**API of the Catalog of Marked Goods**

**API of the catalog of marked goods**

Version 5.40

Contents Оглавление

[What’s new in v.5.40 dd. 20/08/2025 4](#_Toc340326)

[1. GENERAL INFORMATION ABOUT THE CATALOG OF MARKED GOODS 5](#_Toc340327)

[2. GENERAL PARAMETERS FOR WORKING WITH THE NC API 6](#_Toc340328)

[2.1. Limit of API Requests 6](#_Toc340329)

[2.2. Universal Request Parameters 6](#_Toc340330)

[2.3. Allowed Unicode characters when creating and updating the goods cards 7](#_Toc340331)

[2.4. HTTP ETag (Version Control) 7](#_Toc340332)

[2.5. Response Codes 8](#_Toc340333)

[3. METHODS TO WORK WITH API of NC 10](#_Toc340334)

[3.1. Getting information about the card 10](#_Toc340335)

[3.1.1. Method “Get information about your own card” 10](#_Toc340336)

[3.1.2. Method “Get information about published card” 22](#_Toc340337)

[3.1.3. Method “Get brief information about published card” 32](#_Toc340338)

[3.1.4. Method “Get a list of your own cards with brief information about them” 44](#_Toc340339)

[3.1.5. Method “Check changes in cards” 47](#_Toc340340)

[3.1.6. Method “Check whether cards or FEACN codes belong to marked goods groups” 49](#_Toc340341)

[3.2. Create or edit a card 53](#_Toc340342)

[3.2.1. Method “Generate a goods code” 53](#_Toc340343)

[3.2.2. Method “Create or edit a card” 55](#_Toc340344)

[3.2.3. Method “Check update package processing status” 67](#_Toc340345)

[3.2.4. Method “Change the photo size” 74](#_Toc340346)

[3.2.5. Method “Forcibly send a card for moderation” 75](#_Toc340347)

[3.3. Getting information about attributes 76](#_Toc340348)

[3.3.1. Method “Get a tree of categories” 76](#_Toc340349)

[3.3.2. Method “Get an attribute list” 79](#_Toc340350)

[3.3.3. Method “Get a directory of countries of manufacture” 84](#_Toc340351)

[3.3.4. Method “Get a catalog of trademarks” 86](#_Toc340352)

[3.3.5. Method “Check the presence of permit document in the directory” 88](#_Toc340353)

[3.3.6. Method “Get information about a permitting document by a goods code and INN” 90](#_Toc340354)

[3.3.7. Method “Get information about a permitting document by a number and date” 94](#_Toc340355)

[3.4. Signing a card 99](#_Toc340356)

[3.4.1. Method “Get XML for subsequent signing a card” 99](#_Toc340357)

[3.4.2. (Deprecated) Method “Sign a card by using an attached signature” 101](#_Toc340358)

[3.4.3. Method “Sign a card by using the detached signature” 103](#_Toc340359)

[3.5. Working with subaccounts 105](#_Toc340360)

[3.5.1. Method “Get a list of company subaccounts” 105](#_Toc340361)

[3.5.2. Method “Get a list of companies and goods codes for which access has been granted, by subaccount” 106](#_Toc340362)

[3.5.3. Method “Get XML file that is required to control access to subaccounts” 108](#_Toc340363)

[3.5.4. Method “Sign granting or revoking permission to a subaccount to use goods codes for MCs emission” 111](#_Toc340364)

[4. CATALOGS 115](#_Toc340365)

[4.1. Catalog – List of supported goods groups 115](#_Toc340366)

[5. Modifications introduced in the previous versions of the document 117](#_Toc340367)

# What’s new in v.5.40 dd. 20/08/2025

Errors returned during request processing have been added into the description of the “3.3.6. Method “Get information about a permitting document by a goods code and INN”” (v4/rd-info-by-gtin) and “3.3.7. Method “Get information about a permitting document by a number and date”” (v4/rd-info) methods.

*To view a full document revision history, go to the “*5. Modifications introduced in the previous versions of the document*” section.*

# GENERAL INFORMATION ABOUT THE CATALOG OF MARKED GOODS

The National Catalog of Marked Goods of the Russian Federation (hereinafter - the National Catalog) was created to ensure uniformity, standardization and harmonization of data about goods, as well as to maintain all necessary reference information with uniform description thereof in the entire distribution chain, from a manufacturer and/or importer of goods to retail.

Main functions of the National Catalog:

* provide information contained in the registers to Participants and all users of the catalog within their competence and access rights;
* maintain a register of Participants of the National Catalog;
* enable manufacturers and importers to register marked goods items;
* maintain a register of cards of the marked goods (digital data sheets of the marked goods);
* provide information on the permit documents of marked goods items according to the current regulatory legal acts of EAEU countries;
* transfer a unique code of the nomenclature goods item to the marking and traceability system for its inclusion into a structure of the marking code of the item of goods;
* ensure legal value of the information transmitted by goods circulation participants to the Unified Goods Catalog of the Russian Federation.

# GENERAL PARAMETERS FOR WORKING WITH THE NC API

Goods circulation participants’ information exchange in the National Catalog is performed based on the electronic information services with the use of standard protocols and interfaces of electronic interaction that retain the mode of the guaranteed delivery of the data packets:

* via user account;
* via Application Programming Interface (API).

Entering data into the National Catalog via API of the National Catalog is performed according to the methods specified in this document.

* [https://api.nk.sandbox.crptech.ru](https://api.nk.sandbox.crptech.ru/)— test environment address;
* [апи.национальный-каталог.рф](https://апи.национальный-каталог.рф/)— production environment address.

When using the API methods of the National Catalog, you should specify an access key (“apikey”) or TT GIS authentication token (Authorization: Bearer Token) in the request. For more details, see section “2.2. Universal Request Parameters”.

## Limit of API Requests

There are personal limitations applied to each organization, general (for all methods) and private (for the selected method). Time is calculated starting from the first request in a series.

The number of executed requests in the current series is passed in each API response in the API-Usage-Limit HTTP header (example: API-Usage-Limit: 1/500) within the general limit and API-Method-Usage-Limit (example: API-Method-Usage-Limit: 1/10) within the private limit, if it has been set.

## Universal Request Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Mandatory | Default value | Description |
| apikey | - |  | An identifier (key) of the goods owner. The parameter required for authorization and making API queries. The parameter is mandatory if “token” (“TT GIS authentication token”) is not specified.  You can get an identifier (key) in a user account of the National Catalog in the section “Profile” → “Participant’s data” (“API KEY” field). The “API KEY” field is displayed only to a user that has the “Administrator” role. |
| token | - |  | A TT GIS authentication token. It is specified in Authorization: Bearer <token>. The parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified.  You can get a token by interacting with True API, using methods of single authentication.  For more details, see True API documentation |
| format | - |  | A response output format. Possible values: "json", "xml" |

## Allowed Unicode characters when creating and updating the goods cards

|  |  |  |
| --- | --- | --- |
| Unicode block | Allowed characters | Exceptions |
| Basic Latin | U+0000 to U+007F | U+0000 to U+001F, except 000A and 000D |
| Latin-1 Supplement | U+0080 to U+00FF | U+0080 to U+00A0 |
| Latin Extended-A | U+0100 to U+017F |  |
| Latin Extended- B | U+0180 to U+024F |  |
| Cyrillic | U+0400 to U+04FF |  |
| Cyrillic Supplement | U+0500 to U+052F |  |
| General punctuation | U+2000 to U+206F |  |
| Letterlike symbols | U+2116 |  |
| Spacing Modifier Letters | 02BO to 02FF |  |
| Combining diacritical marks | 0300 to 036F |  |
| CJK Unified Ideographs | 4E00 to 9FFF |  |
| Superscripts and Subscripts | 2070 to 209F |  |
| Greek and Coptic | 0370 to 03FF |  |
| Combining diacritical marks | 0300 to 036F |  |
| Important | Unicode characters that are not specified in this table will be cut off and will not be added to the description of the goods | |

## HTTP ETag (Version Control)

The “Entity Tag” header (ETag for short) is used to transfer a page content hash. If the page has been modified, the page hash will also be modified. By comparing the client-side hash with the hash generated on the server side, cache can determine if the page has been modified and should be resent. It is used for the following methods: product, short-product, categories, brands.

When called, the server returns a resource with the respective ETag value, which is in the ETag field of the HTTP header:

ETag: "686897696a7c876b7e"

Then, the resource can be cached along with its ETag. Later, when receiving a page from the same address, the previously saved ETag value can be sent along with the request in the If-None-Match field (it is specified in HTTP header):

If-None-Match: "686897696a7c876b7e"

To this request the server compares the client's ETag with the ETag of the current resource version. If ETag values match, the resource has not been modified, and the server sends back a very brief response with the 304 Not Modified HTTP status. Status 304 means that the cached version is still valid, and no data update is required. However, if ETag values do not match, the resource has been modified, and the server returns full response. If so, the resource cache and its ETag can be updated.

When using the ETag and receiving the response with the 304 Not Modified status, the API usage limit is not applied.

The ETag hash length for the product method depends on the number of goods in the response. Maximum length: 4 Kb.

Example of a header with ETag:

Content-type: application/json; charset=utf-8

API-Usage-Limit: 1/500

Etag: "38fho43p543re634c993ec28581d867" Status Code: 200

Example of the response with status 304:

Content-type: application/json; charset=utf-8 Status Code: 304

## Response Codes

* 200 — OK;
* 304 — Page not modified;
* 400 — Request parameter error;
* 401 — Not authorized, apikey/token is missing in the request;
* 403 — No access to the requested information;
* 404 — Requested information not found;
* 413 — Request size exceeds the limit;
* 429 — Limit of API requests exceeded, Retry-After HTTP header contains a value of time before the beginning of provision of access in seconds;
* 500 — Internal server error;
* 501 — Method does not exist;
* 503 — Service is unavailable, under maintenance, repeat the request later.

# METHODS TO WORK WITH API of NC

## Getting information about the card

### Method “Get information about your own card”

The method “feed-product” returns all available information about a goods card (about yours or about the one to which you have been given access by the right holder as part of the functionality of work with subaccounts), i.e. all filled attributes regardless of a card status (except for the cards that have the "Requires processing" status), unlike the product method that returns information only on the published cards.

**Note:**

* it requires mandatory indication of one of the parameters: “gtin” (“Goods code”), “good\_id” (“Goods ID”), “gtins” (“List of goods codes”) or “good\_ids” (“List of goods identifiers”);
* if two mandatory parameters “gtin” (“Goods code”) and “good\_id” (“Goods ID”) are specified, then a result of the selection by “good\_id” (“Goods ID”) is returned. In this case GTIN is ignored;
* if two mandatory parameters “gtins” (“List of goods codes”) and “good\_ids” (“List of goods

identifiers”) are specified, then a result of the selection is returned for both parameters;

* if the parameters “gtins” (“List of goods codes”) and “good\_ids” (“List of goods identifiers”) or one of them is specified, then the maximum number of goods items in the request should not exceed 25;
* if one of the parameters “gtin” (“Goods code”) or “good\_id” (“Goods ID”) and one of the parameters “gtins” (“List of goods codes”) or “good\_ids” (“List of goods identifiers”) are specified, then a response with code 400 is returned.

**URL:** /v3/feed-product

**Method:** GET

**Request string example:**

GET <url of environment>/v3/feed-product?apikey=XXX&gtin=0000000000001

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | It is mandatory if “token” (“Authentication token”) is not specified |
| gtin | string | - | Goods code | It is mandatory if one of “good\_id” (“Goods item identifier”), “good\_ids” (“List of goods identifiers”), “gtins” (“List of goods codes”) parameters is missing |
| good\_id | string | - | ID of goods item in the catalog | It is mandatory if “gtin” (“Goods code”), “good\_ids” (“List of goods identifiers”), “gtins” (“List of goods codes”) are missing |
| gtins | string | - | List of goods codes in the | It is mandatory if “good\_id” (“Goods item identifier”), |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  | catalog with delimiter in the form of “;” | “good\_ids” (“List of goods identifiers”), “gtin” (“Goods code”) are missing |
| good\_ids | string | - | List of goods identifiers in the catalog with delimiter in the form of “;” | It is mandatory if “gtin” (“Goods code”), “good\_id” (“Goods item identifier”), “gtins” (“List of goods codes”) are missing |
| subaccount | boolean | - | Indicator that the goods card has been requested through the subaccount | Possible values:  true; false |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| identified\_by | array | + | Array containing information on goods identifiers |  |
| \*value | string | + | Identifier value |  |
| \*type | string | + | Identifier type |  |
| \*\*gtin | string | + | Global barcode GTIN |  |
| \*multiplier | integer | + | Number of goods in the packing | Default value is 1 |
| \*level | string | + | Packing level | Possible values:  “trade-unit” — “Consumer”;  “inner-pack” — “Group consumer”;  “box” — “Shipping”;  “layer” — “Pallet layer”;  “pallet” — “Pallet”;  “metro-unit” — “Metro-  unit”;  “show-pack” — “Showpack” |
| good\_id | integer | + | Goods item identifier |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| good\_name | string | - | Goods name |  |
| is\_kit | boolean |  | Indicator that a goods card has the  “Bundle” type | Possible values:  “true” — yes;  “false” — no |
| is\_set | boolean | + | Indicator that a goods card has the “Set” type | Possible values:  “true” — yes;  “false” — no |
| set\_gtins | array | - | Array of nestings in the set | It is specified only when is\_set=1 |
| \*gtin | string | - | Goods code |  |
| \*quantity | integer | - | Number of nestings |  |
| good\_img | string | - | Default photo |  |
| good\_status | string | + | Technological status of goods card |  |
| good\_detailed\_status | array | - | Array of the current statuses of the goods card; | Possible variants:  “draft” — draft;  “moderation” — being moderated;  “errors” — to be changed;  “notsigned” — awaiting for signing;  “published” — published |
| good\_signed | boolean | + | Indicator that a goods card is signed | Possible values:  “true” — yes;  “false” — no |
| good\_mark\_flag | boolean | + | Indicator that the attributes of the first layer have been filled in and MC can be emitted | Possible values:  “true” — yes;  “false” — no |
| good\_turn\_flag | boolean |  | Indicator that the attributes of the second layer have been filled in and the introduction into circulation is possible (in case if the card is published) | Possible values:  “true” — yes;  “false” — no |
| flags\_updated\_date | string | + | Date of flags update |  |
| create\_date | string | + | Card creation date |  |
| update\_date | string | + | Date of card update |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| producer\_inn | string | + | INN of the manufacturer’s / importer’s company |  |
| producer\_name | string | + | Name of the manufacturer’s / importer’s company |  |
| categories | array | + | Array of categories |  |
| \*cat\_id | integer | + | Identifier of the category to which goods belong |  |
| \*cat\_name | string | + | Name of the category to which goods belong |  |
| brand\_id | integer | + | Trade mark identifier |  |
| brand\_name | string | + | Trade mark name |  |
| good\_images | array | - | Array with images |  |
| \*photo\_type | string | - | Type of photo | Possible values:  “default” — default photo (front view);  “facing” — cropped photo for planograms (cropped along goods item contour);  “left” — left-side photo of a goods item;  “right” — right-side photo of a goods item; “back” — back-view photo of a goods item;  “3ds” — 3D series; “text” — photo of the text on a goods item; “marketing” — a marketing photo of a goods item; “ecommerce” — ecommerce photo; “undef” — single shot, goods item photo from random perspective; “cubi” — photo of goods item's dimensions and weights |
| \*photo\_date | string | - | Photo creation date (UTC) |  |
| \*photo\_url | string | - | Link to the med (medium) photo size |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| \*barcode | string | - | Barcode or SKU of the goods for which the photo was made |  |
| good\_attrs | array | + | Array of attributes |  |
| \*attr\_id | integer | + | Attribute identifier |  |
| \*attr\_name | string | + | Attribute name |  |
| \*attr\_value\_id | integer | + | Attribute value identifier | only for attributes attr\_id = 2502 and attr\_id = 2503 |
| \*attr\_value | string | + | Attribute value |  |
| \*value\_id | integer | + | Attribute value identifier |  |
| \*attr\_value\_type | string | - | Attribute value type |  |
| \*attr\_group\_id | integer | - | Identifier of attributes group |  |
| \*attr\_group\_name | string | + | Attributes group name |  |
| \*location\_id | integer | - | Identifier of measurement location |  |
| \*level | string | - | Packing level |  |
| \*gtin | string |  | Barcode |  |
| \*multiplier | integer | - | Number of goods in the packing | Default value is: 1 |
| \*certificate\_number | string | - | Certificate number | It is returned only for the attributes from the “Certificates” group |
| \*certificate\_issued\_date | string | - | Certificate validity start date | It is returned only for the attributes from the “Certificates” group |
| \*certificate\_valid\_until\_date | string | - | Certificate validity end date | It is returned only for the attributes from the “Certificates” group |
| \*certificate\_applicant | string | - | Applicant | It is returned only for the attributes from the “Certificates” group |
| \*certificate\_manufacturer | string | - | Manufacturer | It is returned only for the attributes from the “Certificates” group |
| Parameter | Type | Mandatory | Description | Comment |
| \*certificate\_product\_description | string | - | Products | It is returned only for the attributes from the “Certificates” group |
| remainder\_type | string | + | Type of remaining items description | “full” — Complete description of remaining items;  “short” — Short description of remaining items;  “null” — It is not a remaining item |
| is\_tech\_gtin | boolean | + | Indicator of the goods card with technical code of the goods | Possible values:  “true” — yes;  “false” — no |
| first\_sign\_date | string | - | Date of the first signing of the card |  |

**JSON response example in case of success (code 200):**

{

"apiversion":3,

"result":[

{

"good\_id":720679,

"identified\_by":[

{

"value":"0000000000001",

"type":"gtin",

"multiplier":1,

"level":"trade-unit"

}

],

"good\_name":"Чешки детские",

"is\_kit":false,

"is\_set":false,

"set\_gtins":[],

"good\_url":"string",

"good\_img":null,

"good\_status":"draft",

"good\_detailed\_status":["draft"],

"good\_signed":false,

"good\_mark\_flag":false,

"good\_turn\_flag":false,

"flag\_updated\_date":"2020-08-18 14:18:03" ,

"create\_date":"2020-08-18 10:57:18",

"update\_date":"2020-08-18 10:57:18",

"producer\_inn":null,

"producer\_name":null,

"categories":[

{

"cat\_id":30717,

"cat\_name":"Обувь домашняя"

}

],

"brand\_id":null,

"brand\_name":null,

"good\_rating":null,

"good\_images":[],

"good\_attrs":[

{

"attr\_id":2478,

"attr\_name":"Полное наименование товара",

"attr\_value":"Чешки детские",

"attr\_value\_type":null,

"attr\_group\_id":24,

"attr\_group\_name":"Наименование товара и идентификация",

"value\_id":0,

"gtin":null,

"multiplier":null,

"level":""

},

{

"attr\_id":2630,

"attr\_name":"Страна производства",

"attr\_value":"RU",

"attr\_value\_type":null,

"attr\_group\_id":80,

"attr\_group\_name":"Происхождение и бренды",

"value\_id":0,

"gtin":null,

"multiplier":null,

"level":""

},

{

"attr\_id":2504,

"attr\_name":"Товарный знак",

"attr\_value":"string",

"attr\_value\_type":null,

"attr\_group\_id":80,

"attr\_group\_name":"Происхождение и бренды",

"value\_id":0,

"gtin":null,

"multiplier":null,

"level":""

},

{

"attr\_id":13898,

"attr\_name":"Материал верха, %",

"attr\_value":"ИСКУССТВЕННАЯ КОЖА",

"attr\_value\_type":"100",

"attr\_group\_id":26,

"attr\_group\_name":"Состав",

"value\_id":0,

"gtin":null,

"multiplier":null,

"level":""

},

{

"attr\_id":13901,

"attr\_name":"Материал подкладки, %",

"attr\_value":"ТКАНЬ",

"attr\_value\_type":"100",

"attr\_group\_id":26,

"attr\_group\_name":"Состав",

"value\_id":0,

"gtin":null,

"multiplier":null,

"level":""

},

{

"attr\_id":13902,

"attr\_name":"Материал низа / подошвы, %",

"attr\_value":"КОЖА (БАРАНЬЯ)",

"attr\_value\_type":"100",

"attr\_group\_id":26,

"attr\_group\_name":"Состав",

"value\_id":0,

"gtin":null,

"multiplier":null,

"level":""

},

{

"attr\_id":13905,

"attr\_name":"Вид обуви",

"attr\_value":"Чешки",

"attr\_value\_type":null,

"attr\_group\_id":103,

"attr\_group\_name":"Потребительские свойства",

"value\_id":0,

"gtin":null,

"multiplier":null,

"level":""

