

Manual for Open Icecat JSON Product Requests

JSON (JavaScript Object Notation) is an increasingly popular means of transferring to data, comparable to XML. Under the hood, it's used by [Icecat Live](#)'s JavaScript applications. The main advantages of JSON are that it's more efficient in terms of data transfer and that it's compatible with JavaScript applications requiring JSON. This manual explains the Open Icecat JSON product requests (Pull-API) call formats and the output. [For Full Icecat JSON, we refer to the general Icecat JSON manual](#), which will be reworked for the blog as well.

Other Integration Methods

If you want to integrate product data without frames, conforming your CSS, then use [Icecat Live](#), which is the modern variant of the good-old [URL](#) integration method. If you want to have full control over the data, then make use of the [XML](#), or [CSV](#) download methods. [See the overview of methods and manuals](#).

Do you want to add product reviews? Then check out our [reviews manual](#). If you're looking into pushing brand product data to Icecat, then check out the [Push-API \(JSON\) manual](#). Further, check out the [Add-On page for popular platforms](#) already integrating Icecat data.

Important Notes about Icecat JSON

1. Our JSON interface is relatively new, and certain aspects of our JSON interfaces might be subject to upgrades.
2. The current use of JSON is limited to the retrieval of individual product data-sheets. For the use of indexes and repositories, and more in-depth insight into the Icecat data model, we refer to the [XML documentation](#).

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. We're working on simplifying the authorization process in case of JSON data, making it comparable with the Full Icecat XML data access.
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.
5. Some brands have a restrictive syndication policy, in which they allow access to certain digital assets or other product data only to authorized resellers. In general, it is therefore good to [request authorization from brands](#) with which you have established formal relations, via the Icecat website.

Advantages of upgrading to Full Icecat

You can access the JSON interface by all Icecat Data (XML, CSV, or JSON) users, with at least an Open Icecat account. Open Icecat users can access the data of the [sponsoring brands](#). Full Icecat users can access the data of [all brands](#) in the Icecat database.

However attractive the free Open Icecat database is, the Full-Icecat database still contains many, many more [brands](#). If you want to include specific brands into free Open Icecat, please, recommend Icecat to your manufacturer contacts.

[Read more about the main advantages of Full Icecat over free Open Icecat](#) and [see the Icecat Subscription Plans](#).

If you want to upgrade to Full Icecat, please [contact us](#) via the contact form, or click on the "request Full Icecat access"

link after the [User Login](#).

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, requires to respect the [Open Icecat Fair Use Policy](#).

Explicitly relevant for website development is:

- to mention explicitly the “Specs Icecat” on all product datasheets, with a link to the Icecat.biz web site
- to include explicitly an “AS IS” disclaimer, an example of which can be found at <https://www.icecat.biz/menu/disclaimer/index.html> (available in several languages).

International Standards Supported

Apart from JSON, Icecat is compliant with an increasing number of international standards. [See, here a quick overview of such supported standards](#).

1. JSON Examples for Instant Use

Before elaborating the generic JSON structures, it's good to look into examples. Below some examples of real-time Icecat JSON product requests based on different product identifiers. You can dynamically construct such links for your applications using your Icecat user name and the product identifiers known by your system.

1. ProductCode & Brand = Brand product code + the brand name:

```
https://live.icecat.biz/api/?UserName=openIcecat-live&Language=en&Brand=hp&ProductCode=RJ459AV
```

2. GTIN = the GTIN (EAN, UPC or JAN) bar code:

```
https://live.icecat.biz/api/?UserName=openIcecat-live&Language
```

=en>IN=0882780751682

The third method, will require a bit more complex integration, as you have to read first the Icecat index files to obtain the Icecat product IDs. More information regarding index files can be found [here](#). Recommended in case of large-scale integrations in which you want to predict for which products Icecat has product data or not.

