Icecat Open Catalog Interface (OCI):
Open Icecat and Full Icecat JSON (REST) Service

Revision date: April 1, 2016
Version 2.2

IMPORTANT NOTES

1. The current use of JSON is limited to the retrieval of individual product data-sheets. For the use of indexes and repositories, and a more in-depth insight in the Icecat data model, we refer to the XML documentation.
2. In case of Open Icecat, please check the Open Icecat sections.
3. In case of Full Icecat data access or retrieval of private brand assets, authorization may be required, and a proper server certificate is to be used.
4. The links and files in the download directories are to be downloaded with scripts (from server to server). If you try to download them via your browser, you may sometimes experience memory problems. This does not mean that on our side the data is not accessible or correctly available. In case you experience any problems, contact us via the contact form.

Contents

1. Subject of the document

In this document we explain the difference between Open and Full Icecat subscription, list the advantages of using the Full Icecat and describe how to obtain the Full Icecat subscription and receiving the product content.

1.1. Our Mission

It is our mission to provide e-commerce partners with high-quality product information in the form of open content. More and more manufacturers work together with Icecat to take care of their product content to distribute their product content for free via Open Icecat. Here, you can always find the latest overview of sponsoring (free) brands in Open Icecat: http://www.icecat.biz/en/menu/partners/index.html.

However attractive the free Open Icecat database is, the Full Icecat database still contains a lot of brands: 13,000+ (see http://www.Icecat.biz/en/menu/vendors/index.htm). If you are interested in subscribing to the Full Icecat database, please contact us via the website. Is there a catch with the free product content? No, because in that case the manufacturers take care of the product content and related costs. There is one thing that we ask the users of free Open Icecat, that is: to inform their business partners (distributors, manufacturers, retailers) via a newsletter and/or on the website that they are using Open Icecat, and that they would like their partners to join as well. The more partners join, the better and more complete Open Icecat will become.

Please note that some brands, such as Philips, make certain rich media assets only available for their Authorized Resellers. Please contact us in case you see on the Icecat website more assets than you can access.

1.2. International Standards Supported

- JSON/JavaScript Object Notation conforming to ECMA-404
- The encoding used in our interfaces is UTF8
- HTTP for file transfer
- Gzip, Bzip2 and Zip for file compression
- UNSPSC (see http://www.unspsc.org) for categorization
- GTIN (EAN/UPC) for logistical product codes (when provided by manufacturers)
- ISO 639-1 two-character code for languages
- ISO 3166-1 two character code for countries
- All common graphical formats for images (JPEG, GIF, PNG) and video (MP4).
- Diverse product-specific standards, as mentioned in the feature descriptions.

1.3. Open Icecat Fair Use Policy

The user of Open Icecat product data (in a free Open Icecat subscription or as part of the Full Icecat data) is required to respect the Open Icecat Fair Use Policy, which can be found in the forum: http://Icecat.biz/forum.cgi?tmpl=view_message.html;message_id=3004;thread_id=17

Explicitly relevant for website development is:

- To mention explicitly the “(c) Icecat” copyright notice (or “Source: Icecat.biz”) related to the product specs section, with a link to the Icecat.biz web site;

- To include explicitly the Icecat "AS IS" disclaimer, which can be found here: http://www.icecat.biz/en/menu/disclaimer/index.htm (it is available in several languages).

1.4. Coverage analysis

To decide to make use of Full Icecat or Open Icecat, it may be important to analyze the coverage of your portfolio or the portfolio of your supplier. You can always ask us to provide you with an overview of the coverage of (stock) portfolio of your suppliers (i.e., distributors) at any given moment. We strive to cover at least 90% of the stock portfolio (stock > 1) of mainstream distributors and work together with partners to daily monitor this coverage.

If you want to analyze coverage, you should:
1) Register and upload your catalog to Icecat via the Login. In the match analysis, you get the match percentages back for Full Icecat and Open Icecat, and you can get a list of the unrecognized products. An advantage is that you can also optionally display your offers using this method on the Icecat.biz price comparison sites. **Note!:** if you apply this method, Icecat will automatically provide you with your private Icecat index.
2) Use the following matching file, which is a plain dump of everything in our catalog: https://data.Icecat.biz/prodid/prodid_d.txt.gz (a large file with all known products in the Icecat database, described and standardized or not in gzipped archive).

