.html",
    "uom" : "750 ML",
    "upc_code" : "0416000336108",
    "usage" : "",
    "website" : "www.trianglewineco.com/"
  },
  {
    "brand" : null,
    "businessName" : "theoriginalwineclub.com",
    "businessUrl" : "theoriginalwineclub.com/",
    "description" : "Orin Swift Locations Wine Wa 4 Red",
    "formattedNutrition" : null,
    "image" :
"https://theoriginalwineclub.com/media/catalog/product/cache/1/image/9df78eab33525d08d6e5fb8d27136e95/
w/i/wine_109296.png",
    "ingredients" : null,
    "language" : "en",
    "nutrition" : null,
    "prices" : {
      "entries" : 1,
      "offers" : [
        {
          "currencyCode" : "USD",
          "date" : "2020-05-31 09:00:12",
          "price" : "16.9800",
          "url" : "https://theoriginalwineclub.com/orin-swift-locations-wine-wa-4-red.html"
        }
      ]
    },
    "product_web_page" : "https://theoriginalwineclub.com/orin-swift-locations-wine-wa-4-
red.html",
    "uom" : "750 ML",
    "upc_code" : "0416000336108",
    "usage" : "",
    "website" : "theoriginalwineclub.com/"
  },
  {
    "brand" : "14. 49",
    "businessName" : "Idaho ABC",
    "businessUrl" : "post.idaho.gov/",
```

JSON Single-transaction API Functions

```
"description" : "Wa Washington Red Blend NV",
"formattedNutrition" : null,
"image" : null,
"ingredients" : null,
"language" : "en",
"nutrition" : null,
"prices" : {
  "entries" : 3,
  "offers" : [
    {
      "currencyCode" : "USD",
      "date" : "2020-05-09 02:49:51",
      "price" : "167.9201",
      "url" : "https://post.idaho.gov/AbcReporting/pricePosting/search?page=1225&size=50"
    },
    {
      "currencyCode" : "USD",
      "date" : "2020-05-09 02:49:51",
      "price" : "14.4900",
      "url" : "https://post.idaho.gov/AbcReporting/pricePosting/search?page=1225&size=50"
    },
    {
      "currencyCode" : "USD",
      "date" : "2020-05-09 02:49:42",
      "price" : "12.5900",
      "url" : "https://post.idaho.gov/AbcReporting/pricePosting/search?page=1224&size=50"
    }
  ]
},
"product_web_page" :
"https://post.idaho.gov/AbcReporting/pricePosting/search?page=1225&size=50",
"uom" : null,
"upc_code" : "0416000336108",
"usage" : "",
"website" : "post.idaho.gov/"
}
]
```


UPC/EAN DELETION/CORRECTION REQUEST

The JSON request for deletion is submitted as a DELETE operation to the endpoint:

```
http://digit-eyes.com/gtin/v2_0/?upc_code=x&app_key=x&signature=x&language=x[&reason=x]
```

This transaction does not actually delete the code but flags it for the attention of the data curator. After flagging, the listing becomes invisible to other users until either confirmed or denied. If the code cannot be confirmed as accurate, the curator will delete it.

Required parameters:

1. **upc_code**: is the UPC, EAN, GTIN, APN or JPN code for which deletion is requested;
2. **app_key**: is the application identification key of the person or entity making the request (this is issued with the API key).
3. **signature**: created using the upc_code and auth_key. Code examples are in [Appendix D](#);
4. **language**: the two-character ISO 6389-1 language code for the preferred language of response ("en," "es," etc.)

Optional (but recommended) Parameters:

5. **reason**: up to 512 characters of human-readable text explaining the cause of the request. This is an optional field, but failure to provide it may cause the data curator to deny the request or to send an email inquiring as to the reason if the code appears to be valid.

Demonstration: <http://www.digit-eyes.com/gtin/>

Signature calculator: <http://www.digit-eyes.com/gtin/>

Your keys: <https://www.digit-eyes.com/cgi-bin/digiteyes.fcgi?action=myAccount>

Note to developers: Before activating your account for production, deletion requests will be parsed and returned with appropriate response codes, but the deletion will not be performed.

JSON Single-transaction API Functions

UPC/EAN DELETION/CORRECTION REQUEST RESPONSE

The successful JSON-format UPC deletion request will have http status code 204 and information formatted as shown:

```
{  
  "upc_code": "37600138727",  
}
```

The JSON response for error transactions will include a return code and message that indicates the reason for the failure per Appendix C.

UPC/EAN UPSERT REQUEST

The request for information upsert in JSON format is submitted as a POST operation to the endpoint:

```
http://digit-eyes.com/gtin/v2_0/  
?upc_code=n&app_key=x&signature=x&language=x&description=x[&optional parameters]
```

Required parameters:

1. **upc_code:** is the UPC, EAN, GTIN, APN or JPN code for which information is desired;
2. **app_key:** is the application identification key of the party making the request;
3. **signature:** created using the upc_code and auth_key. Code examples are in [Appendix D](#);
4. **language:** the ISO 639-1 2 character language code for the preferred language of response ("en", "es", etc.)
5. **description:** up to 255 characters of short descriptive text.

Optional parameters:

6. **uom:** up to 128 characters that describe the item unit of measure or size such as '24 ounces', '12 pieces', or even just "1";
7. **website:** the website where more information can be found about this item (512 characters; the root of the product web page will be used if that is supplied and this field is empty.);
8. **product_web_page:** web page where more information can be found about this item (512 characters);
9. **image:** location of an image of the item (512 characters);
10. **usage:** romance description of the item, usage instructions if available (no length limit);
11. **ingredients:** constituent elements of the item (no length limit);
12. **nutrition:** if nutritional data if appropriate (no length limit);
13. **brand:** product brand;
14. **categories:** comma-separated list of categories or tags that should be associated with the item, listed in increasing order of importance (the tag on the right end is treated as more important than the tag on the left when the system evaluated lexical proximity);
15. **manufacturer_name¹:** manufacturer name;
16. **manufacturer_address:** manufacturer location: street address;
17. **manufacturer_address2:** manufacturer location: suite, additional address;
18. **manufacturer_city:** manufacturer location: city;
19. **manufacturer_state:** manufacturer location: state (2 characters max);
20. **manufacturer_postal_code:** manufacturer location: postal code;

¹ At a minimum, city and country must be provided with the manufacturer name if manufacturer information is supplied

JSON Single-transaction API Functions

21. **manufacturer_country:** manufacturer country;
22. **manufacturer_phone:** manufacturer phone;
23. **manufacturer_contact:** manufacturer email or contact name.

Note to developers: Before activating your account for production, upsert requests will be parsed and returned with appropriate response codes, but will not be executed. After activation, all upsert requests are visible but provisional until reviewed and approved by the data curator.

Signature calculator: <http://www.digit-eyes.com/gtin/>

Your keys: <https://www.digit-eyes.com/cgi-bin/digiteyes.fcgi?action=myAccount>

JSON Single-transaction API Functions

UPC/EAN UPSERT REQUEST RESPONSE

The successful JSON-format upsert request will have http status code 201 and information formatted as shown:

```
{  
  "upc_code": "37600138727",  
}
```

The JSON response for error transactions will include a return code and message that indicates the reason for the failure per Appendix C.

XML (SOAP) SINGLE-TRANSACTION API FUNCTIONS

UPC/EAN DATA REQUEST

The request for information in XML format is submitted to the endpoint

In V2 format:

