Corincloud API v.0.5.0.0 documentation

The API is in development stage so there can be errors or missing functions.

The API is made that our partners can easily communicate with our system, if they need data. We set a principle that it must be platform independent, so it can be accessed from any programming language which has JSON capabilities. This documentation's examples based on PHP and cURL PHP extension.

The first step is, that You need access credentials to be able to use the API.

In order to do this, You can either generate your own credentials in your corincloud webshop admin, on the following page:

Settings -> Users -> API partners (Beállítások -> Felhasználók -> API partnerek)

or, if You are a third party, please request these credentials from your corincloud webshop partner.

API url structure:

Our api url structure is very simple to use.

E.g: www.test.domain.hu/api/****module****/****iid****
where:

- ****module**** is the resource you want to use
- ****iid**** is the item id that you want to **get**, **update** or **delete**

There are cases where the module is split between multiple submodules, in that case the follwing structure is used (e.g.: attribute api):

E.g.#2: www.test.domain.hu/api/****module**/**submodule**/**iid****
where:

- ****module**** is the resource you want to use
- ****submodule**** is the sub resource you want to use
- ****iid**** is the item id that you want to **get**, **update** or **delete**

Available resources:

- attribute (**beta**)
- coupon (**beta**)
- image (**beta**)
- order (**beta**)
- product
- variant

Examples:

www.sample.domain.hu/api/product	[GET]	- get product list
www.sample.domain.hu/api/product	[POST]	- insert new product data
www.sample.domain.hu/api/product/123	[GET]	- get product data
www.sample.domain.hu/api/product/123	[POST]	- update product data
www.sample.domain.hu/api/product/123	[DELETE]	- delete product data

Notes (url parameters, important):

Requesting lists:
If you request a list (e.g.: product) from the API you have the option te set a limit and a page parameter to control how many id appears in the response array, and which page are you on, in the lists of pages.
E.g.:
www.sample.domain.hu/api/product?limit=20&page=2

Requesting specific items:

If you request a specific item (e.g.:product) from the API you have the option to filter specific fields that you want to get, instead of the whole data array.

Use the “field” parameter, which is a list of fields separated by colons, to filter the fields you need from the resource. **E.g.:**
www.sample.domain.hu/api/product/123?field=SKU,PRODUCTNAME,CREATEDATE

API Response:

The answers of the API will be an appropriate HTTP header and a JSON encoded array which will hold the requested data, or an array of error status code and error message combination if the API ran into problems.

Response headers:

200 - OK
404 - Not Found
405 - Not Allowed
500 - Internal Server Error

Basic API errors:

101 - Api login unknown error
102 - Not found api key

103 - Api not allowed
104 - Not logged in
105 - Inactive api user
106 - Not found IP
107 - Not found module
108 - Not found event
109 - Module not allowed
110 - Not found iid
111 - Event not allowed
112 - Password error
113 - Unknown module error
114 - Method not allowed
115 - Unauthorized acces
116 - Exists item recall set event
117 - Not found post parameters
118 - Not found put data
119 - Not found delete data

Authorization:

Our API use Basic HTTP authentication to authorize incoming calls. You must provide your credentials in the HTTP headers of your every call that you make. To do this, look at the following example.

Example (cURL basic authentication options) (DEPRECATED):
This method of authentication is no longer supported, please use the newer one below

```
// set HTTP authentication header Accept field to Basic
curl_setopt($curl, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
// set credentials
curl_setopt($curl, CURLOPT_USERPWD, $api_username.":".$api_password);
```

Example (cURL custom api header options):
This is the new way of giving the api credentials to every call, the row in the previous auth process don't needed anymore with this

```
// Set these custom http headers for authentication
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
```

These options will be used in the calls to identify yourself as the api user.

Product API:

Product API errors:
201 - Product not found
202 - Not created products
203 - Not found product sku
204 - Not found product id
205 - Not found product stock
206 - Stock update error

Available GET/POST fields:

Field	Description	Header type
SKU	Product unique identifier	GET, POST
MODELNUMBER	Product model id	GET, POST
PRODUCTNAME	Product name	GET, POST
SOURCESKU	Product unique identifier in system which the product came from	GET, POST
LINECODE	It is a collective noun for (eancode, barcode etc.)	GET, POST
GROUPCODE	Group identifier, different products can be connected by this field	GET, POST
LASTSOURCE	Identifies the source which processed the product	GET, POST
SOURCE	Identifies the original source which the product came from	GET, POST
WEIGHT	Product weight	GET, POST
UNIT	Product unit quantity	GET, POST
UNITNAME	product unit name (eg. qty, pcs, pair)	GET, POST
SHIPPINGCOST	Product unique shipping cost	GET, POST
MINQUANTITY	Product	GET
DISCOUNTPRICE	By default, this is the same as product normal price, if this field is lower than normal price, then this field holds the products discount price. It can be deleted if You put 0 (zero) or ‘X’ in the field.	GET, POST
PRICE	Product normal price	GET, POST
STOCK	Products stock	GET, POST
VIEWS	Product all time views	GET
ITEMSORT	Product ordering number	GET, POST
MODDATE	Last modified date	GET

CREATEDATE	Crated date	GET
ACTIVE	Product is active or inactive	GET, POST
PUBLIC	Product is public or not	GET, POST
DESCRIPTION	Product long description. It can be deleted if You give ‘X’ in the field.	GET, POST
SHORTDESCRIPTION	Product short description. It can be deleted if You give ‘X’ in the field.	GET, POST
TEXTATTRIBUTES	Product attribute list as an array of arrays which holds three fields: - ATTRIBUTE_KEY : name of attribute - ATTRIBUTE_VALUE : value of attribute - ISHIDDEN : can be 0 or 1. If it is 1, then the attribute won’t show up on the product page. It can be deleted if You give ‘X’ in the field.	GET, POST
TAGS	Product labels / tags **BETA** Usable types: (BRAND, COLOR, TYPE, SIZE, MATERIAL, SEX, SHIPPINGDAYS) ***See example and explanation in ’Additional examples’ section (Insert attributes / tags)***	GET, POST
MANUFACTURER	Product manufacturer as string	GET, POST
BRAND	Product brand as string	GET, POST
CATEGORYTREE	Product categories **BETA** Data array which holds 2 fields, **EVENT** and **ITEMS**. **EVENT** is a string that sets the API how to processes the categories **ITEMS** is an array of strings, where the strings are the category tree entries for this product. ***See example in ’Additional examples’ section (Insert categories)***	POST
IMAGES	Product images in array, BIG, THUMB, NORMAL sizes	GET
PRODUCTIMAGE	Add new product images as a string list of URLs in an array	POST
VARIANTS	Product variants list, VARID can be used to get variant data in the variant api.	GET
CATEGORYID	Product maincategory id, It can be obtained via attribute api. It must be a category’s label id.	GET, POST
TYPEID	Product type id, It can be obtained via attribute api. It describes	GET, POST

	the products type, as to which label categories are connected to this particular product. E.g.: We are setting the product type as “Shoes” (TYPEID field must hold the label id for the label “Shoes”) The connected label categories will be the following (these connections can be edited on the administration interface): - size - color	

Call examples:

Get product list:
This function will get the list of product ids in the current webshop.

