Digital Miracles L.L.C.

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	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 at 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

CONTENTS

DIGIT-EYES UPC/EAN DATABASE API

OVERVIEW

The Digit-Eyes UPC / EAN database contains over 45,000,000 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 10 languages, principally for products sold in the Americas, Europe, and Australia.

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 CSV (excel file) interface;
- Upload record groups using CSV (excel file) interface;

Data available about a code:

- always includes the product title and size;
- generally includes the address of the owner 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 nutrition information.

Revenue from licensing our database is used to provide free and 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 JSON format is submitted as a GET operation to the endpoint:

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

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.)

Optional parameters:

5. **field_names:** comma-separated list of the fields requested from the database. Acceptable field names 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:** individual web page that describes the product, if available;
   - **nutrition**: nutrition information in human-readable free text (if available);
   - **formattedNutrition**: nutrition data in 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:** web address of manufacturer's image of the product (if available);
   - **thumbnail:** small image of the item, as well as the dimensions;
   - **categories:** list of tags or descriptive categories associated with this item.

JSON Single-transaction API Functions

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 is used to indicate 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.

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

```
{
"upc_code": "37600138727",
"description": "Spam Ham Classic",
"uom": "12 ounces",
"brand": "Spam",
"language": "en",
"ingredients": "Pork With Ham, Salt, Water, Modified Potato Starch, Sugar, Sodium Nitrite.",
"usage": "To Bake: Place slices on baking pan.   Bake at 425 degrees for 10 minutes.   To Fry:  Fry
slices in skillet until golden brown on both sides. To microwave:  Place slices on microwavable
plate.  Heat on high for 1 and one-half to 2 minutes or until hot.   To broil or Grill:  Place slices
on broiler pan or on a grill.  Broil or grill 2 minutes on each side until golden brown.",
"nutrition": "Nutrition Facts Serving Size 56 g Servings Per Container 6 Amount Per Serving Calories
180 Calories from Fat 140 % Daily Value* Total Fat 16g 25% Saturated Fat 6g 30% Cholesterol 40mg 13%
Sodium 790mg 33% Protein 7g Vitamin A 0% Vitamin C 0% Calcium 0% Iron 2%",
 "formattedNutrition" : {
     "Saturated Fat" : {
        "dv" : "30%",
        "qty" : "6g" },
     "Calcium" : {
        "dv" : "0%",
        "qty" : null },
     "Servings Per Container" : {
        "dv" : null,
        "qty" : "6" },
     "Vitamin A" : {
        "dv" : "0%",
        "qty" : null },
     "Serving Size" : {
         "dv" : null,
         "qty" : "2 oz" },
     "Total Fat" : {
         "dv" : "25%",
         "qty" : "16g" },
     "Iron" : {
         "dv" : "2%",
         "qty" : null },
     "Total carbohydrates" : {
         "dv" : "0%",
         "qty" : "1g" },
     "Trans Fat" : {
         "dv" : null,
         "qty" : "0g" },
     "Calories from Fat" : {
         "dv" : null,
         "qty" : "140" },
     "Sugars" : {
         "dv" : null,
         "qty" : "0g" },
     "Protein" : {
         "dv" : null,
         "qty" : "7g" },
     "Vitamin C" : {
         "dv" : "0%",
         "qty" : null },
     "Sodium" : {
         "dv" : "33%",
         "qty" : "790mg" },
     "Calories" : {
         "dv" : null,
         "qty" : "180" },
     "Cholesterol" : {
         "dv" : "13%",
         "qty" : "40mg" }
    },
"website": "www.rampstop.com",
"product_web_page": "http://www.rampstop.com/product_p/037600138727.htm",
"image": "http://www.wegmans.com/prodimg/727/200/037600138727.jpg",
"thumbnail" : {
     "thumbnail_url": "www.digit-eyes.com/thumbs/0/600/037600138727.jpg",
     "thumbnail_width": "250",
     "thumbnail_height": "187"
},
"manufacturer": {
    "name" : "Hormel",
    "address": "One Hormel Place”,
    "address2" : null,
    "city" : "Austin",
    "state" : "MN",
    "postal_code" : "55912-3680",
    "country" : null,
    "contact" : null,
    "phone": "(507) 437-5129"
},
"gcp" : {
    "company" : "Hormel Foods Corporation",
    "address": "One Hormel Place",
    "address2": null,
    "city": "Austin",
    "state": "MN",
    "country": "US",
    "postal_code": "55912-3680",
    "contact" : null,
    "phone": "(507) 437-5129",
    "fax": null
},
"categories: "Canned & Packaged, Canned Goods, Canned Meat, Galley, Grocery, Meat & Poultry, Meat &
Seafood, Meats "
}
```

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

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 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: Prior to activating your account for production, deletion requests will be parsed and returned with appropriate response codes, but will not be applied to the database.

