ValueChecker API Implementation Guide

Version	Date
0.20.70	March 28th, 2024

API Change	Description
`(response)`	The `category` and `cid` parameters have been deprecated in the `/product_search` response. `client_category` and `client_category_id` have been added.

Additional client category information will now be returned in the extended search.

Read more about Extended Search

- `(response)` `/prices` returns the `client_category` and `client_category_id` of the claimed product. The `cid` field of the claimed product has been deprecated.

Additional client category information will now be returned in the replacement search.

Read more about Replacement Search

PDF download Google docs

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

Index

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

Introduction

Before Getting Started

ValueChecker Service Concepts

ValueChecker is a product appraisal service that includes product identification and Like-Kind-and-Quality replacement suggestions. Using advanced heuristic, matching and data analysis technology we are able to provide superior and faster results than is humanly possible to get using traditional web searching.
We recommend you get in touch with your ValueChecker contact person and request demo credentials for our web GUI as the service concepts described below are presented there in an intuitive manner.

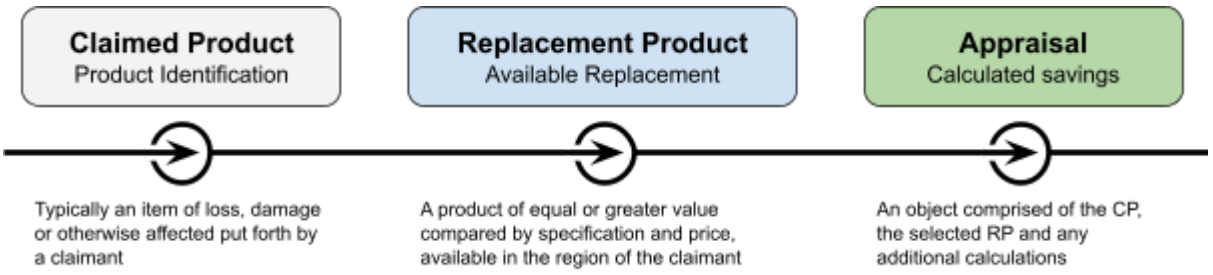
Users and Policyholders

Users are normally the authenticated Claim Handlers / Adjusters who interact with the ValueChecker API. Traditionally, they are doing so on behalf of a third party (such as an Insurance company or a Claim Handling company).
In the case where the ValueChecker is being implemented into a frontend solution (such as First Notice of Loss web forms or mobile apps), the Policyholders are the ‘user’.

Sequential Task Flow

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

The ValueChecker API is a wrapper on top of a robust system of data aggregation, analysis, comparison and matching technology. To use this system in its most powerful form and realize the maximum cost savings this system can provide it should be used in a sequential manner. Each subsequent action adds further value to the flow.



Claimed Product: CP

To deliver the bestLike-Kind-and-Quality replacement suggestions, ValueChecker needs to know what is being claimed (lost, broken or replaced for other reasons). The higher the level of accuracy in describing or selecting the CP, the greater the precision in the replacement suggestions.
For the vast majority, every user action for an individual product should always begin with a query to the Product Search endpoint.

Core Categories

ValueChecker has a curated catalog for many products in the categories of the most claimed products where we aggregate product attributes and apply Product Naming logic.
These categories are referred to as Core Categories. They can include, and are not limited to (depending on the market country):

- Phones
- Tables
- Laptops
- TVs
- Cameras
- Earphones & Headphones*
- Smart watches*
- Dishwashers
- Washing machines
- Hobs
- Bikes
- GoPro's & Video Cameras

**Only popular brands*
ValueChecker is capable of finding almost any standard product in a majority of categories. The only difference is the higher accuracy and correctness of the product information in the Core Categories.

Master Product

Master Products is a normalized group of products that are extremely similar to each other. This usually accounts for product color variations or other trivial deviations that make no difference to the price or viability of the product as replacement candidate.

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

If a claimed product can be matched to one of these Master Products, ValueChecker will return the product name by its non-attributive, normalized name (i.e. *Samsung Galaxy S9 64GB (2018)* instead of *Samsung Galaxy S9 Dual Sim 64GB Black*).
When available, this name is in the `/prices` response in the key:
`grouped_prices.[0].[0].product.product_name`
(0 and 0 being the first (key) product of the first (row) group list in the `grouped_prices` *list for the lowest available price from a recommended shop)*

Replacement Product: RP

At the heart of the ValueChecker service is our Replacement Product technology. It calculates not only what would be suitable Like-Kind-and-Quality replacement products for the claimed product (using our internal product identifier, the PID) but also where it is available for sale and at what price in the country of claim.
The Replacement Product endpoint requires a PID to perform these calculations.

Appraisals

An Appraisal is a dataset of a CP, an RP and any calculations performed on this pair of products. Calculations can be made on:

- metadata attached to the Appraisal (such as date of purchase and/or date of claim to calculate depreciation)
- metadata derived from the product (such as the category of CP, cost value of the RP), or a number of manual metadata manipulations.

Appraisals can be grouped together in a cart with a specific Reference that can be defined by your implementation (and/or the user).
Appraisals can have downloadable reports generated either at the Appraisal single object level or group 'cart' level (multiple Appraisal Objects).

ValueChecker API Essentials

Dynamic Settings

The ValueChecker service is country specific and has many customization and flexible business options making the service client specific.
It is important to build your implementation with this in mind. Settings and access tokens (which are minted per client) should be easy to change in value and between different clients who may be using the same software implementation.
Passing correct query variables with the wrong token will gain you access to the API response but it will be empty in most cases. Please take care in managing the tokens and corresponding settings and configurations.

Client Features

There are a number of features that are available to tailor the ValueChecker API response and results to your needs. Most of these will be set by your ValueChecker representative, but the following settings can be requested to be toggled ON/OFF:

Filtered Prices

Filtered prices are removed from the RP response.
Default is TRUE

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

Category Correction

Requested categories are analyzed using prediction and switched when better results can be found in the CP response.
Default is TRUE

Clustered Prices

Prices in non-core categories are aggregated into price buckets and returned in the CP response.
Default is TRUE

Category Metadata

Category information is returned in the CP response as a metadata object.
Default is TRUE for new clients. FALSE for existing clients until they migrate.