},

{

"attr\_id":13886,

"attr\_name":"Размер в штихмассовой системе",

"attr\_value":"26",

"attr\_value\_type":null,

"attr\_group\_id":103,

"attr\_group\_name":"Потребительские свойства",

"value\_id":0,

"gtin":null,

"multiplier":null,

"level":""

},

{

"attr\_id":36,

"attr\_name":"Цвет",

"attr\_value":"БЕЛЫЙ",

"attr\_value\_type":null,

"attr\_group\_id":103,

"attr\_group\_name":"Потребительские свойства",

"value\_id":0,

"gtin":null,

"multiplier":null,

"level":""

},

{

"attr\_id":3959,

"attr\_name":"Группа ТНВЭД",

"attr\_value":"6402",

"attr\_value\_type":null,

"attr\_group\_id":22,

"attr\_group\_name":"Нормативно-сопроводительная документация",

"value\_id":0,

"gtin":null,

"multiplier":null,

"level":""

},

{

"attr\_id": 13933,

"attr\_name": "Код ТНВЭД",

"attr\_value": "6402000000",

"attr\_value\_type": "",

"attr\_group\_id": 22,

"attr\_group\_name": "Нормативно-сопроводительная документация",

"value\_id": 0,

"gtin": null,

"multiplier": null,

"level": ""

},

{

"attr\_id":3961,

"attr\_name":"Группа ОКПД2",

"attr\_value":"15.20.14",

"attr\_value\_type":null,

"attr\_group\_id":22,

"attr\_group\_name":"Нормативно-сопроводительная документация",

"value\_id":0,

"gtin":null,

"multiplier":null,

"level":""

},

{

"attr\_id": 2439,

"attr\_name": "Ширина",

"attr\_value": "10",

"attr\_value\_type": "см",

"attr\_group\_id": 19,

"attr\_group\_name": "Весогабаритные характеристики",

"value\_id": 0,

"gtin": "0000000000001",

"multiplier": null,

"level": ""

},

{

"attr\_id": 2438,

"attr\_name": "Глубина",

"attr\_value": "35",

"attr\_value\_type": "см",

"attr\_group\_id": 19,

"attr\_group\_name": "Весогабаритные характеристики",

"value\_id": 0,

"gtin": "0000000000001",

"multiplier": null,

"level": ""

},

{

"attr\_id": 2437,

"attr\_name": "Высота",

"attr\_value": "5",

"attr\_value\_type": "см",

"attr\_group\_id": 19,

"attr\_group\_name": "Весогабаритные характеристики",

"value\_id": 0,

"gtin": "0000000000001",

"multiplier": null,

"level": ""

},

{

"attr\_id":2440,

"attr\_name":"Вес брутто",

"attr\_value":"0.15",

"attr\_value\_type":"кг",

"attr\_group\_id": 19,

"attr\_group\_name":"Весогабаритные характеристики",

"value\_id":0,

"gtin":"0000000000001",

"multiplier":null,

"level":""

},

{

"attr\_id": 13756,

"attr\_name": "Расчетный объем",

"attr\_value": "1750",

"attr\_value\_type": "",

"attr\_group\_id": 19,

"attr\_group\_name": "Весогабаритные характеристики",

"value\_id": 0,

"gtin": "0000000000001",

"multiplier": null,

"level": ""

},

{

"attr\_id":4424,

"attr\_name":"Наименование упаковки товара",

"attr\_value":"Чешки детские",

"attr\_value\_type":"",

"attr\_group\_id":24,

"attr\_group\_name": "Наименование товара и идентификация", "value\_id":0,

"gtin":"0000000000001",

"multiplier":null,

"level":""

}

],

"remainder\_type":null,

"is\_tech\_gtin":false

}

] }

**XML response example in case of success (code 200):**

<root>

<apiversion>3</apiversion>

<result>

<item>

<good\_id>2033355</good\_id>

<identified\_by>

<item>

<value>0000000000002</value>

<type>gtin</type>

<multiplier>1</multiplier> <level>trade-unit</level>

</item>

</identified\_by>

<good\_name>Куртка тканная черная XS/160</good\_name>

<is\_kit/>

<is\_set/>

<set\_gtins/>

<good\_url>string</good\_url>

<good\_img></good\_img>

<good\_status>draft</good\_status>

<good\_detailed\_status>

<item>draft</item>

</good\_detailed\_status>

<good\_signed></good\_signed>

<good\_mark\_flag>1</good\_mark\_flag>

<good\_turn\_flag>1</good\_turn\_flag>

<flags\_updated\_date>2020-06-18 14:18:03</flags\_updated\_date>

<create\_date>2020-06-18 10:57:18</create\_date>

<update\_date>2020-06-18 10:57:18</update\_date>

<producer\_inn></producer\_inn>

<producer\_name></producer\_name>

<categories>

<item>

<cat\_id>31326</cat\_id>

<cat\_name>Одежда</cat\_name>

</item>

<item>

<cat\_id>234392</cat\_id>

<cat\_name>6202 Пальто, полупальто, накидки, плащи, куртки (включая лыжные), ветровки, штормовки и аналогичные изделия женские или для девочек, кроме изделий товарной позиции 6204</cat\_name>

</item>

</categories>

<brand\_id></brand\_id>

<brand\_name></brand\_name>

<good\_rating></good\_rating>

<good\_images/>

<good\_attrs>

<item>

<attr\_id>3959</attr\_id>

<attr\_name>Группа ТНВЭД</attr\_name>

<attr\_value>6202</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>22</attr\_group\_id>

<attr\_group\_name>Нормативно-сопроводительная документация</attr\_group\_name>

<value\_id>0</value\_id>

</item>

<item>

<attr\_id>2478</attr\_id>

<attr\_name>Полное наименование товара</attr\_name>

<attr\_value>Куртка тканная черная XS/160</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>24</attr\_group\_id> <attr\_group\_name>Наименование товара и идентификация</attr\_group\_name>

<value\_id>0</value\_id>

</item>

<item>

<attr\_id>2710</attr\_id>

<attr\_name>Тип упаковки</attr\_name> <attr\_value>ОБЕРТКА</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>14</attr\_group\_id>

<attr\_group\_name>Тип и материал упаковки</attr\_group\_name>

<value\_id>0</value\_id>

<gtin>0000000000002</gtin>

<multiplier></multiplier>

<level>trade-unit</level>

</item>

<item>

<attr\_id>2713</attr\_id>

<attr\_name>Материал упаковки</attr\_name>

<attr\_value>ЛИНЕЙНЫЙ ПОЛИЭТИЛЕН НИЗКОЙ ПЛОТНОСТИ

(LLDPE)</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>14</attr\_group\_id>

<attr\_group\_name>Тип и материал упаковки</attr\_group\_name>

<value\_id>0</value\_id>

<gtin>0000000000002</gtin>

<multiplier></multiplier>

<level>trade-unit</level>

</item>

<item>

<attr\_id>13933</attr\_id>

<attr\_name>Код ТНВЭД</attr\_name>

<attr\_value>6202930000</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>22</attr\_group\_id>

<attr\_group\_name>Нормативно-сопроводительная документация</attr\_group\_name>

<value\_id>0</value\_id>

</item>

<item>

<attr\_id>35</attr\_id>

<attr\_name>Размер одежды</attr\_name>

<attr\_value>160-75</attr\_value>

<attr\_value\_type>РОСТ-ОГ</attr\_value\_type>

<attr\_group\_id>103</attr\_group\_id> <attr\_group\_name>Потребительские свойства</attr\_group\_name>

<value\_id>0</value\_id>

</item>

<item>

<attr\_id>36</attr\_id>

<attr\_name>Цвет</attr\_name>

<attr\_value>ЧЕРНЫЙ</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>103</attr\_group\_id>

<attr\_group\_name>Потребительские свойства</attr\_group\_name>

<value\_id>0</value\_id>

</item>

</good\_attrs>

<remainder\_type></remainder\_type>

<is\_tech\_gtin/>

</item>

</result>

</root>

### Method “Get information about published card”

The “product” method returns information only about the published goods cards. It is recommended to use the feed-product method to view information about your goods cards.

A particular limit is set for the method, which is 100 requests in a series.

**Note**:

* it requires mandatory indication of one of the parameters: “gtin” (“Goods code”), “good\_id” (“Goods ID”), “gtins” (“List of goods codes”) or “good\_ids” (“List of goods identifiers”);
* if two mandatory parameters “gtin” (“Goods code”) and “good\_id” (“Goods ID”) are specified, then a result of the selection by good\_id is returned. And “gtin” (“Goods code”) is ignored;
* if two mandatory parameters “gtins” (“List of goods codes”) and “good\_ids” (“List of goods

identifiers”) are specified, then a result of the selection is returned for both parameters;

* if the parameters “gtins” (“List of goods codes”) and “good\_ids” (“List of goods identifiers”) or one of them is specified, then the maximum number of goods items in the request should not exceed 25;
* if one of the parameters “gtin” (“Goods code”) or “good\_id” (“Goods ID”) and one of the parameters “gtins” (“List of goods codes”) or “good\_ids” (“List of goods identifiers”) are specified, then a response with code 400 is returned;
* if the “good\_id” (“Goods ID”) parameter is specified, then a goods item with the corresponding identifier or a response with code 404 is returned;
* if the “gtin” (“Goods code”) parameter is specified, then a goods item with the corresponding goods code or a response with code 404 is returned.

**URL:** /v3/product

**Method:** GET

**Request string example:**

GET <url of environment>/v3/product?apikey=XXX&gtin=0000000000001

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | It is mandatory if "token" is not specified |
| gtin | string | - | Goods code | It is mandatory if one of “good\_id” (“Goods item identifier”), “good\_ids” (“List of goods identifiers”), “gtins” (“List of goods codes”) parameters is missing |
| good\_id | string | - | ID of goods item in the catalog | It is mandatory if “gtin” (“Goods code”), “good\_ids” (“List of goods identifiers”), “gtins” (“List of goods codes”) are missing |
| gtins | string | - | List of goods codes in the catalog with delimiter in the form of “;” | It is mandatory if “good\_id” (“Goods item identifier”), “good\_ids” (“List of goods identifiers”), “gtin” (“Goods code”) are missing |
| good\_ids | string | - | List of goods identifiers in the catalog with delimiter in the form of “;” | It is mandatory if “gtin” (“Goods code”), “good\_id” (“Goods item identifier”), “gtins” (“List of goods codes”) are missing |
| product\_name | string | - | Product name | It is used for missing goods item search request |
| cat\_id | integer | - | Category identifier | It is used for missing goods item search request |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| good\_id | integer | + | Goods item identifier |  |
| good\_name | string | - | Goods name |  |
| is\_kit | boolean | + | Indicator that a goods card has the “Bundle” type | Possible values:  “true” — 1;  “false” — 0 |
| is\_set | boolean | + | Indicator that a goods card has the “Set” type | Possible values:  “true” — 1;  “false” — 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| set\_gtins | array | - | Array of nestings in the set | it is specified only when is\_set=1 |
| \*gtin | string | - | Goods code |  |
| \*quantity | integer | - | Number of nestings |  |
| brand\_id | integer | + | Trade mark identifier |  |
| brand\_name | string | + | Trade mark name |  |
| identified\_by | array | + | Array containing information about goods code |  |
| \*value | string | + | Barcode or local identifier |  |
| \*type | string | + | Barcode type | Possible values:  “gtin” — Global GTIN;  “ntin” — Local NTIN;  “ltin” — Local LTIN (for example, weight barcodes);  “sku” — Local goods item identifier  (SKU);  “barcode” — Barcode (a barcode with an incorrect control figure) |
| \*multiplier | integer | + | Number of goods in the packing | Default value is 1 |
| \*level | string | + | Packing level | Possible values:  “trade-unit” — “Consumer”;  “inner-pack” — “Group consumer”;  “box” — “Shipping”;  “layer” — “Pallet layer”;  “pallet” — “Pallet”;  “metro-unit” — “Metro-unit”; “show-pack” — “Show-pack” |
| good\_img | string | - | Default photo (front view) |  |
| good\_status | string | + | Technological status of goods card |  |
| create\_date | string | + | Card creation date |  |
| update\_date | string | + | Date of card update |  |
| categories | array | + | Array of categories |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| \*cat\_id | integer | + | Identifier of the category to which goods belong |  |
| \*cat\_name | string | + | Name of the category to which goods belong |  |
| good\_attrs | array |  | Array of attributes (only the private attributes of the apikey account are output): |  |
| \*attr\_id | integer | + | Attribute identifier |  |
| \*attr\_name | string | + | Attribute name |  |
| \*attr\_value\_id | integer | + | Attribute value identifier | Only for attributes with attr\_id =  2502 (“Production capacity”) and attr\_id = 2503 (“Manufacturer”) |
| \*attr\_value | string | + | Attribute value |  |
| \*attr\_value\_type | string | - | Attribute value type |  |
| \*attr\_group\_id | integer | - | Identifier of attributes group |  |
| \*attr\_group\_name | string | + | Attributes group name |  |
| \*location\_id | integer | - | Identifier of measurement location (optional) |  |
| \*party\_location\_id | integer | - | Internal identifier of location for a company where the measurement was performed | Only the company to which the location belongs is displayed |
| \*level | string | - | Packing level |  |
| \*gtin | string | - | Barcode |  |
| \*multiplier | integer | - | Number of goods in the packing |  |
| \*certificate\_number | string | - | Certificate number | It is returned only for the attributes from the “Certificates” group |
| \*certificate\_issued\_date | string | - | Certificate validity start date | It is returned only for the attributes from the “Certificates” group |
| \*certificate\_valid\_until\_date | string | - | Certificate validity end date | It is returned only for the attributes from the “Certificates” group |
| \*certificate\_applicant | string | - | Applicant | It is returned only for the attributes from the “Certificates” group |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| \*certificate\_manufacturer | string | - | Manufacturer | It is returned only for the attributes from the “Certificates” group |
| \*certificate\_product\_description | string | - | Products | It is returned only for the attributes from the “Certificates” group |
| good\_images | array | - | Array of images (optional) |  |
| \*photo\_type | string | - | Type of photo | Possible values:  “default” — default photo (front view);  “facing” — cropped photo for planograms (cropped along goods item contour);  “left” — left-side photo of a goods item;  “19” — right-side photo of a goods item;  “13” — back-side photo of a goods item;  “si1” — top-view photo of a goods item;  “si2” — bottom-view photo of a goods item;  “si3” — photo of a packaged goods item;  “si4” — photo of a goods item without packaging;  “si5” — photo of a group packaging;  “3ds” — 3D series;  “text” — photo of the text on a goods item;  “marketing” — a marketing photo of a goods item;  “ecommerce” — e-commerce photo; “undef” — single shot, a goods item photo from random perspective;  “cubi” — photo of goods item's dimensions and weights |
| \*photo\_date | string |  | Photo creation date (UTC) |  |
| \*photo\_url | string |  | Link to the med (medium) photo size |  |
| \*barcode | string | - | Barcode or SKU of the goods for which the photo was made |  |
| remainder\_type | string | + | Type of remaining items description | Possible values:  “full” — Complete description of remaining items;  “short” — Short description of remaining items; |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  |  | “null” — It is not a remaining item |
| is\_tech\_gtin | boolean | + | Indicator of the goods card with technical code of the goods | Possible values:  “true” — yes;  “false” — no |
| first\_sign\_date | string | - | Date of the first signing of the card |  |

**JSON response example in case of success (code 200):**

{

"apiversion": 3,

"result": [

{

"good\_id": 3142181,

"identified\_by": [],

"good\_name": "Комплект: Одеколон, 100 мл + Шампунь для волос и тела, 75 мл

+ Парфюмированный дезодорант для тела, 50 мл",

"is\_kit": true,

"is\_set": false,

"set\_gtins": [],

"good\_url": "https://национальный-каталог.рф/product/0000000000002-runabor-odekolon-100-ml-shampun-dlya-volos-i-tela-75-ml-parfyumirovannyy-dezodorant-dlyatela-50-ml-1",

"good\_img": null,

"good\_status": "published",

"create\_date": "2020-08-20 12:59:32",

"update\_date": "2020-08-20 13:59:32",

"categories": [],

"brand\_id": 27738,

"brand\_name": "string",

"good\_images": [],

"good\_attrs": [

{

"attr\_id": 2504,

"attr\_name": "Товарный знак",

"attr\_value": "string",

"attr\_value\_id": 27738,

"value\_id": 36595997,

"attr\_value\_type": "",

"attr\_group\_id": 80,

"attr\_group\_name": "Происхождение и бренды", "published\_date": "2020-08-12T14:17:40+03:00"

},

{

"attr\_id": 3959,

"attr\_name": "Группа ТНВЭД",

"attr\_value": "3303",

"attr\_value\_type": "",

"attr\_group\_id": 22,

"attr\_group\_name": "Нормативно-сопроводительная документация", "value\_id": 36596000,

"published\_date": "2020-08-12T14:17:40+03:00",

"gtin": null,

"multiplier": null,

"level": ""

},

{

"attr\_id": 2478,

"attr\_name": "Полное наименование товара",

"attr\_value": "Комплект: Одеколон, 100 мл + Шампунь для волос и тела, 75 мл + Парфюмированный дезодорант для тела, 50 мл",

"attr\_value\_type": "",

"attr\_group\_id": 24,

"attr\_group\_name": "Наименование товара и идентификация",

"value\_id": 36647414,

"published\_date": "2020-08-13T17:16:55+03:00",

"gtin": null,

"multiplier": null,

"level": ""

},

{

"attr\_id": 2716,

"attr\_name": "Заявленный объем",

"attr\_value": "НЕ КЛАССИФИЦИРОВАНО",

"attr\_value\_type": "---",

"attr\_group\_id": 24,

"attr\_group\_name": "Наименование товара и идентификация",

"value\_id": 36647417,

"published\_date": "2020-08-13T17:16:55+03:00",

"gtin": null,

"multiplier": null,

"level": ""

},

{

"attr\_id": 1034,

"attr\_name": "Тип парфюмерии",

"attr\_value": "ОДЕКОЛОН",

"attr\_value\_type": "",

"attr\_group\_id": 103,

"attr\_group\_name": "Потребительские свойства", "value\_id": 36647420,

"published\_date": "2020-08-13T17:16:55+03:00",

"gtin": null,

"multiplier": null,

"level": ""

},

{

"attr\_id": 28,

"attr\_name": "Комплектация",

"attr\_value": "Одеколон (1 шт), Шампунь для волос и тела (1 шт), Парфюмированный дезодорант (1 шт)",

"attr\_value\_type": "",

"attr\_group\_id": 26,

"attr\_group\_name": "Состав",

"value\_id": 36647423,

"published\_date": "2020-08-13T17:16:55+03:00",

"gtin": null,

"multiplier": null,

"level": ""

},

{

"attr\_id": 13836,

"attr\_name": "Номер Регламента/стандарта",

"attr\_value": "ТР ТС 009/2011 \"О безопасности парфюмернокосметической продукции\"",

"attr\_value\_type": "",

"attr\_group\_id": 22,

"attr\_group\_name": "Нормативно-сопроводительная документация",

"value\_id": 36647426,

"published\_date": "2020-08-13T17:16:55+03:00",

"gtin": null,

"multiplier": null,

"level": ""

},

{

"attr\_id": 2710,

"attr\_name": "Тип упаковки",

"attr\_value": "КОРОБКА/БОКС",

"attr\_value\_type": "",

"attr\_group\_id": 14,

"attr\_group\_name": "Тип и материал упаковки",

"value\_id": 36647429,

"measure\_date": "2020-08-12T14:17:40+03:00", "published\_date": "2020-08-13T17:16:55+03:00",

"gtin": "0000000000002",

"multiplier": null,

"level": ""

},

{

"attr\_id": 2713,

"attr\_name": "Материал упаковки",

"attr\_value": "КАРТОН ЛАМИНИРОВАННЫЙ",

"attr\_value\_type": "",

"attr\_group\_id": 14,

"attr\_group\_name": "Тип и материал упаковки",

"value\_id": 36647432,

"measure\_date": "2020-08-12T14:17:40+03:00", "published\_date": "2020-08-13T17:16:55+03:00",

"gtin": "0000000000002",

"multiplier": null,

"level": ""

}

],

"remainder\_type": null,

"is\_tech\_gtin": false,

"first\_sign\_date":"2020-08-18 14:18:03"

}

] }

**XML response example in case of success (code 200):**

<root>

<apiversion>3</apiversion>

<result>

<item>

<good\_id>3142181</good\_id>

<identified\_by/>

<good\_name>Комплект: Одеколон, 100 мл + Шампунь для волос и тела, 75 мл +

Парфюмированный дезодорант для тела, 50 мл</good\_name>

<is\_kit>1</is\_kit>

<is\_set/>

<set\_gtins/>

<good\_url>https://национальный-каталог.рф/product/0000000000002-ru-naborodekolon-100-ml-shampun-dlya-volos-i-tela-75-ml-parfyumirovannyy-dezodorant-dlya-tela50-ml-1</good\_url>