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 two-character ISO 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:** 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:** 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 (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_name[1]:** 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

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

Note to developers: Prior to 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 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

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:

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

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.)

Optional parameters:

5. **field_names:** comma-separated list of the fields requested from the database. Acceptable field names 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:** 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:** web address of manufacturer's image of the product (if available);
    - **thumbnail:** small image of the item, as well as the dimensions;
    - **categories:** list of tags or descriptive categories associated with this item.

For backwards compatibility, the parameters used by API versions 1.0 and 1.1 are still supported but 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

The successful XML-format data 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>37600138727</UPC>
        <Uom>12 ounces</Uom>
        <Description><![CDATA[Spam Ham Classic]]</Description>
        <Language>en</Language>
        <Website>www.rampstop.com</Website>
        <ProductWebPage>http://www.rampstop.com/product_p/037600138727.htm</ProductWebPage>
        <Usage><![CDATA[To Bake: Place slices on baking pan. Bake at 425 degrees for 10 minutes.   To
        Fry: Fry slices in skillet until golden brown on both sides. To microwave:  Place slices on
        microwavable plate.  Heat on high for 1 and one-half to 2 minutes or until hot. To broil or
        Grill: Place slices on broiler pan or on a grill.  Broil or grill 2 minutes on each side until
        golden brown.]]></Usage>
        <Ingredients><![CDATA[Pork With Ham, Salt, Water, Modified Potato Starch, Sugar, Sodium
        Nitrite.]]></Ingredients>

        <Nutrition><![CDATA[Nutrition Facts Serving Size 56 g Servings Per Container 6 Amount Per
        Serving Calories 180 Calories from Fat 140 % Daily Value* Total Fat 16g 25% Saturated Fat 6g
        30% Cholesterol 40mg 13% Sodium 790mg 33% Protein 7g Vitamin A 0% Vitamin C 0% Calcium 0% Iron
        2%]]></Nutrition>
        <FormattedNutrition>
            <serving_size>2 oz</serving_size>
            <servings_per_container>6</servings_per_container>
            <calories>180</calories>
            <total_fat_qty>16g</total_fat_qty>
            <total_fat_dv>25%</total_fat_dv>
            <saturated_fat_qty>6g</saturated_fat_qty>
            <saturated_fat_dv>30%</saturated_fat_dv>
            <trans_fat_qty>0g</trans_fat_qty>
            <cholesterol_qty>40mg</cholesterol_qty>
            <cholesterol_dv>13%</cholesterol_dv>
            <sodium_qty>790mg</sodium_qty>
            <sodium_dv>33%</sodium_dv>
            <sugars_qty>0g</sugars_qty>
            <protein_qty>7g</protein_qty>
            <vitamin_a_dv>0%</vitamin_a_dv>
            <vitamin_c_dv>0%</vitamin_c_dv>
            <calcium_dv>0%</calcium_dv>
            <iron_dv>2%</iron_dv>
        </FormattedNutrition>
        <Image>http://www.wegmans.com/prodimg/727/200/037600138727.jpg</Image>
        <Thumbnail>
            <ThumbnailUrl>www.digit-eyes.com/thumbs/0/600/037600138727.jpg</ThumbnailUrl>
            <ThumbnailWidth>250</ThumbnailWidth>
            <ThumbnailHeight>187</ThumbnailHeight>
        </Thumbnail>
        <Brand><![CDATA[Spam]]</Brand>
        <GCP>
            <Company><![CDATA[Hormel Foods Corporation]]</Company>
            <Address>One Hormel Place</Address>
             <Address2>Suite 100</Address2>
            <City>Austin</City>
            <State>MN</State>
            <Postal_Code>55912-3680</Postal_Code>
            <Country>US</Country>
            <Phone>(507) 437-5129</Phone>
            <Fax>(507) 437-5000</Fax>
            <Contact>(507) 437-5000</Contact>
        </GCP>
        <Manufacturer>
            <Company><![CDATA[Hormel Foods Corporation]]</Company>
            <Address>One Hormel Place</Address>
             <Address2>Suite 100</Address2>
            <City>Austin</City>
            <State>MN</State>
            <Postal_Code>55912-3680</Postal_Code>
            <Country>US</Country>
            <Phone>(507) 437-5129</Phone>
            <Fax>(507) 437-5000</Fax>
            <Contact>(507) 437-5000</Contact>
        </Manufacturer>
</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 as Appendix E of this document. The formatted nutrition data includes allergy and other cautionary information.