Client Categories

One of ValueCheckers strengths is its ability to adapt to many types of category configurations.
Some clients may use 2 categories of 'Telephones' and 'Tablets' while others may introduce a third middle range screen size as 'Phablets' between the 2.
To adapt to this, we maintain a separate category taxonomy for each client. Categories are represented by an alphanumeric string (`client_category`) i.e. "Mobile phones"and a corresponding integer category identifier (`client_category_id`) i.e. 12345.

To use the ValueChecker API you can pass either the `client_category` (recommended) or the `client_category_id`. E.g, if the client category "Tablets" has a client category id 123, then it will be equivalent to send a request using either `client_category_id=123` or `client_category=tablets`.

client_category usage

Pass the category to both the Claimed Product and Replacement Product endpoints as a formatted string.

- The string is not case-sensitive
- Special alphabetic characters like "ø" are converted to their ASCII equivalent, in this case "o".
- Spaces and other special characters like "&" and "/" are disregarded, so the following examples are equivalent:
    - "Audio / Video" and "Audio & Video"
    - "Midi/Minisets" and "Midi / Minisets"
    - "Make-Up - Aangebroken" and "Make-Up (Aangebroken)"
    - "Airconditioning (mobiel)" and "Airconditioning - Mobiel"

This method also holds the following advantage: any client category that does not have a mapping (client_category_id) will be automatically processed. There is no need to wait until the *`client_category_id`* is issued, making the implementation process simpler and faster.
How this works:

1. You use a *`client_category`* that has never been passed to the valuechecker API before: Example: *"Android Tablets"*

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

2. The API identifies it as a new category and creates a new mapping for this category using smart analysis (the extracted word "*Tablets"* is equal to our core category *"Tablets"*).
    a. If for some reason a mapping cannot be automatically assigned, the request will continue in the generic category of "All Other Products" until our QA team can manually ascertain the mapping.
3. That mapping (your category of *"Android Tablets"* to our core category *"Tablets"*) gets a `client_category_id` (i.e. 123) and is available in the next request to the Client Categories endpoint.
4. You can then continue using the `client_category` or the `client_category_id` for future requests to this category.

client_category_id usage

To use the ValueChecker API with the identifier you must always pass your `client_category_id` along with the request query to both the Claimed Product and Replacement Product endpoints.
It is absolutely required that these values are correct. Passing an incorrect or outdated `client_category_id` will yield no results!
We provide a Client Categories endpoint where you can fetch these values on demand to store them in a dynamic manner. These identifiers do not change often, but it is good practice to keep them regularly updated in an automated way.

All Other Products

We maintain a generic category named "All Other Products". It may be called a different name according to your locale. You can find it in the Client Categories endpoint.
We recommend if you do not know the category in a particular search to omit sending any category information as our category prediction will find the best result based on analysis of the search string.
If you are not able to omit the category parameter, we recommend the user be able to enter the category as best as they can describe it using the client_category (string) parameter.
In the last resort cases you may use this category for any searches where you are not sure of the category or cannot find a suitable category or do not wish to allow the user to define their category.

Session Identifiers

A session identifier is a unique string passed as a parameter in certain requests that can be used by ValueChecker to aggregate actions by a single end user (i.e. a user of your integration). This string can be:

- a hashed value of a username or account id from your integration
- an application or framework assigned session id
- any other uniquely generated string - per end user

We use this identifier in our system health checks, anonymized usage analysis, and to help detect anomalies in your integration.

Multi-Client Users (MCU)

If a user account is MCU enabled, you may use that user accounts API key to call data for any client that the user has permission to access.

- Use the parameter `client_id` to specify which client you are requesting data from
- If you request a client that you do not have access to, you will receive an error response
- If your user is MCU enabled, the client_id is a required parameter on most endpoints. It is good practise to include the client_id in all your requests by default so that you do not need to make code change when enabling/disabling MCU
- If you send the parameter `client_id` when not MCU enabled, it is ignored with a 200 response
- If your user should be MCU enabled but is not, pls contact support@valuechecker.ai

Minimal Implementation

1. Search by a `query` string and a category to identify the Claimed Product (CP)

`{api_url_path}/query=iphone 11 64&client_category=Mobile%20Phones`

```
{ (excerpt)
    "product_name": "Apple iPhone 11 64GB",
    "pid": 1032089319,
}
```

2. Search using a `pid` to return Replacement Product (RP) suggestions and prices
   (Note: For locale pricing purposes, `country` is also a required variable)

`{api_url_path}/prices?pid=1032089319&country=nl`

```
{ (excerpt)
    "claimed_product": {
        "master_name": "Apple iPhone 11 64GB",
        "pic_url": "https://img.valuechecker.net/,sxDbsr...,
        "product_name": "Apple iPhone 11 64GB Wit"
    },
    "grouped_prices": [
      [
          {
                "currency": "EUR",
                "in_stock": true,
                "offer_url": "https://{offer_url}/abcde123456abf1234",
                "price_date": "2020-08-26 16:45:45",
                "product": {
                    "master_name": "Apple iPhone 11 64GB",
                    "pic_url": "https://img.valuechecker.net/,sxDbsr...,
                    "product_name": "Apple iPhone 11 64GB Wit",
                },
                "shop": {
                    "shop_name": "belsimpel.nl",
                },
               "total_price": 694.00
            }
        ]
    ]
}
```


International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

Get Started

** Note: All response examples are snippets. Some fields may be truncated and should be ignored..*

API URL Path

The HTTP path to the valuechecker API is:

```
https://api.valuechecker.net
```

There is only 1 accessible version and any changes to the path when new generations are deployed will be updated in the documentation accordingly.

Authentication

ValueChecker uses API keys for authentication.
You will receive an API key which you will provide on every request to the API. If you do not have a key, ask your ValueChecker contact.

To pass the API key in a request, use the “Authorization” header. Prefix the key with “apiKey ” Curl example:

```
curl -H "Authorization: apiKey zcwPhya.S23UR5cN....xFWLdKq2KRLX" -X GET
https://api.valuechecker.net/<endpoint>?query=param1&param2=40&param3=-1
```

You may request multiple API keys and call the ValueChecker API concurrently.

Note: If you are still using a (deprecated) OAuth token, see OAuth Authentication.