Header type: **GET**
This call will result in a JSON encoded array which will contain:

- number of products (count)
- product id list (items)
- next x product ids url (next)
- previous x product ids url (prev)

Parameters:

- limit (quantity of elements, default 10)
- page (list offset, which part of the list are we on, based on the limit)
- label (filter products, which has the given label identifier)
- type (flter products, which has the given type identifier)
- filteractive (filter products, which has active status, and has stock > 0 (in case of using stock handling module in the webshop))
- modifydate (filter products, which has a modified date greater or equal than given value, value needs to be a timestamp, use php time() function for this)

Call example:

```php
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'https://sample.domain.hu/api/product?limit=10&page=1');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($curl);
// process json $response here
```

Response example:

```
'{"count":"87","items":{"0":"8398","1":"8397","2":"8396","3":"8395","4":"8394","5":"8393","6":"8392","7":"
8391","8":"8390","9":"8389"},"next":"\/api\/product?limit=10\u0026page=2"}'
```


Get product list method #2:
This method is an alternative implementation of the api product list method. This one will give you the products' basic datas not only the product id. This way you don't need to implement a two step synchronization process.

Header type: **GET**
This call will result in a JSON encoded array which will contain:

- number of products (count)
- product list (items), which contains the basic data for each product
- next x product url (next)
- previous x product url (prev)

Parameters:

- limit (quantity of elements, default 10)
- page (list offset, which part of the list are we on, based on the limit)
- label (filter products, which has the given label identifier)
- type (flter products, which has the given type identifier)
- filteractive (filter products, which has active status, and has stock > 0 (in case of using stock handling module in the webshop))
- modifydate (filter products, which has a modified date greater or equal than given value, value needs to be a timestamp, use php time() function for this)

Call example:

```php
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'https://sample.domain.hu/api/product/list?limit=10&page=1);
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($curl);
// process json $response here
```

Response example:


```
{"count":"2","items":{"0":{"PRODUCTID":"42422","SKU":"21","MODELNUMBER":"21","LINECODE":"","PRODUCTNAME":"Sample
product 1","STOCK":1,"DISCOUNTPRICE":1899,"PRICE":2290,"VAT":"27","WEIGHT":"500","BRAND":"Sample
product","SHORTDESCRIPTION":"shortdescription","DESCRIPTION":"longdescription","ACTIVE":"1","CREATEDATE":"2017-0
4-12 19:54:04","MODDATE":"2018-04-10 15:44:28","CATEGORYTREE":{"59513":"A","59515":"A \u003E A1","59517":"A
\u003E A1 \u003E
A11"}},"1":{"PRODUCTID":"42423","SKU":"22","MODELNUMBER":"22","LINECODE":"","PRODUCTNAME":"Sample product
2","STOCK":1,"DISCOUNTPRICE":1899,"PRICE":2290,"VAT":"27","WEIGHT":"1000","BRAND":"Sample
product","SHORTDESCRIPTION":"shortdescription2","DESCRIPTION":"longdescription2","ACTIVE":"1","CREATEDATE":"2017
-04-12 19:54:04","MODDATE":"2018-04-10 13:53:42","CATEGORYTREE":{"59513":"A","59515":"A \u003E A1","59517":"A
\u003E A1 \u003E A11"}}}}
```

Get specific product data:
This function will get the data of a speific product in the current webshop.

Header type: **GET**
This call will result in a JSON encoded array, which will hold the requested data (all available fields) for this specific product.

Parameters:

- field (list of product fields separated by colons, it filters the data set)

Call example:

```php
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'https://sample.domain.hu/api/product/8398');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($curl);
// process json $response here
```

Response example:


```
'{"PRODUCTID":"8398","SKU":"65264","MODELNUMBER":"","SOURCESKU":"","LINECODE":"","GROUPCODE":"","LASTSOURCE":"0","SOURCE":"0","GALLERYID":"5253","MEDIAID":"0","CATEGORYID":"0","BRANDID":"0","SEOID":"227","PRODUCTNAME":"rename product1","ACTIVE":"1","PUBLIC":"1","UNIT":1,"UNITNAME":"db","WEIGHT":"0","DISCOUNTPRICE":"0","PRICE":"0","SHIPPINGCOST":"0.00","MINQUANTITY":"0","STOCK":"0","VIEWS":"0","ITEMSORT":"0","MODDATE":"2015-08-06 17:25:23","CREATEDATE":"2015-08-06 17:25:15","IMAGES":{"0":{"big":"skin\/0014\/images\/kk_minta_kep.jpg","thumb":"skin\/0014\/images\/kk_minta_kep.jpg","normal":"skin\/0014\/images\/kk_minta_kep.jpg"}},"TEXTATTRIBUTES":"","SHORTDESCRIPTION":"","DESCRIPTION":"","BRAND":""}'
```


Insert new product:
This function will insert a new product into the current webshop.

Header type: **POST**
This call will result in a JSON encoded array which will hold the posted data set.

Call example:

```php
$postdata = array(
```

```
    'SKU' => "A1S2D3_55698",
    'PRODUCTNAME' => 'API test product insert name',
    'DISCOUNTPRICE' => 5500,
    'PRICE' => 5900,
    'STOCK' => 55,
    "ACTIVE" => 1,
    "PUBLIC" => 1,
    "DESCRIPTION" => "Api test long description",
    "SHORTDESCRIPTION" => "Api test short description",
    "TEXTATTRIBUTES" => array(
        array(
            "ATTRIBUTE_KEY" => "Made in",
            "ATTRIBUTE_VALUE" => "Hungary",
            "ISHIDDEN" => 0, // 0 = shown, 1 = hidden
        ),
        array(
             "ATTRIBUTE_KEY" => "Warranty",
             "ATTRIBUTE_VALUE" => "24 months",
             "ISHIDDEN" => 1, // 0 = shown, 1 = hidden
        )
    )
    "MANUFACTURER" => "Samsung",
    "BRAND" => "Samsung",
    'PRODUCTIMAGE' => array(
       'http://image.domain.hu/images/_5e9e1bc089505be5ce86c213c4c1a5e1.jpg',
       'http://image.domain.hu/images/_asdasdasd089505be5ce86c213c4c1a5e1.jpg',
       'http://image.domain.hu/images/_5easdhjasgafhagsce86c213c4c1a5e1.jpg',
    )
  );

  $curl = curl_init();
  curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
  curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
  curl_setopt($curl, CURLOPT_POST, true);
  curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
```

```
$response = curl_exec($curl);
// process json response here
```

Response example:

```
'{"DISCOUNTPRICE":"5900","PRICE":"5900","STOCK":"55","BRANDID":"9287","PRODUCTID":8436}'
```

Update product:
This function will update a specific product in the current webshop.