Note!: Market presence=Y in the matching file indicates that we notice that the product is currently present at least in one distributor’s catalog. Quality=Icecat means that Icecat editors have standardized the data-sheet. Quality=SUPPLIER means that the data-sheet is auto-imported from a manufacturer’s CMS, and may be queued for standardization. Quality=NOEDITOR means that our editors did not create a data-sheet yet, nor did we auto-import manufacturer data; so, it is only the raw data as we received it from channel partners.

TIP: discuss with the Icecat team a service level for your portfolio.

1.5. Advantages of upgrading to Full Icecat

What are the main advantages of Full Icecat over Open Icecat? All product data of all 13,000+ brands (see http://www.icecat.biz/en/menu/vendors/index.htm; if you see the brand logo the brand's product data is also available in the free Open Icecat). Coverage guarantees when we monitor your stock portfolio Support for producing missing data-sheets, categories and brands on request. If you want to upgrade to Full Icecat, you can contact us via the contact form, or click on the "request Full Icecat access" link after the User Login (http://www.icecat.biz/en/menu/openIcecat/index.htm), or fill in and fax the Full Icecat registration form that you can find as an attachment. You will also find in the attached form (pdf) with Full Icecat tariffs.

2. Working with Icecat Rest Service

2.1. Access

To obtain access to the repository, you will need login credentials, provided by Icecat. If you don't have these details, please register online (click on Login) for free Open Icecat.

Note!: During the registration you need to choose from the URL and XML version. If you want to make use of data (JSON), make sure that you choose the Open Icecat XML subscription. You can always change this afterwards, or create a second subscription for the URL (links to hosted data-sheets) method.

For Full Icecat access a separate contract is needed. You can request this contract form here. Make sure that you have provided us with the right IP addresses of your content servers, as the access is secured. You can test the login/password combination by entering it manually in the BASIC HTTP authorization pop-up screen, which appears automatically when the directory URL is entered. In this case, we need to have added the IP address of your workplace as well to your profile. If you don't know this IP address, just type in your browsers http://www.whatismyip.com/ or http://www.getip.com/ and fill it in the IP addresses field after Login via the Icecat website (multiple IPs need to be entered space separated). Of course, for professional use the server access needs to be set up. In case of little experience in this field, we advise you to make use of the URL versions (links to pictures and datasheets) or ask us for assistance.

TIP: for many mainstream solutions standard interfaces are available, including Magento, Prestashop, osCommerce (and siblings) and Batavi.

2.2. General Data Structure of a JSON Product Data-sheet

The JSON/REST service will make it possible for the user to retrieve the needed information for a specific product from an Icecat repository. For the real-time use if product data is in JSON format.

The main principles are the following:

- The requests are based on HTTP/HTTPS protocol.
- The GET method is to be used.
- Each URI identifies one specific resource.
- The Internet media type for the data is JSON (Content-type application/json).

General product data to be presented:

Product title*	Title
Brand*	Brand
Product name*	ProductName
Product code*	BrandPartCode
Category*	CategoryID; NameID; Value; Language
Virtual category	VirtualCategoryID; UNCATID; NameID; Value; Language
Product family	ProductFamilyID; NameID; Value; Language
Product series	SeriesID; NameID; Value; Language
EAN/UPC code	Global Trade Item Number (GTIN)

IMPORTANT!: * means that these parameters are mandatory to be filled in.

2.3. Data Retrieval Algorithm

According to the REST service architecture properties, all data (resources) will be retrieved by URLs using the GET method.

1. Check if the Icecat user (UserName) is present

· UserName (Icecat user) is absent in DB – ERROR “The Icecat user is unknown”

· UserName (Icecat user) is present in DB – you can proceed with further actions

2. Product identification

· Product is absent – Error “The requested product is not present in the Icecat database”

· Product is present – Check if the product is in Open Icecat or Full Icecat

· Full – display of content for users with a Full Icecat subscription level will require the use of a server certificate and a dynamic secret phrase. Please contact your account manager for help with the implementation.

· Open - proceed with further actions

3. Content parameter checkup

· empty – return all product content

· filled in – check the parameter.