<good\_img></good\_img>

<good\_status>published</good\_status>

<create\_date>2020-08-20 12:59:32</create\_date>

<update\_date>2020-08-20 13:59:32</update\_date>

<categories/>

<brand\_id>27738</brand\_id>

<brand\_name>string</brand\_name>

<good\_images/>

<good\_attrs>

<item>

<attr\_id>2504</attr\_id>

<attr\_name>Товарный знак</attr\_name> <attr\_value>string</attr\_value>

<attr\_value\_id>27738</attr\_value\_id>

<value\_id>36595997</value\_id>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>80</attr\_group\_id>

<attr\_group\_name>Происхождение и бренды</attr\_group\_name> <published\_date>2020-08-12T14:17:40+03:00</published\_date>

</item>

<item>

<attr\_id>3959</attr\_id>

<attr\_name>Группа ТНВЭД</attr\_name> <attr\_value>3303</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>22</attr\_group\_id>

<attr\_group\_name>Нормативно-сопроводительная документация</attr\_group\_name>

<value\_id>36596000</value\_id>

<published\_date>2020-08-12T14:17:40+03:00</published\_date> <gtin></gtin>

<multiplier></multiplier>

<level></level>

</item>

<item>

<attr\_id>2478</attr\_id>

<attr\_name>Полное наименование товара</attr\_name>

<attr\_value>Комплект: Одеколон, 100 мл + Шампунь для волос и тела,

75 мл + Парфюмированный дезодорант для тела, 50 мл</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>24</attr\_group\_id> <attr\_group\_name>Наименование товара и идентификация</attr\_group\_name>

<value\_id>36647414</value\_id>

<published\_date>2020-08-13T17:16:55+03:00</published\_date> <gtin></gtin>

<multiplier></multiplier>

<level></level>

</item>

<item>

<attr\_id>2716</attr\_id>

<attr\_name>Заявленный объем</attr\_name>

<attr\_value>НЕ КЛАССИФИЦИРОВАНО</attr\_value>

<attr\_value\_type>---</attr\_value\_type> <attr\_group\_id>24</attr\_group\_id> <attr\_group\_name>Наименование товара и идентификация</attr\_group\_name>

<value\_id>36647417</value\_id>

<published\_date>2020-08-13T17:16:55+03:00</published\_date>

<gtin></gtin>

<multiplier></multiplier>

<level></level>

</item>

<item>

<attr\_id>1034</attr\_id>

<attr\_name>Тип парфюмерии</attr\_name>

<attr\_value>ОДЕКОЛОН</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>103</attr\_group\_id>

<attr\_group\_name>Потребительские свойства</attr\_group\_name>

<value\_id>36647420</value\_id>

<published\_date>2020-08-13T17:16:55+03:00</published\_date>

<gtin></gtin>

<multiplier></multiplier>

<level></level>

</item>

<item>

<attr\_id>28</attr\_id>

<attr\_name>Комплектация</attr\_name>

<attr\_value>Одеколон (1 шт), Шампунь для волос и тела (1 шт),

Парфюмированный дезодорант (1 шт)</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>26</attr\_group\_id>

<attr\_group\_name>Состав</attr\_group\_name>

<value\_id>36647423</value\_id>

<published\_date>2020-08-13T17:16:55+03:00</published\_date>

<gtin></gtin>

<multiplier></multiplier>

<level></level>

</item>

<item>

<attr\_id>13836</attr\_id>

<attr\_name>Номер Регламента/стандарта</attr\_name> <attr\_value>ТР ТС 009/2011 &quot;О безопасности парфюмернокосметической продукции&quot;</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>22</attr\_group\_id>

<attr\_group\_name>Нормативно-сопроводительная документация</attr\_group\_name>

<value\_id>36647426</value\_id>

<published\_date>2020-08-13T17:16:55+03:00</published\_date> <gtin></gtin>

<multiplier></multiplier>

<level></level>

</item>

<item>

<attr\_id>2710</attr\_id>

<attr\_name>Тип упаковки</attr\_name>

<attr\_value>КОРОБКА/БОКС</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>14</attr\_group\_id>

<attr\_group\_name>Тип и материал упаковки</attr\_group\_name>

<value\_id>36647429</value\_id>

<measure\_date>2020-08-12T14:17:40+03:00</measure\_date>

<published\_date>2020-08-13T17:16:55+03:00</published\_date>

<gtin>0000000000002</gtin>

<multiplier></multiplier>

<level></level>

</item>

<item>

<attr\_id>2713</attr\_id>

<attr\_name>Материал упаковки</attr\_name>

<attr\_value>КАРТОН ЛАМИНИРОВАННЫЙ</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>14</attr\_group\_id>

<attr\_group\_name>Тип и материал упаковки</attr\_group\_name>

<value\_id>36647432</value\_id>

<measure\_date>2020-08-12T14:17:40+03:00</measure\_date>

<published\_date>2020-08-13T17:16:55+03:00</published\_date>

<gtin>0000000000002</gtin>

<multiplier></multiplier>

<level></level>

</item>

</good\_attrs>

<remainder\_type/>

<is\_tech\_gtin/>

<first\_sign\_date>2020-08-13T17:16:55+03:00</first\_sign\_date>

</item>

</result>

</root>

### Method “Get brief information about published card”

The method “short-product” returns short or complete information about a product (goods item). This method is available only for the companies for which the role “Integrator / Software developer” has been selected when registering access to the API of the National Catalog.

**Note:**

* it requires mandatory indication of one of the parameters: “gtin” (“Goods code”), “good\_id” (“List of goods identifiers”), “gtins” (“List of goods codes”) or “good\_ids” (“List of goods identifiers”);
* if two mandatory parameters “gtin” (“Goods code”) and “good\_id” (“List of goods identifiers”) are specified, then a result of the selection by good\_id is returned. And “gtin” (“Goods code”) is ignored;
* if two mandatory parameters “gtins” (“List of goods codes”) and “good\_ids” (“List of goods

identifiers”) are specified, then a result of the selection is returned for both parameters;

* if the parameters “gtins” (“List of goods codes”) and “good\_ids” (“List of goods identifiers”) or one of them is specified, then the maximum number of goods items in the request should not exceed 25;
* if one of the parameters “gtin” (“Goods code”) or “good\_id” (“Goods ID”) and one of the parameters “gtins” (“List of goods codes”) or “good\_ids” (“List of goods identifiers”) are specified, then a response with code 400 is returned;
* if the “good\_id” (“List of goods identifiers”) parameter is specified, then the item with the corresponding identifier or the response with code 404 is returned;
* if the “gtin” (“Goods code”) parameter is specified, then a goods item with the corresponding goods code or a response with code 404 is returned.

**URL:** /v3/short-product

**Method:** GET

**Request string example:**

GET <url of environment>/v3/short-product?apikey=XXX&gtin=0000000000001

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | This parameter is mandatory if “token” (“Authentication token”) is not specified |
| gtin | string | - | Goods code | It is mandatory if one of “good\_id” (“Goods item identifier”), “good\_ids” (“List of goods identifiers”), “gtins” (“List of goods codes”) parameters is missing |
| good\_id | string | - | ID of goods item in the catalog | It is mandatory if “gtin” (“Goods code”), “good\_ids” (“List of goods identifiers”), “gtins” (“List of goods codes”) are missing |
| gtins | string | - | List of goods codes in the catalog with delimiter in the form of “;” | It is mandatory if “good\_id” (“Goods item identifier”), “good\_ids” (“List of goods identifiers”), “gtin” (“Goods code”) are missing |
| good\_ids | string | - | List of goods identifiers in the catalog with delimiter in the form of “;” | It is mandatory if “gtin” (“Goods code”), “good\_id” (“Goods item identifier”), “gtins” (“List of goods codes”) are missing |
| product\_name | string | - | Product name | It is used for missing goods item search request |
| cat\_id | integer | - | Category identifier | It is used for missing goods item search request |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| identified\_by | array | - | Array containing barcode information |  |
| \*value | string | - | Barcode or local identifier |  |
| \*type | string | - | Barcode type | Possible values:  “gtin” — Global GTIN;  “ntin” — Local NTIN; “ltin” — Local LTIN (for example, weight barcodes);  “sku” — Local goods item  identifier (SKU);  “barcode” — Barcode (a barcode with an incorrect control figure) |
| \*party\_id | - |  | Identifier of distribution network |  |
| \*multiplier | integer | - | Number of goods in the packing | Default value is 1 |
| \*level | string | + | Packing level | Possible values:  “trade-unit” — “Consumer”; “inner-pack” — “Group consumer”;  “box” — “Shipping”;  “layer” — “Pallet layer”;  “pallet” — “Pallet”;  “metro-unit” — “Metro-unit”;  “show-pack” — “Show-pack” |
| is\_kit | boolean | + | Indicator that a goods card has the  “Bundle” type | Possible values:  “true” — yes;  “false” — no |
| is\_set | boolean | + | Indicator that a goods card has the  “Set” type | Possible values:  “true” — yes;  “false” — no |
| set\_gtins | array | - | Array of nestings in the set | It is specified only when is\_set=1 |
| \*gtin | string | - | Goods code |  |
| \*quantity | string | - | Number of nestings |  |
| updated\_date | string | - | Date of card technological status update |  |
| producer\_inn | string | - | INN of the manufacturer’s / importer’s company |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| producer\_name | string | - | Name of the manufacturer’s / importer’s company |  |
| categories | array | - | Array of categories |  |
| \*cat\_id | integer | - | Category identifier |  |
| \*cat\_name | string | - | Name of the category to which goods belong |  |
| good\_images | array | - | Array with images |  |
| \*photo\_type | string | - | Type of photo | Possible values:  “default” — default photo (front view);  “facing” — cropped photo for planograms (cropped along goods item contour);  “left” — left-side photo of a goods item;  “right” — right-side photo of a goods item;  “back” — back-view photo of a goods item;  “3ds” — 3D series;  “text” — photo of the text on a goods item;  “marketing” — a marketing photo of a goods item; “ecommerce” — e-commerce photo;  “undef” — single shot, goods item photo from random  perspective;  “cubi” — photo of goods item's dimensions and weights |
| \*photo\_date | string | - | Photo creation date (UTC) |  |
| \*photo\_url | string | - | Link to the med (medium) photo size |  |
| \*barcode | string | - | Barcode or SKU of the goods for which the photo was made | Optional |
| good\_attrs | array | - | Array of attributes |  |
| \*attr\_id | integer | - | Attribute identifier |  |
| \*attr\_name | string | - | Attribute name |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| \*attr\_value\_id | integer | - | Attribute value identifier | Only for attributes attr\_id = 2502 and attr\_id = 2503 |
| \*attr\_value | string | - | Attribute value |  |
| \*value\_id | integer | - | Attribute value identifier |  |
| \*attr\_value\_type | string | - | Attribute value type |  |
| \*attr\_group\_id | integer | - | Identifier of attributes group |  |
| \*attr\_group\_name | string | - | Attributes group name |  |
| \*measure\_date | string | - | Date of attribute measurement (UTC) (optional) |  |
| \*published\_date | string | - | Data of attribute publication (UTC) (optional) |  |
| \*effective\_date | string | - | Date from which an attribute value is valid (UTC) |  |
| \*expired\_date | string | - | Date from which an attribute value is invalid (UTC) |  |
| \*location\_id | integer | - | Identifier of measurement location |  |
| \*party\_location\_id | integer | - | Internal identifier of location for a company where the measurement was performed | Only the company to which the location belongs is displayed |
| \*level | string | - | Packing level |  |
| \*gtin | string | - | Barcode |  |
| \*multiplier | integer | - | Number of goods in the packing |  |
| \*certificate\_number | string | - | Certificate number | It is returned only for the attributes from the “Certificates” group |
| \*certificate\_issued\_date | string | - | Certificate validity start date | It is returned only for the attributes from the “Certificates” group |
| \*certificate\_valid\_until\_date | string | - | Certificate validity end date | It is returned only for the attributes from the “Certificates” group |
| \*certificate\_applicant | string | - | Applicant | It is returned only for the attributes from the “Certificates” |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  |  | group |
| \*certificate\_manufacturer | string | - | Manufacturer | It is returned only for the attributes from the “Certificates” group |
| \*certificate\_product\_description | string | - | Products | It is returned only for the attributes from the “Certificates” group |
| remainder\_type | string | + | Type of remaining items description | Possible values:  “full” — Complete description of remaining items;  “short” — Short description of remaining items;  “null” — It is not a remaining item |
| is\_tech\_gtin | boolean | - | Indicator of the goods card with technical code of the goods | Possible values:  “true” — yes;  “false” — no |

**JSON response example in case of success (code 200):**

{

"apiversion": 3,

"result": [

{

"identified\_by": [

{

"value": "0000000000001",

"type": "ntin",

"multiplier": 1,

"level": "trade-unit"

},

{

"value": "0000000000001",

"type": "gtin",

"multiplier": 1,

"level": "trade-unit"

},

{

"value": "0000000000002",

"type": "gtin",

"multiplier": 28,

"level": "box"

}

],

"is\_kit": false,

"is\_set": false,

"set\_gtins": [],

"update\_date": "2019-09-26 17:27:21",

"producer\_inn": null,

"producer\_name": null,

"categories": [

{

"cat\_id": 30302,

"cat\_name": "Кетчуп"

},

{

"cat\_id": 206309,

"cat\_name": "соусы"

}

],

"good\_images": [

{

"photo\_type": "facing",

"photo\_date": "2019-02-22T19:26:56+03:00",

"photo\_url": "string",

"barcode": "0000000000001"

}

],

"good\_attrs": [

{

"attr\_id": 2478,

"attr\_name": "Полное наименование товара",

"attr\_value": "string",

"attr\_value\_type": "",

"attr\_group\_id": 24,

"attr\_group\_name": "Наименование товара и идентификация",

"value\_id": 175, "published\_date": "2018-09-13T15:44:32+03:00",

"gtin": null,

"multiplier": null,

"level": ""

},

{

"attr\_id": 2504,

"attr\_name": "Товарный знак",

"attr\_value": "string",

"attr\_value\_id": 199,

"value\_id": 160752,

"attr\_value\_type": "",

"attr\_group\_id": 80,

"attr\_group\_name": "Происхождение и бренды", "published\_date": "2018-09-13T15:46:54+03:00"

},

{

"attr\_id": 3793,

"attr\_name": "Базовая единица измерения",

"attr\_value": "ШТ",

"attr\_value\_type": "",

"attr\_group\_id": 24,

"attr\_group\_name": "Наименование товара и идентификация",

"value\_id": 13973319,

"published\_date": "2019-02-22T19:27:42+03:00",

"gtin": null,

"multiplier": null,

"level": ""

}

],

"remainder\_type": null,

"is\_tech\_gtin": false

}

] }

**XML response example in case of success (code 200):**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<apiversion>3</apiversion>

<result>

<item>

<identified\_by>

<item>

<value>0000000000001</value>

<type>gtin</type>

<multiplier>1</multiplier> <level>trade-unit</level>

</item>

<item>

<value>0000000000002</value>

<type>gtin</type>

<multiplier>28</multiplier>

<level>box</level>

</item>

</identified\_by>

<is\_kit/>

<is\_set/>

<set\_gtins/>

<update\_date>2019-09-26 17:27:21</update\_date>

<producer\_inn></producer\_inn>

<producer\_name></producer\_name>

<categories>

<item>

<cat\_id>30302</cat\_id>

<cat\_name>Кетчуп</cat\_name>

</item>

<item>

<cat\_id>206309</cat\_id>

<cat\_name>соусы</cat\_name>

</item>

</categories>

<good\_images>

<item>

<photo\_type>facing</photo\_type>

<photo\_date>2019-02-22T19:26:56+03:00</photo\_date>

<photo\_url>string</photo\_url>

<barcode>0000000000001</barcode>

</item>

</good\_images>

<good\_attrs>

<item>

<attr\_id>2478</attr\_id>

<attr\_name>Полное наименование товара</attr\_name>

<attr\_value>string</attr\_value> <attr\_value\_type></attr\_value\_type>

<attr\_group\_id>24</attr\_group\_id> <attr\_group\_name>Наименование товара и идентификация</attr\_group\_name>

<value\_id>175</value\_id>

<published\_date>2018-09-13T15:44:32+03:00</published\_date>

<gtin></gtin>

<multiplier></multiplier>

<level></level>

</item>

<item>

<attr\_id>2504</attr\_id>

<attr\_name>Товарный знак</attr\_name>

<attr\_value>string</attr\_value>

<attr\_value\_id>199</attr\_value\_id>

<value\_id>160752</value\_id>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>80</attr\_group\_id>

<attr\_group\_name>Происхождение и бренды</attr\_group\_name> <published\_date>2018-09-13T15:46:54+03:00</published\_date>

</item>

<item>

<attr\_id>3793</attr\_id>

<attr\_name>Базовая единица измерения</attr\_name>

<attr\_value>ШТ</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>24</attr\_group\_id> <attr\_group\_name>Наименование товара и идентификация</attr\_group\_name>

<value\_id>13973319</value\_id>

<published\_date>2019-02-22T19:27:42+03:00</published\_date>

<gtin></gtin>

<multiplier></multiplier>

<level></level>

</item>

</good\_attrs>

<remainder\_type/>

<is\_tech\_gtin/>

</item>

</result> </root>

**JSON response example in case of success (code 200):**

{

"apiversion": 3,

"result": [

{

"identified\_by": [

{

"value": "0000000000001",

"type": "ntin",

"multiplier": 1,

"level": "trade-unit"

},

{

"value": "0000000000001",

"type": "gtin",

"multiplier": 1,

"level": "trade-unit"

},

{

"value": "0000000000002",

"type": "gtin",

"multiplier": 28,

"level": "box"

}

],

"is\_kit": false,

"is\_set": false,

"set\_gtins": [],

"update\_date": "2019-09-26 17:27:21",

"producer\_inn": null,

"producer\_name": null,

"categories": [

{

"cat\_id": 30302,

"cat\_name": "Кетчуп"

},

{

"cat\_id": 206309,

"cat\_name": "соусы"

}

],

"good\_images": [

{

"photo\_type": "facing",

"photo\_date": "2019-02-22T19:26:56+03:00",

"photo\_url": "string",

"barcode": "0000000000001"

}

],

"good\_attrs": [

{

"attr\_id": 2478,

"attr\_name": "Полное наименование товара",

"attr\_value": "string",

"attr\_value\_type": "",

"attr\_group\_id": 24,

"attr\_group\_name": "Наименование товара и идентификация", "value\_id": 175, "published\_date": "2018-09-13T15:44:32+03:00",

"gtin": null,

"multiplier": null,

"level": ""

},

{

"attr\_id": 2504,

"attr\_name": "Товарный знак",

"attr\_value": "string",

"attr\_value\_id": 199,

"value\_id": 160752,

"attr\_value\_type": "",

"attr\_group\_id": 80,

"attr\_group\_name": "Происхождение и бренды",

"published\_date": "2018-09-13T15:46:54+03:00"

},

{

"attr\_id": 3793,

"attr\_name": "Базовая единица измерения",

"attr\_value": "ШТ",

"attr\_value\_type": "",

"attr\_group\_id": 24,

"attr\_group\_name": "Наименование товара и идентификация",

"value\_id": 13973319,

"published\_date": "2019-02-22T19:27:42+03:00",

"gtin": null,

"multiplier": null,

"level": ""

}

],

"remainder\_type": null,

"is\_tech\_gtin": false

}

] }

**XML response example in case of success (code 200):**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<apiversion>3</apiversion>

<result>

<item>

<identified\_by>

<item>

<value>0000000000001</value>

<type>gtin</type>

<multiplier>1</multiplier> <level>trade-unit</level>

</item>

<item>

<value>0000000000002</value>

<type>gtin</type>

<multiplier>28</multiplier>

<level>box</level>

</item>

</identified\_by>

<is\_kit/>

<is\_set/>

<set\_gtins/>

<update\_date>2019-09-26 17:27:21</update\_date>

<producer\_inn></producer\_inn>

<producer\_name></producer\_name>

<categories>

<item>

<cat\_id>30302</cat\_id>

<cat\_name>Кетчуп</cat\_name>

</item>

<item>

<cat\_id>206309</cat\_id>

<cat\_name>соусы</cat\_name>

</item>

</categories>

<good\_images>

<item>

<photo\_type>facing</photo\_type>

<photo\_date>2019-02-22T19:26:56+03:00</photo\_date>

<photo\_url>string</photo\_url>

<barcode>0000000000001</barcode>

</item>

</good\_images>

<good\_attrs>

<item>

<attr\_id>2478</attr\_id>

<attr\_name>Полное наименование товара</attr\_name>

<attr\_value>string</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>24</attr\_group\_id> <attr\_group\_name>Наименование товара и идентификация</attr\_group\_name>

<value\_id>175</value\_id>

<published\_date>2018-09-13T15:44:32+03:00</published\_date>

<gtin></gtin>

<multiplier></multiplier>

<level></level>

</item>

<item>

<attr\_id>2504</attr\_id>

<attr\_name>Товарный знак</attr\_name>

<attr\_value>string</attr\_value>

<attr\_value\_id>199</attr\_value\_id>

<value\_id>160752</value\_id>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>80</attr\_group\_id>

<attr\_group\_name>Происхождение и бренды</attr\_group\_name> <published\_date>2018-09-13T15:46:54+03:00</published\_date>

</item>

<item>

<attr\_id>3793</attr\_id>

<attr\_name>Базовая единица измерения</attr\_name>

<attr\_value>ШТ</attr\_value>

<attr\_value\_type></attr\_value\_type>

<attr\_group\_id>24</attr\_group\_id> <attr\_group\_name>Наименование товара и идентификация</attr\_group\_name>

<value\_id>13973319</value\_id>

<published\_date>2019-02-22T19:27:42+03:00</published\_date> <gtin></gtin>

<multiplier></multiplier>

<level></level>

</item>

</good\_attrs>

<remainder\_type/>

<is\_tech\_gtin/>

</item>

</result>

</root>

### Method “Get a list of your own cards with brief information about them”

The method “/v4/product-list” returns a list of goods belonging to the owner with brief information about them. The maximum number of goods items in the selection is 10,000. You can navigate through them using the “limit” (“Number of the records in the response”) and “offset” (“Shift relative to the start of the issue”) parameters.