Header type: **POST**
This call will result in a JSON encoded array which will hold those fields which are updated.

Call example:

```
$postdata = array(
    'PRODUCTNAME' => 'API test product update name',
    'STOCK' => 66,
);

$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product/8436');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
$response = curl_exec($curl);
// process json response here
```

Response example:

```
{"PRODUCTNAME":"API test product update name","STOCK":"66","PRODUCTID":8436}
```

if the response array is empty, it means that there were no fields to update.

Delete existing product:
This function will completely delete a product drom the current webshop.

Header type: **DELETE**
This call results in a JSON encoded array which will hold the product id of the deleted product.
Notice: This call will completely remove all product elements from the system.

Call example:

```
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product/8438');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_CUSTOMREQUEST,"DELETE");
$response = curl_exec($curl);
// process response json here
```

Response example:

```
'{"PRODUCTID":8438}'
```

Call examples:

Search by SKU:
Header type: **GET**
Note: This call is the same as the base product id list request, but you can search specific products by SKU, if you do the following call with the **get** parameter:

```
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product?get=Test_sku');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($curl);
// process json response here
```

Search by SKU #2:
Header type: **GET**
Note: We made this function to make searching for a specific SKU easier. It's only purpose, is to provide information about the existence of a product, or variant. It can return two kinds of identifiers. On is the product id which is used by the product api, the other one is the variant id (var id for short) which is used by the variant api.

- By default if you search for a product, and you don't use variants or don't use SKUs for variants (setting) in the shop, then it returns only the product id if it has a match
- The extended functionality is provided when you use variants and set the webshop to use SKU for variants. In this case if the search has found a match, then it returns the matched variant id and product id of the product which has this particular variant. Whit this you don't need to check twice for product and variant match.
- If there aren't any match for the given SKU, then the call will show an appropriate error message.

Call example:
Note: There are cases when the given sku has spaces or special characters inside of them, so the use of *'urlencode'* PHP function is highly recommended.

```php
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product/getid/'.urlencode('teszt sku'));
 curl_setopt($curl, CURLOPT_HTTPHEADER,
     array(
         "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
     )
 );
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($curl);
// process json response here
```

Response example:

```
{'PRODUCTID':'12345'}

{'PRODUCTID':'12345','VARID':'54321'}
```

Insert pictures:
Header type: **POST**
Note: This call is performed when the product is **already in the webshop**, so this counts as an UPDATE. You need to place the PRODUCTID at the end of the call url to upload the images for this specific product.

```php
$postdata = array(
  'PRODUCTIMAGE' => array(
    'http://image.domain.hu/images/_5e9e1bc089505be5ce86c213c4c1a5e1.jpg',
    'http://image.domain.hu/images/_asdasdasd089505be5ce86c213c4c1a5e1.jpg',
    'http://image.domain.hu/images/_5easdhjasgafhagsce86c213c4c1a5e1.jpg',
  )
);
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product/0123');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
$response = curl_exec($curl);
// process json response here
```

Note#2: If you want to insert a new product, and want to insert a picture as well with it, you just have to add the PRODUCTIMAGE field into the post data array. The PRODUCTIMAGE field must be an array of urls to the image files.

This example call will insert a new product, and set the product sku, product name, product stock and product image.

```php
$postdata = array(
  'SKU' => 'Test_SKU',
  'PRODUCTNAME' => 'Test API product name',
  'STOCK' => 32,
   'PRODUCTIMAGE' => array(
      'http://image.domain.hu/images/_5e9e1bc089505be5ce86c213c4c1a5e1.jpg',
      'http://image.domain.hu/images/_asdasdasd089505be5ce86c213c4c1a5e1.jpg',
      'http://image.domain.hu/images/_5easdhjasgafhagsce86c213c4c1a5e1.jpg',
    )
);
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
$response = curl_exec($curl);
// process json response here
```

Insert categories by providing it with the product data:
Header type: **POST**
Note: You can insert categories by providing the CATEGORYTREE field in the post data array of your call.

The CATEGORYTREE field is an array, which contains two fields:

EVENT: this field is used to decide what should the API do with the categories. It has 2 values:

- **add** : this will set the API to append the new entries to the product's categories
- **update** : this will set the API to completely replace the product's categories (**note:** delete previous category entries for this product and add the new ones).

ITEMS: this is an array of strings which holds the category tree entries. The first category tree item is considered as the product main category, and it will be set only on update event. The strings must be in the following format:

Maincategory>First level subcategory>Second level subcategory

If the categories missing from the webshop, then these will be created as given.
E.g.:
Maincategory
- First level subcategory
    - Second level subcategory

Note: This sample call will UPDATE the product manufacturer, brand, and categories.

```php
$postdata = array(
  "MANUFACTURER"=>"Dc Shoes",
  "BRAND"=>"DC",
  'CATEGORYTREE' => array(
      'EVENT' => 'update',
      'ITEMS' => array(
             'Male>Outdoor',
             'Male>Sport>Skate',
             'Male>Endurance',
      ),
  ),
);
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product/0123');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
$response = curl_exec($curl);
// process json response here
```

Insert description by providing it with the product data:
Header type: **POST**
Note: You can insert description by providing the DESCRIPTION field in the post data array of your call.
The DESCRIPTION field is a html formated string:
Note: This sample call will UPDATE the product description and short description (lead).

```php
$postdata = array(
  "DESCRIPTION"=>"Api test long description",
  "SHORTDESCRIPTION"=>"Api test short description",
);
```

```php
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product/0123');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
$response = curl_exec($curl);
// process json response here
```

Insert price or discount price by providing it with the product data:
Header type: **POST**
Note #1: You can insert price by providing the PRICE / DISCOUNTPRICE field in the post data array of your call.
Note #2: If you want to delete the discount price from your product, you can do that by setting the DISCOUNTPRICE field to 0.
Note #3: This sample call will UPDATE the product price and discount price.

```php
$postdata = array(
   'DISCOUNTPRICE'=>5500,
   'PRICE'=>5900,
);
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product/0123');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
$response = curl_exec($curl);
// process json response here
```

Insert product tags/labels by providing it with the product data:
Header type: **POST**
Note: You can insert tags by providing the TAGS field in the post data array of your call.
The TAGS field is a html formated string:
Note: This sample call will INSERT the product tags

```php
$postdata = array(
   "TAGS"=>array(
      “COLOR” => array(
```

```
            'EVENT' => 'add'
            'ITEMS' => array(
               "blue",
               "red"
             ),
       ),
    ),
  );
  $curl = curl_init();
  curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product/0123');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
  curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
  curl_setopt($curl, CURLOPT_POST, true);
  curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
  $response = curl_exec($curl);
  // process json response here
```

Get Product stock batch

Header type: **GET**

This function is working almost exactly like the normal get product list function, except it will give you the product's product id, sku, type id, and stock fields in the items response array to work with.