4. Obtained data for a specific product are limited to a chosen content type (if any).

5. Filtered data are sent to the client in JSON format.

2.3.1. Full Icecat catalog Access (available for users with a Full Icecat subscription level)

Icecat LIVE JS REST web service API checks request with digital signature, if request is OK, REST back-end gives needed JSON for Full Icecat content. Attempting to access a web service with an invalid signature will result in HTTP 403 (Forbidden) error, as you are not allowed to view the content of Full Icecat.

To get Full Icecat subscription means to pass a two-step procedure: to generate a certificate and to generate a signature. Let’s take a closer look on the procedure.

Firstly, CA certificate should be created. There are two possible options:

a) Client generates the RSA keys and the certificate (and shares data with Icecat support). See the step by step guide in the section 2.3.2. How to receive the CA certificate.

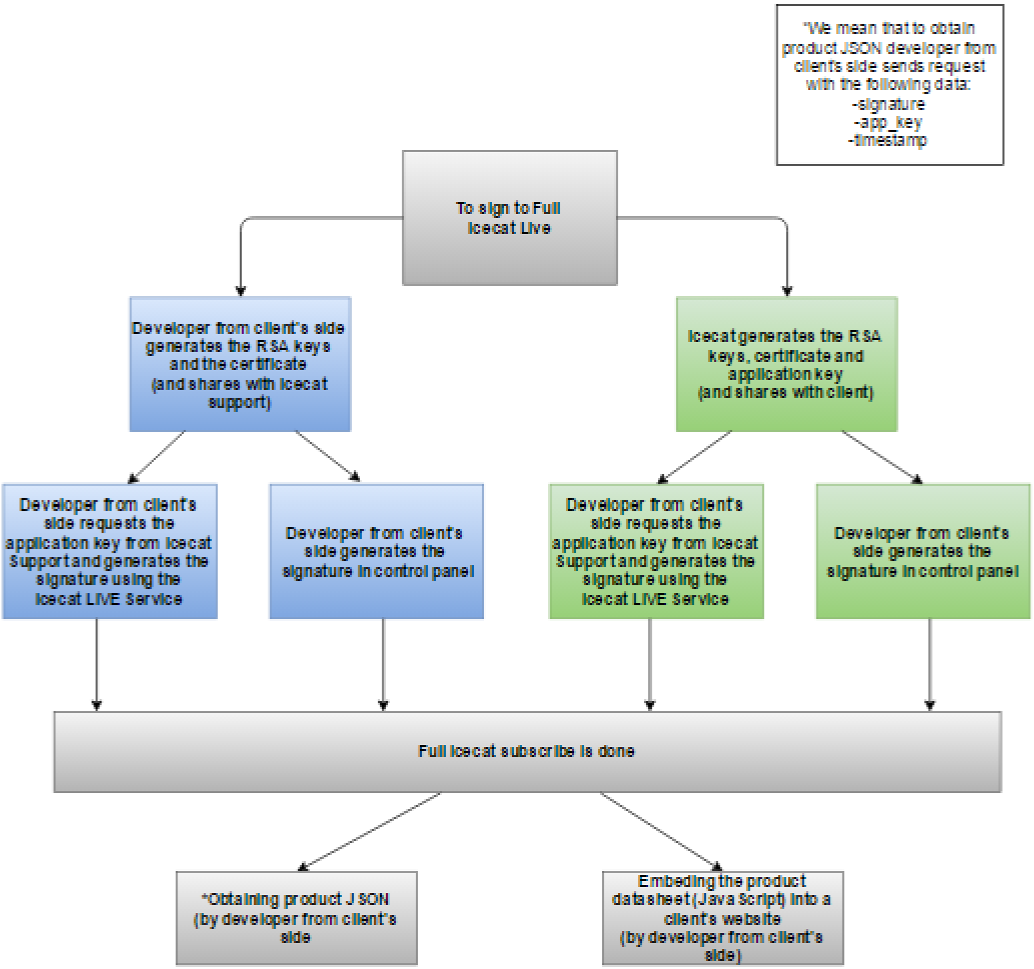
b) Icecat generates the RSA keys, aplication token and the certificate (and shares with client).