Query Parameter Requests

These calls will be used to get parameter values needed for other calls. Since we are continuously improving our service, we recommend that our clients call these endpoints once a month to account for changes or additions we make to the system.

Get client information

Notes: `client_id`, `country_code`, and `language` are the parameters needed for future calls. Other parameters in the response can be ignored.

Request

Method	URL
`GET`	/clients

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

Params	Value-Type	Example Values
`No params needed`		

Response

Status	Response

200	{
	"client_id": 2,
	"country_code": “nl”,
	"language": “nl”
	}

Get Client Categories

Notes: Client categories are not required if you intend to use category strings.
If you decide to use ids, the `client_category_id` values will be used in several other requests. These ids represent each of the product categories that we recognize, define and assign to products in our catalog (translated in your local language). For example, Laptops, Cell Phones, Washing Machines and Tablets are some of the categories that will be returned.

Request

Method	URL
GET	/client_categories

Params	Value-Type	Example Values
`No params needed`		

Response

Status	Response
200	{ "acv_disabled": true, "client_category_id": 2735, "has_replacement": false, "client_category": "All Other Products" },

```
    {
        "acv_disabled": false,
        "client_category_id": 2738,
        "client_category": "Bicycles",
        "cid": 43256,
        "has_replacement": false
    },
    {
        "acv_disabled": false,
        "client_category_id": 2739,
        "client_category": "Cameras",
        "cid": 4376,
        "has_replacement": true
    },
    {
        "acv_disabled": false,
        "client_category_id": 2740,
        "client_category": "Cell Phones",
        "cid": 4364,
        "has_replacement": true
    },
    {
        "acv_disabled": false,
        "client_category_id": 2737,
        "client_category": "Televisions",
        "cid": 4485,
        "has_replacement": true
    }
```


`acv_disabled`: Informs as to whether the category has depreciation calculations associated with it. Depreciation can be applied in the Calculate ACV call.

`client_category_id`: This is an identifier used to help group and categorize the millions of consumer products in our database. It will be used as a request parameter in several of our API calls.

`client_category`: This is the display name for each product category.

`cid`: The `cid` is the identifier that matches the `client_category_id` to ValueChecker's internal product categories. This value is for reference only and can be ignored.

`has_replacement`: Shows whether the `client_category_id` is considered a "**Replacement category**" by ValueChecker. "**Replacement Categories**" represent product groupings that ValueChecker has defined Like-Kind-and-Quality replacement logic for -- which means a comprehensive search to find relevant Replacement Products based on objective specification values can be completed for products in these categories. All `client_category_id`'s where `has_replacement` = **True** are considered "**Replacement Categories**". Where `has_replacement` = **False**, ValueChecker is still able to search for the product and find real-time pricing information; however, there is no ability to search for Replacement Products within these categories.

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

Step 1. Create Client Reference

While not strictly required to use the ValueChecker API, it is good practice to create a Reference record prior to every usage session. Think of it as a shopping cart where Appraisal records are collected and can be modified prior to being retrieved into your system. The claim reference number of a policyholder's claim can also be used as the Client Reference.

Note: If your implementation is only using the CP and RP steps, you may skip creating a reference.

Client references have a user defined label (`client_reference`) and a `client_reference_id`. The `client_reference_id` is used to interact with the ValueChecker API to identify a reference.

Request

Method	URL
`POST`	/client_reference

Params	Value-Type	Example Values
`client_reference`	`string`	Reference 321

Note: This must be unique to your client, otherwise we return an error. The response will contain the client_reference_id which is used in later calls.

Response

`200`	`12345`

1.1 List All References

To list your existing references (or 'Carts'), you can use the same endpoint. This returns all references belonging to the user the token belongs to.
No parameters are required

Request

Method	URL
`GET`	/client_reference

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

Params	Value-Type	Example Values
`No params needed`		

Response

Status	Response

200	<pre>{
	  "client_reference_id": 5000,
	  "client_reference": "A reference name",
	  "user_id": 12345,
	  "client_id": 123,
	  "create_time": "2021-02-23 14:58:33"
	},
	{
	  "client_reference_id": 5001,
	  ...</pre>

1.1 Display Specific Reference

You should have a `client_reference_id` from Step 1: Create Reference to use in the request path.

Request

Method	URL
`GET`	/client_reference/`{client_reference_id}`

Params	Value-Type	Example Values
`No params needed`		

Response

Status	Response

200	<pre>{
	  "client_reference_id": 5000,
	  "client_reference": "A reference name",
	  "user_id": 12345,</pre>

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

	<pre>"client_id": 123,
	"create_time": "2021-02-23 14:58:33"
	}</pre>

Step 2. Product Search (CP)

Search for products in our catalogs using a query string.

Category Prediction

The ValueChecker API is capable of predicting the correct category based on analysis of the search query string. Keep in mind this is highly dependent on the richness of the query string.

Examples of rich search queries:
2008 lg washing machine front loader
Wifi/4G Galaxy Watch Sport T-Buckle graphite

Though it is not required, we recommend you use a category identifier (`client_category_id`) or a category name (`client_category`) where possible (i.e. when a category picker is available so the user can select or define a category).
Read more about Client Categories

Category Correction

When a search is made to a requested category that the ValueChecker predicts with high certainty is the wrong category, a client feature (enabled by ValueChecker) automatically performs the search and returns results from the correct category.
As a feature you can request to enable/disable this automation correction depending on how your application user flow works (i.e. you wish to show the user potentially poor or no results from the requested category by design).
Additionally, if you have Category Correction enabled, you can pass an override parameter of `force_client_category` = `true` for specific search requests that will prevent any category analysis or switching.

2.1 Search Suggestions

For implementations that require fast or continuously updating results for a query search (examples: “search as you type”, instant display of related series/models, drop-down or tree population based on options etc..) you should call the `search_suggestion` endpoint.
It returns structured response data from our Master catalog at a faster rate than the extended search does. Additionally, calls to this endpoint have a much lower rate metering restriction. You can call it without any throttling.

For example: in a “search as you type” scenario, every keystroke (after the first X characters)

can query this endpoint and you will get a response with product data based on the updated string as fast as the network allows which your implementation can then update/render to the user.
This endpoint returns a case insensitive, strictly matched result. Misspellings and mistakes are not considered nor automatically corrected.