Parameters:

- limit (quantity of elements, default 10)
- page (list offset, which part of the list are we on, based on the limit)
- label (filter products, which has the given label identifier)
- type (flter products, which has the given type identifier)

Call example:

```
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'https://sample.domain.hu/api/product/stock?limit=10&page=1);
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($curl);
```

```
// process json $response here
```

Response example:

```
'{"count":"16","items":{"0":{"PRODUCTID":"33004","SKU":"228958","STOCK":"1"},"1":{"PRODUCTID":"33005","SKU
":"85834","STOCK":"2"},"2":{"PRODUCTID":"33006","SKU":"853192","STOCK":"1"},"3":{"PRODUCTID":"33007","SKU"
:"317960","STOCK":"3"},"4":{"PRODUCTID":"33126","SKU":"777685","STOCK":"0"}},"next":"\/api\/product\/stock
?limit=5\u0026page=2"}'
```

Set product stock batch
Header type: **POST**

This function is working with the Get product stock batch function's response items list format. You must provide an array of arrays in which the sub arrays contains the products relevant datas. PRODUCTID for primary identification, SKU if you dont know the PRODUCTID of the product, and STOCK field with correct stock data. The function will give you this array back in the provided format in the call respoonse. If an error occures during a product stock update process, the error message will be given to you in the exact place (array index) of the product data you provided.

Call example:

```
$postdata = array(
  array(
    "PRODUCTID" => "33004",
    "SKU" => "228958",
    "STOCK" => "",
  ),
  array(
    "PRODUCTID" => "33005",
    "SKU" => "85834",
    "STOCK" => "2"
  ),
  array(
    "PRODUCTID" => "33006",
    "SKU" => "853192",
    "STOCK" => "1"
  )
);

$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'https://sample.domain.hu/api/product/stock');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
$response = curl_exec($curl);
```

```
// process json response here
```

Response example:


```
{"0":{"error":205,"msg":"not_found_product_stock"},"1":{"PRODUCTID":"33005","SKU":"85834","STOCK":"2"},"2":{"PRODUCTID":"33006","SKU":"853192","STOCK":"1"}}
```


Product Variant API

Product variant api errors:
301 - Variant not found
302 - Variant not created
303 - Required product id

Available GET/POST fields:

Field	Description	Header type
PRODUCTID	variant product id	GET, POST
LABELCATEGORY1	Variant category id 1	GET, POST
LABELCATEGORY2	Variant category id 2	GET, POST
LABELNAME1	Variant label name 1	GET, POST
LABELNAME2	Variant label name 2	GET, POST
SKU	Variant sku	GET, POST
LINECODE	Variant linecode	GET, POST
PRICE	Variant price	GET, POST
ITEMSORT	Variant item sort	GET, POST
STOCK	Variant stock	GET, POST
ACTIVE	Variant active	GET, POST
DEFAULT	Variant default item	GET, POST
MODIFY	Variant modify date	GET

Get a specific product variant data:
Header type: **GET**
This function will result in a JSON encoded array which will contain the requested product variant data.

Call example:

```
$curl = curl_init();
```

```php
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/variant/5520');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($curl);
// process json response here
```

Response example:

```
'{"PRODUCTID":"8398","SKU":"65264","LABELCATEGORY1":"méret","LABELNAME1":"32","ACTIVE":"1","STOCK":"1","PR
ICE":2000,"ITEMSORT":"10","DEFAULT":"0","MODIFY":"2015-08-18 08:09:10"}'
```

Insert variant:
Header type: **POST**
This function will insert a new product variant into the current webshop.
This call will result in a JSON encoded array which will contain the posted variant data set.

Call example:

```php
$postdata = array(
    'PRODUCTID'=>"5520",
    'LABELCATEGORY1' => 'méret',
    'LABELNAME1'=>'32',
    'SKU'=>'12345',
    'STOCK' => '10',
    "ACTIVE"=>1,
    "PRICE"=>'2000',
    "ITEMSORT"=>10,
    "DEFAULT"=>0,
);
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/variant);
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
$response = curl_exec($curl);
```

```
// process json response here
```

Response example:

```
'{"PRODUCTID":"5520","SKU":"12345","LABELCATEGORY1":"Méret","LABELNAME1":"32","ACTIVE":"1","STOCK":"1","PRICE":2000,"ITEMSORT":"10","DEFAULT":"0"}'
```

Update specific product variant data:
Header type: **POST**
This function will update a product variant data in the current webshop.

This call will result in a JSON encoded array which will contain the fields which were updted.

Call example:

```php
$postdata = array(
    'PRODUCTID'=>"5520",
    'LABELCATEGORY1' => 'méret',
    'LABELNAME1'=>'32',
    'SKU'=>'12345',
    'STOCK' => '10',
    "ACTIVE"=>1,
    "PRICE"=>'2000',
    "ITEMSORT"=>10,
    "DEFAULT"=>0,
 );
 $curl = curl_init();
 curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/variant/3472);
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
 curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
 curl_setopt($curl, CURLOPT_POST, true);
 curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
 $response = curl_exec($curl);
 // process json response here
```

Response example:

```
'{"VARID":"3472"}'
```

Delete specific variant:

Header type: **DELETE**
This function will delete an existing product variant.

This call will result in a JSON encoded array which will contain the deleted variant product id.

Call example:

```
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product/3472);
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_CUSTOMREQUEST,"DELETE");
$response = curl_exec($curl);
// process response json here
```

Response example:

```
'{"VARID":"3472"}'
```

Product Image API: (beta version)

Our system is storing the source files of your images, and generates 3 types of resized versions that will be actually used at various points of the webshop user interface and administration interface.

product image api errors:
401 - product not found
402 - image not found
403 - image not uploaded
406 - source image not created
407 - image already exists
408 - not allowed image format (accepted: jpg, png)
409 - Image not deleted
410 - Galleries not found
411 - Gallery folder not exists
412 - Cannot open gallery folder
413 - Cannot open source folder
414 - Not found postdata media id
415 - Not found media data

Get product images
Header type: **GET**

This function will provide information about pictures of a product. The last part of the call url identifies the product, it is the product id.

This call will result in a JSON encoded array which will hold the product's picture informations.

Call example:

```php
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/image/1234');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($curl);
// process JSON response here
```

Response example:


```
{"ITEMID":1234,"COUNT":1,"ITEMS":{"8124":{"MEDIAID":"8124","MEDIANAME":"apitest-image-api-sample_7a0d720a362c44c
40867330af7d4b79d.jpg","MEDIAURL":"http:\/\/sample.domain.hu\/public\/0538\/product\/001\/234\/source\/apitest-i
mage-api-sample_7a0d720a362c44c40867330af7d4b79d.jpg","ITEMSORT":"0","CREATEDATE":"0000-00-00
00:00:00","MODIFYDATE":"0000-00-00 00:00:00","MEDIASIZE":37426}}}
```


Update product image data:
Header: **POST**
This function will update image data for product image.
Note: This function has a small functionality. At this point, it is only used for setting the order of the images.