3. icecat_id = Numerical product identifier in Icecat catalog (take from Icecat index files):

`https://live.icecat.biz/api/?UserName=openIcecat-live&Language=en&icecat_id=1198270`

The general URL structure is:

– product code & the brand’s name:

`http(s)://live.icecat.biz/api/?UserName=&Language=&Content=&Brand=&ProductCode=`

– GTIN (EAN, UPC or JAN code):

`http(s)://live.icecat.biz/api/?UserName=&Language=&Content=>IN=`

– Icecat ID:

`http(s)://live.icecat.biz/api/?UserName=&Language=&Content=&icecat_id=`

Query parameters:

Query Parameter	Description
ProductCode + Brand or GTIN	The unique product identifier: Brand + Manufacturer Part Code or GTIN (EAN, UPC or JAN)
Language	the language code or locale code (in case of ambiguity) that is used

Content	Content parameter. If the Content parameter is absent in the URL request, the system will return all products content that is available.
UserName	the Icecat user name as registered and authorized for certain repositories

Generally, the JSON REST service operates with the following Content parameters:

Content parameter	Description
All	All content of the product
GeneralInfo	The general info of a product, which includes: Product title, Brand, Product name, Brand product code, category, virtual category, product family, product series, GTIN
Image	The main image of a product (IsMain)
Multimedia	all multimedia files related to a product: videos, PDFs, 360o/3D demos, and etc.
Gallery	all gallery images of product
FeatureLogos	feature is enriched with a feature logo, and this logo is provided.
Description	The marketing text and long summary description
ReasonsToBuy	The main arguments to buy products, formatted as reasons to buy (RTB): RTB image + RTB title + RTB description
Reviews	The expert or user reviews of a product, which are mainly provided by Testseek.

Related	The related products (this data is not used now)
Dictionary (Imports for dictionary: Brand, Category, Product_Family, Product_Series, ReasonsToBuy, UserManual, ProductName, Product code)	The translation of content elements into the local language

Here is the request structure:

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

* you should enter the content parameter you need

Examples:

- With '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/?UserName=openIcecat-live&Language=en&Content=All&Brand=hp&ProductCode=RJ459AV
```

- With 'multimedia' content parameter

The method GET is to be used. For multimedia content parameter the following is to be entered:

```
http(s)://live.icecat.biz/api/?UserName=openIcecat-live&Language=en&Content=Multimedia&Brand=hp&ProductCode=RJ459AV
```

See also [Digital Asset Types as Represented in Icecat's Open Catalog Interfaces](#).

2. Working with JSON Output

Once you enter any request, for example, `http(s)://live.icecat.biz/api/?icecat_id=12462305&UserName=openIcecatlive&Language=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': , // product identifier
        'URL': ,
        'Type': ,
        'ContentType': ,
        'KeepAsUrl': ,
        'Description': ,
        'Size': , (optional) // in bytes
        'IsPrivate': , (optional) // 0/1 or true/false
        'Updated' : , (optional)
        'Language': , // language code or locale code (if ambiguous)
        `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: "https://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: "https://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: "https://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:"https://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
}
}

```

In case user wants to download JSON files of the Full Icecat product(s), he needs to make sure that his IP is whitelisted for the relevant Full Icecat user and he request the JSON files using App-Key. IP can be whitelisted by the user himself via the [Icecat web site](#). App-Key is the Icecat internal entity which we can provide on demand for Full Icecat users. User can request it from relevant Icecat manager or general [contact form](#).

URL structures to request **Full Icecat** products JSON files are the same as for Open Icecat products described above, only difference is that in case of the Full Icecat products URL structure should end with **&app_key=...**

Note!: 'IsPrivate:1' means that this (asset) belongs to our private database, which is accessible for authorized resellers only. 'IsPrivate:0' means that no further authorization is required to access the asset(s) involved.

Possible Error Messages

You can see all the error messages that you can get while using JSON files from Icecat via the following [link](#)