Every request searches for results except when the search query matches a brand name, or the beginning of a brand name. In this case, due to the low specificity of the search, no results will be returned. For instance: if the endpoint is called with the query "*Apple*" for the client category "Mobile Phones", the search query would match any Apple phone, so the chances of returning the product that is indeed being searched for is very low. Therefore, in this case, the search query needs to include at least two more characters after the brand name for the Search Suggestion endpoint to begin returning results. In the example above there will be iPhone results with a search query of "*Apple ip*".

A product category (`client_category_id` or `client_category` from client_categories) should ideally be selected by the user (we recommend a drop-down/list/other similar UI component be used for selection purposes) to be passed as a parameter in the call for the claimed product being searched for.
There is support for the user to also be allowed to define their own category which you pass to the API with the `client_category` parameter.

Important: This feature is only available for our catalog categories where we have pre-calculated replacement data. You can find this flag from the results of the `client_categories` endpoint where the category has a field and value of `"has_replacement": true`
Example category:

```
{
    "client_category_id": 11499,
    "client_category": "Smartwatches",
    "parent_client_category_id": 2427,
    "cid": 431051,
    "has_depreciation": true,
    "has_replacement": true,
    "acv_disabled": false,
    "show_closest_matches": false
},
```

Results Note: The result set may not be identical to the extended search results as we only query our Master products for this endpoint. We recommend you add functionality to allow a user to perform an extended search if they are not satisfied with these results or when we do not have any Master products matching their search query (empty resultset).

Language Note: The specification name (`spec_name`) values are currently only returned in lowercase English (please cast to the preferred format of your locale i.e. "Title Case" or "Camel Case" etc..).
Serving translations for the default locale of the requesting client (token) is under development and will be available in an upcoming non-breaking release.

Request

Method	URL
GET	/search_suggestion

Params	Value-Type	Example Values
query	string	“asus zenb”
client_category_id*	number	1234
client_category*	string	1234
force_client_category*	boolean	true

`query`: User-input string. The **user** should enter the name of the product being claimed / searched for.
`*client_category_id`: Optional correlating ID of the category selected by the **user**, or
`*client_category`: Optional category selected or defined by the **user**
`*force_client_category`: Optional flag to force the use of the requested category

Response

Status	Response
200	{ "results": [{ "product_name": "Asus ZenBook 13 UX325 (13.3-inch, 2020) Intel Core i5-1035G1 512GB 8GB", "pic_url": "https://img.valuechecker.net/,s3kV...jpg", "pid": 1069630315, "specs": [{ "spec_id": 999, "spec_name_display": "Brand", "spec_value_display": "Asus", "spec_value": "10", "spec_value_numeric": 1.0 }, ...] }, {

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

<table>
<tr><td></td><td><pre>
            "product_name": "ASUS Zenbook Flip UX362 (13.3-Inch,
2018) Series Intel Core i5-8265U 512GB 8GB",
            "pic_url": "https://img.valuechecker.net/,sF_TJh.jpg",
            "pid": 1052904939,
            "specs": []
        },
        ...
}
</pre></td></tr>
</table>

2.2 Extended Search

A product category identifier `client_category_id` or a category name `client_category` from client_categories should be selected by the user (we recommend a drop-down/list/other similar UI component be used for selection purposes) to be used in the call for the claimed product being searched for.

This action performs a sequence of tasks (such as auto-correction for spelling typos, matching for Master products etc..) and analyses a large set of potential results. It returns a dataset of potential products within a series or product family (for example “Apple iPhone 11” will return the different storage variants of the standard, Pro and Pro Max models) or non-aggregate data from a variety of data sources (general product data, web results etc..).

This endpoint is metered and should not be called in a continuously updating implementation (such as “search as you type” etc..).

Request

Method	URL
GET	/product_search

Params	Value-Type	Example Values
query	string	“asus zenphone 2015”
client_category_id*	number	1234
client_category*	string	1234
force_client_category*	boolean	true
session-id*	string	“ABCDE12345”

`query`: User-input string. The **user** should enter the name of the product being claimed / searched for.

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

`*client_category_id`: Optional correlating ID of the category selected by the **user**, or
`*client_category`: Optional category selected or defined by the **user**
`*force_client_category`: Optional flag to force the use of the requested category
*`session-id`: Optional session identifier of the **user**

Response

Important legacy note: This response formatted as an object has been introduced in API version 0.19.44. Clients using the list response (see deprecated section) will need to migrate to the response documented below.

Status	Response

200	<pre>{
	  "results": [
	    {
	      "child_products": [],
	      "pid": 1122795690,
	      ~~“cid”: 132~~,
	      ~~“category”: “Laptops”,~~
	      “client_category”: “Laptops”,
	      “client_category_id”: 2000,
	      "product_name": "Asus Zenfone 9 8GB 5G 128GB White",
	      "Product_picture_url": "https://img.valuechecker.net/,..j",
	      "specs": [
	        {
	          "spec_id": 999,
	          "spec_name_display": "Marke",
	          "spec_value_display": "Asus",
	          "spec_value_numeric": 1.0,
	          "spec_name": "Marke"
	        }],
	      "price": {
	        "price": 790.0,
	        "price_url": "https://www.belsimpel.nl/asus-zen...",
	        "shop_count": 1,
	        "price_date": "2022-12-06 15:58:52"
	    }],
	  "metadata": {
	    "category": {
	      “requested_client_category_id”: 1234,
	      "requested_client_category": "Laptops",
	      "prediction_certainty": 0.9978,
	      "predicted_client_category_id": 469,
	      "predicted_client_category": "Mobiele Telefoon",
	      "category_switched": true
	    },
	      "used_query": "asus zenfone 2015"
	}</pre>

Deprecation warnings:

- “cid“ has been deprecated and will be removed in a future update. Please use the client_category_id directly instead.
- “category” has been deprecated and will be removed in a future update. However, the “client_category” will return the same information.

If at least 1 `pid` is returned, use a `pid` to continue to Replacement Suggestions and Prices.

The response example also shows the Category Correction feature being activated where a query for "asus zenfone 2015" was performed in a requested category of “Laptops”. The certainty of 99% that this search should be performed in the category of Mobile phones ("Mobiele Telefoon" according to the client settings) triggered an automatic switch of categories for the search and response.