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 confirmed as accurate; 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: Prior to activating your account for production, deletion requests will be parsed and returned with appropriate response codes, but will not be applied to the database.

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 two-character ISO language code for 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:** 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_name[2]:** 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.

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: Prior to 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 batch download feature is accessed through a special endpoint:

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

Response is an HTML pages 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 common 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:** language of listing;
5. **website:** 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);

8. **thumbnail URL:** location of a thumbnail image of the item (512 characters). Please note this is a 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

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

Processing response is provided in HTML with an http status code of 200 and a link to a downloadable .csv-format file:

- 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 n 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:** language of the listing may be any of the following:
    a. da: Danish
    b. de: German
    c. en: English
    d. es: Spanish
    e. fr: French
    f. it: Italian
    g. no: Norwegian
    h. pl: Polish
    i. pt: Portuguese
    j. 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:** 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;
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: Prior to 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 records that could not be processed are 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, particular search attention is paid to attempt to locate it. The impact of this is that items that are frequently or recently searched tend to have information added to the listing any time they are accessed and found to be missing or incomplete.

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
```

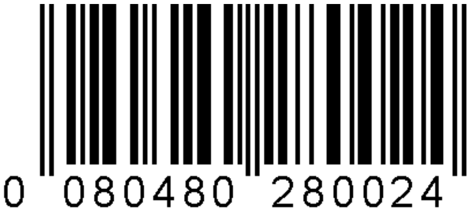
APPENDIX A: A FEW USEFUL FACTS ABOUT UPC / EAN CODES

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 in other countries for tracking trade items in stores. Its most common form, the UPC-A, consists of 12 digits, which are 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 countries[5]. The codes in the US are issued by GS1 US.

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

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



- The above is actually 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 U.S.A., 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

- The last number, **4**, is a check digit for the preceding numbers.
- Looking the entire number up in the database, you'll find that it is Grey Goose Vodka.
- Cheers!

Some UPC's are zero-suppressed. These are 8 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. This is not a topic of interest to most people and we are not going to discuss it 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.

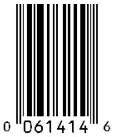
- The check digit is calculated by taking the base number without the check digit: 008048028002 and adding a leading zero if it is necessary to pad the number to an odd number of digits. Starting little-endian numbering of the digit positions at 0,
  - the digits in the even numbered positions are added 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:
    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. This 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));