This function will result in a JSON encoded array which will hold the updated product picture's media id and item order number if updated.

Call example:

```php
$postdata = array(
  "MEDIAID" => "159753",
  "ITEMSORT" => "0",
);
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/image/432112');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
```

```
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
$response = curl_exec($curl);
// parse JSON response here
```

Response example:

```
{"MEDIAID":"159753","ITEMSORT":"0"}
```

Upload base64 encoded string image data:
Header type: **PUT**
This function will upload a base64 encoded string image data as image to the given product. **1 image per cURL exec call!**

This call will result in a JSON encoded array which will hold the newly created image media id.

Call example:

```
$file = "http://sample.image.source.com/image/nice.jpg";
$imagedata = file_get_contents($file);
$encoded = base64_encode($imagedata);
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/image/8500');
 curl_setopt($curl, CURLOPT_HTTPHEADER,
     array(
         "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
     )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_CUSTOMREQUEST,"PUT");
curl_setopt($curl, CURLOPT_POSTFIELDS, $encoded);
curl_setopt($curl, CURLOPT_HTTPHEADER, array("Content-length: ".strlen($encoded)));
$response = curl_exec($curl);
// process JSON response here
```

Response example:

```
{"MEDIAID":"8039"}
```

Delete product image:

Header type: **DELETE**
This function will delete the product's source image and it's size variations, and delete the media elements' datas from the system permanently. **This action cannot be reversed!**

This call will result in a JSON encoded array, which will hold the deleted picture's media id.

Call example:

```
$postdata = array(
  "MEDIAID" => "582035",
);
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/image/412536');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_CUSTOMREQUEST,"DELETE");
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
$response = curl_exec($curl);
// process JSON response here
```

Response example:

```
{"MEDIAID":"582035"}
```

Attribute API: (beta version)

The attribute api consists of two sub modules.
- The category module where you can manage the label categories which describes the labels' groups.
- The label module where you can manage the labels.

Note: You need to create the categories before creating the actual labels.

Attribute category:

Attribute category errors:
701 - attribute category not found
702 - attribute category already exists
703 - not allowed category data type

704 - not allowed category data filterable value
705 - not allowed category data unit value
706 - attribute category not deleted
707 - not allowed category data interval value

Available fields:

Field	Description	Header type
CATEGORYNAME	Attribute category name. **Cannot be empty!**	GET / POST
CATEGORYTYPE	Attribute category type.	GET / POST
CATEGORYDATA [TYPE]	Attribute category data array's type field. It describes the category's elements type. **Accepted values:** - boolean - char - checkbox - number - radio - select - slider	GET / POST
CATEGORYDATA [FILTERABLE]	Attribute category data array's filterable field. It describes, if the category's elements can be filtered. **Accepted values:** - 0 - 1	GET / POST
CATEGORYDATA [UNIT]	Attribute category data array's unit field. It describes the category's unit. **Accepted values:** - miliméter - centiméter - méter - miligramm - gramm - dekagramm - kiló - mililiter - centiliter - deciliter - liter - db - pár - A	GET / POST

	- V - mWh - kWh - W - Hz - kHz - MHz - GHz - bit - B - kB - MB - GB - TB	
CATEGORYDATA [INTERVALMIN]	Attribute category data array's interval minimum field. It describes the category's filter slider minimum value if used. **Accepted values:** - **numeric (int, float)**	GET / POST
CATEGORYDATA [INTERVALMAX]	Attribute category data array's interval maximum field. It describes the category's filter slider maximum value if used. **Accepted values:** - **numeric (int, float)**	GET / POST
CATEGORYDATA [INTERVALSTEP]	Attribute category data array's inteval step field. It describes the category's filter slider change step value if used. **Accepted values:** - **numeric (int, float)** - **must be positive** -	GET / POST

API get category list:
Header type: **GET**
This function will list the available attribute categories which exists in the webshop.

This call will result in a JSON encoded array which will hold the response data.

- **count:** which is the maximum quantity of the categories in the webshop
- **items:** categories list
- **prev:** previous page link if using limit smaller than the maximum count e.g.: api/attribute/category?limit=10&page=1
- **next:** next page link if using limit smaller then the maximum count e.g.: api/attribute/category?limit=10&page=3

Call example:

```
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/attribute/category');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($curl);
// process JSON response here
```

Response example:


```
{"count":21,"items":{"0":"10","1":"11","2":"30","3":"32","4":"33","5":"35","6":"37","7":"39","8":"41","9":"44"},
"next":"\/api\/attribute\/category?limit=10\u0026page=2"}
```


API get category:
Header type: **GET**
This function wil retrive the attribute category's data set by the given category id.

This call will result in a JSON encoded array which will hold the category data set.

Call example:

```
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/attribute/category/30');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($curl);
// process JSON response here
```

Response example:


```
{"CATEGORYID":"30","CATEGORYNAME":"prodmaincategory","CATEGORYTYPE":"0","CATEGORYDATA":{},"ITEMS":{"0":"99680","
1":"99679","2":"105163"}}
```


API insert category:
Header type: **POST**

This function will insert an attribute category into the webshop if its not exists yet.

This call will result in a JSON encoded array which will hold the crated attribute category's category id.

Call example:

```php
$postcategory = array(
  "CATEGORYNAME" => "Test label category",
  "CATEGORYTYPE" => "0",
  "CATEGORYDATA" => array(
    "TYPE" => "number",
    "FILTERABLE" => "1",
    "UNIT" => "db",
    "INTERVALMIN" => "0",
    "INTERVALMAX" => "100",
    "INTERVALSTEP" => "1",
  ),
);

$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/attribute/category');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postcategory));
$response = curl_exec($curl);
// process JSON response here
```

Response example:

```
{"CATEGORYID":9866}
```

API update category:
Header type: **POST**

This function will update an existing attribute category data set if its already exists.

This call will result in a JSON encoded array which will hold the updated attribute category's category id.

Call example:

```php
$postcategory = array(
  "CATEGORYNAME" => "Test label category update",
  "CATEGORYTYPE" => "0",
  "CATEGORYDATA" => array(
    "TYPE" => "char",
    "FILTERABLE" => "0",
    "UNIT" => "db",
    "INTERVALMIN" => "0",
    "INTERVALMAX" => "1000",
    "INTERVALSTEP" => "10",
  ),
);

$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/attribute/category/9866');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postcategory));
$response = curl_exec($curl);
// process JSON response here
```

Response example:

```
{"CATEGORYID":9866}
```

API delete category:
Header type: **DELETE**

This function will delete an existing attribute category and is elements permanently! **This action cannot be reversed!**

This call will result in a JSON encoded array which will hold the deleted attribute category's category id.