Secondly, user needs to generate the signature. Signature could be valid for the one IP address only. If user has several IP addresses and prefers to use Icecat LIVE service from different IP addresses, signature should be generated for every IP address. There are two possible options:

a) Client generates the signature using Icecat Live Service. See the step by step guide in the section 2.3.3. How to get signature from Icecat Live service (server to server scheme).

b) Client generates the signature using his own resource. See the step by step guide in the section 2.3.4.How to generate a signature with external library.

The scheme of the process looks the following way:


*We mean that to obtain product JSON developer from client's side sends request with the following data:
-signature
-app_key
-timestamp
To sign to Full Icecat Live
Developer from client's side generates the RSA keys and the certificate (and shares with Icecat support)
Icecat generates the RSA keys, certificate and application key (and shares with client)
Developer from client's side requests the application key from Icecat Support and generates the signature using the Icecat LIVE Service
Developer from client's side generates the signature in control panel
Developer from client's side requests the application key from Icecat Support and generates the signature using the Icecat LIVE Service
Developer from client's side generates the signature in control panel
Full Icecat subscribe is done
*Obtaining product JSON (by developer from client's side
Embeding the product datasheet (Java Script) into a client's website (by developer from client's side)

After user has generated the signature, he will be able to insert to client's website the product datasheets both of sponsorship and non-sponsorship products. See the step by step guide in the section 2.3.5 Icecat LIVE JS API

2.3.2. How to generate the self-signed certificate

a) Client generates the self-signed certificate

For Linux machine with OpenSSL installed or Keytool, you can Generate a new key without a passphrase.

To generate an RSA private key, use the `genrsa` option. The command below generates a 2048 bit RSA key and saves it to a file called `key.pem`:

```
openssl genrsa -out privatekey.pem 2048
```

Generate the self-signed root CA certificate with the previously generated key:

```
      openssl req -x509 -new -sha256 -nodes -key privatekey.pem -
days 365 -out certificate.pem
```

In the generation command we set 365 days as example of the certificate lasting term, but in fact we can set here any value.

Note!: SHA-256 signature algorithm only is used and checked by default

Check your certificate:

```
openssl x509 -in certificate.pem -text
```

After certificate is generated, user need to share the CA certificate, RSA keys and end-user IP address with Icecat, as this information is mandatory for signature generation.

Short description of the terms:

openssl	A service which provides a full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols
app_key	Application token (secret phrase). Application token is needed when user generates the signiture using the Icecat LIVE service.
RSA public key, private key	A 2048-bit pair ensures that the data can be encrypted by one key but can only be decrypted by the other key pair
CA certificate	A digital certificate which certifies the ownership of a public key by the named subject of the certificate.

b) Icecat generates the self-signed certificate

Alternatively, RSA key, CA certificate and application token may be generated by Icecat and shared with client (client can request CA certificate generation from Icecat support).

2.3.3. How to get signature from Icecat Live service (server to server scheme)

In this section we describe the actions to be done to get signature.

For the signature generation user needs the following data:

app_key	Application token (secret phrase). Application token is needed when user generates the signiture using the Icecat LIVE service.
shopname	Username
lang	Language code (en, nl, fr etc)
icecat_id or brand+part_code or ean_upc	Product identifier

timestamp	Each request must contain the date and time the request was created, represented as a string in timestamp. Format of the timestamp is e.g.: 1454339914. Icecat timestamp is usually valid during one day.
ip	Client IP address from which the user is viewing the current page. It is meant about end user IP.
useragent	Contents of the User-Agent: header from the current request, if there is one. This is a string denoting the user agent which is accessing the page. Useragent is necessary. Useragent example: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/46.0.2490.80 Safari/537.36

Below we give the step by step guide for the signiture generation:

1) To get the app_key.

Note!: Application key could be generated by Icecat only. In case, if client generated the RSA keys and a CA certificate himself, he need to send request for the application key generation to Icecat Support. For example, secret access key is: Tes0123456789t

2) To insert the app_key into the app_key attribute. In our example: app_key = Tes0123456789t

3) To check the useragent, for example, useragent can be taken from this website http://whatsmyuseragent.com/

Useragent example:

```
      useragent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/46.0.2490.80 Safari/537.36
```

Note!: For the signature generation, user need to use the end-user IP address.