Reference Price

The Price returned here is a reference price. In many cases it is a near-live or recent price that ValueChecker has obtained from multiple internet sources (shops, search engines etc..). This price is not to be confused with the Replacement prices from the Replacement Suggestions and Prices.

Child Products

When ValueChecker finds products that are equivalent in technical specifications to a Master Product in our catalog but with minor variance in attributes (for example color), the response will consist of the Master Product with the variant products’ names in a nested list. The specifications of the child product(s) are identical to the Master Product with the only difference being in the product name. There is no difference in the replacement calculation.

Note: By default, the response contains an empty `child_products` key (value: empty list).

An example response for a search for “*iPhone 10 grey*”:

Status	Response

200	<pre>"product_name": "Apple iPhone X 64GB (2017)",
	"child_products": [
	"Apple - iPhone X 64GB - Space Gray",
	"Apple - iPhone X 64GB - Space Gray (AT&T)",
	"Apple - iPhone X 64GB - Space Gray (O2)",
	],
	"specs": [
	...</pre>

Aggregated Prices (a.k.a. Clustered Prices)

For searches in non-core categories, the ValueChecker will search many popular shopping APIs (Google shopping for example) and collect all prices that match the search query. After clustering the prices into 3 buckets of low, medium and high, the data is available in the response for cases where you want to show average prices to the application users.

An example response for a search for “*Levis bluejeans*”:
(note that no category information is requested)

`{api_url_path}/product_search?query=Levis bluejeans`

Status	Response (excerpt)

200	{
	"results": [
	{
	"pid": 1739972236,
	~~“cid”: 135~~,
	~~“category”: “Clothes”,~~
	“client_category”: “Clothes”,
	“client_category_id”: 3000,
	"product_name": "levi's blue jeans",
	"product_picture_url": "https://img.valueecker.net/...jpg",
	"price": {},
	"price_clusters": {
	"low": {
	"udc_id": 163332,
	"pid": 1739972236,
	"group": "low",
	"price": 51,
	"count": 7,
	"percentage": 0.28,
	"min_price": 39,
	"max_price": 61
	},
	"medium": {
	"udc_id": 163333,
	...
	},
	"high": {
	"udc_id": 163334,
	...
	}
	},
	"child_products": [],
	"child_product_contents": []
	}

The first item in the “results” object will have the keys
`price_clusters`: the object with the price cluster information
`udc_id`: This is the identifier of the product with the attached cluster weight: low, medium or high. There are 3 different udc_ids per product that can be used to create an appraisal.
`price`: The median price of the cluster. Use this value as the cluster price.

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

`percentage`: Ratio of the number of prices in the cluster compared to all prices found for the search.

Step 3. Replacement Product Suggestions (RP)

Using the `pid` provided in this request (the returned value of the selected Claimed Product in the Product Search step), ValueChecker will suggest similar products based on relevant specifications, - the product that is selected during this step is referred to as the Replacement Product. It's worth noting that a suggested Replacement Product could have the same `pid` as the Claimed Product if the Claimed Product is still for sale and in many cases the Replacement Product will share a very similar or identical `product_name` with the Claimed Product.

Request

Method	URL
`GET`	/prices

Params	Value-Type	Example Values
pid	number	44204791
country	string	nl
session-id*	string	"ABCDE12345"

`pid`: The product identifier (usually selected from the product search response)
`country`: The country in where prices will be searched
*`session-id`: Optional session identifier of the **user**

Response

Status	Response

200	{
	"claimed_product": {
	"pid": 1008793454,
	~~"cid": 138~~,
	"product_name": "LG 55SM9010PLA",
	"master_name": "",
	"pic_url": "https://img.valuechecker.net/,snkJVOj...",
	"price": {
	"price": 845.0,
	"price_date": "2020-09-07 11:29:11",

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

```
            "last_known_price": true
        },
        "specs": [
            {
                "spec_id": 999,
                "spec_name_display": "Brand",
                "spec_value_display": "LG",
                "spec_value_numeric": 2.0
            },
            ...
        ],
        "in_stock": true,
        “client_category”: “Television”,
        “client_category_id”: 4000
    },
    "grouped_prices": [
        [
            {
                "pid": 1008793454,
                "shop_product_name": "LG 55SM9010PLA Zwart",
                "currency": "EUR",
                "total_price": 1178.0,
                "base_price": 1178.0,
                "price_date": "2020-09-07 11:29:11",
                "offer_url": "https://offer.valuech.../ABC...",
                "international": false,
                "is_refurbished": false,
                "in_stock": true,
                "blocked": false,
                "country_code": "NL",
                "product": {
                    "pid": 1008793454,
                    "sid": 31001,
                    "source_internal_id": "1362936",
                    "product_name": "LG 55SM9010PLA",
                    "master_name": "LG 55SM9010",
                    "pic_url": "https://img.va.../,snkb-o6...",
                    "brand": "LG",
                    "specs": [
                      {
                        "spec_id": 999,
                        "spec_name_display": "Merk",
                        "spec_value_display": "LG",
                        "spec_value_numeric": 2.0
                      },
                    ]
                },
                "offer_id": 18882086,
```

	<pre>"shop": {
	"shop_name": "PlatteTV",
	"shop_status": 1
	},
	"filtered_price": false
	},
	]
	]</pre>

Deprecation warnings:

- The “cid“ field on the claimed product has been deprecated and will be removed in a future update. Please use the client_category_id directly instead.

The response is a list of price groups, where each group is an array of the prices found for one product, sorted first by Recommended Shop and then Price.