Call example:

```php
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/attribute/category/9866');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_CUSTOMREQUEST,"DELETE");
$response = curl_exec($curl);
// process JSON response here
```

Response example:

```
{"CATEGORYID":9866}
```

Attribute label:

Attribute label errors:
720 - attribute label not found
721 - attribute label already exists
722 - not found postdata category id
723 - not found postdata label name
724 - attribute label not deleted

Available fields:

Field	Description	Header type
LABELNAME	Label name. **Cannot be empty!**	GET / POST
CATEGORYID	Label attribute category id. **Must be existing attribute category id!**	GET / POST
PARENTID	Label parent id. It describes	GET / POST

	that which label is the parent of the current element. This field is used for categories to represent Parent children relations, other labels does not use it.	
LABELCONTENT	Label description. Can be HTML formatted. **(optional)**	GET / POST
LABELSKU **(beta)**	Label unique id. It is used to identifiy a single label by its unique id. **Accepted value is 100 chars long varchar.**	GET / POST
ITEMSORT	Label order number.	GET / POST
ACTIVE	Label active field. **Accepted values:** - 0 - 1	GET / POST

API get label list:
Header type: **GET**
This function will list the available attribute labels which exists in the webshop.

This call will result in a JSON encoded array which will hold the response data.

- **count:** which is the maximum quantity of the labels in the webshop
- **items:** labels list
- **prev:** previous page link if using limit smaller than the maximum count e.g.: api/attribute/label?limit=10&page=1
- **next:** next page link if using limit smaller then the maximum count e.g.: api/attribute/label?limit=10&page=3

Call example:

```
1.  $curl = curl_init();
2.  curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/attribute/label');
3.  curl_setopt($curl, CURLOPT_HTTPHEADER,
4.      array(
5.          "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
6.      )
7.  );
8.  curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
9.  $response = curl_exec($curl);
10. // process JSON response here
```

Response example:


```
{"count":1449,"items":{"0":"7712","1":"7713","2":"7717","3":"7721","4":"7722","5":"7726","6":"7727","7":"7730","
8":"7731","9":"7736"},"next":"\/api\/attribute\/label?limit=\u0026page=2"}
```


get label list other usages:
You can add parameters to the label list call.

- **search** : with this parameter you can search in the label name field. The search will retrieve label ids for all those labels where you can find the given phrase as an exact match, or a substring. If the given phrase not found, the api will return with an error (720).
  example url: `http://sample.domain.hu/api/attribute/label?search=sample`
- **get** : with this parameter you can search in the label sku field. The search will give you an exact match. If the given phrase not found, the api will return with an error (720).
  example url: `http://sample.domain.hu/api/attribute/label?get=sample`

API get label:
Header type: **GET**
This function wil retrive the attribute label's data set by the given label id.

This call will result in a JSON encoded array which will hold the label's data set.

Call example:

```php
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/attribute/label/7736');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($curl);
// process json response here
```

Response example:


```
{"LABELID":"7736","PARENTID":"7713","CATEGORYID":"30","CATEGORYNAME":"prodmaincategory","LABELNAME":"Kiegészítők
","LABELCONTENT":"","ITEMSORT":"0","ACTIVE":"1","CREATEDATE":"2014-07-29 15:02:59"}
```


API insert label:
Header type: **POST**
This function will insert an attribute label into the webshop if its not exists yet.

This call will result in a JSON encoded array which will hold the crated attribute label's label id.

Call example:

```php
$postlabel = array(
  "LABELNAME" => "Teszt api insert label",
  "CATEGORYID" => "135",
  "PARENTID" => "0",
  "LABELCONTENT" => "<div>Teszt label content</div>",
  "ITEMSORT"=>"0",
  "ACTIVE" => "1",
);

$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/attribute/label');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postlabel));
$response = curl_exec($curl);
// process JSON response here
```

Response example:

```
{"LABELID":19595}
```

API update label:
Header type: **POST**
This function will update an existing attribute label data set if its already exists.

This call will result in a JSON encoded array which will hold the updated attribute label's label id.

Call example:

```php
$postlabel = array(
  "LABELNAME" => "Teszt api update label",
```

```
  "CATEGORYID" => "135",
  "PARENTID" => "0",
  "LABELCONTENT" => "<div>Teszt label update content</div>",
  "ITEMSORT"=>"1",
  "ACTIVE" => "0",
);

$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/attribute/label/19595');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postlabel));
$response = curl_exec($curl);
// process JSON response here
```

Response example:

```
{"LABELID":19595}
```

API delete label:
Header type: **DELETE**
This function will delete an existing attribute label permanently! **This action cannot be reversed!**

This call will result in a JSON encoded array which will hold the deleted attribute label's label id.

Call example:

```
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/attribute/label/19595');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_CUSTOMREQUEST,"DELETE");
$response = curl_exec($curl);
```

```
11. // process JSON response here
```

Response example:

```
{"LABELID":19595}
```

Coupon API: (beta version)

Coupon api errors:
501 - not found coupon template name
502 - not found coupon type
503 - incorrect coupon type
504 - not found coupon value
505 - incorrect coupon value
506 - not found coupon code
507 - incorrect expiry days value
508 - incorrect categoryid value
509 - incorrect brandid value
510 - incorrect productid value
511 - incorrect allowondiscount value
512 - incorrect userid value

514 - incorrect cartminimum value
515 - incorrect couponuses value
516 - coupon code already exists

Available GET/POST fields:

Field	Description	Header type
TEMPLATENAME	Coupon name, this field is **required** to create a cupon template. **Cannot be empty!**	GET, POST
COUPONTYPE	Coupon type, this field is **required** to create a coupon template. **Cannot be empty! Accepted values (FIX, PERCENT)**	GET, POST
COUPONVALUE	Coupon value, this field is **required** to create a coupon template! **Cannot be empty or negative!**	GET, POST
COUPONCODE	Coupon unique code, this field is **required** to create a coupon **if COUPONFIXCODE is not used**.	GET, POST

COUPONFIXCODE	Coupon fix code, this field is **required** to create a coupon **if COUPONCODE is not used**. It is used with COUPONUSES > 1 if we want to create a reusable coupon. **If given at the same time as COUPONCODE, this will be used!**	GET, POST
CREATEDATE	Coupon use start date, must be in date format (Y-m-d). **Required** to create a coupon template!	GET, POST
EXPIRYDATE	Coupon use end date, must be in date format (Y-m-d). It is used to set the expiry date of the coupon. **Optional.**	GET, POST
EXPIRYDAYS	Coupon expires in given days computed from coupon creation time in the system. **Owerwrites EXPIRYDATE!** **Optional.**	GET, POST
USERID	User id, it can be sent to set a unique user for the coupon. **Cannot be negative!** **Optional.**	GET, POST
USEREMAIL	User email, it can be sent to set a unique user for the coupon. **Optional.**	GET, POST
CATEGORYID	Category id, it can be sent to set a category for the coupon to be used on. **Cannot be negative!** **Optional.**	GET, POST
BRANDID	Brand id, it can be sent to set a brand for the coupon to be used on. **Cannot be negative!** **Optional.**	GET, POST
PRODUCTID	Product id, it can be sent to set a product for the coupon to be used on. **Cannot be negative!** **Optional.**	GET, POST
ALLOWONDISCOUNT	Coupon can be used with	GET, POST

	discounts. **Cannot be empty! Accepted values (true,false)** **Optional!**	
COUPONUSES	Coupon use count. It sets how many times a coupon can be used! default value is 1. **Optional!**	GET, POST
CARTMINIMUM	Coupon minimum cart limit. It sets the limit after the coupon can be used in the cart. **Cannot be negative!** **Optional!**	GET, POST
COUPONNOTE	Coupon note. It sets a small text message for the coupon. **Optional!**	GET, POST

Send coupon into Webshop API:
Header type: **POST**

This function is used to create a coupon in the webshop. The call will result in a JSON encoded array which will hold the created coupon id!