4) To generate the signature with the data listed above:

```
GET
http://live.icecat.biz/api/signatures?app_key=test&shopname=openicecat-
live&lang=en&lang=en=1&icecat_id=24617541&ip=127.0.0.1&useragent=chrome
```

Example with our data:

```
      GET  http://live.icecat.biz/api/signatures?shopname=openicecat-
live&lang=en&icecat_id=11111&app_key= Tes0123456789Tqwertyui&ip=91.218.73.23
&useragent=Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/46.0.2490.80 Safari/537.36
200 OK
```

Signiture example:

```
{
    "signature":
"jj6lZugy1lPYcnGvrb9kD3coNDpv8PgpJUJpVLM%2F4iDTs7sUiusDJWlgRMNKREzM0Zf2zV8
bG3AsNePxBLXcMQ%2BKEQlt0P8GeqlWcWV8Tx05pgk8KWmVkRFG7Q0UqS1QbMer3Kn%2FuSRh2
esAW5kcC6BN60L%2Fnvf9zsessgdH4SwJzX1YYzruaNs%2B8uANariPcoeyfNoZwTIls7JbKPC
KX9eg%2BeHwnFmxGOQO0WjdMyg%2F4UO3ZGiLi%2FELb2z5%2BPgCo5GOQPMJjxFmlCsNcR6iZ
5oTVDlp6wJrSd104%2Fac6gVhqaKSGGhOXIV%2B1EREAAQ5m47E9OT2tmM%2FIZ520ADs9g%3D
%3D",
    "timestamp": 1438863985
}
```

Note!: Signature is valid for the one IP address only. If user has several IP addresses and prefers to use Icecat LIVE service from different IP addresses, signature should be generated for every IP address.

After user has generated the signature, he'll be able to insert to client's website the product datasheets both of sponsorship and non-sponsorship products. See the step by step guide in the section 2.3.5 Icecat LIVE JS API

2.3.4. How to generate a signature (in control panel)

Each request must contain a valid digital signature (see logic below), or the request is rejected. A request signature is calculated using your secret access key (SSL), which is a shared secret key known to user and Icecat live service only.

SHA-256 signature algorithm string explanation:

```
{shopname}\n
{icecat_id or ean_upc or brand_name}\n
{part_code - optional}\n
{lang}\n
{current timestamp}\n
{ip address}\n
{useragent}
```

shopname	Username
lang	Language short code (en, nl, fr etc)
icecat_id or brand+part_code or ean_upc	Product identifier
timestamp	Each request must contain the date and time the request was created, represented as a string in timestamp. Format of the timestamp is e.g.: 1454339914. Icecat timestamp is usually valid during one day.
ip	Client IP address from which the user is viewing the current page. It is meant about end user IP.
useragent	Contents of the User-Agent: header from the current request, if there is one. This is a string denoting the user agent which is accessing the page. Useragent is necessary.

String examples:

Note!: for correct use string must be converted to lowercase:

```
    openicecat-live\n
    24617541\n
    en\n
    1438269191\n
    91.218.73.23\n
    mozilla/5.0 (x11; linux x86_64) applewebkit/537.36 (khtml, like gecko)
chrome/46.0.2490.80 safari/537.36
```

or

```
    openicecat-live\n
    apple\n
    mg4f2ql/a\n
    nl\n
    1438269191
    91.218.73.23\n
    mozilla/5.0 (x11; linux x86_64) applewebkit/537.36 (khtml, like gecko)
chrome/46.0.2490.80 safari/537.36
```

or

```
    openicecat-live\n
    8032325143317\n
    fr\n
    1438269191\n
    91.218.73.23\n
    mozilla/5.0 (x11; linux x86_64) applewebkit/537.36 (khtml, like gecko)
chrome/46.0.2490.80 safari/537.36
```

\n - linefeed (LF or 0x0A (10) in ASCII)

Create digital signature from your string

Signature requirements:

- Base64 and URL encoding
- signature is associated with the attached private key

Request signature must be Base64 encoded. Base64 encoding converts the signature into a simple ASCII string that can be attached to the request.