Grouped prices

`grouped_prices` contains a list(s) of replacement product(s) full of arrays of key value information pertaining to the shops where the replacement product is available.
This list is sorted by Recommended Shop, then Price.
ValueChecker has a default list of shops that are trustworthy (recommended). The recommended shops list can be modified to your specifics. Get in touch with your ValueChecker contact to make shop settings changes.
If your implementation requires only 1 replacement price, take the `total_price` value from the first product of the first group in the list for the lowest available price from a recommended shop.
Example:
`grouped_prices.[0].[0].total_price`
(0 and 0 being the first product (key) of the first group (row) in the `grouped_prices` list)

Response Structure

```
{
  "claimed_product": {
    ...
  },
  "grouped_prices": [
    [
      {
        ...
      },
    ],
    [
      {
        ...
      },
      ...
    ],
    ...
  ]
}
```


International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

Last Known Price

The `claimed_product` may contain an object called `price`. This contains an additional nested key again called `price` (see Response sample below). This is not to be confused with replacement prices in the following object `grouped_prices`. The Last Known Price is the most recent valid price that ValueChecker has stored for the Claimed Product. It can be used as historical data but should not be shown as the current ValueChecker suggested replacement price.
When available it will be included in the `claimed_product` object if no current valid prices for the Claimed Product are found in the current market. It is also included in the `grouped_prices` object as a filtered_price replacement if no other current prices are available for any possible replacement.

We additionally return prices that have been *filtered out* by us but can still be selected as a viable replacement product if the user opts to do so.

Filtered prices

Filtered prices are defined by the boolean `filtered_price` .
The default setting for parsing the prices results should always screen out filtered prices. Price entries with `filtered_price = true` should be hidden in your result set. These prices are included in the response for special business cases and for your historical analysis as the values may change often (for example: the "in stock" status can change often - thereby switching the filtered status).
The 4 cases where `filtered_price = true` (ValueChecker considers as a filtered price but still returns the data in the response) are as follows:

Key	Value	Explanation
"in_stock"	`false`	The shop indicates the product as not in stock
`"international"`	`true`	The shop marked the offer as not locally available
`"is_refurbished"`	`true`	The product is marked as refurbished
`"shop": { "shop_status":`	`-1`	(in the shop array) <0 the shop itself has been filtered out 0 the shop is whitelisted 1 the shop is recommended (ranked higher)

Step 4. Appraisals

4.1 Create Appraisal

Creating an Appraisal record requires a Reference (or 'cart') id. A Reference (or cart) can contain 1 or more Appraisals.
You should have a `client_reference_id` from Step 1. Create Reference (or Step1.1) to use

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

in the request path.

Request

Method	URL
POST	/client_reference/`{client_reference_id}`/appraisal

Body JSON Params	Value-Type	Example Values
cpid	number	1000000
rpid	number	1000001
*replacement_cost_value	number	1000.00
{"cpid":1000000,"rpid":1000001, "replacement_cost_value": 1000.00}		

`cpid`: This is the Claimed Product Identifier. You should have this value from the previous response Step 3: (RP) from field `claimed_product.pid`

`rpid`: This is the Replacement Product Identifier. The previous response should have one or more Replacement Products. You should take the value from whichever product you select from field `grouped_prices.[Y].[X].pid` (where X is the numbered product of the Y numbered group)

`replacement_cost_value`: This is an optional variable parameter. You can either use the price from the previous response from field `grouped_prices.[Y].[X].total_price` or you can set a custom value based on your own logic.

Response

Status	Response
200	{ "appraisal_id": 1234567 }

Note: If an identical Appraisal record exists in the same client_reference (the same CPID and RPID), we will create a new appraisal record with a different identifier (`appraisal_id`).

4.2 View All Appraisals

To view all/any Appraisals within a reference, call the same endpoint without any parameters You should have a `client_reference_id` from Step 1. Create Reference (or Step1a.) to use in the request path.

Request

Method	URL

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

GET	/client_reference/`{client_reference_id}`/appraisal

Params	Value-Type	Example Values
No params needed		

Response

Status	Response

200	[
	{
	"appraisal_id": 1234567,
	"client_reference_id": 1001,
	"cpid": 1000000,
	"rpid": 1000001,
	"create_time": "2021-02-17 13:36:10",
	"update_time": "2021-02-17 13:36:10",
	},
	{
	"appraisal_id": 1234568,
	"client_reference_id": 1001,
	"cpid": 1000002,
	"rpid": 1000003,
	"create_time": "2021-02-17 13:34:48",
	"update_time": "2021-02-17 13:34:48",
	"cp_name_cart": "Apple iPhone X 64GB (2017)",
	"rp_name_cart": "Apple iPhone X 64GB Zilver",
	"cp_pic_url": "https://img.valuechecker.net/,sqYJ...",
	"rp_pic_url": "https://img.valuechecker.net/,aqYJ...",
	"replacement_cost_value": 1000.01,
	"is_acv_cp_price_estimate_based": false
	}
	]

Note: Depending how the appraisal was created, the response may contain additional values.

4.3 Delete All Appraisals

To remove all Appraisals from a reference, call the same endpoint without any parameters.

Request

Method	URL

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

DELETE	/client_reference/`{client_reference_id}`/appraisal

Params	Value-Type	Example Values
No params needed		

Response

Status	Response
204	[]

Note: This empties the Reference of all Appraisals but does not delete the actual Reference itself.

4.4 View An Appraisal

The `appraisal_id` (returned from 4.1) is required in addition to the `client_reference_id`.

Request

Method	URL
GET	/client_reference/`{client_reference_id}`/appraisal/`{appraisal_id}`

Params	Value-Type	Example Values
No params needed		

Response

Status	Response

200	{
	"appraisal_id": 1234567,
	"client_reference_id": 1001,
	"cpid": 1000000,
	"rpid": 1000001,
	"create_time": "2021-02-17 13:36:10",
	"update_time": "2021-02-17 13:36:10",
	}

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

4.5 Modify An Appraisal

Modifications are passed as parameters in the payload (formatted in JSON).

Request

Method	URL
PUT	/client_reference/`{client_reference_id}`/appraisal/`{appraisal_id}`

Params	Value-Type	Example Values
cp_name_cart	string	“A new product name 2 plus”
rpid	number	12345679
rp_name_cart	string	“A new product name 2 plus”
replacement_cost_value	number	638.95
cp_price_estimate	number	321.01
offer_label	string	“Link to shop”

```
{"rpid":12345683, "rp_name_cart":"A name", "replacement_cost_value":638.95}
```

`cp_name_cart`: Custom name for the claimed product (the `cpid` cannot be changed).
`rpid`: This is the Replacement Product Identifier. The previous response should have one or more Replacement Products. You should take the value from whichever product you select from field `grouped_prices.[Y].[X].pid` (where X is the numbered product of the Y numbered group)
`rp_name_cart`: Custom name for the replacement product.
`replacement_cost_value`: Modified product price.
`cp_price_estimate`: The value claimed by the policyholder (Purchase Price / Repair).
`offer_label`: Custom label for the offer URL.