Call example:

This sample will create a coupon with the name “Teszt API template name”. The coupon will be fix 500 valued with the code “QWE123RTZ321”. The coupon can be used once between 2015-11-10 10:00:00 and 2015-11-30 10:00:00 by the user with the email address “sample@somedomain.com”

```
$coupon_data = array(
  "TEMPLATENAME" => "Teszt API template name",
  "COUPONTYPE" => "FIX",
  "COUPONVALUE" => "500",
  "COUPONCODE" => "QWE123RTZ321",
  "CREATEDATE" => "2015-11-10",
  "EXPIRYDATE" => "2015-11-30”,
  "USEREMAIL" => "sample@somedomain.com",
  "COUPONUSES" => "1",
);
```

```php

$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.com/api/coupon');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($coupon_data));

$response = curl_exec($curl);
// parse JSON response here
```

Additional examples for the CorinCloud API:

How to set grouped product color variants with different sizes:

Header type: **POST**

Eg.: You have a product with 3 colors (red, blue, green) and each color has different size variants.

Theory: You need to insert each color as unique products, connected by and identifier (GROUPCODE), and then you need to insert variants for each color.

First step, insert color as unique product:

```php
$postproduct = array(
  'SKU'=>"123ASD_GREEN",
  'PRODUCTNAME' => 'Green api product',
  'GROUPCODE'=>'API_TEST_GROUP',
  'STOCK' => 23,
  "ACTIVE"=>1,
  "TAGS"=>array(
     'COLOR'=>array(
        'EVENT' => 'add',
        'ITEMS' => array(
            'blue'
        )
```

```
    ),
);
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.com/api/product');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postproduct));
$response = curl_exec($curl);
// process JSON response here
// You will need to save the product id from the response to use it to insert the size variations to this product in the second step
```

Second step, insert size variants for product:
You should do this step until you finish processing the sizes for this color.

```
$postvariant = array(
  'PRODUCTID'=>1230, // product id from first step
  "LABELCATEGORY1"=>"Size", // label category, this is the type of the label
  'LABELNAME1' => "41", // label name, this is the actual size
  'STOCK' => 9, // variant unique stock value
  "ACTIVE"=>1, // variant should be active to show it on the front end
  "DEFAULT"=>1, // with this, you can set the product main variant, optional
);
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.com/api/variant');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postvariant));
$response = curl_exec($curl);
// process JSON response here
// it will be the created variant id
```

After you finished with the size variations for this color, you should proceed to the next color and do the steps all over again, until you process all colors for your product. **You should use the same GROUPCODE for all colors.**

If you set all of your colors and variations for this product, you can proceed to the nex product.

How to add, update or delete product attributes / tags:

Header type: **POST**
The TAGS field is used to manage product attributes.

It is an array, which holds sub arrays of label types (COLOR, MATERIAL etc). **Each sub array** consists of 2 fields.
EVENT: is a string and it sets the way that how the api will process the attributes

- **add**: appends the attributes of a specific type (eg. COLOR) to the product
- **update**: completely replaces the attributes of a specific type (eg. COLOR) (deletes the prevoius entries and appends the new ones)
- **delete**: completely deletes the attributes from a specific type (eg. COLOR) from the product

ITEMS: is an array of string which holds the attribute names, (**note:** on delete event, it is not needed to be sent to the api)

Call example:

```php
$postproduct = array(
  'SKU'=>"ct_test",
  "TAGS"=>array(
    'COLOR'=>array(
       'EVENT'=>'add',
       'ITEMS'=>array(
         'blue',
         'red',
       )
    ),
    'MATERIAL'=>array(
      'EVENT'=>'add',
      'ITEMS'=>array(
        'cloth',
    )
  )
);

$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product/0123');
```

```
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postproduct));
$response = curl_exec($curl);
// process JSON response here
```

To update a product tags / attributes you shoud set the EVENT field value to **update** and send the new values which you want to set for the product in the **ITEMS** field
Reminder: update event removes all previous attributes of a specific type (eg. COLOR)

Parameters of the **update** event

```
$postproduct = array(
  'SKU'=>"ct_test",
  "TAGS"=>array(
    'COLOR'=>array(
       'EVENT'=>'update',
       'ITEMS'=>array(
         'green'
       )
    ),
    'MATERIAL'=>array(
      'EVENT'=>'update',
      'ITEMS'=>array(
        'silicon compozite',
    )
  )
);
```

To delete a product tags / attributes you should set the EVENT field to **delete** and call the the api.
Reminder: the ITEMS field is not needed on this EVENT, so it shouldn't be in the parameters of your call.

Parameters of the **delete** event

```
$postproduct = array(
  'SKU'=>"ct_test",
  "TAGS"=>array(
    'COLOR'=>array(
```

```
        'EVENT'=>'delete'
    ),
    'MATERIAL'=>array(
      'EVENT'=>'delete'
  )
);
```

How to add/update/delete categories via tags field for product:

Header type: **POST**

You can set already existing categories for your product by providing the needed category ids. You can GET these id-s via requesting a product datas, the information is in the [TAGS] [CATEGORY] field. You will need the ids of the categories to work with.

As you can see below, the CATEGORY array consists of the same fields as the other TAGS subarrays (eg.: COLOR etc.). The usage is the same too. If You want to **add** categories to your product, You just need to provide the category ids that you want to append for the product. If you want to **update** (completely replace) the categories, you need to provide all the category ids for the product. Also You can **delete** the previously assigned categories from your product. It depends on the **EVENT** field in the example.

NOTE: This type of category assignment DO NOT create new categories, because it needs already existing category ids to work.

Example (add):

```
$postproduct = array(
  'SKU'=>"ct_test",
  "TAGS"=>array(
    'CATEGORY'=>array(
       'EVENT'=>'add',
       'ITEMS'=>array(
         '0123',
         '0234',
       )
    )
  )
);

$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/product/0123');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
```

```
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postproduct));
$response = curl_exec($curl);
// process JSON response here
```

Example (update):

with these parameters you can completely replace the previous category assignments for the product

```
$postproduct = array(
  'SKU'=>"ct_test",
  "TAGS"=>array(
    'CATEGORY'=>array(
       'EVENT'=>'update',
       'ITEMS'=>array(
         '0345'
       )
    )
  )
);
```

Example (delete):

with these parameters you can delete the previous category assignments from the product

```
$postproduct = array(
  'SKU'=>"ct_test",
  "TAGS"=>array(
    'CATEGORY'=>array(
       'EVENT'=>'delete'
    )
  )
  );
```

ORDER API (beta version)

Order api errors:
302 - error storing order to database
350 - order not found
351 - can’t create order
352 - can’t create order item
353 - order id not found

GET SPECIFIC ORDER

This function will get the data of a specific order by id from the webshop.