```
http://digit-eyes.com/gtin/v2_0/xml/?upc_code=x&app_key=x&signature=x&language=x&field_names=x
```

In V3 format:

```
http://digit-eyes.com/gtin/v3\_0/xml/?upc\_code=x&app\_key=x&signature=x&language=x&field\_names=x
```

Where “x” is the value of the upc code, your app_key and the signature you’ve computed from the upc_code and your private key. The parameter field_names should be “all” or the names of the fields you want (see below)

If you want price data (an extra charge item) include the value “,price” in the fieldname list:

```
http://digit-eyes.com/gtin/v3_0/xml/?upc_code=x&app_key=x&signature=x&language=x&field_names=x,price
```

Required parameters:

1. **upc_code:** is the UPC, EAN, GTIN, APN or JPN code for which information is desired;
2. **app_key:** is the application identification key of the party making the request;
3. **signature:** created using the upc_code and auth_key. Code examples are in [Appendix D](#);
4. **language:** the ISO 639-1 2- character language code for the preferred language of response ("en", "es", etc.)

Optional parameters:

5. **field_names:** comma-separated list of the fields requested from the database. Acceptable field names are:

Acceptable field requests for both V2 and V3 are:

- **description:** short product name;
- **uom:** unit of measure;
- **usage:** romance description / instructions (if available);
- **brand:** product labeling (if available);
- **language:** 2-character ISO code for the preferred language of listing;

XML (SOAP) Single-transaction API Functions

- **website:** the authoritative website for the product (typically the seller or manufacturer website);
- **product_web_page:** an individual web page that describes the product, if available;
- **nutrition:** nutrition information for the product (if available);
- **formattedNutrition:** nutrition information divided into discrete elements (US-format nutrition information only and only if available for the item);
- **ingredients:** ingredients of the product (if available);
- **manufacturer:** Manufacturer's name and address elements (if available);
- **gcp_name_address:** GCP owners name and address elements (if available); Please note that if you ask for GCP and the UPC information is not available, the system will return the GCP information and set http code 200. You should check the response_code to determine whether only GCP information was returned.
- **image:** the web address of the manufacturer's image of the product (if available);
- **thumbnail:** a small image of the item, as well as the dimensions;
- **categories:** list of tags or descriptive categories associated with this item.

V3 supports the additional field requests (the V2 endpoint ignores them)

- **prices:** an array of prices, including the currency, vendor business name, vendor address, and date the price data was collected;
- **system4:** value 0 or 1; if 1 and a private/system 4 UPC number is provided. more than one UPC record may be returned if a System 4 number is provided. In V2, data requests for System 4 numbers are ignored.

Please note: for backward compatibility, the parameters used by API versions 1.0 and 1.1 are still supported, but the use is deprecated:

- u: synonym for upc_code ;
- k: synonym for app_key;
- m: synonym for signature;
- l: synonym for language.

Demonstration: <http://www.digit-eyes.com/gtin/>

Signature calculator: <http://www.digit-eyes.com/gtin/>

Your keys: <https://www.digit-eyes.com/cgi-bin/digiteyes.fcgi?action=myAccount>

UPC/EAN DATA REQUEST RESPONSE

V2 RESULTS

The successful XML-format data request to the V2 endpoint will have http status code 200 and information formatted as shown:

```
<?xml version="1.0" standalone="yes"?>
<UPCInformation>
<Version>2.0</Version>
<Header><Sender>
  <Name>Digital Miracles</Name>
</Sender>
</Header>

<Body>
  <ReturnCode>000</ReturnCode>
  <ReturnMessage>Success</ReturnMessage>
  <UPC>0054400015034</UPC>
  <Uom>10.0 ounces Bottle</Uom>
  <Description><![CDATA[Grey Poupon Mustard Dijon]]></Description>
  <Language>en</Language>
  <Website>www.kraftheinz-foodservice.com</Website>
  <ProductWebPage> https://www.kraftheinz-foodservice.com/product/10054400000525/GREY-POUPON-Dijon-
Mustard-Squeeze-Bottle-10-oz-Bottle-Pack-of-12?categoryid=0000070248</ProductWebPage>
  <Usage><![CDATA[Grey Poupon Dijon Mustard. Made with white wine. Directions Refrigerate after
opening. Do not freeze. Shake well before using. Remove inner seal.]]></Usage>
  <Ingredients><![CDATA[Water, Vinegar, Mustard Seed, Salt, White Wine, Fruit Pectin, Citric Acid,
Tartaric Acid, Sugar, Spice.]]></Ingredients>
  <Nutrition><![CDATA[Serving Size 1.0 tsp Servings Per Container 57 Amount Per Serving Calories 5
Calories from Fat 0 % Daily Value* Total Fat 0g 0% Sodium 120mg 5% Total Carbohydrate 0g 0%
Protein 0g Vitamin A 0% Vitamin C 0% Calcium 0% Iron 0%]]></Nutrition>
  <Image> http://images.salsify.com/image/upload/s--V03IoFC1--
/h_500,w_500/xvzvzvkltnt3atmspm0h.jpg</Image>
  <Brand><![CDATA[Grey Poupon]]></Brand>
  <Categories><![CDATA[Amy's Kitchen Inc, Available Items, Canned Dry & Packaged Foods, Condiment &
Sauces, Condiments, Condiments & Salad Dressings, Condiments & Sandwich Toppings, Condiments &
Sauces, Condiments Dressings, Condiments Mustard, Condiments Mustards, Condiments Pickles &
Relishes, Condiments Sauces & Spices, Condiments Vinegars & Oils, Cooking & Baking, Cooking
Sauces, Dressing, Food, Food Beverage, Grocery, Grocery & Gourmet Food, Grocery Food, Grypounp,
Ketchup & Mustard, Ketchups Mustards & Mayo, Kraft Foods, Meal Solutions Grains & Pasta, Mustard &
Horseradish, Sandwich Sides, Sauces]]></Categories>
  <Manufacturer>
    <Company><![CDATA[Kraft Foods North America, Inc]]></Company>
  </Manufacturer>
  <GCP>
    <Company><![CDATA[Kraft Foods Group, Inc.]]></Company>
    <Address>3 Lakes Dr</Address>
    <City>Northfield</City>
    <State>IL</State>
    <Postal_Code>60093</Postal_Code>
    <Country>US</Country>
    <Phone>(847) 646-2454</Phone>
  </GCP>
  <FormattedNutrition>
    <serving_size><![CDATA[1.0 tsp]]></serving_size>
    <servings_per_container><![CDATA[57]]></servings_per_container>
    <calories><![CDATA[5]]></calories>
    <total_fat_qty><![CDATA[0g]]></total_fat_qty>
    <total_fat_dv><![CDATA[0%]]></total_fat_dv>
    <sodium_qty><![CDATA[120mg]]></sodium_qty>
    <sodium_dv><![CDATA[5%]]></sodium_dv>
    <protein_qty><![CDATA[0g]]></protein_qty>
    <vitamin_a_dv><![CDATA[0%]]></vitamin_a_dv>
    <vitamin_c_dv><![CDATA[0%]]></vitamin_c_dv>
    <calcium_dv><![CDATA[0%]]></calcium_dv>
    <iron_dv><![CDATA[0%]]></iron_dv>
  </FormattedNutrition>
</Body>
</UPCInformation>
```


XML (SOAP) Single-transaction API Functions