**Note**:

* if none of the “from\_date” or “to\_date” parameters is passed in the request, then the method searches for cards updated one month ahead of the current date;
* if a company has more than 10,000 goods cards updated within the period specified in the “to\_date” and/or “from\_date” parameters, then a response with code 413 will be returned;
* if the “limit” (“Number of the records in the response”) and “offset” (“Shift relative to the start of the issue”) parameters are specified in the request, then their total value must not exceed 10,000, otherwise a response with code 413 will be returned;
* if the “limit” (“Number of the records in the response”) and “offset” (“Shift relative to the start of the issue”) parameters are not specified in the request, then “limit” (“Number of the records in the response”) is considered equal to 1,000, and “offset” (“Shift relative to the start of the issue”) is equal to 0;
* if the “from\_date” and “to\_date” parameters are simultaneously passed in the request, then the specified period can be more than a month.

**URL:** /v4/product-list

**Method:** GET

**Request string example:**

GET <url of environment>/v4/product-list?apikey=XXX&limit=ZZZ&offset=NNN

**Request parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | + | Identifier (key) of the goods owner | The parameter is mandatory if “token”  (“Authentication token”) is not specified |
| limit | integer | - | Number of the records in the response | Minimum allowed value is 1.  Maximum allowed value is 1,000 |
| offset | integer | - | Shift relative to the start of the issue | Minimum allowed value is 0 |
| to\_date | string | - | Date and time in format YYYYMM-DD HH:ii:ss | All “gtin” updated within a month **before** the specified date will be selected |
| Parameter | Type | Mandatory | Description | Comment |
| from\_date | string | - | Date and time in format YYYYMM-DD HH:ii:ss | All “gtin” updated within a month **after** the specified date will be selected |
| good\_status | string | - | Technological status of goods card | Possible values:  “draft” — draft;  “archived” — archived;  “published” — published |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| limit | integer | - | Maximum number of the records in the response |  |
| offset | integer | - | Shift relative to the start of the issue list |  |
| total | integer | - | Total number of records matching the given parameters | No more than 10,000 |
| goods | array | - | Array of goods cards |  |
| \*good\_id | integer | - | Goods card identifier |  |
| \*gtin | string | - | Goods code |  |
| \*good\_name | string | - | Goods name |  |
| \*tnved | string | - | FEACN group code |  |
| \*brand\_name | string | - | Trade mark name |  |
| \*good\_status | string | - | Technological status of goods card |  |
| \*good\_detailed\_status | array | - | Array of the current statuses of the goods card | Possible values:  “draft” — draft;  “moderation” — being moderated;  “errors” — to be changed; |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  |  | “notsigned” — awaiting for signing;  “published” — published |
| \*to\_date | string | - | Date of the last update of the goods card technological status |  |

**JSON response example in case of success (code 200):**

{

"apiversion": 4,

"result": {

"limit": 1,

"offset": 0,

"total": 1,

"goods": [

{

"good\_id": 720679,

"gtin": "0000000000001",

"good\_name": "Чешки детские",

"tnved": "6402",

"brand\_name": "string",

"good\_status": "draft",

"good\_detailed\_status": ["draft"],

"to\_date": "2020-08-18 10:57:18"

}

]

} }

**XML response example in case of success (code 200):**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<apiversion>4</apiversion>

<result>

<limit>1</limit>

<offset>0</offset>

<total>1</total>

<goods>

<item>

<good\_id>720679</good\_id>

<gtin>0000000000001</gtin>

<good\_name>Чешки детские</good\_name>

<tnved>6402</tnved>

<brand\_name>string</brand\_name>

<good\_status>draft</good\_status>

<good\_detailed\_status>

<item>draft</item>

</good\_detailed\_status>

<to\_date>2020-08-18 10:57:18</to\_date>

</item>

</goods>

</result>

</root>

### Method “Check changes in cards”

The method “etagslist” returns a list of all goods that belong to a certain owner together with a hash of the page content (for more information about hash, see section “HTTP ETag (Version control)”). It is intended to promptly receive the list of hashes, compare them with the local storage and update information on goods (method “product”) with modified hashes, i.e. there are new data. Thus, it is possible to identify goods, the cards of which have been modified.

**URL:** /v3/etagslist

**Method:** GET

**Request string example:**

GET <url of environment>/v3/etagslist?apikey=XXX&brand\_id=YYY&cat\_id=ZZZ&offset=NNN

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |
| brand\_id | integer | - | Trade mark identifier |  |
| owner\_inn | string | - | INN of the goods owner’s company |  |
| cat\_id | integer | - | Identifier of any products category |  |
| offset | integer | - | Shift relative to the start of the issue |  |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | Mandatory if "apikey" is not specified |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| goods\_count | integer | + | Number of products in the current response |  |
| Parameter | Type | Mandatory | Description | Comment |
| offset | integer | + | Shift relative to the start of the issue list |  |
| last\_product\_number | integer | + | Number of the last showed product with respect to the start of the issue list |  |
| total | integer | + | Total number of records that correspond to the specified parameters |  |
| goods | array | + | Goods and their hashes | Maximum 100 goods in one response |
| \*good\_id | integer | - | Goods item identifier |  |
| \*etag | string | - | Hash | For more details, see section “HTTP ETag (Version control)” |

**JSON response example in case of success (code 200):**

{

"apiversion":3,

"result": {

"goods\_count": 3,

"offset": 0,

"last\_product\_number": 3,

"total": 1100,

"goods": [

{

"good\_id": 3,

"etag": "32b3502ff24f7c30"

},

{

"good\_id": 4,

"etag": "8529021f8808aaa9"

},

{

"good\_id": 5,

"etag": "4c23e42d0384a622"

}

]

} }

**XML response example in case of success (code 200):**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<apiversion>3</apiversion>

<result>

<goods\_count>3</goods\_count>

<offset>0</offset>

<last\_product\_number>3</last\_product\_number>

<total>1100</total>

<goods>

<item>

<good\_id>3</good\_id>

<etag>32b3502ff24f7c30</etag>

</item>

<item>

<good\_id>4</good\_id>

<etag>8529021f8808aaa9</etag>

</item>

<item>

<good\_id>5</good\_id>

<etag>4c23e42d0384a622</etag>

</item>

</goods>

</result>

</root>

### Method “Check whether cards or FEACN codes belong to marked goods groups”

The method “mark-check” is used to determine whether the goods with specified goods code or FEACN codes belong to goods groups subject to marking.

One request may contain up to 100 goods codes and FEACN codes in total.

**URL:** /v3/mark-check

**Method:** POST **Request example:**

curl -X POST "<url of environment>/v3/mark-check?apikey=XXX"

-H "Content-Type: application/json; charset=utf-8"

--data-raw "{

"gtins":[

"00000000000001",

"00000000000002",

"00000000000003"

],

"tnveds":[

"63",

"6302"

] }"

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Request body parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| gtins | array | - | List of goods codes | The parameter is mandatory if “tnveds” (“List of FEACN of goods”) is not specified |
| tnveds | array | - | Array containing the FEACN codes or FEACN group codes | The parameter is mandatory if the “gtins” (“List of goods codes”) is not specified |

**Response parameters:**

A response is formed separately for each array.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| gtins | array | - | Array containing the received goods codes and FEACN codes to which goods codes are linked, information about marking, and a response code |  |
| \*gtin | string | + | Goods code | A goods code for which the goods markability attribute was searched |
| \*tnved | string | - | FEACN code | The FEACN code to which the goods code specified in the request is linked.  This parameter is specified only in the  “gtins” array. If there are no goods in the system, the FEACN code will not be returned |
| \*is\_marked | string | + | A response text that contains information about the need to mark goods with the specified goods code or FEACN code |  |
| \*is\_marked\_code | integer | + | Response codes | Possible values:  “0” — Goods not to be marked; “1” — Goods to be marked; “2” — Goods not found.    The following responses will be returned for remaining goods (goods with the goods codes that begin with “029”):  “0” — “Remaining items not to be marked”  “1” — “Remaining items to be marked” |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  |  | “2” — “Remaining items not found” |
| tnveds | array | - | Array that contains the received FEACN codes, marking information and response code |  |
| \*tnved | string | + | The searched FEACN code |  |
| \*is\_marked | string | + | Information about marking of goods item with the specified goods code or FEACN code |  |
| \*is\_marked\_code | integer | + | Response codes | Possible values:  “0” — “Goods with the specified  FEACN code are not to be marked”;  “1” — “Goods with the specified  FEACN code are to be marked”;  “2” — “FEACN not found”;  “3” — “It is impossible to determine whether marking is required for the specified FEACN code.  Please recheck the FEACN code” |

**JSON response example in case of success (code 200):**

{

"apiversion":3,

"result":{

"gtins":[

{

"gtin":"0000000000001",

"tnved":"3303",

"is\_marked":"Товар подлежит маркировке",

"is\_marked\_code":1

},

{

"gtin":"0000000000002",

"tnved":null,

"is\_marked":"Товар не найден",

"is\_marked\_code":2

},

{

"gtin":"0000000000003",

"tnved":null,

"is\_marked":"Товар не найден",

"is\_marked\_code":2

}

],

"tnveds":[

{

"tnved":"63",

"is\_marked":"По указанному коду ТНВЭД невозможно установить необходимость маркировки. Уточните код ТНВЭД товара",

"is\_marked\_code":3

},

{

"tnved":"6302",

"is\_marked":"Товар с указанным кодом ТНВЭД подлежит маркировке",

"is\_marked\_code":1

}

]

} }

**XML response example in case of success (code 200):**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<apiversion>3</apiversion>

<result>

<gtins>

<item>

<gtin>000000000001</gtin>

<tnved>3303</tnved> <is\_marked>Товар подлежит маркировке</is\_marked>

<is\_marked\_code>1</is\_marked\_code>

</item>

<item>

<gtin>0000000000002</gtin>

<tnved></tnved>

<is\_marked>Товар не найден</is\_marked>

<is\_marked\_code>2</is\_marked\_code>

</item>

<item>

<gtin>0000000000003</gtin>

<tnved></tnved>

<is\_marked>Товар не найден</is\_marked>

<is\_marked\_code>2</is\_marked\_code>

</item>

</gtins>

<tnveds>

<item>

<tnved>63</tnved>

<is\_marked>По указанному коду ТНВЭД невозможно установить необходимость маркировки. Уточните код ТНВЭД товара</is\_marked> <is\_marked\_code>3</is\_marked\_code>

</item>

<item>

<tnved>6302</tnved>

<is\_marked>Товар с указанным кодом ТНВЭД подлежит маркировке</is\_marked>

<is\_marked\_code>1</is\_marked\_code>

</item>

</tnveds>

</result>

</root>

## Create or edit a card

### Method “Generate a goods code”

The method “generate-gtins” generates goods codes drafts and returns their goods codes.

**URL:** /v3/generate-gtins

**Method:** GET

**Request string example:**

GET <url of environment>/v3/generate-gtins?apikey=XXX&quantity=3

or:

GET <url of environment>/v3/generate-gtins?apikey=XXX&quantity=3&supplier\_key=YYY

or:

GET <url of environment>/v3/generate-gtins?apikey=XXX&exist=1

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |
| quantity | string | - | Number of new draft goods codes to be generated | The parameter is mandatory if the “exist” (“Attribute that indicates a request for goods codes already generated and available in the user base”) is not specified |
| exist | boolean | - | Attribute that indicates a request for goods codes already generated and available in the user base | The parameter is mandatory if the “quantity” (“Number of new draft goods codes”) is not specified. Possible values:  “true” — yes; “false” — no.  When “exist” = true, no more than 10,000 records from previously generated goods codes will be returned in the response |
| supplier\_key | string | - | Key of supplier or manufacturer of goods | It is designated for laboratories and other suppliers of content that place goods cards in the owner’s account on behalf of the goods owner. In fact, “supplier\_key” is “api\_key” of account of a goods owner |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the | This parameter is mandatory if “apikey”  (“Identifier (key) of the goods owner”) is |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  | authentication token | not specified |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| monthly-  limit | array | - | Monthly limit |  |
| \*limit | string | + | Total number of draft goods codes that can be generated within one month |  |
| \*usage | string | + | Number of goods codes drafts that have already been generated during the current month |  |
| drafts | array | - | List of goods codes drafts |  |
| \*gtin | string | + | Goods code that was generated |  |

**JSON response example in case of success (code 200):**

{

"apiversion": 3,

"result": {

"monthly-limit": {

"limit": 100,

"usage": 6

},

"drafts": [

{

"gtin": "000000000001"

},

{

"gtin": "000000000002"

},

{

"gtin": "000000000003"

}

]

} }

**XML response example in case of success (code 200):**

<root>

<apiversion>3</apiversion>

<result>

<monthly-limit>

<limit>100</limit>

<usage>6</usage>

</monthly-limit>

<drafts>

<gtin>000000000001</gtin>

<gtin>000000000002</gtin>

<gtin>000000000003</gtin>

</drafts>

</result>

</root>

### Method “Create or edit a card”

The method “feed” allows companies – goods owners to create and update goods. The goods cards are created and updated by means of feeds — update packages formed on the side of the user.

|  |  |
| --- | --- |
| Important | When creating and updating goods cards, it is important to take into account the allowed Unicode characters |

The method returns “feed\_id” (identifier of the received feed) or an error.

Limitations:

* feed size - 25 MB;
* number of the goods in a feed - 500;
* number of goods codes in a set — 1,000.

Once the feed size limit or the feed goods number limit has been exceeded, error 413 is returned.

When an error occurs in a data scheme, an error with code 400 will be received.

|  |  |
| --- | --- |
| Note: | Information about sent update packages is stored for one year |

**URL:** /v3/feed

**Method:** POST

**Request string example:**

POST <url of environment>/v3/feed?apikey=XXX&supplier\_key=YYY

**Request parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |
| supplier\_key | string | - | Key of supplier or manufacturer of goods | It is designated for laboratories and other suppliers of content that place goods cards in the owner’s account on behalf of the goods owner. In fact, “supplier\_key” is “api\_key” of account of a goods owner |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

The feed is passed in the request body. The feed can be sent in two formats: JSON and XML.

The Header shall contain:

* "Content-Type: application/xml", if the feed has XML format;
* "Content-Type: application/json", if the feed has JSON format.

The feed is an array of objects called “entry”.

**Description of “entry”:**

To modify an existing goods item, a value of its identifier (“good\_id”) shall be sent. If there is no “good\_id” parameter, the goods item is defined as a new one. The “good\_name” and “gtin” parameters are mandatory for a new item.

**Request body parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| good\_id | integer | - | Goods item identifier | Mandatory for updated or edited goods |
| gtin | string | - | Goods code | Mandatory for a new goods item |
| is\_set | boolean | + | Attribute of creation of the card of goods of the "Set" type | It is mandatory when creating a card of the goods that form a set. Possible values:  “true” — yes;  “false” — no  When a card of the goods item with the “Set” type is described, it will be necessary to indicate the corresponding identifier of the goods group to which the goods card will relate in the attr\_id = 23768 "Goods group” (see “4.1. Catalog – List of supported goods groups”) |
| is\_tech\_gtin | boolean | + | Indicator of creation of the goods card with technical code of the goods | It is mandatory when creating a goods card with technical goods code (in this case the goods code is not specified in the “entry”).  Possible values:  “true” — yes;  “false” — no |
| good\_name | string | - | Goods name | Mandatory for a new goods item |
| is\_kit | boolean | + | Indicator of creation of the bundle | It is mandatory when creating a card of the goods that form a bundle. Possible values:  “true” — yes;  “false” — no |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| tnved | string | - | FEACN code | Mandatory for a new goods item.  If a four-digit code of the FEACN group has been registered in the National Catalog, it is indicated in the “tnved” attribute (in this case attribute 3959 is not filled in), and a ten-digit FEACN code is indicated in attribute 13933.  If a ten-digit FEACN code is registered in the National Catalog for a four-digit code of the FEACN group, then it is indicated in the “tnved” attribute (in this case attributes 3959 and 13933 are not filled in).  You can check whether a four-digit or ten-digit FEACN code is registered in the National Catalog using the method “markcheck” |
| brand | string | - | Goods item identifier | Mandatory for a new goods item |
| moderation | boolean | - | Indicator that a goods item has been sent for moderation | Possible values:  1 — a goods item is sent for moderation, a card is created with the “Being moderated” status;  0 or undefined — a goods item is not sent for moderation, a card is created with the “Draft” status |
| set\_gtins | array | - | Array of nestings in the set | It is specified only when is\_set=1 |
| \*gtin | string | + | A code of the goods item nested into the set | It is mandatory for all goods groups for which a set can be created, except the “Clothing items, bed, table, bath and kitchen linens” goods group |
| \*quantity | integer | - | Number of nestings | It is mandatory when a goods code is specified.  A whole number which is no less than 1 |
| identified\_by | array | - | Array of identifiers | It is mandatory if at least one of the conditions is met:   * the “good\_images” parameter (“Array of images”) is   filled in;   * a goods card is created with a goods code corresponding to a packing of the “trade-unit” type (“Consumer package”) — in such a case, an array of identifiers must contain at least an identifier of this package type (see Example 1).   In other cases, it is optional |
| \*type | string | + | Identifier type | Possible values:  “gtin” — goods code |
| \*value | string | + | A value of the goods code of the package |  |
| \*level | string | + | Package type | Possible values:  “trade-unit” — consumer; “inner-pack” — group consumer; “box” — shipping. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment | | |
|  |  |  |  | For all types of packages, except “trade-unit”, it is recommended to indicate explicitly in the attribute array:  “attr\_id” — attribute identifier (constant in the current implementation: “attr\_id” = 13763);  “attr\_value” — attribute value: a goods code of packing enclosure. The goods code of packing of the previous level is indicated (see Example 2);  “gtin” — a goods code of package.  The method will return an error when you try to create a group package card (“level” = “inner-pack”) if a goods item belongs to unmarked products, i.e. if a value of the “cat\_id” (“Category ID”) belongs to one of the values:  552786 — Smoking tobacco (unmarked products); | | |
|  |  |  |  | 552789 — Pipe tobacco (unmarked | | products); |
|  |  |  |  | 552792 — Shisha tobacco (unmarked | | products); |
|  |  |  |  | 552795 — Chewing tobacco (unmarked | | products); |
|  |  |  |  | 552798 — Snuff | (unmarked | products); |
|  |  |  |  | 552801 — Cigarettes | (unmarked | products); |
|  |  |  |  | 552804 — Mouthpiece | cigarettes (unmarked | products); |
|  |  |  |  | 552807 — Kretek | (unmarked | products); |
|  |  |  |  | 552810 — Cigars | (unmarked | products); |
|  |  |  |  | 552813 — Cigarillos | (unmarked | products); |
|  |  |  |  | 552816 — Beedi | (unmarked | products); |
|  |  |  |  | 552819 — Heated tobacco (unmarked products);  552822 — Tobacco-free hookah mix (unmarked products);  552825 —  Nicotine-containing liquids (unmarked products);  552828 — Disposable nicotine delivery systems (unmarked products);  552831 — Nicotine-free liquids (unmarked products) | | |
| \*multiplier | integer | + | Number of nestings in a package | It always equals 1 for the package type “trade-unit”  (“Consumer package”) | | |
| \*unit | string | - | Type of measurements | Possible values:  "pcs" — pieces | | |
| categories | array | - | Array of categories | Some FEACN codes may correspond to several goods groups in the National Catalog. For such FEACN codes you will need to select a category | | |
| \*cat\_id | integer | + | Category identifier |  | | |
| good\_attrs | array | - | Array of attributes |  | | |
| \*attr\_id | string | - | Attribute identifier | It is mandatory when creating a goods item | | |
| \*attr\_value | string | - | Attribute value | Optional when editing and deleting.  Values for multiplicity attributes are passed as separate entries (multiplicity attribute is an attribute for which several values can be set. For example, values of “RU”, “CH”, “BY”, etc. may be set in one goods card for the “Country of | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  |  | manufacture” attribute).  A composite value is expected to be transmitted through the delimiter “:::” in the format of “number:::date”, where number is a number of the permitting document, and date is a date of the permitting document in the format of “YYYYMM-DD”, for the attributes that have “attr\_id” (“Attribute identifier”) equal to:  “23555” (“Marketing authorization”);  “23890” (“Marketing authorization in the State Register”); “23561” (“Certificate of conformity”); “23557” (“Declaration of conformity”).  And there is no need to put spaces before and after the delimiter “:::” |
| \*attr\_value\_type | string | - | Attribute value type |  |
| \*gtin | string | - | Goods code | It is mandatory when describing attributes of the packages, except a package of the “trade-unit” type (“Consumer package”) |
| \*delete | boolean | - | Deletion attribute | It is available only when editing the existing goods.  Possible values:  “1” – deletion of the goods attribute.  When sending this parameter, “attr\_value” shall be specified with a deleted value |
| good\_images | array | - | Array of images |  |
| \*photo\_type | string | + | Image type | Possible values:  "default" — default photo (front view);  "7" — left-side photo of goods;  "19" — right-side photo of goods;  "13" — back-side photo of goods;  "si1" — top-view photo of goods;  "si2" — bottom-view photo of goods;  "si3" — photo of packaged goods;  "si4" — photo of goods without package;  "si5" — photo of group package;  "3ds" — 3D series;  "marketing" — marketing photo of goods;  "text" — photo of a text on goods;  "ecommerce" — e-commerce photo |
| \*photo\_url | string | + | A link to a photo | URL or an array of URLs when “photo\_type” (“Image type”) = “3ds”.  The links to photos with automatic redirect are not supported |
| \*identifier | string | - | Goods code | It is used to link photos with goods identifiers |
| \*identifier\_type | string | - | Identifier type | It is mandatory if “identifier” (“Goods code”) has been |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  |  | transferred. Possible values:  “gtin” — goods code |