Response

Status	Response
200	*empty*

4.6 Delete An Appraisal

To remove a specific Appraisal from a reference, pass the `appraisal_id` in the path.

Request

Method	URL

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

DELETE	/client_reference/{client_reference_id}/appraisal/{appraisal_id}

Params	Value-Type	Example Values
No params needed		

Response

Status	Response
204	*empty*

Step 5. RCV and ACV Calculations

Note: The endpoint `/client_reference` has an availability setting. Any calls to it without the setting enabled will result in a 404 response. You can request access from your ValueChecker representative.

ValueChecker is able to calculate payouts (before standard deductions apply) by using the depreciation values provided by the insurance company. This is accomplished by providing user-input parameters like the date of damage and the date of purchase in the request -- along with other, previously obtained parameters detailed below.

5.1 Perform an ACV Calculation

Request

Method	URL
PUT	/client_reference/{client_reference_id}/appraisal/{appraisal_id}/acv

Params	Value-Type	Example Values
date_purchase	date	"2017-02-01"
date_claim	date	"2019-03-05"
selected_client_category_id	number	234
is_acv_cp_price_estimate_based	boolean	true or false

```
{"date_purchase": "2017-03-03", "date_claim": "2020-02-20",
"selected_client_category_id": 1000, "is_acv_cp_price_estimate_based": false}
```


International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

`date_purchase`: The date of purchase as reported by the claimant.
`date_claim`: The date of the submitted claim for this product.
`selected_client_category_id`: (Optional) The identifier for the category that the ACV algorithm is to use for calculation. This value is client specific.
Note: This is not necessarily the same as or related to the product category (client category id) used for the Claimed Product search. Please ask your ValueChecker support contact for more information.
`is_acv_cp_price_estimate_based`: Base the ACV calculation on the value from `cp_price_estimate` (Purchase Price / Repair) instead of Replacement Cost

Response

Status	Response
200	*empty*

5.2 Reset an ACV Calculation

Request

Method	URL
DELETE	/client_reference/{client_reference_id}/appraisal/{appraisal_id}/acv

Params	Value-Type	Example Values
No params needed		

Response

Status	Response
204	*empty*

5.3 Retrieve RCV and/or ACV Values

The RCV and ACV values are stored (if available) in each appraisal object and can be retrieved by either the full client reference or an individual appraisal.

Appraisals that have been added by the ValueChecker Customer Portal or by a Sheet Import are automatically processed and will have both the RCV and ACV calculated values available.

Appraisals that your implementation adds will need to be calculated manually by using the step outlined in Step 5.1.

Example for retrieving RCV and ACV values from a client_reference:

Request

Method	URL
GET	/client_reference/`{client_reference_id}`/appraisal

Params	Value-Type	Example Values
No params needed		

Response

Status	Response

200	<pre>{
	  "appraisals": [
	  {
	  "appraisal_id": 275204,
	  …
	  "replacement_cost_value": 1000,
	  "actual_cash_value": 750,
	  …
	  }
	  ]
	}</pre>

Example for retrieving RCV and ACV values from an appraisal:

Request

Method	URL
GET	/client_reference/`{client_reference_id}`/appraisal/`{appraisal_id}`

Params	Value-Type	Example Values
No params needed		

Response

Status	Response

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

200	{
	"appraisal_id": 1234567,
	…
	"replacement_cost_value": 1000,
	"actual_cash_value": 750,
	…
	}

Step 6. Generate Reports

Note: The endpoint `/client_reference` has restricted availability. Any calls to it will result in a 404 response. You can request access from your ValueChecker representative.

ValueChecker reports can be generated at the Client Reference level (multiple Appraisals) or on a specific Appraisal.
The response in both cases is a unique link to the document in PDF format.

6.1 Reference Report

The generated document lists all Appraisals from a Reference:

- Claimed products
- Replacement products
- Offer links
- Product prices
- Actual Cash Values (ACV)

Request

Method	URL
POST	/client_reference/`{client_reference_id}`/report

Params	Value-Type	Example Values
No params needed		

Response

Status	Response

200	{
	…
	}

6.2 Appraisal Report

This document lists all details from a specific Appraisal:

- Claimed products
- Replacement products
- Offer links
- Product prices
- Actual Cash Values (ACV)

The layout is presented in a user-friendly format that can be shared with Claimants.

Request

Method	URL
POST	/client_reference/`{client_reference_id}`/appraisal/`{appraisal_id}`/report

Params	Value-Type	Example Values
`No params needed`		

Response

Status	Response

200	{
	…
	}

International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden

APPENDIX

Definitions

- Claimed Product (CP)
  - The product being claimed by the policyholder

- Item of Loss
  - The item being claimed by the policyholder

- Replacement Product (RP)
  - The product identified by ValueChecker and/or the Claims Handler to replace the product being claimed by the policyholder. Note: This can be the same product as the Claimed Product - if still available on the market at a reasonable price - or a different product of the same Like-Kind-and-Quality (LKQ)

- Replacement Cost Value (RCV)
  - The cost to replace the Claimed Product at the time of damage, thus the price to purchase the Replacement Product (RP).

- Actual Cash Value (ACV)
  - The calculated value of the Claimed Product after applying the insurer's depreciation rules. Thus the ACV = RCV - Depreciation.

- Depreciation
  - A reduction - based on the rules of the insurer - in the value of the Claimed Product over time, due in particular to wear and tear.

- vcid
  - Identifier assigned to very specific groupings of similar products in ValueChecker, that have the same or similar price regardless of model.
    Example: "Apple iPhone 7 *32GB Gold*", "Apple iPhone 7 *32GB Black*", etc.
    Note: One vcid can only have 1 alid.

- alid
  - Identifier assigned to broad groupings of similar products in ValueChecker, that are of the same model family but can have .
    Example: Apple iPhone 7.
    Note: One alid can have many vcid's.