Header type: **GET**
This call will result in a JSON encoded array, which will hold the requested data (all available fields) for this specific order.

Parameters:

- none

Call example:

```php
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'https://sample.domain.hu/api/order/6021);
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($curl);
// process json $response here
```

Response example:

```
'{
    "order_id": 6021,
    "order_number": "20220823181337",
    "user_id": 0,
    "user_external_id": "",
    "user_email": "sample@sample.com",
    "addresses": "",
    "shipping_mode": "courier",
    "payment_mode": "cash",
    "product_cost": "0.00",
    "coupon_cost": "0.00",
    "quantity": 1,
    "shipping_cost": "0.00",
    "discounts": "0.00",
```

```
    "remark_cost": "0.00",
    "sum_cost": "10990.00",
    "currency": "Ft",
    "status": 4,
    "createdate": "2022-08-23 18:13:37",
    "modifydate": "2022-08-24 21:15:32",
    "order_note": "",
    "admin_order_note": "",
    "package_number": "",
    "shipment_data": "",
    "send_shipment": "0000-00-00 00:00:00",
    "delivery_date": "0000-00-00",
    "shipping_date": "0000-00-00",
    "invoiced": 0,
    "conversion": 0,
    "deleted": 0,
    "blocked": 0,
    "blacklist": 0,
    "registered": 0,
    "source": 11,
    "lastsource": 10,
    "subscribed": "",
    "items": {
        "2893": {
            "order_item_id": "2893",
            "order_id": "6021",
            "item_number": "1",
            "item_type": "0",
            "blocked": "0",
            "product_id": "58751",
            "var_id": "52219",
            "item_sku": "1091264476",
            "item_name": "Vaude Wo Manukau polár felső dusty rose (Méret: 42)",
            "product_quantity": "1",
            "outofstock": "0",
            "instock": "0",
            "completed": "0",
            "deleted": "0",
            "normal_price": "20990.00",
            "actual_price": "20990.00",
            "discount": "10000.00",
            "sum_price": "10990.00",
            "currency": "HUF",
            "product_data":
"{\"product_sku\":\"1091264475-MAIN\",\"modelid\":\"1091264475-MAIN\",\"var_sku\":\"1091264476\"
}",
            "createdate": "2022-08-23 10:10:10",
            "modifydate": "0000-00-00 00:00:00",
            "product_variant": ""
        }
    },
    "user": {
        "user_email": "sample@sample.com"
    }
```

```
}'
```

INSERT NEW ORDER

This function will insert a new order to the webshop.

Header type: **POST**
This call will result in a JSON encoded array which will hold the result data set.

Parameters:

- none

Call example:

```php
$postdata = array(
    "order_number" => "TESTEXTID-1819195",
    "createdate" => "2022-08-24 22:37:00",
    "user_email" => "abc@example.com",
    "status" => 1,
    "payment_mode" => "cash",
    "shipping_mode" => "courier",
    "package_number" => 111,
    "currency" => "Ft",
    "items" => array(
       array(
            "item_number" => 1,
            "item_sku" => "1091264476",
            "var_sku" => "1091264476",
            "item_name" => "Vaude Wo Manukau polár felső dusty rose (Méret: 42)",
            "product_quantity" => 1,
            "normal_price" => 20990,
            "actual_price" => 20990,
            "discount" => 10000,
            "createdate" => "2022-08-24 22:34:00",
       ),
    ),
    "order_name" => "Teszt Elek",
    "order_phone" => "+36201234567",
    "shipping_address" => array(
       "full_name" => "Teszt Elek",
       "phone" => "+36201234567",
       "country" => "Magyarország",
```

```
        "county" => "Bács-Kiskun",
        "zip" => "6000",
        "city" => "Kecskemét",
        "street_address" => "Fő u. 1",
    ),
    "billing_address" => array(
        "full_name" => "Teszt Elek",
        "phone" => "+36201234567",
        "country" => "Magyarország",
        "county" => "Bács-Kiskun",
        "zip" => "6000",
        "city" => "Kecskemét",
        "street_address" => "Fő u. 1",
        "taxcode" => "1234567-8-90",
    ),
);

$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/order');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
$response = curl_exec($curl);
// process json response here
```

Response example:

```
'{
  "status": 0,
  "message": "successful",
  "data": 6024
}'
```

UPDATE ORDER

This function will update a specific order in the webshop.

Header type: **POST**

This call will result in a JSON encoded array which will hold the result data set.

Parameters:

- none

Call example:

```
$postdata = array(
    "order_id" => 6024,
    "order_number" => "TESTEXTID-1819195",
    "createdate" => "2022-08-24 22:37:00",
    "user_email" => "abc@example.com",
    "status" => 1,
    "payment_mode" => "cash",
    "shipping_mode" => "courier",
    "package_number" => 111,
    "currency" => "Ft",
    "items" => array(
       array(
            "item_number" => 1,
            "item_sku" => "1091264476",
            "var_sku" => "1091264476",
            "item_name" => "Vaude Wo Manukau polár felső dusty rose (Méret: 42)",
            "product_quantity" => 1,
            "normal_price" => 20990,
            "actual_price" => 20990,
            "discount" => 10000,
            "createdate" => "2022-08-24 22:34:00",
       ),
    ),
    "order_name" => "Teszt Elek",
    "order_phone" => "+36201234567",
    "shipping_address" => array(
       "full_name" => "Teszt Elek",
       "phone" => "+36201234567",
       "country" => "Magyarország",
       "county" => "Bács-Kiskun",
       "zip" => "6000",
       "city" => "Kecskemét",
       "street_address" => "Fő u. 1",
```

```
    ),
    "billing_address" => array(
       "full_name" => "Teszt Elek",
       "phone" => "+36201234567",
       "country" => "Magyarország",
       "county" => "Bács-Kiskun",
       "zip" => "6000",
       "city" => "Kecskemét",
       "street_address" => "Fő u. 1",
       "taxcode" => "1234567-8-90",
    ),
);

$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://sample.domain.hu/api/order');
curl_setopt($curl, CURLOPT_HTTPHEADER,
    array(
        "Authorization: Basic ".base64_encode($api_username.":".sha1($api_username.$api_password)),
    )
);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, http_build_query($postdata));
$response = curl_exec($curl);
// process json response here
```

Response example:

```
'{
  "status": 0,
  "message": "successful",
  "data": 6024
}'
```