**Example 1.** Example of the request for creation of the goods card with a goods code of the consumer package

curl -X POST "<url of environment>/v3/feed?supplier\_key=YYY"

-H "Content-Type: application/json; charset=utf-8"

-H "Content-length: 434"

-H "Authorization: Bearer <ТОКЕН>"

--data-raw "{

"gtin":"000000000001",

"tnved":"3303",

"brand":"string",

"categories":[

30120

],

"good\_name":"Полное наименование товара",

"identified\_by":[

{

"value":"000000000001",

"type":"gtin",

"multiplier":1,

"level":"trade-unit",

"unit":"шт"

}

],

"good\_attrs":[

{

"attr\_id":2630,

"attr\_value":"RU"

},

{

"attr\_id":2716,

"attr\_value":"50",

"attr\_value\_type":"мл"

}

] }"

**Example 2.** Example of the request for creation of the goods card with goods codes of packages of different levels (with nestings)

curl -X POST "<url of environment>/v3/feed?supplier\_key=YYY"

-H "Content-Type: application/json; charset=utf-8"

-H "Content-length: 434"

-H "Authorization: Bearer <ТОКЕН>"

--data-raw "{

"gtin":"000000000001",

"tnved":"3303",

"brand":"string",

"categories":[

30570

],

"good\_name":"Полное наименование товара",

"identified\_by":[

{

"value":"000000000001",

"type":"gtin",

"multiplier":1,

"level":"trade-unit",

"unit":"шт"

},

{

"value":"00000000000002",

"type":"gtin",

"multiplier":10,

"level":"inner-pack",

"unit":"шт"

},

{

"value":"00000000000003",

"type":"gtin",

"multiplier":100,

"level":"box",

"unit":"шт"

}

],

"good\_attrs":[

{

"attr\_id":13763,

"attr\_value\_type":"gtin",

"attr\_value":"000000000001",

"gtin":"00000000000002"

},

{

"attr\_id":13763,

"attr\_value\_type":"gtin",

"attr\_value":"00000000000002",

"gtin":"00000000000003"

},

{

"attr\_id":4424,

"attr\_value ":"Наименование упаковки товара",

"gtin":"00000000000002"

},

{

"attr\_id":2439,

"attr\_value":"40",

"gtin":"00000000000002"

},

{

"attr\_id":2437,

"attr\_value":"15",

"gtin":"00000000000002"

},

{

"attr\_id":2438,

"attr\_value":"40",

"gtin":"00000000000002"

},

{

"attr\_id":2710,

"attr\_value":"КОРОБКА/БОКС",

"gtin":"00000000000002"

},

{

"attr\_id":2713,

"attr\_value":"КАРТОН",

"gtin":"00000000000002"

}

] }"

**Example 3.** Example of the request for creation of the goods card with a technical goods code in \* .json format

curl -X POST "<url of environment>/v3/feed?supplier\_key=YYY"

-H "Content-Type: application/json; charset=utf-8"

-H "Content-length: 434"

-H "Authorization: Bearer <ТОКЕН>"

--data-raw "{

"is\_tech\_gtin":1,

"tnved":"3303",

"brand":"string",

"good\_name":"Полное наименование товара",

"good\_attrs":[

{

"attr\_id":2630,

"attr\_value":"RU"

},

{

"attr\_id":2716,

"attr\_value":"50",

"attr\_value\_type":"мл"

}

] }"

**Example 4.** Example of the request for creation of new goods card in \* .json format

curl -X POST "<url of environment>/v3/feed?supplier\_key=YYY"

-H "Content-Type: application/json; charset=utf-8"

-H "Content-length: 434"

-H "Authorization: Bearer <ТОКЕН>"

--data-raw "{

"gtin":"00000000000001",

"good\_name":"Шоколад с Цельным Миндалём 55% 90г 14шт шоу-бокс",

"identified\_by":[

{

"value":"00000000000001",

"type":"gtin",

"multiplier":1,

"level":"trade-unit",

"unit":"кг"

}

],

"categories":[

{

"cat\_id":30120

}

],

"good\_images":[

{

"photo\_type":"default",

"photo\_url":"https://your-site-name.com/photo.jpg",

"identifier":"00000000000001",

"identifier\_type":"gtin"

},

{

"photo\_type":"3ds",

"photo\_url":[

"https://your-site-name.com/photo-1.jpg", "https://your-site-name.com/photo-2.jpg"

]

}

],

"good\_attrs":[

{

"attr\_id":2630,

"attr\_value":"RU"

},

{

"attr\_id":2501,

"attr\_value":"ДА",

"attr\_value\_type":"кг"

}

]

}"

**Example 5.** Example of the request for creation of new goods card in \* . xml format

curl -X POST "<url of environment>/v3/feed?supplier\_key=YYY"

-H "Content-Type: application/xml; charset=utf-8"

-H "Content-length: 511"

-H "Authorization: Bearer <ТОКЕН>"

--data-raw "

<?xml version="1.0" encoding="UTF-8"?>

<entries>

<entry>

<gtin>00000000000001</gtin>

<tnved>0403</tnved>

<good\_name>Йогурт легкий. Злаки + чернослив</good\_name>

<identified\_by>

<item>

<value>00000000000001</value>

<type>gtin</type>

<multiplier>1</multiplier>

<level>trade-unit</level>

<unit>кг</unit>

</item>

</identified\_by>

<categories>

<item>

<cat\_id>230988</cat\_id>

</item>

</categories>

<good\_images>

<item>

<photo\_type>default</photo\_type>

<photo\_url>https://your-site-name.com/photo.jpg</photo\_url>

<identifier>00000000000001</identifier>

<identifier\_type>gtin</identifier\_type>

</item>

<item>

<photo\_type>3ds</photo\_type>

<photo\_url>

<item>https://your-site-name.com/photo.jpg</item>

<item>https://your-site-name.com/photo.jpg</item>

</photo\_url>

</item>

</good\_images>

<good\_attrs>

<item>

<attr\_id>2630</attr\_id>

<attr\_value>RU</attr\_value>

</item>

<item>

<attr\_id>15448</attr\_id>

<attr\_value>200</attr\_value>

<attr\_value\_type>мл</attr\_value\_type>

</item>

</good\_attrs>

</entry> </entries>

"

**Example 6.** Example of the request for creation of new goods card of the “Set” type in \* .json format

curl -X POST "<url of environment>/v3/feed?supplier\_key=YYY"

-H "Content-Type: application/json; charset=utf-8"

-H "Content-length: 434"

-H "Authorization: Bearer <ТОКЕН>"

--data-raw "{

"gtin":"0000000000004",

"tnved":"3303",

"moderation":1,

"brand":"тест",

"good\_name":"Духи",

"is\_set":true,

"good\_attrs":[

{

"attr\_id":16271,

"attr\_value":"еще салфеточка"

},

{

"attr\_id":23768,

"attr\_value":4

}

],

"set\_gtins":[

{

"quantity":20,

"gtin":"460000000"

},

{

"quantity":5,

"gtin":"460000005"

}

] }"

**Example 7.** Example of the request for creation of new goods card of the “Set” type in \* . xml format

curl -X POST "<url of environment>/v3/feed?supplier\_key=YYY"

-H "Content-Type: application/xml; charset=utf-8"

-H "Content-length: 434"

-H "Authorization: Bearer <ТОКЕН>"

--data-raw "

<?xml version="1.0" encoding="UTF-8"?>

<entries>

<entry>

<gtin>0000000000004</gtin>

<tnved>3303</tnved>

<moderation>1</moderation>

<brand>тест</brand>

<good\_name>Духи</good\_name>

<is\_set>1</is\_set>

<good\_attrs>

<item>

<attr\_id>16271</attr\_id>

<attr\_value>еще салфеточка</attr\_value>

</item>

<item>

<attr\_id>23768</attr\_id>

<attr\_value>4</attr\_value>

</item>

</good\_attrs>

<set\_gtins>

<item>

<quantity>20</quantity>

<gtin>460000000</gtin>

</item>

<item>

<quantity>5</quantity>

<gtin>460000005</gtin>

</item>

</set\_gtins>

</entry> </entries>

"

**Example 8.** Example of the request for update of the previously created goods card in \* .json format

curl -X POST "<url of environment>/v3/feed?supplier\_key=YYY"

-H "Content-Type: application/json; charset=utf-8"

-H "Content-length: 434"

-H "Authorization: Bearer <ТОКЕН>"

--data-raw "{

"good\_id":1939447, "identified\_by":[

{

"value":"0000000000001",

"type":"gtin",

"multiplier":1,

"level":"trade-unit"

}

],

"good\_attrs":[

{

"attr\_id":5,

"attr\_value":"46%"

},

{

"attr\_id":4540,

"attr\_value":"ДА"

},

{

"attr\_id":15290,

"delete":1

}

] }"

**Example 9.** Example of the request for update of previously created goods card in \* . xml format

curl -X POST "<url of environment>/v3/feed?supplier\_key=YYY"

-H "Content-Type: application/json; charset=utf-8"

-H "Content-length: 434"

-H "Authorization: Bearer <ТОКЕН>"

--data-raw "

<?xml version="1.0" encoding="UTF-8"?>

<entries>

<entry>

<good\_id>1939447</good\_id>

<identified\_by>

<item>

<value>0000000000001</value>

<type>gtin</type>

<multiplier>1</multiplier> <level>trade-unit</level>

</item>

</identified\_by>

<good\_images>

<item>

<photo\_type>default</photo\_type>

<photo\_url>https://your-site-name.com/photo.jpg</photo\_url>

</item>

</good\_images>

<good\_attrs>

<item>

<attr\_id>5</attr\_id>

<attr\_value>46%</attr\_value>

</item>

<item>

<attr\_id>15448</attr\_id>

<attr\_value>1.0</attr\_value>

<attr\_value\_type>л</attr\_value\_type>

</item>

<item>

<attr\_id>15290</attr\_id>

<delete>1</delete>

</item>

</good\_attrs>

</entry> </entries> "

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| feed\_id | integer | + | Feed identifier |  |

**JSON response example in case of success (code 200):**

{

"apiversion":3,

"result": {

"feed\_id": "2131"

} }

**XML response example in case of success (code 200):**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<apiversion>3</apiversion>

<result>

<feed\_id>2131</feed\_id>

</result>

</root>

### Method “Check update package processing status”

The method “feed-status” is intended to verify a status of the previously sent update package (feed). A result can be retrieved only for the feeds that have been sent by a company (laboratory).

|  |  |
| --- | --- |
| Attention | If the update package has the “Processing” status for more than a day, it will be automatically transferred to the “Rejected” status and will need to be sent for processing again |

Information on the sent update packages is stored for one year.

**URL:** /v3/feed-status

**Method:** GET

**Request string example:**

GET <url of environment>/v3/feedstatus?apikey=XXX&supplier\_key=YYY&feed\_id=7126&verbose=false

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |
| feed\_id | string | + | Feed identifier |  |
| verbose | boolean | - | Response format | Possible values:  false or it is missing — a standard block of errors in the feed processing will be displayed in the response;  true — an extended response on occurred errors in the feed processing will be displayed in the response |
| supplier\_key | string | - | Key of supplier or manufacturer of goods | It is designated for laboratories and other suppliers of content that place goods cards in the owner’s account on behalf of the goods owner. In fact, “supplier\_key” is “api\_key” of account of a goods owner |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Response parameters (verbose parameter was specified in the request with true value):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| feed\_id | integer | + | Request ID |  |
| status | string | + | Current status | Possible values:  “Received” — a request was received, data are being moderated;  “Moderated” — goods moderation is completed;  “Signed” — goods approved by the moderator were signed;  “Rejected” — a request was rejected;  “Processing” — a request was received and is |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  |  | awaiting for processing |
| status\_id | integer | + | Feed status identifier | Possible values:  “0” — a request was rejected;  “1” — a request was received, data are being moderated;  “2” — goods moderation is completed; “3” — goods approved by the moderator were signed;  “4” — a request was received and is awaiting for processing |
| received\_at | string | + | Feed creation time |  |
| status\_updated\_at | string | + | Time of transition of the feed to the current status |  |
| error\_details | object | - | Errors identified during the content validation |  |
| \*items | array | - | Array of errors in terms of goods codes |  |
| \*\*id | integer | - | Identifier of “entry” | Identifier of the card in the sent update package |
| \*\*gtin | string | - | Goods code |  |
| \*\*errors | object | - | List of errors for the goods code specified |  |
| \*\*\*code | integer | - | Numerical code of error |  |
| \*\*\*text | string | - | Error text |  |
| \*\*\*attr\_id | string | - | Identifier of the attribute in which the error was found |  |
| \*commonError | object | - | A common error, if any, when parsing information |  |
| \*\*code | integer | - | Numerical code of error |  |
| \*\*text | string | - | Error text |  |

**Response parameters (verbose parameter was specified in the request with false value or it was not specified):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| feed\_id | integer | + | Request ID |  |
| status | string | - | Current status | Possible values:  “Received” — a request was received, data are being moderated;  “Moderated” — goods moderation is completed;  “Signed” — goods approved by the moderator were signed;  “Rejected” — a request was rejected;  “Processing” — a request was received and is awaiting for processing |
| status\_id | integer | + | Feed status identifier | Possible values:  “0” — a request was rejected;  “1” — a request was received, data are being moderated;  “2” — goods moderation is completed;  “3” — goods approved by the moderator were signed;  “4” — a request was received and is awaiting for processing |
| received\_at | string | + | Feed creation time |  |
| status\_updated\_at | string | + | Time of transfer of the feed to the current status |  |
| result | object |  | Errors identified during the content validation | The keys are serial numbers of goods in the sent information |
| \*@id | array |  | Identifier of “entry” | An identifier of the card in the sent update package.  An array containing information on the errors identified during processing of information on the goods card specified in a feed under the identifier |
| \*totalErrors | string | - | Total number of errors |  |
| item | array | - | Array containing information about errors | If goods are sent successfully, the parameter is not transferred |
| \*id | integer |  | Identifier of “entry” | Identifier of the card in the sent update package |
| \*gtin | string | - | Goods code | A goods code to which information on the error relates |
| \*good\_id | integer | - | Goods item identifier |  |
| \*attribute\_id | integer | - | Attribute identifier |  |
| \*attribute\_name | string | - | Attribute name |  |
| \*status\_code | integer | - | Numerical code of error |  |
| Parameter | Type | Mandatory | Description | Comment |
| \*status\_message | string | - | Status text |  |
| \*message | string | - | Message text |  |

**JSON response example in case of success (code 200):**

{

"apiversion":3,

"result": {

"feed\_id": 7126,

"status": "Received",

"status\_id": 1,

"received\_at":"2019-08-13T17:03:40Z",

"status\_updated\_at":"2019-08-14T12:03:40Z"

} }

**XML response example in case of success (code 200):**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<apiversion>3</apiversion>

<result>

<feed\_id>7126</feed\_id>

<status>Received</status>

<status\_id>1</status\_id>

<received\_at>2019-08-13T17:03:40Z</received\_at>

<status\_updated\_at>2019-08-14T12:03:40Z</status\_updated\_at>

</result> </root>

**Example of JSON response in case of success (code 200) if the request by feed method has been successfully processed, cards moderation is completed:**

{

"apiversion": 3,

"result": {

"feed\_id": 66707,

"status": "Moderated",

"status\_id": 2,

"received\_at": "2019-10-02T13:42:23Z",

"status\_updated\_at": "2019-10-02T13:51:35Z",

"item": [

{

"id": 0,

"gtin": null,

"good\_id": null,

"attribute\_id": "2716",

"attribute\_name": "Заявленный объем",

"status\_code": 5,

"status\_message": "Отменено",

"message": "Неверный объем"

},

{

"id": 1,

"gtin": "0000000000001",

"good\_id": "789817",

"attribute\_id": "2716",

"attribute\_name": "Заявленный объем",

"status\_code": 5,

"status\_message": "Отменено",

"message": "Неверный объем"

},

{

"id": 1,

"gtin": "0000000000001",

"good\_id": "789817",

"attribute\_id": "2716",

"attribute\_name": "Заявленный объем",

"status\_code": 5,

"status\_message": "Отменено",

"message": "Неверный объем"

},

{

"id": 3,

"gtin": null,

"good\_id": null,

"attribute\_id": "13918",

"attribute\_name": "Селектив", "status\_code": 5,

"status\_message": "Отменено",

"message": "Ошибка"

},

{

"id": 5,

"gtin": null,

"good\_id": null,

"attribute\_id": "13918",

"attribute\_name": "Селектив", "status\_code": 5,

"status\_message": "Отменено",

"message": "Ошибка"

}

]

} }

**Example of XML response in case of success (code 200) if the request by feed method has been successfully processed, cards moderation is completed:**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<apiversion>3</apiversion>

<result>

<feed\_id>66707</feed\_id>

<status>Moderated</status>

<status\_id>2</status\_id>

<received\_at>2019-10-02T13:42:23Z</received\_at>

<status\_updated\_at>2019-10-02T13:51:35Z</status\_updated\_at>

<item>

<item>

<id>0</id> <gtin/>

<good\_id/>

<attribute\_id>2716</attribute\_id>

<attribute\_name>Заявленный объем</attribute\_name>

<status\_code>5</status\_code>

<status\_message>Отменено</status\_message>

<message>Неверный объем</message>

</item>

<item>

<id>1</id>

<gtin>0000000000001</gtin>

<good\_id>789817</good\_id>

<attribute\_id>2716</attribute\_id>

<attribute\_name>Заявленный объем</attribute\_name>

<status\_code>5</status\_code>

<status\_message>Отменено</status\_message>

<message>Неверный объем</message>

</item>

<item>

<id>1</id>

<gtin>0000000000001</gtin>

<good\_id>789817</good\_id>

<attribute\_id>2716</attribute\_id>

<attribute\_name>Заявленный объем</attribute\_name>

<status\_code>5</status\_code>

<status\_message>Отменено</status\_message>

<message>Неверный объем</message>

</item>

<item>

<id>3</id> <gtin/>

<good\_id/>

<attribute\_id>13918</attribute\_id>

<attribute\_name>Селектив</attribute\_name>

<status\_code>5</status\_code>

<status\_message>Отменено</status\_message>

<message>Ошибка</message>

</item>

<item>

<id>5</id> <gtin/>

<good\_id/>

<attribute\_id>13918</attribute\_id>

<attribute\_name>Селектив</attribute\_name>

<status\_code>5</status\_code>

<status\_message>Отменено</status\_message>

<message>Ошибка</message>

</item>

</item>

</result>

</root>

**Example of JSON response in case of success (code 200) if the request by feed method has been rejected:**

{

"apiversion":3,

"result": {

"feed\_id": 71206,

"status": "Rejected",

"received\_at": "2020-12-06T13:03:20Z",

"status\_updated\_at": "2020-12-06T13:03:30Z",

"result": {

"0": [

"Указанный ТН ВЭД не относится к числу ваших активных товарных групп в

Едином Личном Кабинете ГИС МТ. Пожалуйста, проверьте настройки или обратитесь в службу технической поддержки "

],

"totalErrors": "1"

}

} }

**Example of XML response in case of success (code 200) if the request by feed method has been rejected:**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<apiversion>3</apiversion>

<result>

<feed\_id>71206</feed\_id>

<status>Rejected</status>

<received\_at>2020-08-13T17:03:40Z</received\_at>

<status\_updated\_at>2020-08-14T12:03:40Z</status\_updated\_at>

<result>

<item>Указанный ТН ВЭД не относится к числу ваших активных товарных групп в Едином Личном Кабинете ГИС МТ. Пожалуйста, проверьте настройки или обратитесь в службу технической поддержки</item>

<item>Атрибут #44 недоступен для редактирования</item>

</result>

<totalErrors>2</totalErrors>

</result>

</root>

### Method “Change the photo size”

The method “image” allows to modify proportionally image size and to fill in missing fields with white color. The method accepts and returns JPEG format.