```
</FormattedNutrition>
</Body>
</UPCInformation>
```

Usage, Ingredients, and Nutrition may contain character data but will not contain HTML. A complete list of possible data fields in the "formattedNutrition" element is included in Appendix E of this document. Formatted nutrition data includes allergy and other cautionary information.

V3 RESULTS

The V3 endpoint allows access to System 4 numbers. System 4 numbers are not unique, which means that multiple products may be available for a single number. The UPC and a count of the products is provided before any product data. The product data follows, structured into a "products" segment. The GCP and manufacturer segments are meaningless in the context of System 4 numbers; instead, the "issuer" segment identifies the business entity that provided the information about the product. Thumbnail data is not available for System 4 products, but if the issuer provided an image, the address of that image is available.

If you want private URLs (system 4 URLs), include the value ",system4" in the fieldname list:

```
http://digit-
eyes.com/gtin/v3_0/xml/?upc_code=x&app_key=x&signature=x&language=x&field_names=x,system4

<?xml version="1.0" standalone="yes"?>
<UPCInformation>
<Version>3.0</Version>
<Header><Sender>
  <Name>Digital Miracles</Name>
</Sender>
</Header>

<Body>
  <ReturnCode>000</ReturnCode>
  <ReturnMessage>Success</ReturnMessage>
  <UPC>4160003361084</UPC>
  <Entries>3</Enries>
  <Products>
    <Product>
      <Uom>2 LB</Uom>
      <Description><![CDATA[Bgi Tomatogain Plant Food Granules 8-16-16 12ea /
2LB]]></Description>
      <Language>en</Language>
      <Website>www.centralgarden.com</Website>
      <ProductWebPage> https://www.centralgarden.com/fertilizer/granules/bgi-tomatogain-
tomato-food-12ea-8-16-16.html </ProductWebPage>
      <Usage><![CDATA[TomatoGain granular plant food is easy to use and formulated to provide
immediately available nutrients to promote vigorous growth and high yields while
preventing blossom-end rot and splitting. Works well on all vegetables and some
fruits]]></Usage>
      <Ingredients><Ingredients>
      <Nutrition>Nutrition>
      <Image>https://www.centralgarden.com/media/catalog/product/cache/1/image/229x151/9df78ea
b33525d08d6e5fb8d27136e95/1/0/100532892-a.jpg </Image>
      <Brand></Brand>
      <Categories></Categories>
      <Issuer>
        <Company><![CDATA[Central Garden Distribution Company>
        <Address>http://www.centralgarden.com/contact-us </Address>
```

XML (SOAP) Single-transaction API Functions

```
        <City></City>
        <State></State>
        <Postal_Code>60093</Postal_Code>
        <Country>US</Country>
        <Phone>"(866) 811-1395</Phone>
    </Issuer>
</Product>
<Product>
    <Uom></Uom>
    <Description><![CDATA[Eno Leave No Trace Doublenest Hammock | Rei Co-op]]></Description>
    <Language>en</Language>
    <Website>www.rei.com</Website>
    <ProductWebPage>https://www.rei.com/product/886801/eno-leave-no-trace-doublenest-
    hammock</ProductWebPage>
    <Usage></Usage>
    <Ingredients><Ingredients>
    <Nutrition>Nutrition>
    <Image>https://www.rei.com/media/product/886801g</Image>
    <Brand>REI</Brand>
    <Categories></Categories>
    <Issuer>
        <Company><![CDATA[Central Garden Distribution Company.]]></Company>
        <Address></Address>
        <City>Seattle</City>
        <State>WA</State>
        <Postal_Code></Postal_Code>
        <Country>US</Country>
        <Phone>866) 811-1395</Phone>
    </Issuer>
</Product>
<Product>
    <Uom>750 ML</Uom>
    <Description><![CDATA[Eno Leave No Trace Doublenest Hammock | Rei Co-op]]></Description>
    <Language>en</Language>
    <Website>www.thecorkscrew.com</Website>
    <ProductWebPage> https://www.trianglewineco.com/catalog/product/12423/locations-wa-
    washington-red.html"</ProductWebPage>
    <Usage></Usage>
    <Ingredients><Ingredients>
    <Nutrition>Nutrition>
    <Image>https://storage-
    trianglewine.comcash.com/images/products/thumb_b425dea28f0060688b01570068d9c477.png
    </Image>
    <Brand>Locations</Brand>
    <Categories>Orin Swift</Categories>
    <Issuer>
        <Company><![CDATA[Central Garden Distribution Company.]]></Company>
        <Address>2613 Chatham Road</Address>
        <City>Springfield</City>
        <State>IL</State>
        <Postal_Code>62704</Postal_Code>
        <Country>US</Country>
        <Phone></Phone>
    </Issuer>
</Product>
</Products>
</Body>
</UPCInformation>
```

Additionally, the V3 endpoint may return a price segment if prices are requested:

If you want price data (an extra charge item) include the value “,price” in the fieldname list:

http://digit-eyes.com/gtin/v3_0/xml/?upc_code=x&app_key=x&signature=x&language=x&field_names=x,price

```
<Offers>
<Offer>
    <CurrencyCode>AUD</CurrencyCode>
```

XML (SOAP) Single-transaction API Functions

```
<Date>2020-06-09 15:13:42</Date>
<Price>52.1600</Price>
<URL>https://www.fishpond.com.au/Kitchen/Grey-Poupon-Dijon-Mustard-10-oz/0054400015034</URL>
</Offer>
<Offer>
  <CurrencyCode>USD</CurrencyCode>
  <Date>2020-06-05 09:57:40</Date>
  <Price>4.5900</Price>
  <URL>https://shopfoodex.com/grey-poupon-dijon-mustard-squeeze-p-7575.html</URL>
</Offer>
<Offer>
  <CurrencyCode>USD</CurrencyCode>
  <Date>2020-06-02 04:03:13</Date>
  <Price>29.5900</Price>
  <URL>https://www.fishpond.com/Kitchen/Grey-Poupon-Dijon-Mustard-10-oz/0054400015034</URL>
</Offer>
<Offer>
  <CurrencyCode>USD</CurrencyCode>
  <Date>2020-04-12 09:08:07</Date>
  <Price>2.9800</Price>
  <URL>https://www.walmart.com/ip/Grey-Poupon-Dijon-Mustard-10-0-oz-Squeeze-Bottle/10801675</URL>
</Offer>
</Offers>
```

UPC/EAN CORRECTION/DELETION REQUEST

The request for deletion to which an XML response is requested includes the "r" parameter:

```
http://digit-eyes.com/gtin/v2_0/xml/?upc_code=n&app_key=n&signature=x&language=x&r=1&reason=x
```

This transaction does not actually delete the code but flags it for the attention of the data curator. After flagging, the listing becomes invisible to other users until either confirmed or denied. If the code cannot be verified, the curator will delete it.

Required parameters:

1. **upc_code:** is the UPC, EAN, GTIN, APN or JPN code for which information is desired;
2. **app_key:** is the application identification key of the party making the request;
3. **signature:** created using the upc_code and auth_key. Code examples are in [Appendix D](#);
4. **language:** the two-character ISO language code for the preferred language of response ("en", "es", etc.)
5. **r:** suspense requested flag: the only acceptable value is '1'.

Optional (but recommended) Parameters:

6. **reason:** up to 512 characters of human-readable text explaining the cause of the request. This is an optional field, but failure to provide it may cause the data curator to deny the request or to send an email inquiring as to the reason if the code appears to be valid.

Signature calculator: <http://www.digit-eyes.com/gtin/>

Your keys: <https://www.digit-eyes.com/cgi-bin/digiteyes.fcgi?action=myAccount>

Note to developers: Before activating your account for production, deletion requests will be parsed and returned with appropriate response codes, but no records will actually be deleted.

UPC/EAN DELETION/CORRECTION REQUEST RESPONSE

The successful XML-format UPC deletion request will have http status code 204 and information formatted as shown:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<UPCDelete>
  <Version>2.0</Version>
  <Header>
    <Sender>
      <Name>Digital Miracles</Name>
    </Sender>
  </Header>
  <Body>
    <ReturnCode>100</ReturnCode>
    <UPC>9780545137645</UPC>
  </Body>
</UPCDelete>
```

UPC/EAN UPSERT REQUEST

An upsert request for which an XML response is required is submitted to the endpoint in the following format:

```
http://digit-eyes.com/gtin/v2_0/xml/  
?upc_code=x&app_key=x&signature=x&language=x&description=x[&optional parameters]
```

Required parameters

1. **upc_code:** is the UPC, EAN, GTIN, APN or JPN code for which information is desired;
2. **app_key:** is the application identification key of the party making the request;
3. **signature:** created using the upc_code and auth_key. Code examples are in [Appendix D](#);
4. **language:** the ISO 639-1 2-character language code identifying the preferred language of response ("en", "es", etc.);
5. **description:** up to 255 characters of short descriptive text for the item.

Optional Parameters

6. **uom:** Unit of measure: up to 128 characters that describe the item size such as '24 ounces', '12 pieces', or even just "1".
7. **website:** the web site where more information can be found about this item (512 characters; the root of the product web page will be used if that is supplied and this field is empty.)
8. **product_web_page:** web page where more information can be found about this item (512 characters);
9. **image:** location of an image of the item (512 characters);
10. **usage:** romance description of the item, usage instructions if available (no length limit);
11. **ingredients:** elements from which the item is made (no length limit);
12. **nutrition:** if nutritional data if appropriate (no length limit);
13. **brand:** product brand
14. **categories:** comma-separated list of categories or tags that should be associated with the item, listed in increasing order of importance (most important tag is on the right end of the string, least important on the left);
15. **manufacturer_name²:** manufacturer name;
16. **manufacturer_address:** manufacturer location: street address;
17. **manufacturer_address2:** manufacturer location: suite, additional address;
18. **manufacturer_city:** manufacturer location: city;
19. **manufacturer_state:** manufacturer location: state (2 characters max);

² At a minimum, city and country must be provided with the manufacturer name when supplying manufacturer information.

XML (SOAP) Single-transaction API Functions

20. **manufacturer_postal_code**: manufacturer location: postal code;
21. **manufacturer_country**: manufacturer country;
22. **manufacturer_phone**: manufacturer phone;
23. **manufacturer_contact**: manufacturer email or contact name.

Note to developers: Before activating your account for production, upsert requests will be parsed and returned with appropriate response codes, but will not be applied to the database. After activation, all upsert requests are provisional until reviewed and approved by the data curator.

Signature calculator: <http://www.digit-eyes.com/gtin/>

Your keys: <https://www.digit-eyes.com/cgi-bin/digiteyes.fcgi?action=myAccount>

UPC/EAN UPSERT REQUEST RESPONSE

The successful XML-format UPC deletion request will have http status code 201 and information formatted as shown:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<UPCUpsert>
  <Version>2.0</Version>
  <Header>
    <Sender>
      <Name>Digital Miracles</Name>
    </Sender>
  </Header>
  <Body>
    <ReturnCode>0</ReturnCode>
    <ReturnCode>Success</ReturnCode>
    <UPC>9780545137645</UPC>
  </Body>
</UPCUpsert>
```


CSV MULTIPLE-TRANSACTION API FUNCTIONS

UPC/EAN DATA REQUEST

This batch feature allows appropriately authorized and authenticated users to submit a list of UPC or EAN numbers for which information is requested. The endpoint is accessed:

```
https://www.digit-eyes.com/cgi-bin/digiteyes/api.cgi?action=Download
```

The response is an HTML page with a link to a .csv file with the results of the transaction.

The numbers are submitted in a file formatted as comma-separated values (.CSV format). CSV is a standard file format that can be produced by Excel and other similar programs. A sample download-format CSV file is here:

<http://www.digit-eyes.com/data/demoDownloadList.csv>

The first row of the .csv file conventionally contains the titles of each column of data in the spreadsheet; the rest of the rows contain the data.

Only one field is required for retrieval:

1. **upc code:** UPC or EAN code. Should be 12 or 13 characters (leading zeros may be omitted). If this field cannot be validated, an error message will appear in the output file.

In the first row of the input file, include the names of the fields that should be downloaded, one field name per column. Spacing and capitalization are ignored within the titles, but spelling is important. Fields names that are not recognized will be ignored and an error message shown after processing. Field names that may be requested:

1. **description:** The title (short description) of the item (up to 255 characters)
2. **uom:** Unit of measure: up to 128 characters that describe the item size or weight. Examples might be '24 ounces', '750 ml', '12 pieces', or even just "1".
3. **brand:** product brand
4. **language:** the language of listing (ISO 639-1 2-character language code)
5. **website:** the web site where more information can be found about this item (512 characters; the root of the product web page will be used if that is supplied and this field is empty.)
6. **product web page:** web page where more information can be found about this item (512 characters);
7. **image:** location of the authoritative image of the item (512 characters);

CSV Multiple-Transaction API

8. **thumbnail URL:** location of a thumbnail image of the item (512 characters). Please note this is an extra charge feature, it is charged only if the requested thumbnail is available and supplied;
9. **thumbnail width:** location of an image of the item (512 characters);
10. **thumbnail height:** location of an image of the item (512 characters);
11. **usage:** romance description of the item, usage instructions if available (no length limit);
12. **ingredients:** elements from which the item is made (no length limit);
13. **nutrition:** nutritional data if available (no length limit);
14. **formattedNutrition:** JSON-formatted nutrition data if available (no length limit; applies to data with US-format nutrition information only.)
15. **categories:** provide a list of product categories and tags that apply to this item in a comma-separated list (no length limit);
16. **manufacturer name:** manufacturer name
17. **manufacturer address:** manufacturer location: street address
18. **manufacturer address 2:** manufacturer location: suite, additional address
19. **manufacturer city:** manufacturer location: city;
20. **manufacturer state:** manufacturer location: state or political entity;
21. **manufacturer postal code:** manufacturer location: postal code
22. **manufacturer country:** manufacturer country
23. **manufacturer phone:** manufacturer phone
24. **manufacturer contact:** manufacturer email or contact name
25. **GCP:** Global Company Prefix
26. **GCP Company:** Company name of the CGP holder
27. **GCP GLN:** Global Location Number
28. **GCP address 1:** GCP holder's location: street address
29. **GCP address 2:** GCP holder's location: suite, additional address
30. **GCP city:** GCP holder's location: city;
31. **GCP state:** GCP holder's location: state or political entity;
32. **GCP postal code:** GCP holder's location: postal code
33. **GCP country:** GCP holder's country
34. **GCP phone:** GCP holder's phone
35. **GCP contact:** GCP holder's email or contact name
36. **GCP phone:** GCP holder's email or contact name
37. **GCP fax:** GCP holder's email or contact name

V3 data (price data and System 4 numbers) are not currently available in the CSV content.

CSV Multiple-Transaction API

Demonstration: <http://www.digit-eyes.com/gtin/>

Signature calculator: <http://www.digit-eyes.com/gtin/>

Your keys: <https://www.digit-eyes.com/cgi-bin/digiteyes.fcgi?action=myAccount>

UPC/EAN DATA REQUEST RESPONSE

The status and results of the processing response appear in an HTML page with an http status code of 200. The page includes a link to a downloadable .csv-format file. Data is also provided to show:

- The on-screen report lists and UPC/EAN codes that could not be processed due to checksum errors or the presence of non-numeric characters and gives processing totals;
- Within the output:
 - codes that could not be processed have an error messages in the "description" field;
 - codes that were not found have an "item not found" error message in the "description" field;
 - input codes appear in column "A"; if the code had to be expanded or corrected prior to lookup, the corrected code is listed in column "B"

UPC/EAN UPSERT REQUEST

This batch feature allows appropriately authorized and authenticated users to submit a file of UPC/EAN data to be added or updated into the database. The data is formatted as a comma-separated value (.csv) file. CSV is a common file format that can be produced by spreadsheet programs such as Excel.

The first row of the .csv file conventionally contains the titles of each column in the spreadsheet; the rest of the rows contain the data.

Required fields:

1. **code:** UPC or EAN code. Should be 12 or 13 characters (leading zeros may be omitted). If this field cannot be validated, the entire row of data will be omitted;
2. **description:** The title of the item (up to 255 characters);
3. **language:** the ISO 639-1 2- character language code. The following languages are currently supported:

a. da: Danish	e. fr: French	i. no: Norwegian
b. de: German	f. fi: Finnish	j. pl: Polish
c. en: English	g. it: Italian	k. pt: Portuguese
d. es: Spanish	h. nl: Dutch	l. sv: Swedish

Optional fields

4. **uom:** Unit of measure: up to 128 characters that describe the item size such as "24 ounces", "750 ml", "12 pieces", or even just "1".
5. **website:** the web site where more information can be found about this item (512 characters; the root of the product web page will be used if that is supplied and this field is empty.)
6. **product_web_page:** web page where more information can be found about this item (512 characters);
7. **image:** location of an image of the item (512 characters);
8. **usage:** romance description of the item, usage instructions if available (no length limit);
9. **ingredients:** elements from which the item is made (no length limit);
10. **nutrition:** if nutritional data if appropriate (no length limit);
11. **brand:** product brand
12. **categories:** comma-separated list of categories or tags that should be associated with the item, listed in increasing order of importance (most important tag is on the right end of the string);
13. **manufacturer_name:** manufacturer name;
14. **manufacturer_address:** manufacturer location: street address;

CSV Multiple-Transaction API

15. **manufacturer_address2**: manufacturer location: suite, additional address;
16. **manufacturer_city**: manufacturer location: city;
17. **manufacturer_state**: manufacturer location: state (2 characters max);
18. **manufacturer_postal_code**: manufacturer location: postal code;
19. **manufacturer_country**: manufacturer country;
20. **manufacturer_phone**: manufacturer phone;
21. **manufacturer_contact**: manufacturer email or contact name.

Note to developers: Before activating your account for production, upsert requests of this type will not be processed. After activation, all upsert requests are provisional until reviewed and approved by the data curator.

A sample CSV upload file is available here: <http://www.digit-eyes.com/data/demoDownloadList.csv>

Signature calculator: <http://www.digit-eyes.com/gtin/>

Your keys: <https://www.digit-eyes.com/cgi-bin/digiteyes.fcgi?action=myAccount>

UPC/EAN UPSERT REQUEST RESPONSE

The result of a batch file upsert operation is an on-screen report that totals the number of records added. Any requests that could not be processed will be shown with an error message explaining the error.

DELAYED RETRIEVAL

Growth of the database is stimulated, in part, by unsuccessful search activity. When an item is not found, particularly if the action is initiated through the API, the hypersearch operation attempts to locate it. The impact of this process is that items that are frequently or recently searched tend to have information added to the database shortly after a failing access.

This batch feature enables customers who are using the Single-Transaction API to receive notification and updates when the indexers locate information for an item number that had been previously requested and returned as "not found".

The request for delayed retrieval in CSV format is submitted as a POST operation to the endpoint:

```
http://digit-eyes.com/gtin/v2\_0/csv/delayedRetrieval  
?app_key=x&auth_key=x&language=x&field_names=x
```


APPENDIX A: A FEW USEFUL FACTS ABOUT UPC / EAN CODES

The Universal Product Code (UPC) is a barcode symbology (i.e., a specific type of barcode) that is widely used in North America, in the United Kingdom, Australia, New Zealand, and other countries for tracking trade items in stores. Its most common form, the UPC-A, consists of 12 digits, uniquely assigned to each trade item. Along with the related EAN barcode, the UPC is the barcode mainly used for scanning of trade items at the point of sale, per GS1 specifications.³ UPC data structures are a component of GTINs (Global Trade Item Numbers). All of these data structures follow the global GS1 specification, which bases on international standards. Some retailers (clothing, furniture) do not use the GS1 System but use other bar code symbologies such as code 128 or code 39 and other article number systems. Other retailers use the EAN/UPC bar code symbology but without using a GTIN (for products brands sold at such retailers only)⁴.

The UPC/EAN numbers are managed worldwide by GS1, an international not-for-profit association with member organizations in most countries⁵. The GS1 association issues the codes in the US.

The UPC/EAN does not identify the maker of the product (as many people believe) but, instead, identifies the party that has trade responsibility for the item. While these are often the same party, this is not necessarily always the case.

UPC/EAN codes are composed of several sections, each of which is useful. Consider this code: 080480280024



- The above is a 13-digit version of a full 14-byte GTIN number 000 80408 28002 4;
- The first three digits tell you which country (in this case, **000** or **00** means the US). A complete list of code ranges and their assignments is in Appendix C.
- The value **80480** indicates who the party is within the address space of the US numbers who have trade responsibility for the item. In this case, it is *Bacardi USA, Inc. 2701 South Le Juene Road, Coral Gables FL, 33134, US*⁶
- The number **28002** is the stocking number for the item.

³ EAN/UPC <http://www.gs1us.org/resources/standards/ean-upc>

⁴ Universal Product Code, Wikipedia http://en.wikipedia.org/wiki/Universal_Product_Code

⁵ Overview of GS1, GS1, <http://www.gs1.org/>

⁶ For a useful source of such information: <http://gepir.gs1.org/v32/xx/gtin.aspx?Lang=en-US>

Appendix A: About UPC/EAN Codes

- The last number, **4**, is a check digit for the other numbers.
- [Looking up the number in the database](#), you'll find that it is Grey Goose Vodka.
- Cheers!

Some UPC's are zero-suppressed. These are eight characters in length (or less) and start with the number 0. They are used in locations where there is not much space and need to be expanded before use. Compressed UPCs are uncommon, so the method of their expansion not discussed here.

Calculating the check digit for a UPC is simple and should be routinely performed as part of scanning to ensure the scan occurred correctly.

- The check digit is calculated by taking the base number without the check digit: 008048028002 and adding a leading zero if it is necessary to pad the number to an odd number of digits. Starting little-endian numbering of the digit positions at 0,
 - the digits in the even-numbered positions are added and the sum multiplied by 3;
 - The digits in the odd-numbered positions are added;
 - The sum of the two operations is divided by 10, and the remainder subtracted from the next highest multiple of 10;
 - If the answer is 10 (because the remainder is zero), the check digit is 0;
 - Given a 14-byte array named @parts with individually-addressable bytes numbered 0-13:
$$\text{checkDigit} = (10 - (((@parts[0] + @parts[2] + @parts[4] + @parts[6] + @parts[8] + @parts[10] + @parts[12]) * 3) + (@parts[1] + @parts[3] + @parts[5] + @parts[7] + @parts[9] + @parts[11] + @parts[13])) \% 10));$$

- In the case above:
 - Start with the base number: 080480280024
 - Remove the existing check digit (4): 08048028002
 - zero pad to the left to create a 13-byte root: 0008048028002
 - Perform the following calculation to determine whether the check digit is correct

0008048028002

The digits in the odd-numbered (green) positions are added together and multiplied by 3; the sum is then added to the other digits. The result is divided by 10 and the remainder is subtracted from 10. The calculation result is truncated to the right-most byte:

```
checkDigit = (10 - (((0+0 + 8 + 2 + 0 + 2) * 3) + (0 + 8 + 4 + 0 + 8 + 0)) \% 10));
checkDigit = (10 - (((12* 3) + 20) \% 10));
checkDigit = (10 - ((36 + 20) \%10));
checkDigit = (10 - (46 \% 10));
```

Appendix A: About UPC/EAN Codes

checkDigit = (10 – 6);

checkDigit = 4;

- the 14-digit GTIN number is 000 80480 28002 4
- the number that will probably appear on this item in the US is 0 80480 28002 4

Some UPC/EAN codes have special meaning and will not be in the database:

- Codes that start with "02" or "2" are for private use. They are locally-assigned codes that the stores used for conveying price and/or weight information. This means anybody (typically a retailer) that needs to assign a UPC to an item that doesn't already have one, can use any number system 4 UPC and be sure that it won't conflict with one assigned formally. So, such a UPC can and does mean something different depending on who you ask, and there's no reason to try to keep track of what products these correspond to. Typical examples would be random-weight packaged meat or produce.
- Codes that start with "4" are also locally-assigned codes (also sometimes called "System 4" or "Private UPC codes") and typically indicate a product that the store has on temporary special, where the store wants to override the public code or which has, for some reason, no public code at all. Examples might be in-store-produced sandwiches, a product manufactured to the business entity's specification or a product where the store does not want to obscure the public number.
- Codes that start with "05" are for coupons. It is not uncommon that the balance of the code indicates the product to which the coupon applies, but this is by no means certain. Most coupons today in the US do not use UPC, but instead use the [GS1 DataBar](#) code.



As a parenthetical note, some UPC/EAN codes are compressed and have to be unpacked before use. These are typically located in very small spaces and may consist of as few as 5 digits.

The digit in the last position indicates where zeros are to be interpolated into the number and how many zeros should be interpolated. The code, once expanded, is the same as any other UPC/EAN code in format.

APPENDIX B: UPC/EAN COUNTRY CODES

Beginning	End	Country Name or Usage	BeginningEnd	Country Name or Usage
000000000000	019000000000	United States	611000000000	611000000000 Morocco
020000000000	029000000000	Restricted distribution	613000000000	613000000000 Algeria
030000000000	039000000000	United States	615000000000	615000000000 Nigeria
040000000000	049000000000	Restricted distribution	616000000000	616000000000 Kenya
050000000000	059000000000	Coupons	618000000000	618000000000 Côte d'Ivoire
060000000000	139000000000	United States	619000000000	619000000000 Tunisia
200000000000	299000000000	Restricted distribution	621000000000	621000000000 Syria
300000000000	379000000000	France	622000000000	622000000000 Egypt
380000000000	380000000000	Bulgaria	624000000000	624000000000 Libya
383000000000	383000000000	Slovenia	625000000000	625000000000 Jordan
385000000000	385000000000	Croatia	626000000000	626000000000 Iran
387000000000	387000000000	Bosnia and Herzegovina	627000000000	627000000000 Kuwait
389000000000	389000000000	Montenegro	628000000000	628000000000 Saudi Arabia
400000000000	440000000000	Germany	629000000000	629000000000 UAE
450000000000	459000000000	Japan	640000000000	649000000000 Finland
460000000000	469000000000	Russian Federation	690000000000	695000000000 China
470000000000	470000000000	Kyrgyzstan	700000000000	709000000000 Norway
471000000000	471000000000	Taiwan	729000000000	729000000000 Israel
474000000000	474000000000	Estonia	730000000000	739000000000 Sweden
475000000000	475000000000	Latvia	740000000000	740000000000 Guatemala
476000000000	476000000000	Azerbaijan	741000000000	741000000000 El Salvador
477000000000	477000000000	Lithuania	742000000000	742000000000 Honduras
478000000000	478000000000	Uzbekistan	743000000000	743000000000 Nicaragua
479000000000	479000000000	Sri Lanka	744000000000	744000000000 Costa Rica
480000000000	480000000000	Philippines	745000000000	745000000000 Panama
481000000000	481000000000	Belarus	746000000000	746000000000 Dominican Republic
482000000000	482000000000	Ukraine	750000000000	750000000000 Mexico
484000000000	484000000000	Moldova	754000000000	755000000000 Canada
485000000000	485000000000	Armenia	759000000000	759000000000 Venezuela
486000000000	486000000000	Georgia	760000000000	769000000000 Switzerland
487000000000	487000000000	Kazakhstan	770000000000	770000000000 Colombia
488000000000	488000000000	Tajikistan	773000000000	773000000000 Uruguay
489000000000	489000000000	Hong Kong	775000000000	775000000000 Peru
490000000000	499000000000	Japan	777000000000	777000000000 Bolivia
500000000000	509000000000	United Kingdom	779000000000	779000000000 Argentina
520000000000	520000000000	Global Office	780000000000	780000000000 Chile
528000000000	528000000000	Lebanon	784000000000	784000000000 Paraguay
529000000000	529000000000	Cyprus	786000000000	786000000000 Ecuador
530000000000	530000000000	Albania	789000000000	790000000000 Brazil
531000000000	531000000000	Macedonia	800000000000	839000000000 Italy
535000000000	535000000000	Malta	840000000000	849000000000 Spain
539000000000	539000000000	Ireland	850000000000	850000000000 Cuba
540000000000	549000000000	Belgium	858000000000	858000000000 Slovakia
560000000000	560000000000	Portugal	859000000000	859000000000 Czech Rep.
569000000000	569000000000	Iceland	860000000000	860000000000 Serbia
570000000000	579000000000	Denmark	865000000000	865000000000 Mongolia
590000000000	590000000000	Poland	867000000000	867000000000 Korea, Peoples Rep
594000000000	594000000000	Romania	868000000000	869000000000 Turkey
599000000000	599000000000	Hungary	870000000000	879000000000 Netherlands
600000000000	601000000000	South Africa	880000000000	880000000000 KOREA, Rep. Of
603000000000	603000000000	Ghana	884000000000	884000000000 Cambodia
608000000000	608000000000	Bahrain	885000000000	885000000000 Thailand
609000000000	609000000000	Mauritius	888000000000	888000000000 Singapore

Appendix B: UPC / EAN Country Codes

Beginning	End	Country Name or Usage
8900000000000	8900000000000	India
8930000000000	8930000000000	Vietnam
8960000000000	8960000000000	Pakistan
8990000000000	8990000000000	Indonesia
9000000000000	9190000000000	Austria
9300000000000	9390000000000	Australia
9400000000000	9490000000000	New Zealand
9500000000000	9500000000000	Global Office
9550000000000	9550000000000	Malaysia
9580000000000	9580000000000	Macau
9770000000000	9770000000000	Serial publications (ISSN)
9780000000000	9790000000000	Bookland (ISBN)
9800000000000	9800000000000	Refund receipts
9810000000000	9820000000000	Common Currency
Coupons		
9900000000000	9990000000000	Coupons

DATA RETRIEVAL OPERATION

DATA RETRIEVAL OPERATION: HTTP STATUS CODES

It is expected that the users of JSON will use the HTTP status codes to determine success or failure. The API uses the following HTTP status codes:

- 200 ok
- 400 UPC/EAN Code invalid
- 401 Signature invalid
- 402 Requires funding
- 404 UPC/EAN not found

DATA RETRIEVAL OPERATION: RETURN CODES

Return codes are supplied to clarify the cause of failures and to provide information about the data returned:

- 0 = Successful transaction (http status code 200). Omitted in JSON response; use HTTP status code to determine success or failure.
- 1 = UPC information not on file, but GCP information is, returning GCP company data (http status code 200)
- 2 = UPC / GCP numbers not on file; information provided from private database; (http status code 200). Private databases are available by special arrangement only; this return code will not be found in most implementations.
- 3 = UPC not found, information provided from GCP + private database; (http status code 200). Private databases are available by special arrangement only; this return code will not be found in most implementations.
- 4 = (Used only in the V2 API) This is a locally-assigned UPC (a 13-digit code starting with "04" or a 12-digit code starting with "4".) Each store uses this number differently, so this code does not have a defined meaning. (http status code is 200)
- 5 = This is a coupon code (a 13-digit code starting with "05" or a 12-digit code starting with "5".) Each manufacturer codes their coupons differently, so there is no name or value defined for this code; (http status code 200). Most coupons today do not use UPC coding, but, instead, use [GS1 DataBar](#) (which permits more explicit information about coupons.)
- 6 = This is a locally-assigned UPC used for items of random weight such as meat, fish or product. (It is a 13-digit code starting with "02" or a 12-digit code starting with "2".) Random

Appendix C: Error Messages

weight UPCs are a way of price-marking an item. This item cost \$price. (http status code 200)
7 = This is a private code (a 13-digit code starting with "000000" or a 12-digit code starting with "000000".) Each store uses these numbers differently, so the code does not have a defined meaning. (http status code 200)

999 = UPC/EAN code not found (http status code 404)

998 = UPC/EAN code not supplied (http status code 400)

996 = Application key not supplied or invalid (http status code 401)

995 = UPC/EAN code not valid (http status code 400)

994 = Unknown operation requested (http status code 400)

992 = Valid signature required (http status code 401)

666 = Account requires funding, data cannot be delivered (http status code 402)

DELETE OPERATION

DELETE OPERATION: HTTP STATUS CODES:

It is expected that the users of JSON will use the HTTP status codes to determine success or failure. The codes that may be returned are:

204 ok

304 Not modified

400 UPC/EAN Code invalid

401 Signature invalid

402 Requires funding

404 UPC/EAN not found

DELETE OPERATION RETURN: CODES

Return Codes are supplied to clarify the cause of failures and to provide information about the data returned:

0 = Successful transaction (http status code 204). Omitted in JSON response.

999 = UPC/EAN code not found (http status code 404)

998 = UPC/EAN code not supplied (http status code 400)

996 = Application key not supplied or invalid (http status code 401)

995 = UPC/EAN code not valid (http status code 400)

992 = Valid signature required (http status code 401)

Appendix C: Error Messages

991 = Deletions not allowed at this time (http status code 304)

990 = Language must be specified (http status code 400)

UPSERT OPERATION

UPSERT OPERATION: HTTP STATUS CODES:

It is expected that the users of JSON will use the HTTP status codes to determine success or failure.

The codes that may be returned are:

201 ok

400 Code not supplied

401 Signature invalid

402 Requires funding

404 UPC/EAN not found

UPSERT OPERATION: RETURN CODES

Return Codes are supplied to clarify the cause of failures and to provide information about the data returned:

0 = Successful transaction (http status code 201). This element is omitted in JSON response; use HTTP status code to determine success or failure.

995 = UPC/EAN code not valid (http status code 400)

996= Application key not supplied or invalid (http status code 401)

992 = Valid signature required (http status code 401)

990 = Language must be specified (http status code 400)

DELAYED RETRIEVAL OPERATION

0 = Successful transaction (http status code 201). This element is omitted in JSON response; use HTTP status code to determine success or failure.

993 = Authorization key required (http status code 401)

666 = Account requires funding, data cannot be delivered (http status code 402)

665 = Request must be processed in batch, please contact support@digit-eyes.com (http status code 413)

APPENDIX D: SIGNING THE REQUEST

Each request for information must be signed. The "signature" is an encrypted version of the UPC or EAN code that is SHA-1 hashed with your authorization key. Examples are shown below, and you can use the demo at <http://digit-eyes.com/gtin/> to validate your signature creation results.

OBJECTIVE C EXAMPLE

To create a signature, we provide a category in objective C that takes a string and a key and returns the encoded value where UPCCode is the UPC or EAN code, and AuthKey is your authorization key:

```
#import "NSString+SHA1.h"
NSString *hashedValue = [UPCCode hashedValue:AuthKey];
```

Download: <http://www.digit-eyes.com/specs/codeLibrary/NSString+SHA1.zip>

C# EXAMPLE

Where UpcCode is the UPC or EAN code, and AuthKey is your authorization key:

```
/// Product code that is to be used in the query is UpcCode
/// Hash that is to be used to create the signature is AuthKey
private string GetDigitEyesVerificationCode(string UpcCode)
{
    var hmac = new HMACSHA1(Encoding.UTF8.GetBytes(AuthKey));
    var m = hmac.ComputeHash(Encoding.UTF8.GetBytes(UpcCode));
    return Convert.ToBase64String(m);
}
```

PHP EXAMPLE

Where \$upc_code is the UPC or EAN code, and \$auth_key is your authorization key:

```
<?php
#
# Digit-Eyes Code to Generate and Output JSON
#
#****Dynamic UPC from form****
//$upc_code = $_GET["upc"];
#****Hard Code Test UPC****
$upc_code = '016000483668';
#****Substitute Your Auth Key****
$auth_key = 'xxxxxxxxxxx';
#****Substitute Your App Key****
$app_key = 'yyyyyyyyyyyy';
#****Generates API Signature****
$signature = base64_encode(hash_hmac('sha1', $upc_code, $auth_key, $raw_output = true));