```
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 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.
- 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
0000000000000	0190000000000	United States
0200000000000	0290000000000	Restricted distribution
0300000000000	0390000000000	United States
0400000000000	0490000000000	Restricted distribution
0500000000000	0590000000000	Coupons
0600000000000	1390000000000	United States
2000000000000	2990000000000	Restricted distribution
3000000000000	3790000000000	France
3800000000000	3800000000000	Bulgaria
3830000000000	3830000000000	Slovenia
3850000000000	3850000000000	Croatia
3870000000000	3870000000000	Bosnia and Herzegovina
3890000000000	3890000000000	Montenegro
4000000000000	4400000000000	Germany
4500000000000	4590000000000	Japan
4600000000000	4690000000000	Russian Federation
4700000000000	4700000000000	Kyrgyzstan
4710000000000	4710000000000	Taiwan
4740000000000	4740000000000	Estonia
4750000000000	4750000000000	Latvia
4760000000000	4760000000000	Azerbaijan
4770000000000	4770000000000	Lithuania
4780000000000	4780000000000	Uzbekistan
4790000000000	4790000000000	Sri Lanka
4800000000000	4800000000000	Philippines
4810000000000	4810000000000	Belarus
4820000000000	4820000000000	Ukraine
4840000000000	4840000000000	Moldova
4850000000000	4850000000000	Armenia
4860000000000	4860000000000	Georgia
4870000000000	4870000000000	Kazakhstan
4880000000000	4880000000000	Tajikistan
4890000000000	4890000000000	Hong Kong
4900000000000	4990000000000	Japan
5000000000000	5090000000000	United Kingdom
5200000000000	5200000000000	Global Office
5280000000000	5280000000000	Lebanon
5290000000000	5290000000000	Cyprus
5300000000000	5300000000000	Albania
5310000000000	5310000000000	Macedonia
5350000000000	5350000000000	Malta
5390000000000	5390000000000	Ireland
5400000000000	5490000000000	Belgium
5600000000000	5600000000000	Portugal
5690000000000	5690000000000	Iceland
5700000000000	5790000000000	Denmark
5900000000000	5900000000000	Poland
5940000000000	5940000000000	Romania
5990000000000	5990000000000	Hungary
6000000000000	6010000000000	South Africa
6030000000000	6030000000000	Ghana
6080000000000	6080000000000	Bahrain
6090000000000	6090000000000	Mauritius

BeginningEnd		Country Name or Usage
6110000000000	6110000000000	Morocco
6130000000000	6130000000000	Algeria
6150000000000	6150000000000	Nigeria
6160000000000	6160000000000	Kenya
6180000000000	6180000000000	Côte D'Ivorie
6190000000000	6190000000000	Tunisia
6210000000000	6210000000000	Syria
6220000000000	6220000000000	Egypt
6240000000000	6240000000000	Libya
6250000000000	6250000000000	Jordan
6260000000000	6260000000000	Iran
6270000000000	6270000000000	Kuwait
6280000000000	6280000000000	Saudi Arabia
6290000000000	6290000000000	U.A.E.
6400000000000	6490000000000	Finland
6900000000000	6950000000000	China
7000000000000	7090000000000	Norway
7290000000000	7290000000000	Israel
7300000000000	7390000000000	Sweden
7400000000000	7400000000000	Guatemala
7410000000000	7410000000000	El Salvador
7420000000000	7420000000000	Honduras
7430000000000	7430000000000	Nicaragua
7440000000000	7440000000000	Costa Rica
7450000000000	7450000000000	Panama
7460000000000	7460000000000	Dominican Republic
7500000000000	7500000000000	Mexico
7540000000000	7550000000000	Canada
7590000000000	7590000000000	Venezuela
7600000000000	7690000000000	Switzerland
7700000000000	7700000000000	Colombia
7730000000000	7730000000000	Uruguay
7750000000000	7750000000000	Peru
7770000000000	7770000000000	Bolivia
7790000000000	7790000000000	Argentina
7800000000000	7800000000000	Chile
7840000000000	7840000000000	Paraguay
7860000000000	7860000000000	Ecuador
7890000000000	7900000000000	Brazil
8000000000000	8390000000000	Italy
8400000000000	8490000000000	Spain
8500000000000	8500000000000	Cuba
8580000000000	8580000000000	Slovakia
8590000000000	8590000000000	Czech Rep.
8600000000000	8600000000000	Serbia
8650000000000	8650000000000	Mongolia
8670000000000	8670000000000	Korea, Peoples Rep
8680000000000	8690000000000	Turkey
8700000000000	8790000000000	Netherlands
8800000000000	8800000000000	KOREA, Rep. Of
8840000000000	8840000000000	Cambodia
8850000000000	8850000000000	Thailand
8880000000000	8880000000000	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

APPENDIX C: ERROR MESSAGES

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 HTTP status codes that may be returned are:

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 = 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

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)

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 siguature 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 = 'xxxxxxxxxx';
#****Substitute Your App Key****
$app_key = 'yyyyyyyyyyy’;
#****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 .'';
```

```
$ch = curl_init($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 JSON format, being an array, is unordered; the XML version is ordered. 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. This is because there is, 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
choloride	choloride	choloride (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*

JSON Field Name	XML Field Name	Nutrient	Qty	Unit	%DV
magnesium	magnesium	magnesium (trace nutrient)	no*	mg	yes*
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).