**URL:** /v3/image

**Method:** GET

**Request string example:**

GET <url of environment>/v3/image?apikey=XXX&name=https://национальныйкаталог.рф/i/300x200/5a7eb614-13d3-69ed-caf7-420624d1bdd3.jpg&width=300&height=400

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |
| name | string |  | Full URI of the image received in the API response |  |
| width | integer | - | Width of an output image | It shall be within [100, 1000] (resulting height in pixels) |
| height | integer | - | Height of an output image | It shall be within [100, 1000] (resulting width in pixels) |
| nobackground | boolean | - | Attribute of disabling the background | Possible values:  true — do not add background; false — add white background (default value).  White background will be added to the resulting image by default in order to strictly comply with the required “width” and “height” sizes; however, if “no-background” = true, no background will be added |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Response:**

The method response is a photo with modified sizes.

### Method “Forcibly send a card for moderation”

The method “feed-moderation” is used to send the previously created goods card in the "Draft" status for moderation for the “good\_id” parameter specified in the request or for a group of parameters (“gtin” and “inn”).

**URL:** /v3/feed-moderation

**Method:** GET

**Request string example:**

GET <url of environment>/v3/feed-moderation?apikey=XXX&good\_id=7126

or:

GET <url of environment>/v3/feedmoderation?apikey=XXX&gtin=00000000000001&inn=123455678

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |
| good\_id | string | - | Goods item identifier | The parameter is mandatory if “gtin” and “inn” are missing |
| gtin | string | - | Goods code | The parameter is mandatory if “good\_id” is missing |
| inn | string | - | INN of account | The parameter is mandatory if “good\_id” is missing |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| good\_ id | string | + | Goods item identifier |  |
| error | string | + | Error text | It returns in case of error |

**Response example in \* .json format:**

{

"apiversion": 3,

"result": {

"good\_id": 123456,

"error": "Черновик GTIN 02XXXXXXXX601050000383 не в том статусе."

}

}

## Getting information about attributes

### Method “Get a tree of categories”

The method “categories” is used to get a tree of categories (the tree root is not returned), to get a tree of subcategories of a given category (the given category will also be returned), get categories related to a specified goods group, or get categories related to a specified FEACN code.

**URL:** /v3/categories

**Method:** GET

**Request string example:**

GET <url of environment>/v3/categories?apikey=XXX

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |
| cat\_id | string | - | Category identifier | It can be used together with “gismt\_code” (“Code of the marked goods group”) |
| gismt\_code | string | - | Code of the marked goods group | For more details, see “4.1. Catalog – List of supported goods groups”.  It can be used together with “cat\_id” (“Category identifier”) |
| tnved | string | - | FEACN code | It cannot be used together with “gismt\_code” (“Code of the marked goods group”) and/or “cat\_id” (“Category identifier”).  The FEACN code specified must be active in the National catalog, must be a code of the final level and contain from 4 to 10 digits |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| cat\_id | integer | + | Category identifier |  |
| cat\_name | string | + | Category name |  |
| cat\_parent\_id | integer | + | Parent category identifier |  |
| cat\_level | integer | + | Level in the category tree | Possible values:  “1” — top level;  “2” — underlying level; and so on |
| category\_active | boolean | + | Attribute of category activity | Possible values:  “true” — yes;  “false” — no |
| Parameter | Type | Mandatory | Description | Comment |
| gismt\_codes | array[integer] | - | A code of the marked goods group, in the presence of dependence | A list of possible values can be found in "4.1. Catalog – List of supported goods groups" |

**JSON response example in case of success (code 200):**

{

"apiversion": 3,

"result": [

{

"cat\_id": 30064, "cat\_name": "Продовольственные товары",

"cat\_parent\_id": 30062,

"cat\_level": 2,

"category\_active": false,

"gismt\_codes": []

},

{

"cat\_id": 30066,

"cat\_name": "Косметика и парфюмерия",

"cat\_parent\_id": 30062,

"cat\_level": 2,

"category\_active": false,

"gismt\_codes": []

},

{

"cat\_id": 30068,

"cat\_name": "Одежда, Обувь, Персональные принадлежности",

"cat\_parent\_id": 30062,

"cat\_level": 2,

"category\_active": false,

"gismt\_codes": []

}

] }

**XML response example in case of success (code 200):**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<apiversion>3</apiversion>

<result>

<item>

<cat\_id>30068</cat\_id>

<cat\_name>Косметика и парфюмерия</cat\_name>

<cat\_parent\_id>30062</cat\_parent\_id>

<cat\_level>2</cat\_level>

<category\_active></category\_active>

<gismt\_codes/>

</item>

<item>

<cat\_id>30068</cat\_id>

<cat\_name>Одежда, Обувь, Персональные принадлежности</cat\_name> <cat\_parent\_id>30062</cat\_parent\_id>

<cat\_level>2</cat\_level>

<category\_active></category\_active>

<gismt\_codes/>

</item>

</result>

</root>

### Method “Get an attribute list”

The method “attributes” returns a list of attributes to create a goods card with a specified FEACN code or category identifier. If the FEACN code or category identifier is not specified, a complete list of the attributes that are available for the account from which a request is made, is returned.

If FEACN does not uniquely determine a goods group, to obtain a correct list of the attributes, you should first request a list of categories corresponding to this FEACN (see “categories”), then request an attribute model by using this method (“attributes”) by an identifier of the appropriate category (“cat\_id” parameter).

Up-to-date information on the marked goods groups and FEACN codes is presented on the website честныйзнак.рф.

**URL:** /v3/attributes

**Method:** GET

**Request string example:**

Example of the request with a category identifier:

GET <url of environment>/v3/attributes?apikey=XXX&cat\_id=30933&attr\_type=m

Example of the request with FEACN code:

GET <url of environment>/v3/attributes?apikey=XXX&tnved=3303&attr\_type=m

Example of the request with an indicator of the card of the goods that have the “Set” type:

GET <url of environment>/v3/attributes?apikey=XXX&is\_set=1

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |
| cat\_id | integer | - | Category identifier | A category identifier for which a set of attributes is requested.  It is mandatory if “attr\_type” (“Attribute type”) parameter is specified |
| tnved | string | - | FEACN code | It cannot be used together with “cat\_id” (“Category identifier”).  It is mandatory if “attr\_type” (“Attribute type”) parameter is |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  |  | specified.  The FEACN code specified must be active in the National catalog, must be a code of the final level and contain from 4 to 10 digits |
| is\_set | boolean | - | Indicator of the card of the goods that have the “Set” type | It is used to receive a set of attributes |
| attr\_type | string | - | Attribute type | If “attr\_type” parameter is specified, then either “cat\_id”, or “tnved”, or “is\_set” shall be indicated. Available values:  "a" — (default value) return all attributes;  "m" — return mandatory attributes only;  "r" — return recommended attributes only;  "o" — return optional attributes only |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| attr\_group\_id | integer | + | Identifier of the group to which an attribute belongs |  |
| attr\_name | string | + | Attribute name |  |
| preset\_url | string | - | A link to a method to receive an array of possible attribute values |  |
| attr\_preset\_only | boolean | + | Attribute indicating that only listed values of the preset are used | Possible values:  “true” — yes;  “false” — no |
| attr\_multiplicity | boolean | + | Attribute of multiplicity attribute | Possible values:  “true” — yes; “false” — no.  Multiplicity attribute is an attribute for which several values can be set. For example, values of  “RU”, “CH”, “BY”, etc. may be set in one goods card for the “Country of manufacture” |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  |  | attribute. |
| attr\_multiplicity\_type | string | + | Type of attribute multiplicity | Possible values:  “regular” is a multiplicity attribute without a check of the attribute type for uniqueness (several attribute values with the same type can be specified when describing a goods card); “unique” is a multiplicity attribute with a check of the type for uniqueness (several attribute values can be specified when describing a goods card, but their types shall be different);  “null” - nonmultiplicative attributes do not have multiplicity type |
| attr\_id | integer | + | Attribute identifier |  |
| attr\_group\_name | string | + | Name of the group to which an attribute belongs |  |
| attr\_field\_type | string | - | Attribute value type | Possible values:  “number”;  “text”;  “date” |
| attr\_value\_type | array | - | Array of possible values of the attribute type |  |
| dependent\_attributes | array | - | Array of context-sensitive attributes |  |
| \*value | string | - | When selecting this value, the  attributes from the “atters” array become mandatory |  |
| \*atters | array | - | Array of attributes |  |
| \*\*attr\_id | integer | - | Attribute identifier |  |
| \*\*first\_layer | boolean | + | Indicator that the attribute belongs to the first layer of attributes (attributes required to order the marking code) | Possible values:  “true” — yes;  “false” — no |
| \*\*second\_layer | boolean | + | Indicator that the attribute belongs to the second layer of attributes (attributes required to introduce goods into circulation) | Possible values:  “true” — yes;  “false” — no |
| \*\*attr\_type | string | - | Attribute type |  |
| first\_layer | boolean | + | Indicator that the attribute belongs to the first layer of attributes (attributes required | It accepts values true/false |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  | to order the marking code) |  |
| second\_layer | boolean | + | Indicator that the attribute belongs to the second layer of attributes | It accepts values true/false |
| attr\_type | string | - | Attribute type | This parameter is returned if “cat\_id” or “tnved” parameter is specified in the request |
| attr\_preset | array | - | Array of possible values of the attribute |  |

If “attr\_id” = 15653 (“Subject to veterinary checks (supervision)” attribute is filled in with “YES” value, filling of other attributes of “Data for FGIS VetIS” group automatically becomes mandatory.

When you fill in the attribute of “attr\_id” = 23768 “Goods group”, it is necessary to specify one of identifiers of the goods group in a body of the request to the method feed (see “4.1. Catalog – List of supported goods groups”).

**JSON response example in case of success (code 200):**

{

"apiversion": 3,

"result": [

{

"attr\_group\_id": 103,

"attr\_name": "Тип парфюмерии",

"attr\_preset\_only": false,

"attr\_multiplicity": false,

"attr\_multiplicity\_type": null,

"attr\_id": 1034,

"attr\_group\_name": "Потребительские свойства",

"attr\_field\_type": "text",

"attr\_value\_type": [],

"dependent\_attributes": [],

"first\_layer": false,

"second\_layer": true,

"attr\_type": "m",

"attr\_preset": [

"ДУХИ",

"ЛАВАНДОВАЯ ВОДА",

"ОДЕКОЛОН",

"ПАРФЮМЕРНАЯ ВОДА",

"СПРЕЙ ДЛЯ ТЕЛА",

"ТУАЛЕТНАЯ ВОДА",

"ДУШИСТАЯ ВОДА"

]

},

{

"attr\_group\_id": 24,

"attr\_name": "Полное наименование товара",

"attr\_preset\_only": false,

"attr\_multiplicity": false,

"attr\_multiplicity\_type": null,

"attr\_id": 2478,

"attr\_group\_name": "Идентификация товара",

"attr\_field\_type": "text",

"attr\_value\_type": [],

"dependent\_attributes": [],

"first\_layer": true,

"second\_layer": false,

"attr\_type": "m",

"attr\_preset": []

},

{

"attr\_group\_id": 24,

"attr\_name": "Товарный знак",

"attr\_preset\_only": false,

"attr\_multiplicity": false,

"attr\_multiplicity\_type": null,

"attr\_id": 2504,

"attr\_group\_name": "Идентификация товара",

"attr\_field\_type": "text",

"attr\_value\_type": [],

"dependent\_attributes": [],

"first\_layer": true,

"second\_layer": false,

"attr\_type": "m",

"attr\_preset": []

}

] }

**XML response example in case of success (code 200):**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<apiversion>3</apiversion>

<result>

<item>

<attr\_group\_id>103</attr\_group\_id>

<attr\_name>Тип парфюмерии</attr\_name>

<attr\_preset\_only></attr\_preset\_only>

<attr\_multiplicity></attr\_multiplicity>

<attr\_multiplicity\_type></attr\_multiplicity\_type>

<attr\_id>1034</attr\_id>

<attr\_group\_name>Потребительские свойства</attr\_group\_name>

<attr\_field\_type>text</attr\_field\_type>

<attr\_value\_type/>

<dependent\_attributes/>

<first\_layer></first\_layer>

<second\_layer>1</second\_layer>

<attr\_type>m</attr\_type>

<attr\_preset>

<item>ДУХИ</item>

<item>ЛАВАНДОВАЯ ВОДА</item>

<item>ОДЕКОЛОН</item>

<item>ПАРФЮМЕРНАЯ ВОДА</item>

<item>СПРЕЙ ДЛЯ ТЕЛА</item>

<item>ТУАЛЕТНАЯ ВОДА</item>

<item>ДУШИСТАЯ ВОДА</item>

</attr\_preset>

</item>

<item>

<attr\_group\_id>24</attr\_group\_id>

<attr\_name>Полное наименование товара</attr\_name>

<attr\_preset\_only></attr\_preset\_only>

<attr\_multiplicity></attr\_multiplicity>

<attr\_multiplicity\_type></attr\_multiplicity\_type>

<attr\_id>2478</attr\_id>

<attr\_group\_name>Идентификация товара</attr\_group\_name>

<attr\_field\_type>text</attr\_field\_type>

<attr\_value\_type/>

<dependent\_attributes/>

<first\_layer>1</first\_layer>

<second\_layer></second\_layer>

<attr\_type>m</attr\_type>

<attr\_preset/>

</item>

<item>

<attr\_group\_id>24</attr\_group\_id>

<attr\_name>Товарный знак</attr\_name>

<attr\_preset\_only></attr\_preset\_only>

<attr\_multiplicity></attr\_multiplicity>

<attr\_multiplicity\_type></attr\_multiplicity\_type>

<attr\_id>2504</attr\_id>

<attr\_group\_name>Идентификация товара</attr\_group\_name>

<attr\_field\_type>text</attr\_field\_type>

<attr\_value\_type/>

<dependent\_attributes/>

<first\_layer>1</first\_layer>

<second\_layer></second\_layer>

<attr\_type>m</attr\_type>

<attr\_preset/>

</item>

</result>

</root>

### Method “Get a directory of countries of manufacture”

The method “isocountry” is used to obtain a directory of countries of manufacture that are registered in the National Catalog.

**URL:** /v3/isocountry

**Method:** GET

**Request string example:**

GET <url of environment>/v3/isocountry?apikey=XXX

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods | The parameter is mandatory if “token” (“Authentication token”) |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  | owner | is not specified |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| \_etag | string | + | Hash | For more details, see section “HTTP ETag (Version control)” |
| \_list | array | + | Array of objects |  |
| \*сountry\_iso | string | - | Country code | Country code (in Alpha-2 encoding) |
| \*сountry\_name | string | - | Country name |  |

**JSON response example in case of success (code 200):**

{

"apiversion": 3,

"result": {

"\_etag": "0a23f98d522e7c05",

"\_list": [

{

"сountry\_iso": "AD",

"сountry\_name": "Андорра"

},

{

"сountry\_iso": "AE", "сountry\_name": "ОАЭ"

},

{

"сountry\_iso": "AF",

"сountry\_name": "Афганистан"

},

{

"сountry\_iso": "AG",

"сountry\_name": "Антигуа и Барбуда"

}

]

} }

**XML response example in case of success (code 200):**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<apiversion>3</apiversion>

<result>

<\_etag>0a23f98d522e7c05</\_etag>

<\_list>

<item>

<сountry\_iso>AD</сountry\_iso>

<сountry\_name>Андорра</сountry\_name>

</item>

<item>

<сountry\_iso>AE</сountry\_iso>

<сountry\_name>ОАЭ</сountry\_name>

</item>

<item>

<сountry\_iso>AF</сountry\_iso>

<сountry\_name>Афганистан</сountry\_name>

</item>

<item>

<сountry\_iso>AG</сountry\_iso>

<сountry\_name>Антигуа и Барбуда</сountry\_name>

</item>

</\_list>

</result>

</root>

### Method “Get a catalog of trademarks”

The method “brand” is used to get a list of trademarks (brands) registered in the National Catalog.

**URL:** /v3/brands

**Method:** GET

**Request string example:**

GET <url of environment>/v3/brands?apikey=XXX

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |
| name | string | - | Full or partial name of the trade mark |  |
| limit | integer | - | Number of the records in the response | Maximum allowed value is 10,000.  Minimum allowed value is 1 |
| offset | integer | - | Shift relative to the start of the issue | Minimum allowed value is 0 |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| brand\_id | integer | + | Trade mark identifier |  |
| name | string | + | Trade mark name |  |

**JSON response example in case of success (code 200):**

{

"apiversion": 3,

"result": [

{

"brand\_id": 8117,

"brand\_name": "string"

},

{

"brand\_id": 6262,

"brand\_name": "string"

},

{

"party\_brand\_id": "12345",

"brand\_id": 7105,

"brand\_name": "string"

},

{

"brand\_id": 6035,

"brand\_name": "string"

}

] }

**XML response example in case of success (code 200):**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<apiversion>3</apiversion>

<result>

<item>

<brand\_id>8117</brand\_id>

<brand\_name>string</brand\_name>

</item>

<item>

<brand\_id>6262</brand\_id>

<brand\_name>string</brand\_name>

</item>

<item>

<party\_brand\_id>12345</party\_brand\_id>

<brand\_id>7105</brand\_id>

<brand\_name>string</brand\_name>

</item>

<item>

<brand\_id>6035</brand\_id>

<brand\_name>string</brand\_name>

</item>

</result>

</root>

### Method “Check the presence of permit document in the directory”

The method “rd/suggest” is used to check the presence of the required permit document in the directory of the National catalog and the ability of its indication in the goods card.

**URL:** /v3/rd/suggest

**Method:** POST

**Request string example:**

POST <url of environment>/v3/rd/suggest?apikey=XXX

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Request body parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| attr\_id | integer | + | Identifier of the attribute of the permit document |  |
| number | string | + | Number of the permit document | Part of the permit document number or full number |
| model | string | - | Model name | Full or partial name of the model. Applicable for search for marketing authorization |
| search\_date\_from | boolean | - | Flag for searching for the date of | It is used only with the full number of the |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  | registration of the permit document | permit document.  Possible values:  “true” — 1;  “false” — 0 |

**Example of the request to search for MA by a part of the number:**

curl -X POST "<url of environment>/v3/rd/suggest?apikey=XXX"

-H "Content-Type: application/json; charset=utf-8"

--data-raw "{

"attr\_id": 23555, "number": "ФСЗ" }"

**Example of the request body to search for MA by full number:**

curl -X POST "<url of environment>/v3/rd/suggest?apikey=XXX"

-H "Content-Type: application/json; charset=utf-8"

--data-raw "{

"attr\_id": 23555,

"number": "ФСЗ 2011/11111",

"model": "бинт" }"

**Example of the request body to search for a certificate of conformity or declaration of conformity by full number:**

curl -X POST "<url of environment>/v3/rd/suggest?apikey=XXX"

-H "Content-Type: application/json; charset=utf-8"

--data-raw "{

"attr\_id": 23557,

"number": "ЕАЭС N RU Д-RU.РА01.В.35075/20",

"search\_date\_from": 1 }"

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| number | string | + | Number of permit document |  |
| models | array | - | Array of model names |  |
| \*model | string | - | Model name |  |
| dates | array | - | Array of dates |  |
| Parameter | Type | Mandatory | Description | Comment |
| \*date\_from | string |  | Date of registration of permit document |  |

**JSON response example in case of success (code 200):**

{

"apiversion":3,

"result":[

{

"number":"ФСЗ 2011/11111",

"models":[

{

"model":"бинт короткий"

},

{

"model":"бинт длинный"

}

]

}

]

}

### Method “Get information about a permitting document by a goods code and INN”

The method “v4/rd-info-by-gtin” allows to obtain full information about permitting documents (Declaration of conformity, Certificate of conformity, State registration certificate) that have been issued in the Russian Federation and in other EAEU countries and have been specified in a goods card that has the “Published” status.