Characters that can appear in the signature string like plus (+), forward slash (/), and equals (=) must be encoded if used in URL:

(+)	>	%2B
(/)	>	%2F
(=)	>	%3D

For example, if the authentication code includes a plus (+) sign, encode it as %2B in the request. Encode a forward slash as %2F and equals as %3D.

e.g. in PHP:

```
    $timestamp = time();
    $string = 'openicecat-live' . "\n" . '24617541' . "\n" . 'en' . "\n" .
$timestamp . “\n” . $ip . “\n” . $userAgent;
    openssl_sign($string, $signature, $privateKey, OPENSSL_ALGO_SHA256);
    $signBase64 = urlencode(base64_encode($signature));
```

Signiture example:

```
{
    "signature":
"jj6lZugy1lPYcnGvrb9kD3coNDpv8PgpJUJpVLM%2F4iDTs7sUiusDJWlgRMNKREzM0Zf2zV8
bG3AsNePxBLXcMQ%2BKEQlt0P8GeqlWcWV8Tx05pgk8KWmVkRFG7Q0UqS1QbMer3Kn%2FuSRh2
esAW5kcC6BN60L%2Fnvf9zsessgdH4SwJzX1YYzruaNs%2B8uANariPcoeyfNoZwTIls7JbKPC
KX9eg%2BeHwnFmxGOQO0WjdMyg%2F4UO3ZGiLi%2FELb2z5%2BPgCo5GOQPMJjxFmlCsNcR6iZ
5oTVDlp6wJrSd104%2Fac6gVhqaKSGGhOXIV%2B1EREAAQ5m47E9OT2tmM%2FIZ520ADs9g%3D
%3D",
    "timestamp": 1438863985
```

Note!: Signature is valid for the one IP address only. If user has several IP addresses and prefers to use Icecat LIVE service from different IP addresses, signature should be generated for every IP address.

2.3.5. Icecat LIVE JS API

After user subscribed to Full Icecat, he will receive an access both to sponsorship and non-sponsorship products. For adding the product datasheet to a client's website user need to do the inline JS call for the product data-sheet `IcecatLive.getDatasheet:`

app_key	Application token (secret phrase). Application token is needed when user generates the signiture using the Icecat LIVE service.
signature	Valid digital signature encdoded base64 and URL encode created by you own secret key
timestamp	Current timestamp which you add in signature Rest service checking signature when product from full catalog. For open catalog signature doesn't needed.

Request example:

```
IcecatLive.getDatasheet('#loadliveicecat',{'UserName':'openicecat-
 live','brand':'brandname','part_code':'123456789','signature':'abcdef0123
 456789',‘timestamp':'1234567890'},'en');
```

Or

http://live.icecat.biz/api/?username=username&lang=de&content=&timestamp=1454065393&signature={signature}&icecat_id=100

2.4. Individual JSON Product Requests

The General URI structure is:

http(-s)://live.icecat.biz/api/?UserName=<Icecat user name>&lang=<language code>&content=<content>&icecat_id=<product_id>

Alternative attributes used to identify a unique product data-sheet:

1. IcecatId= =<the Icecat Product ID>
2. Part_Code=<a Brand's part code>&brand=<the brand's name>
3. Ean_Upc=<the GTIN code>

Query parameters:

IcecatId or Part_Code&Brand or Ean_Upc	the unique product identifier: IcecatID or Brand + Manufacturer Part Code or GTIN (EAN, UPC or JAN)
Lang	the language code or locale code (in case of ambiguity) that is used
Content	content parameter
UserName	the Icecat user name as registered and authorized for certain repositories

If the **content** parameter is absent in the URL request, the system will return all products content that is available.

Generally, the JSON REST service operates with the following **content parameters**:

All	all content of product
GeneralInfo	general info of product, which includes: Product title, Brand, Product name, Brand product code, category, virtual category, product family, product series, gtin

Content parameter	parameter set to select the content: all content, images etc
Image	main image of product (IsMain)
Multimedia	all multimedia files related to a product: videos, PDFs, 360o/3D demos, and etc.
Gallery	gallery images of product
FeatureLogos	feature is enriched with a feature logo, and this logo is provided.
FeatureGroups	features are logically grouped, user can get just the needed features by request for getting only a featuregroup
Description	marketing text, long summary description
ReasonsToBuy	main arguments to buy products, format of reasons to buy (RTB) is: RTB image + RTB title + RTB description
Reviews	reviews of product, which are mainly provided through Testseek
Related	related products (this data is not used now)
Dictionary Imports for dictionary: Brand Category Product_Family Product_Series Reasons to buy User manual Product name Product code	translation of content elements into a local language