- Like-Kind-and-Quality (LKQ)
  - A condition in property insurance policies that states that the insurer would cover the cost of repairing or replacing a covered loss ("Claimed Product") with

property (“Replacement Product”) that is similar in composition and quality. The ValueChecker is calculating the Replacement Products for all Claimed Products that are in the Replacement Categories, based on multiple relevant product specifications defined for each individual category. As a general rule, for all Replacement Categories the Replacement Products should be of the same Brand and not of an older model family compared to the Claimed Product.

- User
    - The individual who uses ValueChecker during the claim handling process. This could be a Claim Handler or a Policyholder.

Event Codes

Event Segments

Events codes are 6 character long strings constructed using segments that can be parsed to identify the error reason and API resource.

There are 3 segments totalling 6 characters:
ABCDEF

The segment comprising the 1st and 2nd character denotes the Event Severity.

Segment	Description
POS(1,2)	Position 1 and 2 of the Event Code: SEVERITY
ER	Prefix segment for events of severity ‘ERROR’
WA	Prefix segment for events of severity ‘WARN’
IN	Prefix segment for events of severity ‘INFO’

Note: WARN and INFO events are not currently implemented as of this documentation. They will be added in an upcoming release.

ERROR Events

Calls to the ValueChecker API that result in an error for any reason will result in an error response. These event codes begin with **ER**.

Example Response

Status	Response
400	{ "event_code": "ERIPCS", "event_dict": { "event_severity": "ERROR",

	<pre>"title": "Invalid parameter",
	"description": "The \"client_category_id\" parameter is invalid. The value must be an integer."
	},
	"error": "Invalid parameter",
	"description": "The \"client_category_id\" parameter is invalid. The value must be an integer.",
	}</pre>

Note: The root keys of 'error' and 'description' are maintained for backwards compatibility and can be safely ignored. They are deprecated and will be removed in an upcoming release.

Error Segments

The segment comprising the 3rd and 4th character denotes the Error type (when the first segment is **ER**).
The segment comprising the 5th and 6th character denotes the Error location (resource).

Segment	Description
POS(3,4)	Position 3 and 4 of the Event Code: Type
AE	Resource already exists
IP	A passed parameter(s) is invalid (i.e. expecting INT, got STRING)
IT	The bearer token is invalid
MP	A required parameter is missing
NA	Method requested is not allowed
NF	Resource is not found
WP	The value of passed parameter is wrong
POS(5,6)	Position 3 and 4 of the Event Code: Location
SS	The error was raised from /search_suggestion
PS	The error was raised from /product_search
PR	The error was raised from /prices

Status Codes

All status codes are standard HTTP status codes.
The following status codes are used by the ValueChecker API.

Status Code	Description
200	OK
204	No Content
400	Bad Request

Deprecated

Cid and Category in Response object for product_search

Deprecated at version 0.20.70

Status	Response

200	{
	"results": [
	{
	"child_products": [],
	"pid": 1122795690,
	~~“cid”: 132~~,
	~~“category”: Laptops,~~
	“client_category”: Laptops,
	“client_category_id” 2000
	"product_name": "Asus Zenfone 9 8GB 5G 128GB White",
	"Product_picture_url": "https://img.valuechecker.net/,..j",
	"specs": [
	{
	"spec_id": 999,
	"spec_name_display": "Marke",
	"spec_value_display": "Asus",
	"spec_value_numeric": 1.0,
	"spec_name": "Marke"
	}],
	"price": {
	"price": 790.0,
	"price_url": "https://www.belsimpel.nl/asus-zen...",
	"shop_count": 1,
	"price_date": "2022-12-06 15:58:52"
	}],
	"metadata": {
	"category": {
	“requested_client_category_id”: 1234,
	"requested_client_category": "Laptops",
	"prediction_certainty": 0.9978,

	"predicted_client_category_id": 469, "predicted_client_category": "Mobiele Telefoon", "category_switched": true }, "used_query": "asus zenfone 2015" }

Deprecation warnings:

- “cid“ has been deprecated and will be removed in a future update. Please use the client_category_id directly instead.
- “category” has been deprecated and will be removed in a future update. However, the “client_category” will return the same information.

List Response for product_search

Deprecated at version 0.20.70

Status	Response

200	{
	"claimed_product": {
	"pid": 1008793454,
	~~“cid”: 132~~,
	…
	“client_category”: Television,
	“client_category_id” 2000
	},
	"grouped_prices": [
	[
	{
	"pid": 1008793454,
	"shop_product_name": "LG 55SM9010PLA Zwart",

Deprecation warnings:

- The “cid“ field on the claimed product has been deprecated and will be removed in a future update. Please use the client_category_id directly instead.

The ValueChecker response for the product_search formatted as a list response has been migrated to an object response.
All clients currently using the list response should migrate to the object response before April 1st, 2023.

Example list response. See the current object response.

Status	Response
200	{

```
   "child_products": [],
   "pid": 1122795690,
   "product_name": "Asus Zenfone 9 8GB 5G 128GB White",
   "product_picture_url": "https://img.valuechecker.net/,...jpg",
   "specs": [
       {
          "spec_id": 999,
          "spec_name_display": "Marke",
          "spec_value_display": "Asus",
          "spec_value_numeric": 1.0,
          "spec_name": "Marke"
       },
   "price": {
        "price": 790.0,
        "price_url": "https://www.belsimpel.nl/asus-zen...",
        "shop_count": 1,
        "price_date": "2022-12-06 15:58:52"
   },
   "source": "Belsimpel"
...
```


OAuth Authentication

Deprecated at version 0.12.45

ValueChecker uses the OAuth 2.0 standard for authentication.
Currently the authentication process of a new user to the system is handled internally by ValueChecker. You will be sent an access_token. (Type: **Bearer Token**.)
To get started using ValueChecker's API, use the access_token as required for every request.
Ask your ValueChecker contact for more information.

To include the access_token in a request, include it as an "Authorization" header.
Curl example:

```
curl -X GET -H "Authorization: Bearer eyJhbGciOiJIUzUxMiIsInR5cCI6I...Q"
'https://api.valuechecker.net:8443/api/<endpoint>?query=param1&param2=40
&param3=-1'
```


International Consumer Services Sweden AB
SE556680-3572 | Söder Mälarstrand 21, 2 tr. | SE-118 20 Stockholm, Sweden