**URL:** v4/rd-info-by-gtin

**Method:** POST **Request example:**

curl -X POST <url of environment>/v4/rd-info-by-gtin?apikey=XXX

--data-raw "{

"gtin": "00000000000001",

"inn": "0123456789"

}"

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | The parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |

**Request body parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| gtin | string | + | Goods code |  |
| inn | string | - | INN of the goods card owner |  |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| documents | array | + | A list of found permitting documents |  |
| \*attr\_id | integer | + | A type of the permitting document | Possible values: 23561 — Certificate of conformity; 23557 — Declaration of conformity; 23765 — State  registration certificate |
| \*number | string | + | Number of permitting document |  |
| \*from\_date | string | - | Date of drawing up the document |  |
| \*to\_date | string | - | Document expiration date |  |
| \*product\_type | integer | - | Type of permitting document | It is returned for:  declaration of conformity;  certificate of conformity |
| \*product\_name | string | - | Product name |  |
| \*product\_country | string | - | Country of origin of goods | It is returned for:  declaration of conformity;  certificate of conformity |
| \*applicant\_product\_name | string | - | Applicant name | It is returned for:  declaration of conformity; |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  |  | certificate of conformity |
| \*applicant\_product\_type | string | - | Applicant type | It is returned for:  declaration of conformity;  certificate of conformity |
| \*manufacturer\_product\_name | string | - | Manufacturer (producer) name |  |
| \*manufacturer\_product\_country | string | - | Country of manufacturer (producer) | Only for the state registration certificate |
| \*manufacturer\_product\_type | string | - | Manufacturer (producer) type | It is returned for:  declaration of conformity;  certificate of conformity |
| \*get\_product\_name | string | - | Recipient name | Only for the state registration certificate |
| \*get\_product\_country | string | - | Country of the recipient | Only for the state registration certificate |
| \*product\_tnved | string | - | EEU FEACN code | It is returned for:  declaration of conformity;  certificate of conformity |
| \*product\_identification | string | - | Data on products providing their identification | It is returned for:  declaration of conformity;  certificate of conformity |
| \*product\_tech\_regulations | string | - | Technical regulation (on the basis of which products were manufactured) | It is returned for:  declaration of conformity;  certificate of conformity |
| \*status | string | - | Status of the permitting document | It is returned for:  declaration of conformity;  certificate of conformity |
| \*active | boolean | - | Attribute of certificate activity | Only for the state registration certificate |
| \*update\_date | datetime | - | Date of change of the data in the TT GIS |  |
| errors | array | - | Array of errors |  |
| \*code | integer | - | Error code |  |
| Parameter | Type | Mandatory | Description | Comment |
| \*message | string | - | Error text |  |
| \*number | string | - | Number of permitting document |  |

**JSON response example in case of success (code 200):**

{

"apiversion":4,

"result":{

"documents":[

{

"attr\_id":"23557", "number":"ЕАЭС N RU Д-UZ.0000.В.0000/00",

"from\_date":"2022-10-06",

"to\_date":"2027-10-04",

"product\_type":"Декларация о соответствии требованиям технического регламента Евразийского экономического союза (технического регламента Таможенного союза)",

"product\_name":"string",

"product\_country":null,

"applicant\_product\_name":"string",

"applicant\_product\_type":"Индивидуальный предприниматель",

"manufacturer\_product\_name":"string",

"manufacturer\_product\_type":"Иностранное юридическое лицо",

"product\_tnved":"6110",

"product\_identification":null,

"product\_tech\_regulations":"ТР ТС 017/2011 О безопасности продукции легкой промышленности",

"status":"Действует",

"update\_date":"2024-03-11 11:06:34"

},

{

"attr\_id":"23765",

"number":"RU.00.00.00.000.R.000000.00.00",

"from\_date":"2024-01-29",

"to\_date":null,

"product\_name":"string",

"manufacturer\_product\_name":"string",

"manufacturer\_product\_country":null,

"get\_product\_name":"string", "get\_product\_country":null,

"active":true,

"update\_date":"2024-03-11 11:06:34"

}

]

} }

**In case of errors:**

* Error code 09: If the declaration of conformity or certificate of conformity is not found:

{

"errors\_message":"Документ <переданное значение number> от <переданное значение from\_date> не найден."

}

* Error code 10: If the state registration certificate is not found:

{

"errors\_message":"Документ <переданное значение number> не найден." }

* Error code 14: If the permitting document number does not match the required format:

{

"errors\_message":"<Текст ошибки>" }

* Error code 18: If a one-time request has been made for the declaration of conformity or certificate of conformity:

{

"errors\_message":"Документ <переданное значение number> от <переданное значение from\_date> не найден в справочнике. Направлен запрос в реестр ФОИВ.

Пожалуйста, осуществите попытку позже." }

* Error code 19: If a one-time request has been made for the state registration certificate:

{

"errors\_message":"Документ <переданное значение number> не найден в справочнике. Направлен запрос в реестр ФОИВ. Пожалуйста, осуществите попытку позже."

}

### Method “Get information about a permitting document by a number and date”

The method “v4/rd-info” allows to obtain full information about permitting documents (Declaration of conformity, Certificate of conformity, State registration certificate) issued in the Russian Federation and in other EAEU countries:

* by a number and date of the document (Declaration of conformity / Certificate of conformity);
* by a document number (State registration certificate).

An array of permitting documents is specified in the method request (no more than 25).

**URL:** v4/rd-info

**Method:** POST **Request example:**

curl -X POST <url of environment>/v4/rd-info?apikey=XXX --data-raw "[

{

"attr\_id": 23765,

"number": "RU.77.99.88.003.Е.005115.11.17"

},

{

"attr\_id": 23557,

"number": "ЕАЭС N RU Д-UZ.РА07.В.13720/22",

"from\_date": "2022-10-06"

}

]"

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | The parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |

**Request body parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| attr\_id | integer | + | Type of the permitting document | Possible values:  23561 — Certificate of conformity;  23557 — Declaration of conformity;  23765 — State registration certificate |
| number | string | + | Number of permitting document |  |
| from\_date | string | - | Date of drawing up the document | It is used only for:  Certificate of conformity (attr\_id = 23561); Declaration of conformity (attr\_id = 23557).  Format: "YYYY-MM-DD" |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| documents | array | + | A list of found permitting documents |  |
| \*attr\_id | integer | + | Type of the permitting document | Possible values:  23561 — Certificate of |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  |  | conformity; 23557 — Declaration of conformity; 23765 — State  registration certificate |
| \*number | string | + | Number of permitting document |  |
| \*from\_date | string | - | Date of drawing up the document |  |
| \*to\_date | string | - | Document expiration date |  |
| \*product\_type | integer | - | Type of permitting document | It is returned for:  declaration of conformity;  certificate of conformity |
| \*product\_name | string | - | Product name |  |
| \*product\_country | string | - | Country of origin of goods | It is returned for:  declaration of conformity;  certificate of conformity |
| \*applicant\_product\_name | string | - | Name of the applicant | It is returned for:  declaration of conformity;  certificate of conformity |
| \*applicant\_product\_type | string | - | Applicant type | It is returned for:  declaration of conformity;  certificate of conformity |
| \*manufacturer\_product\_name | string | - | Manufacturer (producer) name |  |
| \*manufacturer\_product\_country | string | - | Country of manufacturer (producer) | Only for the state registration certificate |
| \*manufacturer\_product\_type | string | - | Manufacturer (producer) type | It is returned for:  declaration of conformity;  certificate of conformity |
| \*get\_product\_name | string | - | Recipient name | Only for the state registration certificate |
| \*get\_product\_country | string | - | Country of the recipient | Only for the state registration certificate |
| \*product\_tnved | string | - | EEU FEACN code | It is returned for: |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  |  | declaration of conformity;  certificate of conformity |
| \*product\_identification | string | - | Data on products providing their identification | It is returned for:  declaration of conformity;  certificate of conformity |
| \*product\_tech\_regulations | string | - | Technical regulation (on the basis of which products were manufactured) | It is returned for:  declaration of conformity;  certificate of conformity |
| \*status | string | - | Status of the permitting document | It is returned for:  declaration of conformity;  certificate of conformity |
| \*active | boolean | - | Attribute of certificate activity | Only for the state registration certificate |
| \*update\_date | datetime | - | Date of change of the data in the TT GIS |  |
| errors | array | - | Array of errors |  |
| \*code | integer | - | Error code |  |
| \*message | string | - | Error text |  |
| \*number | string | - | Number of permitting document |  |

**JSON response example in case of success (code 200):**

{

"apiversion":4,

"result":{

"documents":[

{

"attr\_id":"23557", "number":"ЕАЭС N RU Д-UZ.0000.В.0000/00",

"from\_date":"2022-10-06",

"to\_date":"2027-10-04",

"product\_type":"Декларация о соответствии требованиям технического регламента Евразийского экономического союза (технического регламента Таможенного союза)",

"product\_name":"string",

"product\_country":null,

"applicant\_product\_name":"string",

"applicant\_product\_type":"Индивидуальный предприниматель", "manufacturer\_product\_name":"string",

"manufacturer\_product\_type":"Иностранное юридическое лицо",

"product\_tnved":"6110",

"product\_identification":null,

"product\_tech\_regulations":"ТР ТС 017/2011 О безопасности продукции легкой промышленности",

"status":"Действует",

"update\_date":"2024-03-11 11:06:34"

},

{

"attr\_id":"23765",

"number":"RU.00.00.00.000.R.000000.00.00",

"from\_date":"2024-01-29",

"to\_date":null,

"product\_name":"string",

"manufacturer\_product\_name":"string",

"manufacturer\_product\_country":null,

"get\_product\_name":"string", "get\_product\_country":null,

"active":true,

"update\_date":"2024-03-11 11:06:34"

}

]

} }

**In case of errors:**

* Error code 09: If the declaration of conformity or certificate of conformity is not found:

{

"errors\_message":"Документ <переданное значение number> от <переданное значение from\_date> не найден."

}

* Error code 10: If the state registration certificate is not found:

{

"errors\_message":"Документ <переданное значение number> не найден." }

* Error code 14: If the permitting document number does not match the required format:

{

"errors\_message":"<Текст ошибки>" }

* Error code 18: If a one-time request has been made for declaration of conformity / certificate of conformity:

{

"errors\_message":"Документ <переданное значение number> от <переданное значение from\_date> не найден в справочнике. Направлен запрос в реестр ФОИВ.

Пожалуйста, осуществите попытку позже." }

* Error code 19: If a one-time request has been made for the state registration certificate:

{

"errors\_message":"Документ <переданное значение number> не найден в справочнике. Направлен запрос в реестр ФОИВ. Пожалуйста, осуществите попытку позже."

}

## Signing a card

### Method “Get XML for subsequent signing a card”

The method “feed-product-document” returns XML of goods for signing on request with indication of an identifier of goods and / or goods code, as well as consent to publish (it is not specified for sets).

The maximum number of items in a request should not exceed 25.

**URL:** /v3/feed-product-document

**Method:** POST

**Request string example:**

POST <url of environment>/v3/feed-product-document?apikey=XXX

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Request body parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| goodIds | array[number] | - | Array of goods IDs |  |
| gtins | array[string] | - | Array of goods codes | It is specified as a string |
| publicationAgreement | boolean | - | Indicator of agreement to publish goods on the web site: национальный-каталог.рф | Possible values:  “true” — agree;  “false” — disagree (default |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  |  | value).  It is not specified for the cards of goods of the "Set" type |

**Request example:**

curl -X POST "<url of environment>/v3/feed-product-document?apikey=XXX"

-H "Content-Type: application/json; charset=utf-8"

--data-raw "{

"goodIds": [345],

"gtins": [

"00000000000001",

"123123123123"

],

"publicationAgreement": false }"

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| xmls | array | + | Array of objects |  |
| \*goodId | number | + | Goods item identifier |  |
| \*xml | string | + | XML of goods for signing |  |
| errors | array | + | Array of objects |  |
| \*goodId | number | - | Goods item identifier |  |
| \*message | string | - | Error text |  |
| \*GTIN | string | - | Goods code | In case of unsuccessful search of goods by a goods code |
| \*message | string | - | Error text | In case of unsuccessful search of goods by a goods code |

**JSON response example in case of success (code 200):**

{

"apiversion": 3,

"result": {

"xmls": [

{

"goodId": 1446768,

"xml": "xml для подписания"

}

],

"errors": [

{

"goodId": 345,

"message": "Информация о товаре ... устарела или скомпроментирована.

Попробуйте еще раз."

},

{

"GTIN": "123123123123",

"message": "Не удалось получить товар по GTIN"

}

]

}

}

### (Deprecated) Method “Sign a card by using an attached signature”

The method “feed-product-sign” accepts an objects array that contains goods identifiers (“good\_id”) and signed XML for these goods items. You can receive XML to sign by using the method feed-productdocument. The received XMLs are signed by means of the attached signature.

The maximum number of items in a request should not exceed 25.

**URL:** /v3/feed-product-sign

**Method:** POST

**Request string example:**

POST <url of environment>/v3/feed-product-sign?apikey=XXX

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Request body parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| goodId | integer | + | Identifier of a goods item for which XML is transmitted |  |
| Parameter | Type | Mandatory | Description | Comment |
| xml | string | + | XML of goods that is signed with attached signature | You can generate XML to sign by using the method feed-product-document |

**Request body parameters:**

[

{

"goodId": 5000,

"xml": "xml для товара с goodId = 5000"

},

{

"goodId": 3,

"xml": "xml для товара с goodId = 3"

},

{

"goodId": 5,

"xml": "xml для товара с goodId = 5"

} ]

Response parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| signed | array | - | Array of identifiers of goods with XMLs that have been validated, saved, and the goods item has been transferred to “Posted” status |  |
| errors | array | - | Array of objects containing the goods identifiers and a text of error occurred during processing of sent XMLs |  |
| \*goodId | integer | + | Goods item identifier |  |
| \*message | string | + | Error text |  |

**JSON response example in case of success (code 200):**

{

"apiversion": 3,

"result": {

"signed": [

5000,

3

],

"errors": [

{

"goodId": 5,

"message": "Информация о товаре ... устарела или скомпроментирована.

Попробуйте еще раз."

}

]

}

}

### Method “Sign a card by using the detached signature”

The method “feed-product-sign-pkcs” is designed to sign a goods card or several cards using a certificate in PKCS#7 format. It differs from the feed-product-sign method in that the signature (which, when using the current method, is passed as a separate parameter in the request) (a card is signed by using the detached signature) is excluded from the content of the XML goods card. Data are encoded in base64.

The maximum number of items in a request should not exceed 25.

**URL:** /v3/feed-product-sign-pkcs

**Method:** POST

**Request string example:**

POST <url of environment>/v3/feed-product-sign-pkcs?apikey=XXX

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Request body parameters:**

[

{

"goodId": 3, "base64Xml":

"PD94bWwgdmVyc2lvbj1cIjEuMFwiIGVuY29kaW5nPVwiVVRGLThcIj8+XG48Z29vZD4=...",

"signature": "..."

},

{

"goodId": 3119690,

"base64Xml":

"PD94bWwgdmVyc2lvbj1cIjEuMFwiIGVuY29kaW5nPVwiVVRGLThcIj8+XG48Z29vZD4=...",

"signature": "..."

},

{

"goodId": 521,

"base64Xml": "PD94bWwgdmVyc2lvbj1cIjEuMFwiIGVuY29kaW5nPVwiVVRGLThcIj8+XG48Z29vZD4=...",

"signature": "..."

} ]

**Request body parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| goodId | integer | + | Goods item identifier |  |
| base64Xml | string | + | Goods card (in XML format) coded in base64 | You can generate XML to sign by using the method feed-product-document |
| signature | string | + | base64 encoded detached signature corresponding to CAdES type in pkcs7 format | Metadata of XML received by means of the feedproduct-document method that is signed with detached signature |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| signed | array[integer] | - | An array of numerical identifiers of goods for which the goods card has been validated and checked (the goods card has been transferred to "Posted" status) |  |
| errors | array | - | Array of errors |  |
| \*goodId | integer | + | Goods item identifier |  |
| \*message | string | + | Error text |  |

**JSON response example in case of success (code 200):**

{

"apiversion":3,

"result":{

"signed":[

3,

521

],

"errors":[

{

"goodId":3119690,

"message":"Товар goodId: 3119690 не готов к подписанию"

}

]

}

}

## Working with subaccounts

### Method “Get a list of company subaccounts”

The method “linked-accounts” is designed to get a list of company subaccounts. The method returns a list of subaccounts with full or partial access.

**URL:** /v3/linked-accounts

**Method:** GET

**Request string example:**

GET <url of environment>/v3/linked-accounts?apikey=XXX

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |
| inn | integer | - | INN of subaccount | INN of the subaccount for which you want to display a list of available goods codes |
| offset | integer | - | Offset relative to the start of issue | If “inn” and “offset” are specified in the request, a value of the “offset” is used to move through the goods codes list |
| limit | integer | - | Number of the records in the response | Maximum allowed value is 10,000. Minimum allowed value is 1.  If “inn” and “limit” are specified in the request, a value of the  “limit” is used to move through the goods codes list |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| linked\_accounts | array | + | Array of company subaccounts | If there are no sub-accounts, an empty array will be returned |
| \*account\_inn | string | + | INN of sub-account |  |
| \*account\_name | string | + | Sub-account name |  |
| \*full\_access | boolean | + | Attribute of full or | Possible values: |
| Parameter | Type | Mandatory | Description | Comment |
|  |  |  | limited access | “true” — for accounts that have full or mixed access; “false” — for accounts that have access to separate goods (partial access) |
| \*access\_allowed | array | - | Array of goods codes | It is returned if the “inn” (“INN of subaccount”) parameter is specified in the request |
| \*\*gtin | string | - | Goods code |  |
| erros | array | - | Array of errors | It returns if there are errors |
| \*message | string | - | Error text |  |
| \*code | integer | - | Error code |  |

**JSON response example in case of success (code 200):**

{

"apiversion":3,

"result":{

"linked\_accounts":[

{

"account\_inn":"inn1",

"account\_name":"ООО \"Ромашка\"",

"full\_access":false,

"access\_allowed":[

{

"gtin":"0000000000001"

},

{

"gtin":"0000000000002"

}

]

}

]

}

}

### Method “Get a list of companies and goods codes for which access has been granted, by subaccount”

The method “linked-gtins” is designed for a subaccount to get a list of companies (which granted access to their goods cards to the subaccount (with indication of codes of such goods).

|  |  |
| --- | --- |
| Attention | Functionality for working with subaccounts allows companies to provide subaccounts with access to Russian goods codes only |

**URL:** /v3/linked-gtins

**Method:** GET

**Request string example:**

GET <url of environment>/v3/linked-gtins?apikey=XXX

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |
| inn | string | - | INN of the goods owner | If this parameter is specified, a response will return a list of goods codes belonging to an owner only with the specified INN |
| gtin | string | - | Goods code | The parameter for checking the availability of a certain goods code.  You can specify several values through the symbol “;”.  Maximum allowed number of values: 1,000 |
| limit | integer | - | Number of the records in the response | Maximum allowed value is 10,000. Minimum allowed value is 1 |
| offset | integer | - | Offset relative to the start of issue | Minimum allowed value is 0 |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| linked\_gtins | array | + | An array of available goods codes | It returns empty if the organization was not granted full or partial access as a subaccount |
| \*producer\_inn | string | + | INN of the goods owner |  |
| \*producer\_name | string | + | Name of the goods owner |  |
| \*gtin | string | + | Goods code | A goods code to which access was granted by the owner |
| errors | array | - | Array of errors |  |
| \*message | string | - | Error text |  |
| \*code | integer | - | Error code |  |

**Example of JSON response (successfully parsed information):**

{

"apiversion":3,

"result":{

"linked\_gtins":[

{

"producer\_inn":"inn1",

"producer\_name":"ООО Ромашка",

"gtin":"0000000000001"

},

{

"producer\_inn":"inn2",

"producer\_name":"ООО Василек",

"gtin":"0000000000002"

}

]

} }

**Example of JSON response (rejected information):**

{

"apiversion":3,

"result":{

"linked\_gtins":[

],

"errors":[

{

"message":"Значение ИНН:012345 должно быть указано в цифровом формате и содержать 10 или 12 цифр",

"code":97

}

]

}

}

### Method “Get XML file that is required to control access to subaccounts”

The method “linked-accounts-documents” is intended **to generate** an XML file containing information about granting or revocation of permission to use own goods codes to order marking codes and to introduce goods into circulation to other organizations (mechanism of subaccounts).