Here is the request:

http(s)://live.icecat.biz/api/?icecat_id=12462305&UserName=openIcecat-live&lang=EN&content={content parameter}*

***** you should enter the content parameter you need

ALL content parameter

The method GET is to be used. For all content parameter the following is to be entered:
http(s)://live.icecat.biz/api/?icecat_id=12462305&UserName=openIcecat-live&lang=EN&content=**all**

Multimedia content parameter

The method GET is to be used. For **mutlimedia** content parameter the following is to be entered:
http(s)://live.icecat.biz/api/?icecat_id=12462305&UserName=openIcecat-live&lang=EN&content=**multimedia**

3. Working with results

Once you enter any request, for example,
http(s)://live.icecat.biz/api/?icecat_id=12462305&UserName=openIcecat-live&lang=EN&content=multimedia
you can get two results:

1. the request succeeds
2. the request fails.

If the request succeeds, the server responds with a 200 OK HTTP status code and the data properties:

```
JSON
________________________________________________________________
200 OK
{
        message: ‘OK’,
‘data’ :
{
‘multimedia’:
[
{
‘ID’: <id>, // product identifier
‘URL’: <url>,
‘Type’: <type>,
‘ContentType’: <content_type>,
‘KeepAsUrl’: <keep_as_url>,
'Description': <description>,
‘Size’: <size>, (optional) // in bytes
‘IsPrivate’: <is_private>, (optional) // 0/1 or true/false
‘Updated’ : <updated>, (optional)
‘Language’: <langid>, // language code or locale code (if ambiguous)
`isVideo`:<isvideo> // 0/1 or true/false
},]
},
}
```

If mandatory fields weren’t filled in, the server responds with 404 HTTP status and the corresponding error message:

```
JSON
________________________________________________________________
404 Not Found
{
Status code:1
 Message: "Error: UserName, lang are mandatory fields"
}
```

When content, for example, multimedia object (MMO), is absent and product is sponsored (Open catalog), the server responds with a 200 OK HTTP status code:

```
JSON
________________________________________________________________________________
200 OK
{
message: ‘OK’,
‘data’ :
{
'Multimedia': []
}
}
```

If the request is failed due to any reason mentioned above (see ‘Data retrieving algorithm’), the server responds with 404 HTTP status and with the corresponding error message:

```
JSON
________________________________________________________________________________
404 Not Found
{
message : “Shop The Icecat user is unknown”
}
```

This is a live example from our database:

```
JSON
________________________________________________________________________________
message:"OK",
•     data:
{
•     Multimedia:
[
•     {
•     IcecatID:"1978657",
•     URL:"http://objects.icecat.biz/objects/12462305_1978657.mp4",
•     Type:"video/mp4",
•     ContentType:"video/mp4",
•     KeepAsUrl:0,
•     Description:"Video",
•     Size:"14623468",
•     IsPrivate:0*,
•     Updated:1426261310,
•     Language:"EN",
•     isVideo:1
},
•     {
•     IcecatID:"1978658",
•     URL:"http://objects.icecat.biz/objects/12462305_1978658.mp4",
•     Type:"video/mp4",
•     ContentType:"video/mp4",
•     KeepAsUrl:0,
•     Description:"Video",
•     Size:"22487335",
•     IsPrivate:0,
•     Updated:1426261310,
•     Language:"EN",
•     isVideo:1
},
•     {
•     IcecatID:"1978659",
•     URL:"http://objects.icecat.biz/objects/12462305_1978659.mp4",
•     Type:"video/mp4",
•     ContentType:"video/mp4",
•     KeepAsUrl:0,
•     Description:"Video",
•     Size:"20575036",
•     IsPrivate:0,
•     Updated:1426261310,
•     Language:"EN",
•     isVideo:1
},
•     {
•     IcecatID:"2142148",
•     URL:"http://objects.icecat.biz/objects/12462305_2319.swf",
•     Type:"360",
•     ContentType:"application/x-shockwave-flash",
•     KeepAsUrl:0,
•     Description:"360",
•     Size:"776006",
•     IsPrivate:0,
•     Updated:1426261310,
•     Language:"EN",
•     isVideo:0
}
],
•     demoAccount:false
}
          }
```

Note!: * ‘IsPrivate’: <is_private> means that this belongs to our rich content, which is accessible for assigned resellers.