#****Dynamic UPC Production****
//$url = 'https://digit-eyes.com/gtin/v2_0/?upc_code='.$_GET["upc"].'&app_key='.$app_key
.'&language=en&field_names=all&signature='.$signature .';

#****Hard Code UPC Test****
$url = 'https://digit-eyes.com/gtin/v2_0/?upc_code='.$upc_code .'&app_key='.$app_key
.'&language=en&field_names=all&signature='.$signature .';
```

Appendix D: Signing the Request

```
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_TIMEOUT, 5);
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 5);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$data = curl_exec($ch);
curl_close($ch);
echo $data;
?>
```

PERL EXAMPLE

Where `$upc_code` is the UPC or EAN code and `$auth_key` is your authorization key:

```
use Digest::HMAC_SHA1 qw(hmac_sha1 hmac_sha1_hex);
my $hmac = Digest::HMAC_SHA1->new($auth_key);
$hmac->add($upc_code);
my $signature = $hmac->b64digest;
my $paddingRequired = 4-(length($signature) % 4);
$signature .= substr("====", 0, $paddingRequired) if ($paddingRequired);
```

PYTHON EXAMPLE

Where `upc_code` is the UPC or EAN code, and `auth_key` is your authorization key:

```
import base64
import hashlib
import hmac

def make_auth_token(upc_string, auth_key):
    sha_hash = hmac.new(auth_key, upc_code, hashlib.sha1)
    return base64.b64encode(sha_hash.digest())
```

VBScript EXAMPLE

Where `UpcCode` is the UPC or EAN code, and `AuthKey` is your authorization key:

```
Private Function GetDigitEyesVerificationCode(UpcCode As String) As String
Dim hmac = New HMACSHA1(Encoding.UTF8.GetBytes(AuthKey))
Dim m = hmac.ComputeHash(Encoding.UTF8.GetBytes(UpcCode))
Return Convert.ToBase64String(m)
End Function
```

APPENDIX E: USING AND FORMATTING NUTRITION DATA

Nutrition labeling data varies by country. For products that have US –format nutrition labeling, the information may be obtained in the "formattedNutrition" variable. The US standard is explained in this online article from the FDA: [How to Understand and Use the Nutrition Facts Label](#). Standards for displaying the label are explained in this FDA article: [Guidance Documents: Labeling & Nutrition](#).

The formatted nutrition data is presented in JSON or XML format. The data elements consist of a nutrient name (such as "biotin") and up to two values. The value "qty" contains the quantity of the nutrient present, and the "dv" field contains the percent daily value for the nutrient:

- The % Daily Values are based on the daily value recommendations for key nutrients for a 2,000 calorie daily diet. As a general rule, trace nutrients such as vitamins and minerals have only the %DV field. However, in the case of products designed for infants and very young children, the trace nutrients may have the quantity field populated rather than the %DV field because the number of calories per day may vary widely by the child's age.
- Protein does not have a % DV. Per the FDA, this is because current scientific evidence indicates that protein intake in the US is not a public health concern.
- Trans fats and sugars have no % DV. Per the FDA, there is insufficient evidence to establish any recommended daily minimum of these substances.

JSON Field Name	XML Field Name	Nutrient	Qty	Unit	%DV
biotin	biotin	biotin (trace nutrient)	no*	mcg	yes*
calcium	calcium	calcium (trace nutrient)	no*	mg	yes*
calories	calories	calories	yes	calories	no
calories from fat	calories_from_fat	calories from fat	yes	calories	no
calories per serving	calories_per_serving	calories per serving	yes	calories	no
carbohydrates	carbohydrates	carbohydrates	yes	g	yes*
cholesterol	cholesterol	cholesterol	yes	mg	yes
chloride	chloride	chloride (trace nutrient)	no"	mcg	yes*
chromium]	chromium	chromium (trace nutrient)	no"	mcg	yes*
copper	copper	copper (trace nutrient)	no"	mg	yes
dietary fiber	dietary_fiber	dietary fiber	yes	g	yes
fiber	fiber	fiber (synonym for dietary fiber)	yes	g	yes
folic acid	folic_acid	folic acid (trace nutrient)	no*	mcg	yes*
insoluble fiber	insoluble_fiber	insoluble fiber	yse	g	no
iodine	iodine	iodine (trace nutrient)	no*	mcg	yes*
iron	iron	iron (trace nutrient)	no*	mg	yes*
magnesium	magnesium	magnesium (trace nutrient)	no*	mg	yes*

Appendix E: Using and Formatting Nutrition Data

JSON Field Name	XML Field Name	Nutrient	Qty	Unit	%DV
manganese	manganese	manganese (trace nutrient)	no*	mg	yes*
molybdenum	molybdenum	molybdenum (trace nutrient)	no"	mcg	yes*
monounsaturated fat	monounsaturated_fat	monounsaturated fat	yes	g	no
niacin	niacin	niacin (trace nutrient)	no*	mg	yes*
other carbohydrates	other_carbohydrates	other carbohydrates	yes	g	no
pantothenic acid	pantothenic_acid	pantothenic acid (trace nutrient)	no*	mg	yes*
phosphorus	phosphorus	phosphorus (trace nutrient)	no*	mg	yes*
polyunsaturated fat	polyunsaturated_fat	polyunsaturated fat	yes	g	no
potassium	potassium	potassium	yes	mg	yes
protein	protein	protein	yes	g	np
riboflavin	riboflavin	riboflavin (trace nutrient)	no*	mg	yes*
saturated fat	saturated_fat	saturated fat	yes	g	yes*
selenium	selenium	selenium (trace nutrient)	no*	mcg	yes*
serving size	serving_size	serving size	yes	text	no
servings per container	servings_per_container	servings per container	yes	text	no
sodium	sodium	sodium	yes	mg	yes
soluble fiber	soluble_fiber	soluble fiber	yes	g	no
sugar	sugar	sugar	yes	g	no
sugar alcohol	sugar_alcohol	sugar alcohol	yes	g	no
sugars	sugars	sugars	yes	g	no
thiamin	thiamin	thiamin (trace nutrient)	no*	mg	yes*
total calories	total_calories	total calories (synonym for calories)	yes	calories	no
total calories from fat	total_calories_from_fat	total calories from fat	yes	calories	no
total carbohydrates	total_carbohydrates	total carbohydrates	yes	g	yes
total fat	total_fat	total fat	yes	g	yes
trans fat	trans_fat	trans fat	yes	g	no
vitamin a	vitamin_a	vitamin a (trace nutrient)	no*	IU	yes*
vitamin b12	vitamin_b12	vitamin b12 (trace nutrient)	no*	mcg	yes*
vitamin b6	vitamin_b6	vitamin b6 (trace nutrient)	no*	mg	yes*
vitamin c	vitamin_c	vitamin c (trace nutrient)	no*	mb	yes*
vitamin d	vitamin_d	vitamin d (trace nutrient)	no*	IU	yes*
vitamin e	vitamin_e	vitamin e (trace nutrient) "	no*	IU	yes*
vitamin k	vitamin_k	vitamin k (trace nutrient)	no*	mcg	yes*
vitamin k1	vitamin_k1	vitamin k1 (trace nutrient)	no*	mcg	yes*
zinc	zinc	zinc (trace nutrient)	no*	mg	yes*

Two additional fields are also available in the formattedNutrition segment: Allergy warnings and Additional warnings (allergy_warnings and additional_warnings for the XML version).