To use permissions, it is necessary to use the method “linked-accounts-sign” that allows to sign the generated XML file and to use the required permissions for a subaccount.

Access to goods by the mechanism of the subaccount can be full or partial (access according to a fixed list of the goods codes).

The maximum number of items (organizations) in a request should not exceed 100.

The maximum number of the goods codes for an organization should not exceed 1,000.

**Note:**

* full access can be granted before or after partial access is granted. If you have full access, partial access is not taken into account;
* if an organization has been granted partial and full access, then when full access is revoked, the partial access remains active;
* functionality for working with subaccounts allows companies to provide subaccounts with access to Russian goods codes only.

**URL:** /v3/linked-accounts-documents

**Method:** POST

**Request string example:**

POST <url of environment>/v3/linked-accounts-documents?apikey=XXX

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**JSON request body parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| linked\_account\_inn | string | + | INN of subaccount | It is specified as a string |
| linked\_goods | string | - | An array of goods cards for which permission is granted or revoked | It is specified to control partial access.  If it is not specified, full access will be granted or revoked |
| is\_access\_allowed | boolean | + | Attribute of revocation / granting access | Possible values:  “false” — to revoke permission from a subaccount to use goods codes to order marking codes and introduce goods into circulation;  “true” — to grant permission to a sub-account to use goods codes to order marking codes and introduce goods into circulation |

**Example of JSON request:**

curl -X POST "<url of environment>/v3/linked-accounts-documents?apikey=XXX"

-H "Content-Type: application/json; charset=utf-8"

--data-raw "{

"linked\_account\_inn": "inn1",

"linked\_goods": ["04600000000001", "04600000000002"],

"is\_access\_allowed": true }"

**Parameters of XML request body:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| mainAccountInn | string | + | INN of the goods owner | INN of the organization that makes request |
| linkedAccountInn | integer | + | INN of subaccount |  |
| linkedGoods | array | - | An array of goods cards for which access is granted or revoked | If it is not specified, full access will be granted or revoked |
| isAccessAllowed | boolean | + | Consent code and a text corresponding to the code | Possible values:  Value[0] — I revoke permission from <INN> to use my goods codes for ordering marking codes and putting goods into circulation;  Value[1] — I authorize <INN> to use my goods codes for ordering marking codes and putting goods into circulation. |

**Example of XML request body:**

curl -X POST "<url of environment>/v3/linked-accounts-documents?apikey=XXX"

-H "Content-Type: application/xml; charset=utf-8"

--data-raw "

<?xml version="1.0" encoding="UTF-8"?> <account>

<schemaVersion>5.0</schemaVersion>

<mainAccountInn>9712121212</mainAccountInn>

<linkedAccountInn>6310000012</linkedAccountInn>

<linkedGoods>

<gtin>04600000000001</gtin>

<gtin>04600000000002</gtin>

</linkedGoods>

<isAccessAllowed value="1">Я разрешаю 6310000012 использовать свои коды товара для заказа кодов маркировки и ввода товаров в оборот</isAccessAllowed>

</account> "

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| xmls | array | + | Array of objects |  |
| \*linked\_account\_inn | string | + | INN of subaccount |  |
| \*xml | string | + | XML for signing |  |
| errors | array | - | Array of errors |  |
| \*linked\_account\_inn | string | - | INN of subaccount |  |
| \*gtin | string | - | Goods code | A goods code to which access will be granted / revoked |
| \*message | string | - | Error text |  |
| \*code | integer | - | Numerical code of the error |  |

**Example of JSON response (code 200):**

{

"result":{

"xmls":[

{

"linked\_account\_inn":"inn1",

"xml":"xml1"

},

{

"linked\_account\_inn":"inn2",

"xml":"xml2"

}

],

"errors":[

{

"linked\_account\_inn":"inn3",

"message":"Для ИНН: inn3 уже предоставлен доступ ко всем Кодам Товаров.",

"code":98

}

]

}

}

### Method “Sign granting or revoking permission to a subaccount to use goods codes for MCs emission”

The method “linked-accounts-sign” is intended **to sign** an XML file containing information about granting or revocation of permission to use own goods codes to order marking codes and to introduce goods into circulation to other organizations (a mechanism of subaccounts).

Source XML file for signing is formed by the method “linked-accounts-documents”, after that it is encoded in base64 in order to be used in the method “linked-accounts-sign” (this method).

Signing is done by using a certificate in PKCS#7 format.

The maximum number of items (organizations) in a request should not exceed 100.

The maximum number of the goods codes for an organization should not exceed 1,000.

**Note:**

* full access can be issued before or after partial access is granted, and if you have full access, partial access is not taken into account;
* if an organization has been granted partial and full access, then when full access is revoked, the partial access remains active;
* functionality for working with subaccounts allows companies to provide subaccounts with access to Russian goods codes only.

**URL:** /v3/linked-accounts-sign

**Method:** POST

**Request string example:**

POST <url of environment>/v3/linked-accounts-sign?apikey=XXX

**Request string parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| apikey | string | - | Identifier (key) of the goods owner | The parameter is mandatory if “token” (“Authentication token”) is not specified |

**Request heading parameters:** Authorization: Bearer <token>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| token | string | - | TT GIS authentication token received as a result of the work of the method for getting the authentication token | This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is not specified |

**Request body parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| linked\_account\_inn | string | + | INN of subaccount |  |
| base64\_xml | string | + | XML coded in base64 with a form of granting / revoking access |  |
| Parameter | Type | Mandatory | Description | Comment |
| signature | string | + | Detached signature (in base64 format) corresponding to CAdES type in pkcs7 format |  |

**Request body example:**

[

{

"linked\_account\_inn": "inn1",

"base64\_xml":"PD94bWwgdmVyc2lvbj1cIjEuMFwiIGVuY29kaW5nPVwiVVRGLThcIj8+XG48YWNjb3VudD48c

2NoZW1hVmVyc2lvbj42LjA8L3NjaGVtYVZlcnNpb24+PG1haW5BY2NvdW50SW5uPg==...",

"signature": "..."

},

{

"linked\_account\_inn": "inn2",

"base64\_xml":"PD94bWwgdmVyc2lvbj1cIjEuMFwiIGVuY29kaW5nPVwiVVRGLThcIj8+XG48YWNjb3VudD48c

2NoZW1hVmVyc2lvbj42LjA8L3NjaGVtYVZlcnNpb24+PG1haW5BY2NvdW50SW5uPg==...",

"signature": "..."

},

{

"linked\_account\_inn": "inn3",

"base64\_xml":"PD94bWwgdmVyc2lvbj1cIjEuMFwiIGVuY29kaW5nPVwiVVRGLThcIj8+XG48YWNjb3VudD48c

2NoZW1hVmVyc2lvbj42LjA8L3NjaGVtYVZlcnNpb24+PG1haW5BY2NvdW50SW5uPg==...",

"signature": "..."

} ]

**Response parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Type | Mandatory | Description | Comment |
| signed | array | + | Array of validated data |  |
| \*linked\_account\_inn | string | + | INN of subaccount |  |
| \*access | string | + | Attribute of granting or recalling the permission | Possible values:  “allowed” — access granting;  “deleted” — access revoking |
| errors | array | - | List of errors |  |
| \*linked\_account\_inn | string | - | INN of subaccount |  |
| \*message | string | - | Error text |  |
| \*code | integer | - | Numerical code of error |  |

**JSON response example in case of success (code 200):**

{

"apiversion":3,

"result":{

"signed":[

{

"linked\_account\_inn":"inn1",

"access":"allowed"

},

{

"linked\_account\_inn":"inn2",

"access":"deleted"

}

],

"errors":[

{

"linked\_account\_inn":"inn3",

"message":"Для ИНН inn3 еще не был предоставлен доступ ко всем Кодам Товаров.",

"code":99

},

{

"linked\_account\_inn":"inn4",

"message":"Для ИНН inn4 уже предоставлен доступ к Коду Товара

4600101006008.",

"code":112

}

]

}

}

# CATALOGS

## Catalog – List of supported goods groups

|  |  |  |
| --- | --- | --- |
| Code in DB | Name | Description |
| 1 | lp | Clothing items, bed, table, bath and kitchen linens |
| 2 | shoes | Footwear |
| 3 | tobacco | Tobacco products |
| 4 | perfumery | Perfume and eau de toilette |
| 5 | tires | New pneumatic rubber tires and tire casings |
| 6 | electronics | Photo cameras (except cine cameras), flash lights and flash bulbs |
| 8 | milk | Dairy products |
| 9 | bicycle | Bicycles and bicycle frames |
| 10 | wheelchairs | Medical products |
| 11 | alcohol | Low-alcohol drinks |
| 12 | otp | Alternative tobacco products |
| 13 | water | Packaged water |
| 14 | furs | Products made of real fur |
| 15 | beer | Beer, beer-based and low-alcohol beverages |
| 16 | ncp | Nicotine products |
| 17 | bio | Biologically active food additives |
| 19 | antiseptic | Antiseptic/antibacterial skin cleansers and hand sanitizers |
| 20 | petfood | Pet foods |
| 21 | seafood | Seafood |
| 22 | nabeer | Non-alcoholic beer |
| 23 | softdrinks | Juice products and non-alcoholic beverages |
| 26 | vetpharma | Veterinary medicines |
| 27 | toys | Games and toys for children |
| 28 | radio | Radio-electronic products |
| 31 | titan | Titanium metal products |
| 32 | conserve | Canned foods |
| 33 | vegetableoil | Vegetable oils |
| 34 | opticfiber | Optical fiber and fiber optic products |
| 35 | chemistry | Perfumes and toilet preparations and household chemicals |
| 36 | books | Printed goods |
| 37 | grocery | Groceries |

|  |  |  |
| --- | --- | --- |
| Code in DB | Name | Description |
| 38 | pharmaraw | Pharmaceutical raw materials, medicines |
| 39 | construction | Building material |
| 40 | fire | Pyrotechnics and fire-fighting equipment |
| 41 | heater | Heaters |
| 42 | cableraw | Cabling and wiring products |
| 43 | autofluids | Engine oils |
| 44 | polymer | Polymer pipes |
| 45 | sweets | Confectionery products |
| 48 | carparts | Auto parts and components for vehicles |

# Modifications introduced in the previous versions of the document

v.5.39 dated 12.08.2025

You can now obtain full information about permitting documents issued not only in the Russian Federation, but also in other EAEU countries, by using the following methods:

• “Get information about permitting document by a goods code and INN” (v4/rd-info-by-gtin); • “Get information about permitting document by a number and date” (v4/rd-info).

v.5.38 dated 03.03.2025

Codes of the “Auto parts and components for vehicles”, “Confectionery products” goods groups were added into the “4.1. Catalog – List of supported goods groups”.

v.5.37 dated 23.12.2024

A clarification has been added into a comment of the “identified\_by” request body parameter (“Array of identifiers”) in the description of the method “Create or edit a card” (/v3/feed) that the parameter becomes mandatory if the “good\_images” parameter (“Array of images”) is specified.

v.5.36 dated 15.11.2024

A code of the “Polymer pipes” goods group was added into the “4.1. Catalog – List of supported goods groups”.

v.5.35 dated 31.10.2024

A clarification has been added into a description of the “3.2.2. Method “Create or edit a card”” method (/v3/feed) into a comment of the “attr\_value” request body parameter (“Attribute value”) that transfer of a compound value is expected for the “attr\_id” (“Attribute identifier”) attribute that is equal to “23890” (“Marketing authorization in the State Register”).

v.5.34 dated 22.10.2024

A code of the “Cabling and wiring products” goods group was added into the “4.1. Catalog – List of supported goods groups”.

v.5.33 dated 19.09.2024

* When creating and editing goods cards by using the method “3.2.2. Method “Create or edit a card”” (/v3/feed), attention should be paid to allowed Unicode characters.
* In the description of the method “3.3.2. Method “Get an attribute list”” (/v3/attributes) the “attr\_to\_cat\_id” parameter (“Identifier of connection of an attribute with a category”) was deleted; response examples were updated.
* A request example was updated in the description of the method “3.3.7. Method “Get information about a permitting document by a number and date”” (v4/rd-info).

v.5.32 dated 05.08.2024

A code of the “Engine oils” goods group was added into the “4.1. Catalog – List of supported goods groups”.

v.5.31 dated 01.08.2024

A method of obtaining an access key (“apikey”) was changed. For more details, see section “2.2. Universal Request Parameters”.

v.5.30 dated 05.07.2024

A code of the “Pyrotechnics and fire-fighting equipment” goods group was added into the “4.1. Catalog – List of supported goods groups”.

v.5.29 dated 03.07.2024

A code of the “Groceries” goods group was added into the “4.1. Catalog – List of supported goods groups”.

v.5.28 dated 28.06.2024

Request body parameters were corrected in the method “3.3.6. Method “Get information about a permitting document by a goods code and INN”” (v4/rd-info-by-gtin)

v.5.27 dated 25.06.2024

A code of the “Heaters” goods group was added into the “4.1. Catalog – List of supported goods groups”.

v.5.26 dated 28.05.2024

Codes of the “Building materials”, “Printed goods” goods groups were added into the “4.1. Catalog – List of supported goods groups”.

v.5.25 dated 21.05.2024

* A note has been added for the “level” (“Package type”) request body parameter of the method “feed” that the method will return an error when you try to create goods cards of the group packages if a goods item belongs to a category of unmarked products.
* A new method - “3.3.6. Method “Get information about a permitting document by a goods code and INN”” (v4/rd-info-by-gtin) has been implemented. The method allows to obtain full information about Russian permitting documents that have been specified in a goods card that has the “Published” status.
* A new method - “3.3.7. Method “Get information about a permitting document by a number and date”” (v4/rd-info) has been implemented. The method allows to obtain full information about Russian permitting documents by a number and date of the document.

v.5.24 dated 16.04.2024

An address of the test environment was changed from <https://api.integrators.nk.crptech.ru/>to [https://api.nk.sandbox.crptech.ru.](https://api.nk.sandbox.crptech.ru/)

v.5.23 dated 29.01.2024

A request body parameter was added (see section “3.2.2. Method “Create or edit a card””).

v.5.22 dated 23.01.2024

A catalog of the goods groups codes was updated.

v.5.21 dated 03.11.2023

A description of the attributes/ method has been supplemented.

v.5.20 dated 04.09.2023

URLs of the methods of authorization request (GET /auth/key) and authentication token obtainment (POST /auth/simpleSignIn) were updated. Examples of the request body were updated for the method /rd/suggest.

v.5.19 dated 25.06.2023

A description of the method /rd/suggest was added. A limit on the number of the goods cards in the request body of the feed method was changed (from 5,000 to 500). A request body example of creation of the goods card of the “Set” type was updated for the feed method. A description of the response parameters of the method categories was updated. A note was added for the response parameters of the method attributes.

v.5.18 dated 25.05.2023

pos-number and сertificate attribute types were deleted from a description of the attr\_field\_type response parameter of the method attributes. Names of the technological statuses that are provided in a description of the request parameters of the method product-list, were corrected.

v.5.17 dated 25.04.2023

A note was updated for the identified\_by request parameter of the method feed. A hyperlink to TT GIS SUA was updated.

v.5.16 dated 15.02.2023

A description of the preset\_url response parameter was added for the attributes method. A description of the method feed was supplemented in terms of transfer of the values for multiplicity attributes and permit document. A description of the method feed was supplemented in terms of transfer of the values for the categories request parameter. A description of the method categories was updated regarding a list of the goods group (values of the response parameter gismt\_codes).

v.5.15 dated 06.09.2022

A description of the feed-product and product methods was updated, first\_sign\_date and create\_date response parameters were added.

v.5.14 dated 22.08.2022

A description of the method feed was updated in terms of example of update of previously created card.

v.5.13 dated 28.04.2022

A description of the method feed-product-sign-pkcs was updated. Response parameters were added and response examples were updated in the method attributes.

v.5.12 dated 23.03.2022

An address of the test environment was changed from <https://api.integrators.nk.crpt.tech/>to [https://api.integrators.nk.crptech.ru/.](https://api.integrators.nk.crptech.ru/) Method /v3/product-list was decommissioned. The good\_status request parameter was added into the method /v4/product-list, a method description was updated.

v.5.11 dated 25.02.2022

A description of the attributes method was updated, attr\_multiplicity\_type response parameter was added.

The “Terms and definitions” section was supplemented.

v.5.10 dated 02.02.2022

A description of the method categories was updated, the request parameters (gismt\_code, tnved) were added, a list of possible values of the gismt\_codes response parameter was supplemented. A value of partial limitation of the requests was updated for the method product. A description of the method /v4/product-list was updated.

v.5.9 dated 22.12.2021

Method /v3/product-list will be decommissioned. A description of the method /v4/product-list was added. A note was added into the feed and feed-status methods. A note for the tnved request parameter was added into the method feed.

v.5.8 dated 17.09.2021

A description of the “Limit of API Requests” section was updated. A description of the method product was updated. The first\_layer response parameter was added into the method attributes.

v.5.7 dated 16.07.2021

A note was supplemented in the methods feed-product, product, short-product.

v.5.6 dated 28.06.2021

gtins and good\_ids request parameters were added into the feed-product, product, short-product methods. A note was added into the feed-status method. A description of the parameters of the feed-product-sign and feed-product-sign-pkcs methods was updated.

v.5.5 dated 28.05.2021

The cat\_id request parameter was added into the method categories. The request parameters (name, limit, offset) were added into the method brands.

v.5.4 dated 26.04.2021

The good\_draft\_ id response parameter was deleted from the feed-moderation method and the good\_id parameter was added. A note for a request with the exist parameter was added into the method linked-gtins and a limit of the gtin request parameter was changed.

v.5.3 dated 26.03.2021

The response examples were updated in the feed-product method. A list of the packages was updated in the request parameters of the method feed and response parameters of the methods product, feed-product, shortproduct. A response parameter was added for the method feed-status.

v.5.2 dated 20.02.2021

A response parameter was added for the method categories. The product-list method to obtain a list of goods cards of the organization with short information about them was added. A description of the @id parameter (entry identifier) was deleted from a description of the method feed, the parameter is not used when processing the update package.

v.5.1 dated 21.01.2021

A description of the etagslist and attributes methods was changed, new parameters were added. The subaccount request parameter was added into the feed-product method. Functionality of the method linkedaccounts was expanded. The linked-gtins method to view information on accesses provided to subaccount by other companies was added. Examples were updated and a description of the signature methods was supplemented.

v.5.0 dated 21.12.2020

The methods have been grouped. A note was added into the method attributes. A description was corrected in the Feed-product. Feed-status, information on a new status was added. The request parameters were corrected in accordance with adding a possibility to use Authorization: Bearer <token> (it is used to work with TT GIS API). Information on control of partial access by means of the linked-accounts-documents and linkedaccounts-sign methods was added.

v.4.9 dated 13.11.2020

Attributes, a description was corrected. Feed, a description was corrected in line with improved package functionality. A description of the method isocountry to obtain a catalog of countries of manufacture was added.

v.4.8 dated 16.10.2020

Feed-product, values of the good\_detailed\_status attribute were detailed

v.4.7 dated 22.09.2020

Attributes, the is\_set request parameter was added. The is\_set, set\_gtins attributes were added into the response parameters of the product, short-product and feed-product. Feed-product, good\_detailed\_status, good\_signed, good\_mark\_flag, good\_turn\_flag, flags\_updated\_date attributes were added into the response parameters. Feed, a request example of the set creation was added and a description of the request parameters was added as well. Linked-accounts, a description of the methods to work with subaccounts was added.

v.4.6 dated 25.08.2020

The terms and definitions section was supplemented;

Response parameters and request examples were updated in the product methods group. Feed-status, a request parameter was added, response parameters were supplemented.

v.4.5 dated 13.08.2020

A description of the feed-product-sign-pkcs method was added. Generate-gtins, request listing was changed

(an attribute set was corrected). Feed, request listing was changed (attr\_type was changed to attr\_value\_type).

Version 5.40

Last updated 2025-08-20 09:40:09 UTC